



Universidad Politécnica de Victoria

HERRAMIENTAS MULTIMEDIA

DOCUMENTO TÉCNICO PARA LA REALIZACIÓN DE UN SERPIENTES Y ESCALERAS CON ACTIONSCRIPT 3.0

Integrantes:

Saldívar Izaguirre Elvira Marisol

Profesor: Mario Humberto Rodríguez Chávez

Cd. Victoria, Tamaulipas a 06 de marzo de 2018

Contenido

1. Introducción	3
2. Desarrollo	3
2.1 Portada.....	3
2.2 Jugadores.....	4
2.3 Selección de peones.....	6
2.4 Reglas	8
2.5 Juego.....	8
2.6 Resultados	12

1. Introducción

En este manual se especificará cómo se realizó el juego de Serpientes y Escaleras.

Aunque el .fla no cumplió con la esencia del juego clásico de Serpientes y Escaleras, se realizó hasta obtener los resultados finales y su exportación en PDF como fue requerido desde un inicio.

2. Desarrollo

2.1 Portada



```

1  import flash.events.MouseEvent;
2  import fl.transitions.Tween;
3  import fl.transitions.TweenEvent;
4  import fl.transitions.easing.*;
5
6
7  stop();
8  var skool:Tween = new Tween(upv,"x",Regular.easeOut,-200,81,2,true);
9  var hm:Tween = new Tween(materia,"y",Regular.easeOut,-200,51,2,true);
10 var loguillo:Tween = new Tween(logo,"x",Regular.easeOut,1000,291,2,true);
11 var namename:Tween = new Tween(nombre,"y",Regular.easeOut,700,396,2,true);
12 function entrar (event:MouseEvent):void{
13     gotoAndStop(2);
14 }
15 entrarbtn.addEventListener(MouseEvent.CLICK,entrar);
16
17 var j:int;
  
```

Se importaron las librerías necesarias y se declararon los tweens correspondientes a los gráficos colocados en la portada.

La función "entrar" se activa al darle clic al botón amarillo, y este nos pasa al siguiente frame.

2.2 Jugadores



PREIONE EL BOTÓN "LISTO" CUANDO TERMINE DE INGRESAR LOS NOMBRES DE LOS JUGADORES

Ingrese nombre del jugador:

Jugador 1

Jugador 2

Jugador 3

Guardar

Regresar

En esta pantalla se ingresan los nombres de los jugadores. El juego necesita únicamente 3 jugadores, y por cada uno se le da clic al botón "Guardar". Cuando se termina de ingresar los nombres de los 3 jugadores, aparecerá un botón "Listo".



PREIONE EL BOTÓN "LISTO" CUANDO TERMINE DE INGRESAR LOS NOMBRES DE LOS JUGADORES

Ingrese nombre del jugador:

Jugador 1

Jugador 2

Jugador 3

Guardar

Listo

Escoger peones

Regresar

Al darle clic al botón "Listo", aparecen los nombres de los jugadores en los rectángulos izquierdos, así como el botón de "Escoger peones".

```

1  import flash.events.MouseEvent;
2
3  var nombresArr: Array = new Array; //array que guarda nombre de jugadores
4  advice.visible = false; //aviso de ingresar nombre
5  choosebtn.visible = false; //botón escoger peones
6  readybtn.visible = false; //botón de listo
7  readytxt.visible = false; //texto de listo
8  nametxt.restrict = "A-Z.a-z"; //restricción de letras
9
10 //función para guardar nombres
11 function guardarnames(event: MouseEvent): void {
12     if (nametxt.text == "") {
13         nombresArr.pop();
14         advice.visible = true;
15
16     } else {
17         nombresArr[j] = String(nametxt.text); //guarda el nombre en el array
18         nametxt.text = "";
19         j++; //pasa a la siguiente posición del contador
20         advice.visible = false;
21         //cuando llega a tres jugadores, se deshabilitan los botones
22         //y el textfield para ingresar nombres
23         //y se habilitan los botones y el texto de "Listo"
24         if (j == 3) {
25             savebtn.enabled = false;
26             nametxt.visible = false;
27             readybtn.visible = true;
28             readytxt.visible = true;
29         }
30     }
31 }
32 savebtn.addEventListener(MouseEvent.CLICK, guardarnames);
33

