

Manual Técnico

MADAGASCAR



Integrantes:

- Lumbreras Ruíz Jesús Alberto Matrícula: 1730049
- Carrizales Piña Claudia Lizbeth Matrícula: 1730048
- Córdova Alanís Cruz Alejandro Matrícula: 1730516

Maestro: Mario Humberto Rodríguez Chávez

Materia: Herramientas Multimedia

ÍNDICE

- [Introducción](#).....3
- [Desarrollo](#).....4
 - [Frame1](#).....5
 - [Frame2](#).....6
 - [Frame3](#).....8
 - [Frame4](#).....9
 - [Frame5](#).....11
 - [Frame9](#).....15
 - [Frame10](#).....16
- [Conclusiones](#).....17
 - [Alberto Ruiz](#).....18
 - [Cruz Alanís](#).....19
 - [Claudia Carrizales](#).....20

Introducción

En este juego que es de Gravedad, trata sobre la película de Madagascar hablaremos paso a paso de cómo fue la creación para llegar al objetivo del juego. Paso a paso con capturas de lo que hace cada parte del código y en cómo funciona.

Este juego tiene algo peculiar ya que te da la opción de que elijas el personaje con el que quieras jugar de la película en este caso tenemos 4 opciones (Gloria, Marty, Rey Julien y Malman), solo es de dos jugadores este juego y el jugador número 2 ya no podrá elegir el personaje que eligió el jugador número uno y al final los pasos que seguimos para que mostrará el jugador ganador (en este caso el nombre del jugador) y todo explicaremos a continuación.



Desarrollo



Fotograma 1

En este fotograma lo que hicimos fue definir nuestras variables que utilizaremos en el juego y nuestros arrays que vamos a ocupar para guardar el nombre de los dos jugadores, los precipicios saltados, el tiempo transcurrido y la distancia de igual manera.

Las librerías que vamos a utilizar y nuestra función de botón que se llama “ingresar” para que nos lleve al fotograma dos que es donde se llenaran los nombres de los dos jugadores y que a continuación se va a mostrar.



```
acciones:11
1 //NUESTRAS LIBRERÍAS PARA NUESTRO MOUSE EVENTS Y NUESTRO TWEENS
2 import flash.events.MouseEvent;
3 import fl.transitions.Tween;
4 import fl.transitions.easing.*;
5 import fl.transitions.TweenEvent
6
7 //NUESTRAS VARIABLES QUE UTILIZAREMOS EN EL JUEGO y nuestros arrays
8 var jugador1:String;
9 var jugador2:String;
10 var nombresJugadores: Array = new Array();
11 var distanciaJugadores: Array = new Array();
12 var tiempoJugadores: Array = new Array();
13 var precipiciosJugadores: Array = new Array();
14 var i:int=0; //NUESTRO CONTADOR QUE SERÁ PARA EL NUMERO QUE LE CORRESPONDA CADA NOMBRE EN EL ARRAY
15
16 //VARIABLE PARA SABER CON QUE PERSONAJE SE JUGÓ
17 var Personajes:int=0;
18 //nuestro tween para nuestro nombre de la universidad Y nuestro logo
19 var bounceEaseIn:Tween =new Tween (upv_mc, "x", Bounce.easeIn, 750, 55, 2, true);
20 var bounceEaseInOut:Tween =new Tween (logo_mc, "y", Bounce.easeInOut, 500, 56, 2, true);
21 //NUESTRA FUNCIÓN PARA EL BOTÓN DE INGRESAR DE NUESTRA PORTADA
22 function ingresar (e:MouseEvent):void{
23     gotoAndStop(2);//PARA IRNOS AL FRAME 2
24 }
25 ingresar_btn.addEventListener(MouseEvent.CLICK, ingresar);
26
27 //PARA QUE SE DETENGA
28 stop();
```

