



Manual técnico

Juego de Serpientes y Escaleras

Herramientas Multimedia

06/03/2018

Autores:

Eduar Rafael López Zúñiga (1730289)

Kate Aracely Rodríguez Estrada (1730335)

Damarys Mendoza Vázquez (1730317)

MANUAL TÉCNICO

Tabla de contenido

Introducción.....	1
Objetivo general del programa	1
Objetivos específicos del programa	1
Reglas.....	1
Funcionalidad	2
Desarrollo(código).....	2
Conclusión.....	12

Introducción

En esta actividad se logró diseñar e implementar el ya conocido juego de serpientes y escaleras el cual cumple con la función de entretener a los usuarios siendo un juego completamente al azar, sin saber a ciencia cierta quien ganara.

El juego de serpientes y escaleras consiste básicamente que es un tablero con cien posiciones y algunas casillas se muestran imágenes de serpientes que significa descender hasta el final de dichas imágenes o se pueden presentar escaleras ascender por ellas ganando ventaja de los oponentes

Objetivo general del programa

Objetivo principal es desarrollar el juego de serpientes y escaleras en el programa de flash usando de soporte las herramientas que se han proporcionado en el lapso de la unidad.

Objetivo específicos del programa

- Se logra diseñar una interfaz dinámica con el usuario.
- Un juego que no tiene una restricción de edad.
- Si se ingresan tres usuarios entonces se permite acceder a jugar.
- Que dependiendo de las imágenes predispuestas en el tablero se pueda definir el número de casilla que tomaran cualquiera de los tres jugadores.

Reglas

- 1.-No se permite la repetición de nombre de usuario.
- 2.-El juego es completamente por turnos, no se puede realizar un movimiento, si se ese momento se está efectuando otro.
- 3.- El juego termina solo hasta que alguno de los usuarios este exactamente en la casilla cien.

Funcionalidad

- El juego se debe de mostrar en un entorno gráfico, el cual permitirá que el usuario pueda visualizar dentro del fotograma del juego el nombre que ingreso, el tiempo transcurrido, la casilla

actual que se encuentra el jugador, el número random que obtuvo al presionar el botón con la imagen del dado.

Desarrollo (código)

La función que cumplirá nuestro botón de guardar es comprobar que se ingresen exactamente tres nombres para los usuarios, se evaluara que ningún nombre se repita entre los usuarios, si llega a ocurrir se borrara ese dato y se mostrara el mensaje de nombre repetido.

```
function fguardar(e:MouseEvent):void{
    error_txt.text="";
    if (contador<3){
        if (nombre_txt.text==""){
            error_txt.text="Ingrese un Nombre";
        }else if(Arraynombres.length>=0 && Arraynombres.length<=3){
            nombre=String(nombre_txt.text);
            Arraynombres.push(nombre);
            contador++;
            nombre_txt.text = ""; // en cuanto se ingrese el nombre, se limpiara la caja de texto, para el siguiente jugador
            trace(contador);
            trace(Arraynombres); // validamos que se impriman los nombres
            //nombres_txt.text=nombres_txt.text+(" "+nombre + "\n");

            if (contador==1){
                jugador1.visible=true;
                nombres_txt.text=nombre;
            }
            if (contador==2){
                nombres2_txt.text=nombre
                if(nombres_txt.text==nombres2_txt.text){
                    contador=contador-1
                    error_txt.text="Nombre Repetido";
                    nombres2_txt.text=""
                    Arraynombres.pop();
                }else{
                    nombres2_txt.text=nombre
                    jugador2.visible=true;
                }
            }
        }
    }
}
```