```

Línea 11-32: función utilizada para almacenar los nombres de los jugadores que se van ingresando en el TextField junto a "Ingresa nombre del jugador:". Se comienza validando que el cuadro de texto no esté en blanco, y si no está en blanco se guarda la cadena ingresada en el array "nombresArr", se limpia el cuadro de texto y se aumenta el contador que se declaró en el frame anterior (j). Cuando se haya terminado de ingresar el nombre de los tres jugadores, el botón de guardar se deshabilitará y el botón de listo se habilitará.

```

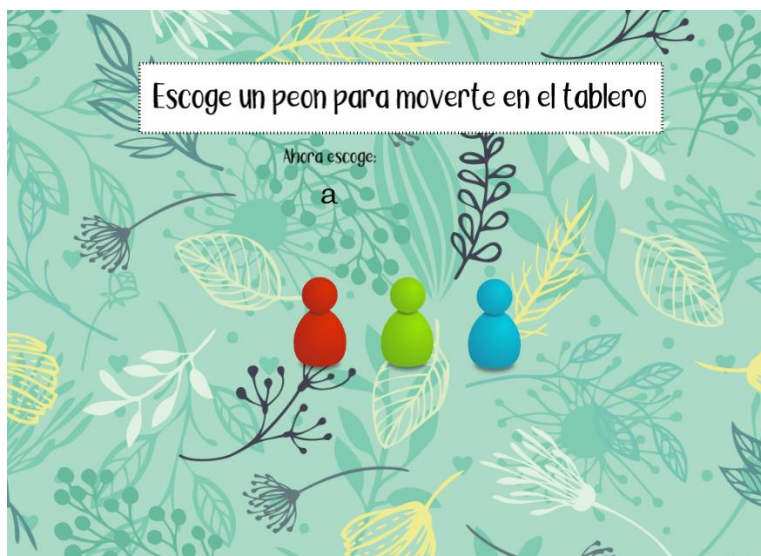
35 //Función para imprimir nombres de jugadores
36 function listo00(event: MouseEvent): void {
37     player1.text = String(nombresArr[0]); //se imprime junto a "Jugador 1"
38     player2.text = String(nombresArr[1]); //se imprime junto a "Jugador 2"
39     player3.text = String(nombresArr[2]); //se imprime junto a "Jugador 3"
40     advice.visible = false;
41     readybtn.enabled = false;
42     choosebtn.visible = true; //botón para pasar a escoger peones
43     nametxt.visible = false;
44 }
45 readybtn.addEventListener(MouseEvent.CLICK, listo00);
46 readytxt.addEventListener(MouseEvent.CLICK, listo00);
47

```

Línea 36-46: función utilizada para imprimir los nombres de los jugadores. Cuando se presiona el botón "Listo", se imprimen en los textfield ubicados a la izquierda los

nombres que se acaban de ingresar, y aparece el botón que nos llevará al siguiente frame a escoger los peones.

2.3 Selección de peones



Se muestran los peones disponibles, y comienza a escoger el jugador que se ingresó primero en los nombres. Al darle clic a un peón, este desaparece (ya que fue seleccionado).



Cuando se selecciona el último peón, aparece la pantalla de arriba, la cuál nos muestra la relación de jugador-peón. Ojo: no significa que ese sea el orden de los turnos. El botón continuar nos manda al frame de las reglas.

```

1      import flash.events.MouseEvent;
2
3      var red: int = 0; //variable del peón rojo
4      var green: int = 0; //variable del peón verde
5      var blue: int = 0; //variable del peón azul
6      var e: int = 0; //contador de nombres de array
7      //textfield que muestra el nombre del player que va a escoger peón
8      playerescoge.text = String(nombresArr[e]);
9      reglasbtn.visible = false; //botón de reglas
10     contxt.visible = false; //texto del botón de reglas
11     turnostxt.visible = false; //texto de Turno
12     turnoltxt.visible = false; //textfield del primer turno
13     turno2txt.visible = false; //textfield del segundo turno
14     turno3txt.visible = false; //textfield del tercer turno
15     rojin.visible = false; //minipeón rojo
16     verdin.visible = false; //minipeón verde
17     azulin.visible = false; //minipeón azul
18

```

```

19     //función del peón rojo (como botón)
20     function rojito(event: MouseEvent): void {
21         //la variable de este peón toma el valor de la posición que tiene
22         //dentro del array el jugador que lo escogió
23         red = e;
24         projo.visible = false;
25         e++; //e aumenta para pasar al siguiente jugador
26         playerescoge.text = String(nombresArr[e]);
27         //si ya escogieron los 3 jugadores, aparece la relación
28         //de jugador-peón
29         if (e == nombresArr.length) {
30             projo.enabled = false;
31             pverde.enabled = false;
32             pazul.enabled = false;
33             reglasbtn.visible = true;
34             playerescoge.visible = false;
35             contxt.visible = true;
36             title.visible = false;
37             choosetxt.visible = false;
38             turnostxt.visible = true;
39             turnoltxt.visible = true;
40             turno2txt.visible = true;
41             turno3txt.visible = true;
42             rojin.visible = true;
43             verdin.visible = true;
44             azulin.visible = true;
45             turnoltxt.text = String(nombresArr[red]);
46             turno2txt.text = String(nombresArr[green]);
47             turno3txt.text = String(nombresArr[blue]);
48         }
49     }
50     projo.addEventListener(MouseEvent.CLICK, rojito);
51