Fotograma 2

```
1 //NUESTRAS LIBRERÍAS QUE UTILIZAREMOS
2 import flash.events.MouseEvent;
3 //AL INICIO PONEMOS NUESTRO BOTÓN EL DE GUARDAR EL PRIMER NOMBRE SOLO EN VISIBLE Y EL EL DE GUARDAR EL SEGUNDO
4 //NOMBRE Y EL DE EMPEZAR EL INVISIBLE
5 guardarN1_btn.visible = true;
6 guardarN2_btn.visible = false;
7 empezar_btn.visible = false;
8 //NUESTRA FUNCIÓN PARA GUARDAR EL PRIMER NOMBRE
9 function guardarNombre1(event: MouseEvent): void {
10     jugador1 = String(nombreJ1_txt.text); //para que sea puro texto el que se escriba como nombre en el text in
11     if ((Number(jugador1)) || (jugador1 == "")) { //validar que lo que ingrese en el texto sean letras y no num
12         aviso_txt.text = "SÓLO LETRAS"; //si lo que escribió fueron números o vacío nos mostrará el aviso
13         nombreJ1_txt.text = "";
14     } else { //si no, si lo que escribió fue correcto se guarda en el array en su respectiva posición
15         nombresJugadores[i] = jugador1;
16         trace(nombresJugadores[i]); //para que nos imprima el nombre en la salida
17         nombreJ1_txt.text = " ";
18         //SE PONDRAN NUESTRAS CAJAS DE TEXTO Y NUESTRO BOTON EN INVISIBLE PARA QUE SIGA EL REGISTRO DEL JUGADOR
19         guardarN1_btn.visible = false;
20         nombreJ1_txt.visible = false;
21         NombreJ1_txt.visible = false;
22         aviso_txt.visible = false; //para que el aviso si se equivoco desaparezca
23         //SE PONE VISIBLE EL BOTON PARA EL REGISTRO DEL JUGADOR 2
24         guardarN2_btn.visible = true;
25     }
26 }
27 guardarN1_btn.addEventListener(MouseEvent.CLICK, guardarNombre1);
28
29 //FUNCION PARA GUARDAR EL JUGADOR 2
30 function guardarNombre2(event: MouseEvent): void {
31     jugador2 = String(nombreJ2_txt.text); //para que sea puro texto el que se escriba como nombre
32     if ((Number(jugador2)) || (jugador2 == "")) { //validar que lo que ingrese en el texto sean letras y no num
33         //si se escribieron numeros o se dejó vacío se manda un aviso
34         avisol_txt.text = "SÓLO LETRAS";
35         nombreJ2_txt.text = "";
36     } else { //si no, si lo que escribió fue correcto se guarda en el array en su respectiva posición
37         nombresJugadores[i + 1] = jugador2;
38         trace(nombresJugadores[i + 1]); //para que imprima el nombre en la salida
39         nombreJ2_txt.text = " ";
40         //SI TODO FUE CORRECTO ENTONCES SE PONE EN INVISIBLE LOS TEXTOS Y EL BOTON DEL REGISTRO 2
41         guardarN2_btn.visible = false;
42         nombreJ2_txt.visible = false;
43         NombreJ2_txt.visible = false;
44         avisol_txt.text = " ";
45         //SE PONE EN VISIBLE EL BOTON DE EMPEZAR PARA PODER IR AL FRAME DE LAS INSTRUCCIONES Y INICIAR EL JUEGO
46         empezar_btn.visible = true;
47     }
48 }
49 guardarN2_btn.addEventListener(MouseEvent.CLICK, guardarNombre2);
50
51 //funcion para el boton de empezar, se va al fotograma de las instrucciones
52 function empezar(event: MouseEvent): void {
53     gotoAndStop(3); //fotograma en donde tenemos nuestras instrucciones
54 }
55 empezar_btn.addEventListener(MouseEvent.CLICK, empezar);
```

Primero agregamos nuestras librerías que íbamos a ocupar, pusimos solo nuestro botón el de ingresar el primero nombre en visible y el de empezar y guardar 2 en invisibles, creamos nuestras funciones que la primera es para agregar el primer nombre y validamos que lo que haya ingresado el usuario no haya sido números o haya dejado el espacio en blanco, se desaparecerá el botón hasta que el nombre que haya ingresado sean solo letras de cas contrario se mostrará un aviso que dice “solo letras” y este se guardará en la posición del array que definimos en el primer fotograma, y este botón se pondrá invisible para posterior poner el de guardar2 en visible para que ingrese el nombre del jugador.

Igual validamos para que solo lo que haya ingresado sean solo letras, una vez hecho estos dos de nombres y que lo que haya ingresado fueron solo letras se nos pondrán en invisibles los dos botones, los dos avisos, el de ingresar nombre de tal jugador y nuestros text inputs donde ingresará los nombres, para posterior se nos ponga visible solo el botón de “EMPEZAR”.

Nuestra función de abajo es del botón de EMPEZAR este lo que hará es mandarnos a las instrucciones del fotograma 3.

```
28
29 //FUNCION PARA GUARDAR EL JUGADOR 2
30 function guardarNombre2(event: MouseEvent): void {
31     jugador2 = String(nombreJ2_txt.text); //para que sea puro texto el que se escriba como nombre
32     if ((Number(jugador2)) || (jugador2 == "")) { //validar que lo que ingrese en el texto sean letras y no num
33         //si se escribieron numeros o se dejó vacío se manda un aviso
34         avisol_txt.text = "SÓLO LETRAS";
35         nombreJ2_txt.text = "";
36     } else { //si no, si lo que escribió fue correcto se guarda en el array en su respectiva posición
37         nombresJugadores[i + 1] = jugador2;
38         trace(nombresJugadores[i + 1]); //para que imprima el nombre en la salida
39         nombreJ2_txt.text = " ";
40         //SI TODO FUE CORRECTO ENTONCES SE PONE EN INVISIBLE LOS TEXTOS Y EL BOTON DEL REGISTRO 2
41         guardarN2_btn.visible = false;
42         nombreJ2_txt.visible = false;
43         NombreJ2_txt.visible = false;
44         avisol_txt.text = " ";
45         //SE PONE EN VISIBLE EL BOTON DE EMPEZAR PARA PODER IR AL FRAME DE LAS INSTRUCCIONES Y INICIAR EL JUEGO
46         empezar_btn.visible = true;
47     }
48 }
49 guardarN2_btn.addEventListener(MouseEvent.CLICK, guardarNombre2);
50
51 //funcion para el boton de empezar, se va al fotograma de las instrucciones
52 function empezar(event: MouseEvent): void {
53     gotoAndStop(3); //fotograma en donde tenemos nuestras instrucciones
54 }
55 empezar_btn.addEventListener(MouseEvent.CLICK, empezar);
```


I

Es así como se ve nuestro fotograma 2 para ingresar los dos nombres de los dos jugadores.