Se muestran algunas de las librerías importadas, la declaración de las variables globales y se oculta la imagen de serpiente y escalera, las cuales se mostraran cuando algún de los tres usuarios este en la casilla correspondiente. Se crea un array con las cien posiciones de nuestro tablero, antes de que se ponga el nombre de la instancia de cada posición se pone tab que es la instancia de nuestro tablero, la razón por lo que se realiza esto es porque se invoca, ya que se encuentran dentro del símbolo de la imagen en dado caso que se mueva el tablero las posiciones se moverán con él. Igualmente se declaran los arrays donde se encuentra la posición inicial y final de las serpientes o escaleras


```

import flash.events.TimerEvent;
import flash.display.MovieClip;
import flash.events.MouseEvent;
import fl.transitions.Tween;
import fl.transitions.easing.*;
import fl.transitions.TweenEvent;
import flash.events.Event;
import flash.utils.Timer;
var contP1:int=0; //valores de los contadores para jugadores
var contP2:int=0;
var contP3:int=0;
var retroceder:int=0;
var i:int; //variable que se asigna para cambiar de turno
//Ocultamos las animaciones de las escaeras
escalera_mc.visible = false;
serpiente_mc.visible = false;

//array para asignar las posiciones de nuestro escenario
var ubicacion:Array=new Array(tab.p1,tab.p2,tab.p3,tab.p4,tab.p5,tab.p6,tab.p7,tab.p8,tab.p9,tab.p10,tab.p11,tab.p12,tab.p13,tab.p14,tab.p15,tab.p16,tab.p17,tab.p18,tab.p19,tab.p20,tab.p21,tab.p22,tab.p23,tab.p24,tab.p25,tab.p26,tab.p27,tab.p28,tab.p29,tab.p30,tab.p31,tab.p32,tab.p33,tab.p34,tab.p35,tab.p36,tab.p37,tab.p38,tab.p39,tab.p40,tab.p41,tab.p42,tab.p43,tab.p44,tab.p45,tab.p46,tab.p47,tab.p48,tab.p49,tab.p50,tab.p51,tab.p52,tab.p53,tab.p54,tab.p55,tab.p56,tab.p57,tab.p58,tab.p59,tab.p60,tab.p61,tab.p62,tab.p63,tab.p64,tab.p65,tab.p66,tab.p67,tab.p68,tab.p69,tab.p70,tab.p71,tab.p72,tab.p73,tab.p74,tab.p75,tab.p76,tab.p77,tab.p78,tab.p79,tab.p80,tab.p81,tab.p82,tab.p83,tab.p84,tab.p85,tab.p86,tab.p87,tab.p88,tab.p89,tab.p90,tab.p91,tab.p92,tab.p93,tab.p94,tab.p95,tab.p96,tab.p97,tab.p98,tab.p99,tab.p100);
//Arrays para los valores del las serpientes y escaleras para poder verificarlas despues en un for mas abajo
var escalerasArray:Array=new Array(5,18,30,59,67,70);
var escalerasArribArray:Array= new Array(35,39,53,83,73,89);
var serpientesArray:Array= new Array(54,68,82,96);
var serpientesAbajoArray:Array= new Array(17,47,21,43);
//Array para que el jugador gane
var NombreGanador = new Array();

```

Se declara el array de contador dado que es la variable auxiliar que almacenara el avance por cada jugador, declaración del tiempo. Y por último la declaración de la variable movimientos que se le asigna el valor del dado que se usara más adelante para poder realizar el movimiento por las casillas.

```

var contDado:Array= new Array(contP1,contP2,contP3) // en este array se guardaran los valores de nuestras variables auxiliares
var jugadores:Array= new Array(tab.jugador1,tab.jugador2,tab.jugador3) //array para nuestros jugadores
//variables para el timer1
var tmp:int=0;
var cont1:int;
var timer:Timer=new Timer (1000, cont1++);
var minutos:int; //incluimos una variable para los minutos para que en el segundo 60 nos pase directamente
var hrs:int;
//arranca el tiempo, iniciamos a contar
timer.start ();
timer.addEventListener (TimerEvent.TIMER, tiempo);

//una funcion para el tiempo
function tiempo(e:TimerEvent):void{
    tmp++; //se incrementa el tiempo
    if(tmp > 59){ //se incluye una condicion para que cuando se llegue a los 60 segundos empiece el contador de los minutos
        minutos++; //se incrementan los minutos
        tmp = 0; //reinicio los segundos para evitar que se ponga 61, si no que empiece de 0
    }
    /*if (minutos>59){ //se incluye la opcion que el tiempo tenga horas
        hrs++;
        minutos=0;
    }
}
*/
tiempo_txt.text= minutos + " m. " + tmp + " s. "; //se imprime en la caja de texto los minutos, segundos
//hrs + " h. " + tmp
}
var movimientos:int; //variable que se le asigna el valor de nuestro dado para poder verificar las posiciones que recorrera en e
//funcion que nos servira para asignar un numero random al dado

