Proyecto: Unidad I – Unidad II

Herramientas Multimedia

Serpientes y escaleras

Manual Técnico

HM-ITI-07116

Alumnos:

Jonathan Garcia Gonzalez (1730380)

Jovan Garcia Gonzalez (1730381)

Profesor:

MSI Mario Humberto Rodríguez Chávez

Introducción

El juego de serpientes y escaleras es un antiguo juego de azar en tablero de origen indio, actualmente este juego es considerado como un clásico a nivel mundial. El juego se juega de dos o más personas, se juega en un tablero con casillas enumeradas (el número de casillas puede variar), este contiene un número determinado de serpientes y escaleras que conectan dos casillas separadas por una o más líneas de casillas, Los jugadores se mueven por medio de fichas, casilla por casilla, el jugador puede avanzar de 1 a 6 casillas por medio de un dado de 6 lados. El objetivo del juego es lograr que la ficha del jugador llegue hasta la casilla final, al llegar a una casilla en donde empiece una escalera este subirá por esta dándole una posición con más ventaja para ganar, y al llegar a una casilla en donde comience la cola de una serpiente bajara disminuyendo sus posibilidades de ganar.

En este documento se presentará la forma en que se realizó el Proyecto de la Unidad 2 (Juego de serpientes y escaleras), el cual consiste de 100 casillas y un numero de 2 a 3 jugadores, explicando detalladamente cada una de las acciones que se realizaron en el documento .fla para el funcionamiento del juego.

El código del juego se realizó apoyándose en los conocimientos aprendidos en la materia de Herramientas Multimedia y adecuando una lógica en la que se complementara la forma de pensar y de hacer las cosas de los desarrolladores del proyecto.

Al final de este documento se dará a conocer la conclusión, los puntos de vista y aprendizajes adquiridos a lo largo de la realización de este proyecto.

Índice

Desarrollo (scripts)

Fotograma 1. (Portada)

En el código de la portada se muestra las animaciones con Tweens de los diferentes elementos que la conforman, también la función y listener del botón “Iniciar” que llevara al siguiente fotograma para el registro de jugadores.

```
import flash.events.MouseEvent;
stop();

cuadro2.visible=false;
inst.visible=false;
inst2.visible=false;
ok.visible=false;

var movep:Tween = new Tween(name2,"x",Elastic.easeInOut,800,280.05,3,true);
var move1:Tween = new Tween(name1,"x",Strong.easeInOut,-450,282.05,2,true);

var move2:Tween = new Tween(universidad,"y",Elastic.easeInOut,-800,35,3,true);
var move3:Tween = new Tween(cuadro,"x",Strong.easeInOut,-450,223.8,2,true);

var move4:Tween = new Tween(materia,"x",Elastic.easeInOut,800,251.8,3,true);

var move5:Tween = new Tween(profesor,"x",Elastic.easeInOut,800,203.8,3,true);
var move6:Tween = new Tween(grupo,"x",Strong.easeInOut,-450,361.05,3,true);

var move7:Tween = new Tween(portada_btn,"y",Elastic.easeInOut,800,474,3,true);
var move8:Tween = new Tween(ayuda_btn,"y",Elastic.easeInOut,800,474,3,true);

portada_btn.addEventListener(MouseEvent.CLICK,fportada);
function fportada (event:MouseEvent):void{
    nextFrame();
}

ayuda_btn.addEventListener(MouseEvent.CLICK,fayuda);
function fayuda (event:MouseEvent):void{
    cuadro2.visible=true;
    inst.visible=true;
    inst2.visible=true;
    ok.visible=true;
}

ok.addEventListener(MouseEvent.CLICK,fok);
function fok (event:MouseEvent):void{
    cuadro2.visible=false;
    inst.visible=false;
    inst2.visible=false;
    ok.visible=false;
```

Fotograma 2. (Registro de jugadores)

Registro de Jugadores.