The screenshot shows a game interface with a background of a snowy, icy landscape. At the top, the text **PRIMER JUGADOR** is displayed. Below it is a white rectangular input field for the first player's name. Underneath the input field is a dashed rectangular box, and below that is a button labeled **GUARDAR**. In the center, the text **SEGUNDO JUGADOR** is displayed. Below it is another white rectangular input field for the second player's name. Underneath this input field is a button labeled **¡ EMPEZAR !**. Below the **¡ EMPEZAR !** button is another dashed rectangular box, and at the bottom is a button labeled **GUARDAR**. At the very bottom of the interface, four cartoon penguins are visible, looking towards the left.

Fotograma 3

Agregamos las librerías que vamos a utilizar, en este frame en acciones solo agregamos la función que es el que nos mandará al frame cuatro donde tendremos que elegir el personaje con el que queremos jugar.

En el lado izquierdo es el frame donde está el botón que le pusimos su función el botón llamado “next” y fue todo lo que hicimos en este frame.

INSTRUCCIONES

1. Elige el personaje con el que quieres jugar.
2. Con la barra espaciadora podrás elevarte al sig precipicio.
3. Si te caes pierdes.
4. Se contará tu distancia, tiempo y precipicios saltados.
5. Al final se mostrarán tus estadísticas y se mostrará cual de los dos jugadores es el ganador.
6. El siguiente jugador en turno ya no podrá elegir el mismo personaje para jugar.
7. Da next una vez leída las instrucciones.



NEXT

acciones :3

```
1 //NUESTRAS LIBRERÍAS QUE VAMOS A OCUPAR
2 import flash.events.MouseEvent;
3 import fl.transitions.Tween;
4 import fl.transitions.easing.*;
5 import fl.transitions.TweenEvent
6 //FUNCIÓN PARA IRSE AL SIGUIENTE FOTOGRAMA, FOTOGRAMA DE ELEGIR EL PERSONAJE
7 function next (event:MouseEvent):void{
8     gotoAndStop(4); //irnos al 4
9 }
10 next_btn.addEventListener(MouseEvent.CLICK, next);
```


Fotograma 4

```
acciones :4
1 import flash.events.MouseEvent;
2 import flash.utils.Timer;
3 //VARIABLES PARA EL TIMER PARA LOS FRAMES DE NUESTROS PERSONAJES, S
4 //TODOS LOS CONTADORES VUELVAN A INICIALIZARSE EN 0
5 var tmp: int = 0;
6 var cont1: int = 0;
7 var min: int = 0;
8 var timer: Timer = new Timer(1000, cont1++);
9
10 //VARIABLES QUE UTILIZAREMOS PARA EL CONTEO DE LA DISTANCIA RECORRI
11 var Distancia:int = 2000;
12 var mts: Number = 0;
13 var contDist: int = 0;
14 var Dist: Timer = new Timer(Distancia, contDist++);
15
16 //VARIABLES PARA EL SALTO
17 var vel:Number=6;
18 var no:Number=-1;
19 var key:Boolean=false;
20 var toque:Boolean=false;
21 var toque1:Boolean=false;
22 var toque2:Boolean=false;
23 var toque3:Boolean=false;
24 var posY:Number=0;
25 var salir:Boolean=false;
26 var par:Number=0;
27 var cont2:int=0;
28
29 //NUESTRAS VARIABLES QUE UTILIZAREMOS
30 var NumRandom: int;
31 var BanderaGloria: Boolean = false;
32 var BanderaMarty: Boolean = false;
33 var BanderaReyJulien: Boolean = false;
34 var BanderaMelman: Boolean = false;
35 var ContPrecipicios1: int = 0;
```

Agregamos las librerías que vamos a utilizar y definimos nuestras variables que vamos a utilizar para nuestro timer y nuestras variables que vamos a utilizar para el conteo de la distancia, así mismo nuestras variables que vamos a ocupar para el salto del personaje y definimos nuestras banderas que vamos a utilizar y esto se nos hará más fácil para que no nos marque errores y nuestra variable que definimos como número random como entero.

```
acciones :4
31 var BanderaGloria: Boolean = false;
32 var BanderaMarty: Boolean = false;
33 var BanderaReyJulien: Boolean = false;
34 var BanderaMelman: Boolean = false;
35 var ContPrecipicios1: int = 0;
36 var BanderaPerdiste: Boolean;
37 var BanderaPerdiste2: Boolean;
38 var BanderaPerdiste3: Boolean;
39 var BanderaPerdiste4: Boolean;
40 var Vel:int=5;
41
42 //SE VALIDA PARA SABER CON QUE PERSONAJE SE JUGÓ Y SE PONE EN INVISIBLE
43 if (Personajes==1){
44     gloria_btn.visible=false;
45 }
46 if (Personajes==2){
47     marty_btn.visible=false;
48 }
49 if (Personajes==3){
50     reyJulien_btn.visible=false;
51 }
52 if (Personajes==4){
53     melman_btn.visible=false;
54 }
55
56 //IRNOS AL FRAME PARA JUGAR CON EL HIPOPOTAMO
57 //le asignamos una posición o más bien un número para que al haber ele
58 //personaje que eligió el primero este desaparezca
59 function gloria(event:MouseEvent):void{
60     gotoAndStop(5);
61     Personajes=1;//le asignamos esta posición
62 }
63 gloria_btn.addEventListener(MouseEvent.CLICK, gloria);
64 //IRNOS AL FRAME PARA JUGAR CON LA CEBRA
65 function marty(event:MouseEvent):void{
```