```

Con la función del dado se realiza el random de los números, y lo que salga en este se asigna a la variable movimientos que se imprimirá en su caja de texto correspondiente. Se incluye la condición que si alguno de los usuarios se encuentre exactamente en la casilla número 100 se guardara el nombre del ganador para poder mostrarse en el último frame y se manda llamar al tiempo terminar. En dado caso de que sea mayor a cien, con la variable retroceder se realiza la resta entre el contador de dado y cien. Se muestra el tween y que tendrá como posición final el retroceso necesario según lo que se haya obtenido al tirar el dado.

```
function fdado(MouseEvent):void{
    var escalera:int = 0; //variable para que pueda funcionar una escalera
    var dado:int = Math.random()*(7-1)+1; // se indica el math random para el dado
    movimientos=dado; //la variable movimientos sera igual al numero que salga en el dado
    dado_txt.text= String(dado); //en la caja de texto, se imprimira el valor de dado
    contDado[i]+=dado; //con una variable auxiliar guardamos el valor de dado
    escalera_mc.visible = false;
    serpiente_mc.visible = false;

    if(contDado[i]==100){
        player_txt.text = String(Arraynombres[i]);
        NombreGanador.push(player_txt.text);
        var movimiento1x:Tween = new Tween(jugadores[i], "x", Strong.easeIn, jugadores[i].x, ubicacion[99].x, 2, true);
        var movimiento2y:Tween = new Tween(jugadores[i], "y", Strong.easeIn, jugadores[i].y, ubicacion[99].y, 2, true);
        contDado[i]=100;
        casilla_txt.text = String(contDado[i]);
        timer.stop();
        timerterminar.start();
    }else if(contDado[i]>100){ //condicion en dado caso que el contador del dado sea mayor que 100
        retroceder = contDado[i] -100;
        trace(retroceder);
        var movimiento1x:Tween = new Tween(jugadores[i], "x", Strong.easeIn, ubicacion[99].x, ubicacion[99-retroceder].x, 2, true);
        var movimiento2y:Tween = new Tween(jugadores[i], "y", Strong.easeIn, ubicacion[99].y, ubicacion[99-retroceder].y, 2, true);
        UbicacionJugador[i]=ubicacion[99-retroceder]
        contDado[i]=100-retroceder;
        serpientes();
        trace(contDado[i])
        casilla_txt.text = String(contDado[i]);
        i++;
    }
}
```

Se aumenta la variable para cambiar de turno (i) y si esta variable es igual a 3 se reinicia. Y a continuación se empieza el delay que indicara el movimiento del jugador y el botón del dado se mantiene oculto en este lapso para evitar ser oprimido. Se declaran los contadores que se utilizaran dentro de la función, estos mismos se aumentan inmediatamente y con una condición se evalúa si el contoo es igual a uno se realiza el tween evaluando casilla por casilla donde en la posición final es la ubicación de nuestro jugador más uno.

Se resetea el contador y se imprime la casilla actual del usuario.

```
i++;
if(i == 3){
    i=0;
}
}
else{
    delay.start() //se inicia el tiempo para que realice el movimiento por cada posicion del array
    casilla_txt.text = String(contDado[i]);
    player_txt.text = String(Arraynombres[i]);
    dado_btn.visible = false; //oculta mientras se realiza el movimiento del jugador
}
}
dado_btn.addEventListener(MouseEvent.CLICK, fdado)
//Declaracion del delay
var delay:Timer = new Timer(300, cont0++);
var cont0:int=0;
var cont00:int=0;
var cont02:int=0;
function MO(event:TimerEvent):void{ //funcion MovimientoDeJugador
    cont00++
    //Suma a los contadores para las comparaciones
    cont02++
    if(cont00==1){ // se realizara el tween correspondiente casilla por casilla
        var movimiento1x:Tween = new Tween(jugadores[i], "x", Strong.easeIn, jugadores[i].x, ubicacion[UbicacionJugador[i]+1].x, 0.1,
        var movimiento2y:Tween = new Tween(jugadores[i], "y", Strong.easeIn, jugadores[i].y, ubicacion[UbicacionJugador[i]+1].y, 0.1,
        cont00=0;
        casilla_txt.text = String(contDado[i])
        UbicacionJugador[i]++
    }
}
```