En el registro de jugadores declaramos arrays para los nombres de los jugadores, en el registro se validó para que solo puedan ser introducidos máximo 3 jugadores y mínimo 2 jugadores. Presionando el botón registrar se ejecuta una función en la cual se almacena en el array jugadores los nombres que se ingresaron.

```
//variables y arrays para guardar datos de jugadores
var njugadores:int=0;
var jugadores:Array = new Array();

j1.visible=false;
j2.visible=false;
j3.visible=false;

var iconos:Array= new Array (j1,j2,j3);
var textoj:Array= new Array (tj1,tj2,tj3);

//jugadores registrados: 0
numjugadores.text="Jugadores registrados: " + njugadores;

//boton para registrar validado (listener y funcion)
registrarbtn.addEventListener(MouseEvent.CLICK,registrar);
function registrar (event:MouseEvent) :void{
    //validacion de nombre de jugador
        if(pnametxt.text==""||pnametxt.text=="Ingresa por lo menos dos jugadores"){
        }else{
            //si se ingresan cuatro jugadores no sera permitido ingresar mas
            if(njugadores==3){
                pnametxt.text="Numero maximo de jugadores ingresado";
            }else{
                iconos[njugadores].visible=true;
                textoj[njugadores].text=pnametxt.text+"";
                njugadores++;
                jugadores.push(pnametxt.text);
                trace(jugadores);
                pnametxt.text="";
                numjugadores.text="Jugadores registrados: " + njugadores;
            }
        }

}
```

Al presionar el botón de Empezar y habiendo ingresado a todos los jugadores dará acceso al siguiente fotograma que dará inicio al juego.

```
//boton empezar juego validaciones()
btnempezar.addEventListener(MouseEvent.CLICK,juego);
function juego (event:MouseEvent) :void{
    //validaciones si no hay jugadores registrados no pasara al juego
        if(njugadores==0||njugadores==1){
            pnametxt.text="Ingresa por lo menos dos jugadores";
        }else{
            gotoAndStop(3);
        }

}

```

Fotograma 3. (Juego)

Ubicación de posiciones.

Para ubicar las posiciones de las casillas del tablero en donde pasará y se detendrá el jugador colocamos símbolos de cuadros invisibles que nos servirán como referencia para las coordenadas de las casillas del tablero.

Una vez ubicados los símbolos creamos un array en donde almacenamos cada de las instancias de los símbolos de ubicaciones (de p1 a p100). Asi de esta forma nosotros podemos acceder a cada una de las posiciones por medio del mismo array.

```
import fl.transitions.TweenEvent;
import flash.events.MouseEvent;
import fl.transitions.Tween;
import flash.events.MouseEvent;
import flash.events.Event;

//variables para identificar a que casilla se moveran los jugadores
var ji:int=0;
var ji2:int=0;
var ji3:int=0;
//array con las instancias de las 100 casillas estas son cuadrados ubicados en la matriz
var array:Array = new Array (p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18,p19,p20,p21,p22,p23,p24,p25,
stop();
```

Creación del dado con números aleatorios.

Para implementar el dado con números aleatorios se utilizaron 3 funciones: fdado, flanzar y fdetener..

En la función flanzar que se activa presionando el botón “lanzar” del dado lo que se hace es mandar llamar la función fdado y ocultar el botón para no crear conflictos.

En la función fdado lo que se hace generar un numero aleatorio del 1 al 6, después dependiendo del número que se genere hace una comparación para mostrar la imagen con su correspondiente número de dado, para dar la animación del dado aleatorio.