Se válida para saber qué personaje eligió el primer jugador y que este se ponga en invisible para que el segundo jugador ya no lo pueda elegir el mismo y que tenga la oportunidad de elegir otro. Para eso se hace lo siguiente...

```

53     melman_btn.visible=false;
54 }
55
56 //IRNOS AL FRAME PARA JUGAR CON EL HIPOPOTAMO
57 //le asignamos una posición o más bien un número para que al haber elegido
58 //personaje que eligió el primero este desaparezca
59 function gloria(event:MouseEvent):void{
60     gotoAndStop(5);
61     Personajes=1;//le asignamos esta posición
62 }
63 gloria_btn.addEventListener(MouseEvent.CLICK, gloria);
64 //IRNOS AL FRAME PARA JUGAR CON LA CEBRA
65 function marty(event:MouseEvent):void{
66     gotoAndStop(6);
67     Personajes=2;//le asignamos esta posición
68 }
69 marty_btn.addEventListener(MouseEvent.CLICK, marty);
70 //IRNOS AL FRAME PARA JUGAR CON REY JULIEN
71 function reyJulien(event:MouseEvent):void{
72     gotoAndStop(7);
73     Personajes=3;//le asignamos esta posición
74 }
75 reyJulien_btn.addEventListener(MouseEvent.CLICK, reyJulien);
76 //IRNOS AL FRAME PARA JUGAR CON LA JIRAFÁ
77 function melman(event:MouseEvent):void{
78     gotoAndStop(8);
79     Personajes=4;//le asignamos esta posición
80 }
81 melman_btn.addEventListener(MouseEvent.CLICK, melman);
82 //FUNCIÓN PARA IR DE NUEVO A LEER LAS INSTRUCCIONES
83 function atras (event:MouseEvent):void{
84     gotoAndStop(3);//IRNOS AL TRES QUE ES DODE ESTÁN
85 }
86 atrasInstrucciones_btn.addEventListener(MouseEvent.CLICK, atras);

```

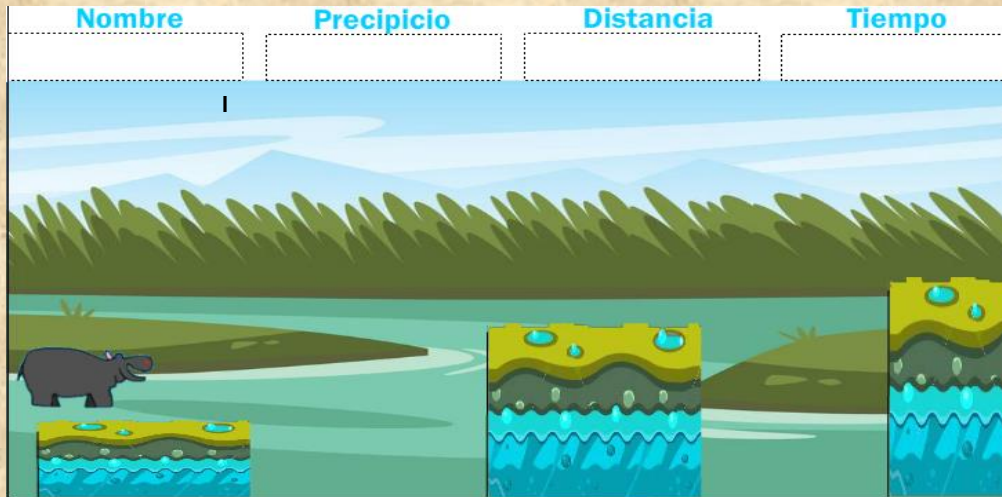


Así se ve nuestro fotograma con los personajes respectivos que ya mencionamos y nuestro botón de atrás.

Ingresamos nuestras cuatro funciones estas son para cada uno de los personajes en este caso tenemos a Gloria, Marty, Rey Julien y Melman a estos dentro de la función de cada uno se le agrega “personaje=1”, “personaje=2” y así sucesivamente y esto hace que el anterior se cumpla y se ponga en invisible el respectivo que eligió el jugador anterior.

Y nuestra función del botón “atrás” que nos hará regresarnos al fotograma de las instrucciones.