```

Línea 20-50: función que se activa al seleccionar el peón rojo. Este peón contiene una variable llamada "red" que toma el valor de la posición del Array del jugador que lo seleccionó. El peón desaparece y se aumenta la variable "e" para pasar al siguiente jugador. Si el último jugador es el que selecciona el peón rojo, aparece la lista de la relación jugador – peón.

2.4 Reglas

Reglas

- Los jugadores comienzan con una ficha y se turnan para lanzar un dado que les indicará la cantidad de casillas que deben avanzar.
- Las fichas se mueven según la numeración del tablero, en sentido ascendente.
 - Si al finalizar un movimiento un jugador cae en una casilla donde comienza una escalera, sube por ella hasta la casilla donde ésta termina.
 - Si, por el contrario, cae en una en donde comienza la cabeza de una serpiente, desciende por ésta hasta la casilla donde finaliza su cola.
- Si un jugador obtiene un 6 podrá mover y tirar nuevamente el dado.
- Si un jugador obtiene tres 6 consecutivos, deberá regresar a la casilla inicial y no podrá mover su ficha hasta obtener nuevamente un 6.
- El jugador que logra llegar a la casilla final es el ganador.

Jugar

En este frame se presentan las reglas que, debido a falta de lógica al momento de realizar el código, no se podrán cumplir.

2.5 Juego

Siguiente turno
b

Turno actual
a

Avanza
5

Tirar dado

Tiempo
0 m. 7 s.

El tablero consta de 100 casillas, 4 serpientes y 4 escaleras. Se muestra el turno del siguiente jugador, el turno del jugador que acaba de hacer clic a Tirar dado, el número que arrojó el dado, el cuál se refiere al número de casillas que avanza, y el tiempo de duración de todo el juego.


```

1  import fl.transitions.Tween;
2  import fl.transitions.easing.*;
3  import fl.transitions.TweenEvent;
4  import flash.events.Event;
5  import flash.events.TimerEvent;
6  import flash.utils.Timer;
7
8  terminotxt.visible=false; //texto de que el juego terminó
9  stoptxt.visible = false; //botón en forma de texto invisible (el del dado)
10 var numerosran: int; //variable para los números random del dado
11 var contr: int = -1; //contador de casilla del peón rojo
12 var contrv: int = -1; //contador de casilla del peón verde
13 var conta: int = -1; //contador de casilla del peón azul
14 var rr: int = 0; //variable para el peón rojo y su movimiento en serpientes y escaleras
15 var vv: int = 0; //variable para el peón verde y su movimiento en serpientes y escaleras
16 var aa: int = 0; //variable para el peón azul y su movimiento en serpientes y escaleras
17 //Array que contiene todas las casillas
18 var casillas: Array = [c1, c2, c3, c4, c5, c6, c7, c8, c9, c10,
19 c11, c12, c13, c14, c15, c16, c17, c18, c19, c20, c21, c22,
20 c23, c24, c25, c26, c27, c28, c29, c30, c31, c32, c33, c34,
21 c35, c36, c37, c38, c39, c40, c41, c42, c43, c44, c45, c46,
22 c47, c48, c49, c50, c51, c52, c53, c54, c55, c56, c57, c58,
23 c59, c60, c61, c62, c63, c64, c65, c66, c67, c68, c69, c70,
24 c71, c72, c73, c74, c75, c76, c77, c78, c79, c80, c81, c82,
25 c83, c84, c85, c86, c87, c88, c89, c90, c91, c92, c93, c94,
26 c95, c96, c97, c98, c99, c100
27 ];