La función fdetener se ejecuta al presionar el botón “detener” del dado, lo que hace es detener la función fdado para que deje de generar números aleatorios y tomar el último número random generado que será el número de posiciones que el jugador avanzara. Al final se ejecutará la función de movimiento (1,2 y 3) que moverá al jugador actual.

```
//se pone invisible el boton detener
detenerbtn.visible=false;

//listener y funcion del boton lanzar dado
lanzarbtn.addEventListener(MouseEvent.CLICK,flanzar);
function flanzar (event:MouseEvent):void{
    //listener de escenario
    stage.addEventListener(Event.ENTER_FRAME,fdado);
    lanzarbtn.visible=false;
    detenerbtn.visible=true;
}
```

```
//funcion y listener para detener dado, el usuario decide cuando ejecutarlo
detenerbtn.addEventListener(MouseEvent.CLICK,fdetener);
function fdetener (event:MouseEvent):void{
    //remueve el listener de escenario del numero aleatorio
    stage.removeEventListener(Event.ENTER_FRAME,fdado);
    numdado=Number(dadotxt.text);
    //se suma la variable i con el numero random

    detenerbtn.visible=false;
    lanzarbtn.visible=true;
    //la ficha del jugador se posicionara en la ubicacion del cuadro por medi
    if(ja==0){
        stage.addEventListener(Event.ENTER_FRAME,movimiento);
    }
    if(ja==1){
        stage.addEventListener(Event.ENTER_FRAME,movimiento2);
    }
    if(ja==2){
        stage.addEventListener(Event.ENTER_FRAME,movimiento3);
    }
}
```

```
//funcion del dado escogera un numero
function fdado (event:Event):void{

    num=Math.random()*6;
    if(num==0){
        dadotxt.text=6+"";
    }else{
        dadotxt.text=num+"";
    }
    if(dadotxt.text=="1"){
        d1.visible=true;
        d2.visible=false;
        d3.visible=false;
        d4.visible=false;
        d5.visible=false;
        d6.visible=false;
    }
    if(dadotxt.text=="2"){
        d1.visible=false;
        d2.visible=true;
        d3.visible=false;
        d4.visible=false;
        d5.visible=false;
        d6.visible=false;
    }
    if(dadotxt.text=="3"){
        d1.visible=false;
        d2.visible=false;
        d3.visible=true;
        d4.visible=false;
        d5.visible=false;
        d6.visible=false;
    }
    if(dadotxt.text=="4"){
        d1.visible=false;
        d2.visible=false;
        d3.visible=false;
        d4.visible=true;
        d5.visible=false;
        d6.visible=false;
```

Movimiento del jugador.

La parte del movimiento de los jugadores consiste en mas de 3000 líneas de código ya que usamos 3 funciones diferentes para los movimientos de los posibles 3 jugadores, por lo que intentaremos resumir la función del movimiento de un jugador.

Una vez que tengamos el número del dado que avanzaremos y las ubicaciones de las casillas, al detenerse el dado se ejecuta la función movimiento (sea para el jugador 1, 2 o 3) que consiste en mover el símbolo del jugador actual por el eje de las x hasta llegar a su destino o sea su posición final que se almacena en la variable ji.

```
var señal:int=0;
//lai sirve para guardar la posicion en el la que se encuentra la ficha
var lai:int=0;
//senal para actualizar el valor de lai
var i2:int=0;
function movimiento (event:Event):void{
    lanzarbtn.enabled=false;
    //Condicion para moviemiento de la fila 1
    if((jugador.x<=1000 && jugador.x>=int(array[9].x) && (jugador.y>=int(array[0].y)))){
        if(i2==0){
            lai=ji;
            ji=ji+numdado;
        }
        texto.text="Casilla actual: "+ji;
        i2=1;


        if(bandera==1){
            bandera=0;
            stage.removeEventListener(Event.ENTER_FRAME,movimiento);
            stage.addEventListener(Event.ENTER_FRAME,movimientoarriba);

        }else{
            jugador.x -=3;
        }
```