Fotograma 5



```

acciones :5
1  //NUESTRAS LIBRERÍAS
2  import flash.events.TimerEvent;
3  BanderaPerdiste = false;
4  //LE ASIGNAREMOS A NUESTRO TEXTO DEL NOMBRE DEL JUGADOR
5  gloriaNombre_txt.text = nombresJugadores[i];
6  //ARRANCAREMOS EL TIMER AL INICIAR EL JUEGO
7  timer.start();
8  Dist.start();
9  function tiempol(tiempoevent: TimerEvent): void {
10     tmp++; //ESTO ES PARA QUE SE VAYAN INCREMENTANDO LOS SEGUNDOS
11     if (tmp > 59) { //SI ESTO SE CUMPLE CUANDO NUESTRO CONTADOR DE SEGUNDOS LLEGUE A 59 ENTO
12         tmp = 0; //SE INICIALIZA EL CONTADOR DE SEGUNDOS EN 0
13         min++; //SE INCREMENTA EN 1 EL CONTADOR DE MINUTOS
14     }
15     if (tmp < 10) { //SI LOS SEGUNDOS SON MENOR A 10 ENTONCES
16         tiempol_txt.text = min + ":" + "0" + tmp + " s."; //SE LE AGREGA UN 0 ANTES DE EL CO
17     } else {
18         //SI NO, ENTONCES SOLAMENTE SE MUESTRAN LOS MINUTOS Y SEGUNDOS
19         tiempol_txt.text = min + ":" + tmp + " s.";
20     }
21     //CONDICION QUE HACE QUE LA VELOCIDAD AUMENTE CADA 30 O 59 SEGUNDOS
22     if ((tmp == 30) || (tmp == 59)) {
23         Vel = Vel + 2;
24     }
25 }
26 timer.addEventListener(TimerEvent.TIMER, tiempol);
27
28 //FUNCION PARA DETECTAR QUE SE PRESIONÓ EL BOTÓN
29 stage.addEventListener(KeyboardEvent.KEY_DOWN, mover1);
30
31 function mover1(event: KeyboardEvent) {
32     if (event.keyCode == 32) {
33         //SE EJECUTARÁ SOLO CUANDO EL MONO ESTÉ EN CONTACTO
34         if (toque1 == true || toque2 == true || toque3 == true) {
35             key = true;

```

Aquí al principio de nuestro frame corremos nuestros 2 timer uno para medir los segundos y el otro para la distancia, colocamos nuestra función del timer la cual lo que hace es mostrar los segundos al entrar al frame y al llegar a 59 convertir a minutos y volver a contar los segundos y así sucesivamente hasta que se pierda, también se condiciona para que cada 30 segundos la velocidad de nuestras plataformas incremente y sea más difícil para el usuario.


```

acciones :5
34         if (toque1 == true || toque2 == true || toque3 == true) {
35             key = true;
36         }
37     }
38 }
39
40 //FUNCION PARA EL CONTEO DE LA DISTANCIA RECORRIDA
41 function dist1(tiempoevent: TimerEvent): void {
42     mts = mts + 1.5;
43     distancial_txt.text = mts + " Metros";
44 }
45 Dist.addEventListener(TimerEvent.TIMER, dist1);
46 //FUNCION PARA EL MOVIMIENTO DE LAS PLATAFORMAS
47 function MoverPiedras1(event: Event): void {
48     if (BanderaPerdiste == false) {
49         var vel: Number = 6;
50         if (hipo_mc.y > 450) {
51             distanciaJugadores[i] = distancial_txt.text;
52             precipiciosJugadores[i] = PrecipiciosGloria_txt.text;
53             tiempoJugadores[i] = tiempo1_txt.text;
54             i++;
55             if (i == 2) {
56                 timer.stop();
57                 Dist.stop();
58                 gotoAndStop(9);
59                 BanderaPerdiste = true;
60                 salir = true;
61             } else {
62                 timer.stop();
63                 Dist.stop();
64                 gotoAndStop(10);
65                 BanderaPerdiste = true;
66                 salir = true;
67             }
68         }

```

Se tiene la función que se utiliza para contar la distancia, la cual va aumentando funcionando como un timer.

Tenemos nuestra función para que nuestras plataformas vayan aumentando y al principio condicionamos para que si nuestra bandera está en falso (aún no se pierde) entonces se coloca otra condición la cual es para cuando nuestro objeto sale del frame lo cual hace que nuestra distancia, tiempo y precipicios saltados se guarden en su respectivo array para después poder mostrarlo, al igual que se condiciona para saber si ya pasaron los 2 jugadores y si es así se va al frame de resultados y si no entonces se va al frame de elección de personajes para el siguiente jugador.

```

acciones :5
67     }
68 }
69 //VARIABLE PARA GRAVEDAD EN EL MONO
70 hipo_mc.y += vel;
71 //LAS ROCAS SE MUEVEN GENERANDO UN EFECTO DE QUE EL MONO AVANZA
72 piedraGloria1_mc.x -= Vel;
73 piedraGloria2_mc.x -= Vel;
74 piedraGloria3_mc.x -= Vel;
75 linea1.x -= Vel;
76 linea2.x -= Vel;
77 linea3.x -= Vel;
78 //ESTO EN CASO DE QUE LA TECLA ESPACIO ESTÉ OPRIMIDA
79 if (key == true) {
80     //AQUÍ SE GENERA EL SALTO, UNA PARABOLA
81     par += Vel;
82     hipo_mc.y = hipo_mc.y - ((-0.2 * par) ^ 2);
83     //CUÁNDO SALTE Y VUELVA A TOCAR UNA ROCA HACE QUE SE PONGA ANCIEN DE LA ROCA
84     if (hipo_mc.hitTestObject(piedraGloria1_mc) || hipo_mc.hitTestObject(piedraGloria2_mc) ||
85         hipo_mc.hitTestObject(piedraGloria3_mc)) {
86         cont2++;
87         hipo_mc.y -= (vel * 2);
88         posY = hipo_mc.y;
89         toque = true;
90         BanderaGloria = false;
91     }
92     //AQUÍ HACE QUE LA VARIABLE DE PULSAR TECLA SE HAGA FALSA
93     if (toque == true && cont2 == 2) {
94         key = false;
95     }
96 }
97 //PARA CUANDO LA TECLA ESPACIO NO SE OPRIMA
98 if (key == false) {
99     //SE RESTABLECE EL VALOR ORIGINAL PARA LA FUNCION DE SALTO
100     par = -130;
101     cont2 = 0;