```

Línea 10-27: se declararon variables necesarias utilizadas a lo largo del juego, así como un array que contiene las casillas del tablero.

```

39 //Función del dado
40 function dados(event: MouseEvent): void {
41     //cálculo del número que saldrá en el dado
42     var dado = Math.floor(Math.random() * 6) + 1;
43     //aparece el nombre del dueño del peón que se está moviendo
44     turnotxt.text = String(nombresArr[e]);
45     //si en el text de turno siguiente está el nombre del último
46     //jugador, al presionar el botón se regresa al nombre del primer jugador
47     if (turnosigtxt.text == nombresArr[2]) {
48         turnosigtxt.text = String(nombresArr[0]);
49     } else { //sino, se le suma 1 al contador para que aparezca el siguiente inmediato
50         turnosigtxt.text = String(nombresArr[e + 1]);
51     }
52     stoptxt.visible = true; //se habilita el botón como texto de tirar dado
53     starttxt.visible = false; //se deshabilita el botón como texto de "Comenzar"
54     dadotxt.text = String(dado); //se imprime el número de casillas que se avanzará
55     timersin.start(); //timer entre cada turno de jugador
56     escaleri.start(); //timer de escaleras
57     viborin.start(); //timer de serpientes
58     stopbtn.visible = false; //desaparece el botón de "Tirar dado"
59     stoptxt.visible = false; //desaparece el texto como botón de "Tirar dado"
60     e++; //se suma al contador para pasar al siguiente jugador
61     //Si es el turno del jugador con el peón rojo, se cumple lo siguiente
62     if (turnotxt.text == nombresArr[red]) {
63         contr = contr + dado; //el contador de casilla toma su valor actual más el que lanzó el dad
64         p = red; //p se utiliza en los timers de escaleras y serpientes para aplicar los tweens
65         rr = rr + dado; //toma el mismo valor que el contador de casilla para saber la posición a 1
66         //tweens de movimiento
67         equisr = new Tween(projo, "x", Regular.easeInOut, projo.x, casillas[contr].x, 20);
68         yer = new Tween(projo, "y", Regular.easeInOut, projo.y, casillas[contr].y, 20);
69         redone(); //función del peón rojo
70         //Si el jugador llega a 100, termina el juego
71         if (contr >= 100) {
72             gudbai.start();
73         }
74     }
75     if (turnotxt.text == nombresArr[green]) {
76         contrv = contrv + dado;
77         p = green;
78         vv = vv + dado;
79         equisv = new Tween(pverde, "x", Regular.easeInOut, pverde.x, casillas[contrv].x, 20);
80         yev = new Tween(pverde, "y", Regular.easeInOut, pverde.y, casillas[contrv].y, 20);
81         greenone();
82         if (contrv >= 100) {
83             gudbai.start();
84         }
85     }
86     if (turnotxt.text == nombresArr[blue]) {
87         conta = conta + dado;
88         p = blue;
89         aa = aa + dado;
90         equisa = new Tween(pazul, "x", Regular.easeInOut, pazul.x, casillas[conta].x, 20);
91         yea = new Tween(pazul, "y", Regular.easeInOut, pazul.y, casillas[conta].y, 20);
92         blueone();
93         if (conta >= 100) {
94             gudbai.start();
95         }
96     }
97     //cuando el contador sea mayor o igual a la longitud del array
98     //el contador regresa a 0
99     if (e >= nombresArr.length) {
100         e = 0;
101     }
102 }
103
104

```

Línea 40-107: función que se activa al hacer clic al botón "Tirar dado". Dentro de esta función se realiza el cálculo del número al azar que tomará el valor del dado. En las líneas siguientes se establecen unas condiciones para determinar si el jugador actual es el del peón rojo, el peón verde o el peón azul. Dentro de cada condición se encuentra el contador que refleja la casilla actual del peón, el valor p que toma la posición en el array de los nombres, así como los tweens del peón al moverse sobre el tablero. Si la posición del peón es 100, el timer para finalizar el juego termina.

```

188 //Timer juego
189 var seg: int = 0;
190 var cont1: int;
191 var timer: Timer = new Timer(1000, cont1++);
192 var min: int = 0;
193 timer.start();
194 timer.addEventListener(TimerEvent.TIMER, tiempo);
195
196 //Función del tiempo total del juego
197 function tiempo(tiempoevent: TimerEvent): void {
198     seg++;
199     if (seg > 59) {
200         min++;
201         seg = 0;
202     }
203     tiempotxt.text = min + " m. " + seg + " s.";
204 }

```

Línea 189-204: timer general del juego.

```

207 //Timer entre cada jugador
208 var jejox: int = 0;
209 var cont0: int;
210 var timersin: Timer = new Timer(1000, cont0++);
211 timersin.addEventListener(TimerEvent.TIMER, playerstime);
212
213 //Función para esperar 2 segundos entre cada turno
214 function playerstime(tiempoevent: TimerEvent): void {
215     jejox++;
216     if (jejox == 2) {
217         stopbtn.visible = true;
218         stoptxt.visible = true;
219         timersin.stop();
220         jejox = 0;
221         viborin.stop();
222         escaleri.stop();
223     }
224 }