En la función movimiento se evalúan condiciones por si ha llegado a su posición destino, si ha llegado a un extremo de una fila o bien si ha caído en una serpiente o escalera, en este ultimo caso se ejecutan las funciones “subirescalera” o “bajarserpiente”. Cabe mencionar que se realizan todas las mismas condiciones en cada fila, si tiene serpientes o escaleras.

```
        //escalera 1
        if(jugador.x==int(array[8].x) && jugador.y==int(array[8].y) && ji==9){
            escalera=1;
            subirescalera();
            ji=33;
            texto.text="Casilla actual: "+ji;
        }

        //funcion extremo 10
        if(jugador.x==int(array[9].x) && jugador.y==int(array[9].y)){
            if(ji==10){
                trace("se hace la condicion");
                i2=0;
                bandera=1;
                stage.removeEventListener(Event.ENTER_FRAME,movimiento);
                ja++;
                señal=1;
                if(njugadores==3){
                    if(ja>2){
                        ja=0;
                    }
                }
                if(njugadores==2){
                    if(ja>1){
                        ja=0;
                    }
                }

            }else{
                i2=1;
                stage.addEventListener(Event.ENTER_FRAME,movimientoarriba);
                stage.removeEventListener(Event.ENTER_FRAME,movimiento);

            }
        }
```

Al llegar a la posición destino del jugador actual, se detiene el listener de la función movimiento y cambia el turno al siguiente jugador (cuando la variable ja se incrementa).

```
264         if((jugador.x==int(array[(lai+numdado)-1].x)) && (jugador.y==int(array[(lai+numdado)-1].y))){
265             i2=0;
266             stage.removeEventListener(Event.ENTER_FRAME,movimiento);
267             lanzarbtn.enabled=true;
268             if(señal==0){
269                 ja++;
270             if(njugadores==3){
271                 if(ja>2){
272                     ja=0;
273                 }
274             }
275             if(njugadores==2){
276                 if(ja>1){
277                     ja=0;
278                 }
279             }
280             }else{
281                 señal=0;
282             }
283 
284             //gotoAndStop(3);
285         }
286     }
```

Fin de turno de jugador actual y siguiente jugador.

Para pasar de turno del jugador actual a otro insertamos una condición cuando el jugador actual llega a su posición destino, en esta se incrementa el número de jugador actual "ja" que nos indica la posición de los array del jugador actual, luego se evalúan las condiciones que dicen si el "ja" ha superado el número máximo de jugadores (sea 2 o 3), el contador del jugador actual volverá a inicializarse en 0, o sea la posición del primer jugador y volverán a reiniciarse los turnos.

Asimismo se creó la variable señal para indicar si un cambio de turno ya fue realizado y en caso de que sea correcto, entonces no se realizara el cambio de turno.

```
if(señal==0){
    ja++;
if(njugadores==3){
    if(ja>2){
        ja=0;
    }
}
if(njugadores==2){
    if(ja>1){
        ja=0;
    }
}
}else{
    señal=0;
}
```

Serpientes y Escaleras.

En caso de que el jugador caiga en la posición de una serpiente o escalera, se ejecutara la función “subirescalera” o “bajarserpiente” en ellas se aplica una animación con Tween en donde se desplaza la posición inicial de la escalera o serpiente hasta la posición final, también se actualiza la posición a la que se desplazó el jugador.

Condición Escaleras:

```
227         //escalera 1
228         if(jugador.x==int(array[8].x) && jugador.y==int(array[8].y) && ji==9){
229             //se define el numero de escalera
230             escalera=1;
231             //se ejecuta la funcion subirescalera
232             subirescalera();
233             //se actualiza la posicion del jugador
234             ji=33;
235             texto.text="Casilla actual: "+ji;
236         }
```