```

Dentro de la función del ENTER_FRAME le dimos la gravedad al personaje con la variable vel. Después con la variable Vel hacemos que 3 plataformas y 3 líneas se muevan generando el efecto de que se mueven.

Creamos además una condición en caso de que la tecla espacio del teclado se haya presionado, aquí se implementó lo del salto del personaje, utilizamos una ecuación matemática para la para la parábola, y una condición para que cuando el personaje cayera se mantuviera en la plataforma. Con una bandera hicimos que la variable key se hiciera falsa.

```

acciones :5
97 //PARA CUANDO LA TECLA ESPACIO NO SE OPRIMA
98 if (key == false) {
99 //SE RESTABLECE EL VALOR ORIGINAL PARA LA FUNCION DE SALTO
100 par = -130;
101 cont2 = 0;
102 hipo_mc.y += vel;
103 //EN CASO DE QUE EL MONO TOQUE CON LA PIEDRA 1
104 if (hipo_mc.hitTestObject(piedraGloria1_mc) && salir == false) {
105 hipo_mc.y -= (vel * 2);
106 posY = hipo_mc.y;
107 toque1 = true;
108 if (BanderaGloria == false) {
109 PrecipiciosGloria_txt.text = String(ContPrecipicios1++);
110 BanderaGloria = true;
111 }
112 } else {
113 toque1 = false;
114 }
115 //EN CASO DE QUE EL MONO TOQUE CON LA PIEDRA 2
116 if (hipo_mc.hitTestObject(piedraGloria2_mc) && salir == false) {
117 hipo_mc.y -= vel * 2;
118 posY = hipo_mc.y;
119 toque2 = true;
120 if (BanderaGloria == false) {
121 PrecipiciosGloria_txt.text = String(ContPrecipicios1++);
122 BanderaGloria = true;
123 }
124 } else {
125 toque2 = false;
126 }
127 //EN CASO QUE EL MONO TOQUE CON LA PIEDRA 3
128 if (hipo_mc.hitTestObject(piedraGloria3_mc) && salir == false) {
129 hipo_mc.y -= vel * 2;
130 posY = hipo_mc.y;
131 toque3 = true;
132 if (BanderaGloria == false) {
133 PrecipiciosGloria_txt.text = String(ContPrecipicios1++);
134 BanderaGloria = true;
135 }
136 } else {
137 toque3 = false;
138 }
139 //EN CASO QUE EL MONO TOQUE CON LA LATERAL DE LA PIEDRA
140 if (hipo_mc.hitTestObject(linea1)) {
141 hipo_mc.x -= Vel;
142 hipo_mc.y += vel;
143 toque1 = false;
144 }
145 //EN CASO QUE EL MONO TOQUE CON LA LATERAL DE LA PIEDRA
146 if (hipo_mc.hitTestObject(linea2)) {
147 hipo_mc.x -= Vel;
148 hipo_mc.y += vel;
149 toque2 = false;
150 }
151 //EN CASO QUE EL MONO TOQUE CON LA LATERAL DE LA PIEDRA
152 if (hipo_mc.hitTestObject(linea3)) {
153 hipo_mc.x -= Vel;
154 hipo_mc.y += vel;
155 toque3 = false;
156 }
157 }
158 //ESTO ES PARA CUANDO LA PRIMER PIEDRA SALGA DEL BORDE
159 if ((piedraGloria1_mc.x + 170) < 0) {
160 //SE COLOCA LA PIEDRA EN EL BORDE DE LA DERECHA PARA QUE PUEDA VOLVER
161 piedraGloria1_mc.x = 800;

```

Cuando la variable key es falsa, es decir que ya se presionó la tecla espacio y saltó nuestro personaje, se detecta si el personaje está tocando con la plataforma y le mandamos a que se mantenga encima de está creando una oposición a la variable de gravedad, en este caso multiplicamos por 2 la gravedad y se la añadimos a nuestro personaje en sentido contrario a la gravedad, es decir hacia arriba.