Si el contador 02 es igual a la variable movimientos se detiene el delay y se reinicia el contador para la siguiente evaluación, el dado se vuelve visible y se manda llamar la función de escaleras y serpientes para realizar dicha evaluación.

```

if(cont02>=movimientos){
    //Detiene el tiempo
    delay.stop();
    //Reinicia el contador para el siguiente movimiento
    cont02=0;
    //Vuelve a poner el boton
    dado_btn.visible = true;
    //Inicia las funciones de las comparaciones
    escaleras();
    serpientes();
    i++;
    //Reinicia el valor de los jugadores cuando este llega a 3
    if(i == 3){
        i=0;
    }
}
}
delay.addEventListener(TimerEvent.TIMER,MO);
}

var UbicacionJugador:Array = new Array(-1,-1,-1);
//Los datos se irán añadiendo en el orden que establezcamos
//Se colocan 3 -1 ya que eso sirve para que al momento de hacerle la suma al valor del
//array y mover los jugadores esta no se vea afectada y los ponga 1 casilla adelante
//Funcion para la verificar si el jugador callo en una escalera
var movimientoterminado:Boolean;
function escaleras(){
    //Hacemos un for para que verifique todos los datos de un array que ha sido colocado previamente
    //Que es donde estan todos los valores de las escaleras
    for(var v:int=0; v<escalerasArray.length;v++){
        //Si es que la variable del contador de casillas del contador llega a ser igual a una dentro del array de 1
        if(contDado[i]==escalerasArray[v]){
            var lugar:int;
            //Declara la variable que le dara el movimiento en el tween
            //Esta variable toma el valor de un array, que contiene los valores de donde termina la escalera
            lugar=escalerasArribArray[v]
            //Hace los tween
            //Primero "Termina el movimiento haciendo que el jugador quede abajo en la escalera"
            if(movimientoterminado==false){
                jugadores[i].x = ubicacion[escalerasArray[v]-1].x
                movimientoterminado= true;
            }
        }
    }
}

```

Con el array UbicaciónJugador llevamos un control de almacenamiento, se le restan -1 tres veces para que al momento de hacer la suma el valor del array no se vea afectado. Y creamos la variable movimientoterminado de tipo boolean (esto lo que hace es darle un tiempo al jugador para terminar de llegar a la casilla antes de que entre la función de la serpiente y lo haga descender). Y hacemos la función de las escaleras, que lo que hace es con un for verifica el array de las escaleras, si nuestra casilla es igual a alguna de las posiciones dentro del array de las escaleras, la variable lugar2 toma el valor de esta. Con la variable movimientoterminado en falso hace que el jugador llegue a la casilla, al llegar pone la variable movimientoterminado en true.

```

        if(movimientoteterminado==true){
            //Desde abajo de la escalera hace el movimiento
            var movimientox:Tween = new Tween(jugadores[i], "x", Strong.easeIn, jugadores[i].x, ubicacion[lugar-1],
            var movimientoy:Tween = new Tween(jugadores[i], "y", Strong.easeIn, jugadores[i].y, ubicacion[lugar-1],
            movimientoteterminado=false;
        }
        //Modifica la ubicacion del jugador
        UbicacionJugador[i]=lugar-1
        //Actualiza el valor del dado
        contDado[i]=lugar
        escalera_mc.visible = true;
        //Y muestra donde es que ahora esta el jugador
        casilla_txt.text = String(contDado[i])
    }
}
}

```

Al poner esta variable en true, entra el tween de la caída por la serpiente y se vuelve a poner el movimiento terminado en falso. Después actualiza las cajas de texto del tablero, y la imagen de escalera es visible.

```
function serpientes() {
    for (var v2: int = 0; v2 < serpientesArray.length; v2++) {
        if (contDado[i] == serpientesArray[v2]) {
            var lugar2: int;
            lugar2 = serpientesAbajoArray[v2]
            if (movimientoterminado == false) {
                jugadores[i].x = ubicacion[serpientesArray[v2] - 1].x
                movimientoterminado = true;
            }
            if (movimientoterminado == true) {
                var movimiento1x: Tween = new Tween(jugadores[i], "x", Strong.easeIn, jugadores[i].x, ubicacion[lugar2 - 1].x, 0.5);
                var movimiento2y: Tween = new Tween(jugadores[i], "y", Strong.easeIn, jugadores[i].y, ubicacion[lugar2 - 1].y, 0.5);
                movimiento1x.start();
                movimiento2y.start();
                movimientoterminado = false;
            }
            UbicacionJugador[i] = lugar2 - 1
            contDado[i] = lugar2
            serpiente_mc.visible = true;
            casilla_txt.text = String(contDado[i])
        }
    }
}
```