Función “subirescalera”:

```
155 function subirescalera (){
156     if(escalera==1){
157         var escx:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[8].y),int(p33.y),1,true);
158         var escy:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[8].x),int(p33.x),1,true);
159     }
160     if(escalera==2){
161         var escx2:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[14].y),int(p46.y),1,true);
162         var escy2:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[14].x),int(p46.x),1,true);
163     }
164     if(escalera==3){
165         var escx3:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[18].y),int(p43.y),1,true);
166         var escy3:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[18].x),int(p43.x),1,true);
167     }
168     if(escalera==4){
169         var escx4:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[58].y),int(p76.y),1,true);
170         var escy4:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[58].x),int(p76.x),1,true);
171     }
172 }
```

Condición Serpientes:

```
390         //serpiente 1
391         if(jugador.x==int(array[20].x) && jugador.y==int(array[20].y) && ji==21){
392             serpiente=1;
393             bajarserpiente();
394             ji=1;
395             texto.text="Casilla actual: "+ji;
396         }
```

Función “bajarserpientes”:

```
function bajarserpiente (){
    if(serpiente==1){
        var escx5:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[20].y),int(p1.y),1,true);
        var escy5:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[20].x),int(p1.x),1,true);
    }
    if(serpiente==2){
        var escx6:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[43].y),int(p5.y),1,true);
        var escy6:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[43].x),int(p5.x),1,true);
    }
    if(serpiente==3){
        var escx7:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[44].y),int(p3.y),1,true);
        var escy7:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[44].x),int(p3.x),1,true);
    }
    if(serpiente==4){
        var escx8:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[82].y),int(p55.y),1,true);
        var escy8:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[82].x),int(p55.x),1,true);
    }
    if(serpiente==5){
        var escx9:Tween = new Tween(njugador[ja],"y",Regular.easeInOut,int(array[86].y),int(p49.y),1,true);
        var escy9:Tween = new Tween(njugador[ja],"x",Regular.easeInOut,int(array[86].x),int(p49.x),1,true);
    }
}
```

En cada condición de movimiento de cada fila se le inserta las condiciones de sus respectivas serpientes y escaleras, además se indica que numero de serpiente o escalera es, para identificar qué movimiento se realizara.

Movimientos hacia arriba.

En caso de que el jugador caiga en un extremo de una fila de casillas se realizara la condición de que si es esa su posición final o destino entonces permanecerá en ella, y cambiara de turno, si no entonces se ejecutara una función llamada “movimientoarriba” en esta función el símbolo del jugador actual se moverá verticalmente hacia arriba en el eje “y” para avanzar a la siguiente posición, una vez que que llegue a la posición de arriba se verificaran la condición de que si esa es la posición final para jugador actual entonces permanecerá ahí, si no, entonces se detendrá el listener de la función movimientoarriba y se ejecutara nuevamente la función movimiento para que siga nuevamente su recorrido en la siguiente fila hasta su posición final.

```
        //funcion extremo 10
        if(jugador.x==int(array[9].x) && jugador.y==int(array[9].y)){
            if(ji==10){
                trace("se hace la condicion");
                i2=0;
                bandera=1;
                stage.removeEventListener(Event.ENTER_FRAME,movimiento);
                ja++;
                señal=1;
                if(njugadores==3){
                    if(ja>2){
                        ja=0;
                    }
                }
                if(njugadores==2){
                    if(ja>1){
                        ja=0;
                    }
                }

            }else{
                i2=1;
                stage.addEventListener(Event.ENTER_FRAME,movimientoarriba);
                stage.removeEventListener(Event.ENTER_FRAME,movimiento);

            }
        }
```