Utilizamos una variable llamada toque para cada plataforma, la función de estas son para detectar si nuestro personaje está sobre una plataforma, para que solo pueda saltar cuando está sobre estás y no en el aire.

Además colocamos varias líneas enfrente de nuestras plataformas, y con el hitTestObject detectamos cuando el personaje toque la parte lateral de la plataforma, si pasa esto significa que el personaje no alcanzó a llegar a la otra plataforma por lo que cae y la plataforma lo empuja.

```

acciones:5
157 }
158 //ESTO ES PARA CUANDO LA PRIMER PIEDRA SALGA DEL BORDE
159 if ((piedraGlorial_mc.x + 170) < 0) {
160     //SE COLOCA LA PIEDRA EN EL BORDE DE LA DERECHA PARA QUE PUEDA VOLVER A SALIR
161     piedraGlorial_mc.x = 800;
162     //SE PONE MERO ABAJO EN EL EJE DE LAS Y, PARA POSTERIORMENTE COLOCARLE NUEVA ALTURA
163     piedraGlorial_mc.y = 450;
164     //NUMERO RANDOM PARA MODIFICAR LA ALTURA DE NUESTRA PIEDRA
165     NumRandom = (Math.random() * 200) + 100;
166     //SE LE PONE NUEVA ALTURA A NUESTRA PIEDRA
167     piedraGlorial_mc.height = NumRandom;
168     //Y SE PONE LA PIEDRA EN EL EJE DE LAS Y SEGUN SU ALTURA PARA NO DEJAR BORDES POR DEF
169     piedraGlorial_mc.y = 450 - NumRandom;
170     //PARA QUE LA LINEA SALGA
171     lineal.height = piedraGlorial_mc.height - 15;
172     lineal.y = piedraGlorial_mc.y + 15;
173     lineal.x = piedraGlorial_mc.x;
174 }
175 //ESTO ES PARA CUANDO LA SEGUNDA PIEDRA SALGA DEL BORDE
176 if ((piedraGloria2_mc.x + 170) < 0) {
177     //SE COLOCA LA PIEDRA EN EL BORDE DE LA DERECHA PARA QUE PUEDA VOLVER A SALIR
178     piedraGloria2_mc.x = 800;
179     //SE PONE MERO ABAJO EN EL EJE DE LAS Y, PARA POSTERIORMENTE COLOCARLE NUEVA ALTURA
180     piedraGloria2_mc.y = 450;
181     //NUMERO RANDOM PARA MODIFICAR LA ALTURA DE NUESTRA PIEDRA
182     NumRandom = (Math.random() * 200) + 100;
183     //SE LE PONE NUEVA ALTURA A NUESTRA PIEDRA
184     piedraGloria2_mc.height = NumRandom;
185     //Y SE PONE LA PIEDRA EN EL EJE DE LAS Y SEGUN SU ALTURA PARA NO DEJAR BORDES POR DEF
186     piedraGloria2_mc.y = 450 - NumRandom;
187     //PARA QUE LA LINEA SALGA
188     linea2.height = piedraGloria2_mc.height - 15;
189     linea2.y = piedraGloria2_mc.y + 15;
190     linea2.x = piedraGloria2_mc.x;
191 }

```

En esta condición le sumamos en x lo que tenga nuestra piedra de ancho para que cuando salga del escenario se coloca en la posición de 800 para que vuelva a salir al principio de nuestro frame y la ponemos mero abajo en Y porque vamos a asignarle un valor random de altura y después de esto se lo agregamos a nuestra plataforma y lo que hacemos es ponerle de altura el random y también le restamos a 450 su altura para que de esta forma no deje bordes por debajo de la plataforma y se vea mejor al igual modificamos las líneas que nos ayudan para los choques colocándoles también altura al igual que a las plataformas

```

}
//ESTO ES PARA CUANDO LA TERCERA PIEDRA SALGA DEL BORDE
if ((piedraGloria3_mc.x + 170) < 0) {
    //SE COLOCA LA PIEDRA EN EL BORDE DE LA DERECHA PARA QUE PUEDA VOLVER A SALIR
    piedraGloria3_mc.x = 800;
    //SE PONE MERO ABAJO EN EL EJE DE LAS Y, PARA POSTERIORMENTE COLOCARLE NUEVA ALTURA
    piedraGloria3_mc.y = 450;
    //NUMERO RANDOM PARA MODIFICAR LA ALTURA DE NUESTRA PIEDRA
    NumRandom = (Math.random() * 200) + 100;
    //SE LE PONE NUEVA ALTURA A NUESTRA PIEDRA
    piedraGloria3_mc.height = NumRandom;
    //Y SE PONE LA PIEDRA EN EL EJE DE LAS Y SEGUN SU ALTURA PARA NO DEJAR BORDES POR DEBAJO
    piedraGloria3_mc.y = 450 - NumRandom;
    //PARA QUE LA LINEA SALGA
    linea3.y = piedraGloria3_mc.y + 15;
    linea3.height = piedraGloria3_mc.height - 15;
    linea3.x = piedraGloria3_mc.x;
}
}
//CUANDO INICIE EL FRAME INICIARA LA FUNCION QUE HARA MOVER AL CIRCULO
piedraGlorial_mc.addEventListener(Event.ENTER_FRAME, MoverPiedras1);

```


Fotograma 9

En nuestro frame de resultados colocamos algunos tweens para nuestros textos y lo que hacemos es asignarle a cada texto dinámico los valores según correspondan al jugador sobre su tiempo su recorrido y sus plataformas saltadas para que tengan una visualización de sus logros durante su turno de juego al igual que también guardamos los precipicios saltados por cada jugador en 2 variables para después hacer una condición para comparar las plataformas saltadas por el primer y el segundo jugador para de esta manera sacar al jugador más destacado y también comparar cuando haya algún empate y todo esto se muestra en nuestro texto dinámico de ganador.