Con la función llamada serpientes evaluamos la caída por las serpientes, con un for que va verificando el Array de las serpientes. Se crea una variable lugar2 que toma el valor de serpientes Abajo (array donde se encuentran las posiciones finales de todas las serpientes) y después verificamos que si movimiento terminado es falso la ubicación del jugador sigue siendo donde se encuentra el inicio de la serpiente, se pone el movimiento terminado en true para ahora si hacer el descenso del jugador y se vuelve a poner el movimiento terminado en falso. Y se actualizan las cajas de texto de casilla actual, la ubicación del jugador y la imagen de las serpientes se ponen visibles.


```

var contterminar:int=0;
var timerterminar:Timer=new Timer (1000, contterminar++);
//arranca el tiempo, iniciamos a contar
timerterminar.addEventListener (TimerEvent.TIMER, timepoter);

//una funcion para el tiempo que hace ganar al jugador
function timepoter(e:TimerEvent):void{
    contterminar++;
    //Despues de 3 segundos inicia el contador de los jugadores
    if(contterminar>=3){
        //Mete al array del ganador el que gano
        NombreGanador.push(Arraynombres[i])
        //Detiene este contador
        timerterminar.stop()
        //Va al fotograma 5 que es donde esta la animacion para ganar y exportar texto
        gotoAndStop(5);
    }
}

```

Este timer manda a llamar la función timepoter, esta lo que hace es que después de 3 segundos de que alguien gane lo almacena en el array del ganador, detiene el tiempo y te lleva al fotograma 5.

Se empieza importando las librerías de siempre, y posteriormente se pone la línea para imprimir el ganador, que se encuentra en la posición 0 del array creado en el frame 4 específicamente solo para el ganador.

```

stop();
import flash.utils.*;
import flash.events.TimerEvent;
import flash.display.MovieClip;
import flash.events.MouseEvent;
import fl.transitions.Tween;
import fl.transitions.easing.*;
import fl.transitions.TweenEvent;
import flash.display.Sprite;
import flash.events.Event;
import flash.utils.Timer;

ganador_txt.text = NombreGanador[0];

```

Aquí se programa el botón para exportar en .txt los datos de los jugadores, se imprime primero el ganador del juego con `NombreGanador[0]` y después se imprimen todos los jugadores de la ronda según el orden de registro.

```

txt_btn.addEventListener(MouseEvent.CLICK, fexportar);
function fexportar(event:MouseEvent):void{
    //Crea un nuevo documento .txt
    var archivo:FileReference = new FileReference();
    //Le pide al usuario un lugar donde guardar el documento el cual se genera con los pa
    archivo.save("#####
    "
        "Ganador de la ronda:" + "Nombre: " + NombreGanador[0] +
        "Tabla de los jugadores" +
        "Jugador 1:" + "Nombre:" + Arraynombres[0] +
        "Jugador 2:" + "Nombre:" + Arraynombres[1] +
        "Jugador 3:" + "Nombre:" + Arraynombres[2] +
        "#####
    "
}

pdf_btn.addEventListener(MouseEvent.CLICK, fpdf);
function fpdf(event:MouseEvent):void{
    //Crea un nuevo documento .txt
    var archivo:FileReference = new FileReference();
    //Le pide al usuario un lugar donde guardar el documento el cual se genera con los pa
    archivo.save("#####
    "
        "Ganador de la ronda:" + "Nombre: " + NombreGanador[0] +
        "Tabla de los jugadores" +
        "Jugador 1:" + "Nombre:" + Arraynombres[0] +
        "Jugador 2:" + "Nombre:" + Arraynombres[1] +
        "Jugador 3:" + "Nombre:" + Arraynombres[2] +
        "#####
    "
}

```

Se hace lo mismo con el botón para exportar a pdf.

```
fregresar_btn.buttonMode=true;
fregresar_btn.addEventListener(MouseEvent.CLICK, fsalir)
function fsalir (MouseEvent):void{
    gotoAndStop(1);
}
```

Este es el botón de regresar, su única función es llevarte a la portada.

Conclusión

En la elaboración de este manual técnico se logró desarrollar ampliamente la explicación sobre cómo se implementaron las distintas herramientas que brinda el programa de flash en el lenguaje actionscript 3.0, todo esto con el objetivo final de elaborar el juego de serpientes y escaleras. Cumpliendo las especificaciones solicitadas por el docente, en tiempo y forma.