Función “movimientoarriba”:

```
function movimientoarriba (event:Event):void{
    jugador.y -=1;
    if(jugador.y==int(array[10].y)){
        if(ji==11){
            stage.removeEventListener(Event.ENTER_FRAME,movimientoarriba);
            lanzarbtn.enabled=true;
            i2=0;
            ja++;
                if(njugadores==3){
                    if(ja>2){
                        ja=0;
                    }
                }
                if(njugadores==2){
                    if(ja>1){
                        ja=0;
                    }
                }
        }else{
            stage.addEventListener(Event.ENTER_FRAME,movimiento);
            stage.removeEventListener(Event.ENTER_FRAME,movimientoarriba);
        }
    }
```

```
    if(jugador.y==int(array[20].y)){
        if(ji==21){
            stage.removeEventListener(Event.ENTER_FRAME,movimientoarriba);
            serpiente=1;
            ji=1;
            bajarserpiente();
            lanzarbtn.enabled=true;
            i2=0;
            ja++;
                if(njugadores==3){
                    if(ja>2){
                        ja=0;
                    }
                }
                if(njugadores==2){
                    if(ja>1){
                        ja=0;
                    }
                }
        }else{
            trace("Se hace el else");
            stage.addEventListener(Event.ENTER_FRAME,movimiento);
            stage.removeEventListener(Event.ENTER_FRAME,movimientoarriba);
        }
    }
```

Y así sucesivamente hasta la posición 91, que sería el último movimiento hacia arriba que hace en el tablero.

Todas las funciones y condiciones mostradas anteriormente fueron aplicadas para cada uno de los 3 jugadores posibles. A eso se debe lo extenso del código.

Guardar nombre del ganador.

Al final de cada fila se realiza una condición en la que dice que si el jugador alcanza la posición 99 del array (100 en el caso de la tabla). Se guardará su nombre en la variable ganador y se pasara al último fotograma donde se muestran los resultados.

```
911         if(jugador.x==int(array[99].x) && jugador.y==int(array[99].y)){
912                 trace("se hace la condicion");
913                 i2=0;
914                 bandera=1;
915                 stage.removeEventListener(Event.ENTER_FRAME,movimiento);
916             ja++;
917             señal=1;
918                 if(njugadores==3){
919                     if(ja>2){
920                         ja=0;
921                     }
922                 }
923                 if(njugadores==2){
924                     if(ja>1){
925                         ja=0;
926                     }
927                 }
928                 ganador=jugadores[0];
929                 gotoAndStop(4);
930         }
```

Fotograma 4.

El fotograma 4 consiste en la pantalla de resultados donde se plasma en un texto dinámico el nombre del jugador que llega a la posición 100 del tablero, dicho nombre se encuentra almacenado en la variable ganador así que lo único que se hace es imprimir el texto de la variable en el texto dinámico.

```
1   Ganador.text=ganador+"";
2
```

Exportación de los resultados a txt.

En la parte final del código se implementa la exportación de los resultados a un documento txt. Mediante una función de un botón llamada “fexportaciontxt”, dentro de la funcion se declara una variable tipo cadena que almacenara el resultado que queremos exportar, despues mediante un FileRefence indicamos la cadena que se exportara y el nombre del archivo que se generara.

Exportación de los resultados a pdf.

Igualmente se utiliza la misma función y método para exportar a pdf solamente cambiando la extensión de txt a pdf.

Conclusión.

En conclusión, de este proyecto y manual técnico, las funciones y técnicas aprendidas, por medio de las prácticas y proyectos anteriores durante el curso de herramientas multimedia nos fueron de mucha utilidad al realizar el juego con resultados satisfactorios. Gracias a la realización del proyecto pudimos adquirir mejor conocimiento sobre el uso de los métodos que también fueron usados en proyectos y practicas anteriores.

Este juego puede parecer que es muy simple de realizar, pero tiene su propia complejidad una vez que se realiza para que de los resultados deseados y que mereció dedicación de nuestro tiempo para poder realizarlo correctamente.

A lo largo de todo el desarrollo del juego se pudo apreciar muchos fallos, que a medida que pasaba el tiempo gracias a aplicar la lógica entre los desarrolladores y el trabajo en equipo se pudieron solucionar a tiempo. En la ultima actualización del proyecto se extendió mucho el código principal del tablero (fotograma 3), pero gracias a esto se pudo solucionar la mayor parte de los fallos.