```

Línea 208-224: timer para esperar entre los turnos de cada jugador.

```

226 //Timer al finalizar el juego
227 var fin: int = 0;
228 var cont6: int;
229 var gudbai: Timer = new Timer(1000, cont6++);
230 gudbai.addEventListener(TimerEvent.TIMER, findelgame);
231 //Función al finalizar el juego
232 function findelgame(tiempoevent:TimerEvent):void{
233     fin++;
234     terminotxt.visible=true;
235     var mensaje:Tween = new Tween (terminotxt, "x", Bounce.easeOut, -200, 137, 2, true);
236     stopbtn.enabled=false;
237     stoptxt.enabled=false;
238     timersin.stop();
239     timer.stop();
240     if (fin == 3){
241         gotoAndStop(6);
242         gudbai.stop();
243     }
244 }
245

```

Línea 227-244: timer para finalizar el juego. Este timer detiene el timer general y el timer entre cada jugador, y muestra un mensaje diciendo que el juego terminó. Cuando su duración es de 3 segundos, se detiene el timer y pasa al frame de los resultados.

```

248 //Función del peón rojo
249 function redone() {
250     for (var i: int = 0; i <= contr; i++) {
251         projo.x = casillas[i].x;
252         projo.y = casillas[i].y;
253         if (e == nombresArr.length) {
254             e = 0;
255         }
256         p = 0;
257     }
258     trace("Casilla actual", i);
259 }
260
261 //Función del peón verde
262 function greenone() {
263     for (var j: int = 0; j <= contrv; j++) {
264         pverde.x = casillas[j].x;
265         pverde.y = casillas[j].y;
266         if (e == nombresArr.length) {
267             e = 0;
268         }
269         p = 0;
270     }
271     trace("Casilla actual", j);
272 }
273
274 //Función del peón azul
275 function blueone() {
276     for (var k: int = 0; k <= conta; k++) {
277         pazul.x = casillas[k].x;
278     }
279 }
280

```

Línea 249-288: funciones para cada peón. "redone" pertenece al peón rojo, "greenone" pertenece al peón verde, y "blueone" pertenece al peón azul. Las tres funciones contienen lo mismo, simplemente se cambian los nombres de los peones. Dentro de ellas se encuentra un ciclo for, el cuál le asigna la posición al peón en la misma posición que la casilla a la que se tiene que mover. Estas funciones se activan en los ifs respectivos en las líneas 40-107.

2.6 Resultados



En esta pantalla final, se imprime el tiempo total del juego, los lugares (del mejor al peor) y el número de casilla en el que terminaron. Cuenta con un botón de exportar el cuál exporta a .txt.

```

1  import flash.net.FileReference;
2  import flash.events.MouseEvent;
3
4  //variable para exportar a txt
5  var txt: FileReference = new FileReference();
6  //array que se va a imprimir
7  var places: Array = new Array();
8
9  //mostrar el tiempo final
10 timetxt.text= min + " m. " + seg + " s.";
11
12 //Las condiciones siguientes emplean lo que son
13 //comparaciones lógicas para saber qué jugador
14 //fue el que llegó a la casilla 100 y quien
15 //le seguía por detras, se guardan los datos
16 //en orden dentro del array
17 if ((contr > contv) && (contr > conta)){
18   cas1.text=String(contr);
19   firstp.text=String(nombresArr[red]);
20   if (contv > conta){
21     cas2.text=String(contv);
22     secondp.text=String(nombresArr[green]);
23     cas3.text=String(conta);
24     thirdp.text=String(nombresArr[blue]);
25     places.push(nombresArr[red]);
26     places.push(contr);
27     places.push(nombresArr[green]);
28     places.push(contv);
29     places.push(nombresArr[blue]);
30     places.push(conta);
31   }else{
32     cas2.text=String(conta);
33     secondp.text=String(nombresArr[blue]);
34     cas3.text=String(contv);
35
36     places.push(nombresArr[red]);
37     places.push(contr);
38   }
39 }
40
41 //Función que imprime en formato pdf el array con los resultados
42 function exportar (event:MouseEvent):void{
43   txt.save(places, "Resultado.txt");
44 }
45 exportarbtn.addEventListener(MouseEvent.CLICK, exportar);
  
```

Línea 17-99: dentro de estas líneas se encuentran una serie de condiciones para validar qué jugador fue el que ganó, quién quedó en segundo lugar y quién en tercer lugar. Dependiendo de la condición que se cumpla, se imprimirán respectivamente en sus cuadros de texto tanto su nombre como el número de la casilla en la que terminaron, pero en orden del mejor al peor.

Línea 102-105: función para exportar a txt.