RESULTADOS


JUGADOR 1

Nombre	
Distancia	
Precipicios	
Tiempo	

JUGADOR 2

Nombre	
Distancia	
Precipicios	
Tiempo	

GANADOR:



acciones :9

```

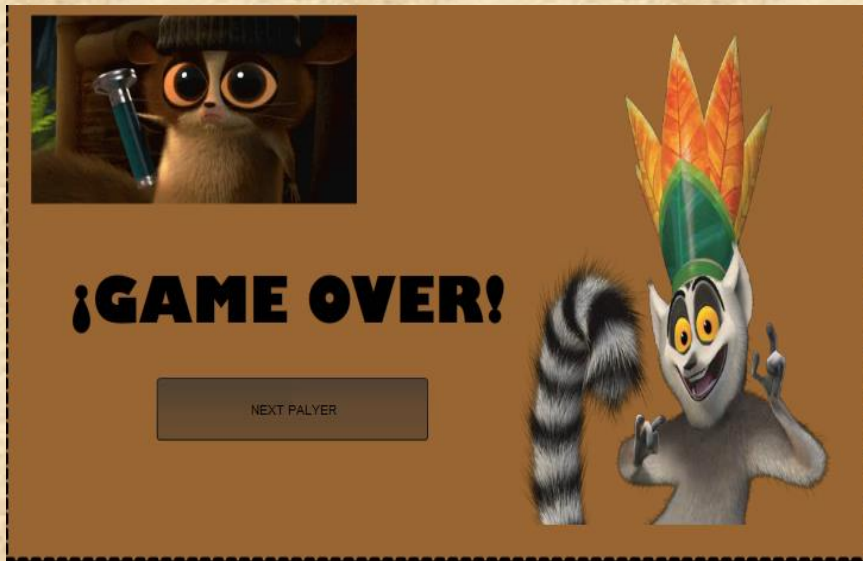
1 //NUESTRAS LIBRERIAS
2 import flash.events.MouseEvent;
3 import fl.transitions.Tween;
4 import fl.transitions.easing.*;
5 import fl.transitions.TweenEvent;
6
7 //le pondremos movimiento a los globos
8 var bounceEaseIn1:Tween =new Tween (global_mc, "y", Bounce.easeIn, 750, 11, 2, true);
9 var bounceEaseIn2:Tween =new Tween (globo2_mc, "y", Bounce.easeIn, 750, 99, 2, true);
10 var bounceEaseIn3:Tween =new Tween (globo3_mc, "y", Bounce.easeIn, 750, 193, 2, true);
11 var NoneEaseInOut1:Tween =new Tween (marlene_mc, "y", None.easeIn, 800, 290, 2, true);
12 var NoneEaseInOut2:Tween =new Tween (resultados_mc, "x", None.easeIn, 0, 223, 2, true);
13 //PARA MOSTRAR LOS RESULTADOS ESTOS PRI,EROS PARA QUE MUESTRE EL NOMBRE RESPECTIVO
14 nombreResultado1_txt.text=nombrresJugadores[0];
15 nombreResultado2_txt.text=nombrresJugadores[1];
16 //PARA QUE MUESTRE LA DISTANCIA DE CADA UNO
17 distanciaResultado1_txt.text=distaciaJugadores[0];
18 distanciaResultado2_txt.text=distaciaJugadores[1];
19 //PARA QUE MUESTRE LOS PRECIPICIOS DE LOS QUE LOGRÓ PASAR
20 precipicioResultado1_txt.text=precipiciosJugadores[0];
21 precipicioResultado2_txt.text=precipiciosJugadores[1];
22 //PARA QUE MUESTRE EL TIEMPO EN EL QUE LLEGÓ CADA UNO DE LOS DOS JUGADORES
23 tiempoResultado1_txt.text=tiempoJugadores[0];
24 tiempoResultado2_txt.text=tiempoJugadores[1];
25
26 if (precipiciosJugadores[0]<precipiciosJugadores[1]){
27     Ganador_txt.text=nombrresJugadores[1];
28 }
29 else
30 {
31     Ganador_txt.text=nombrresJugadores[0];
32 }
33 if (precipiciosJugadores[0]==precipiciosJugadores[1]){
34     Ganador_txt.text="Empate";
35 }

```

Fotograma 10

I

Aquí lo que hacemos es solamente una función para el botón de Next Player la cual nos manda al frame en donde se elige el personaje con el que el 2 jugador quiere jugar.



acciones:10

```
1 //nuestras librerias
2 import flash.events.MouseEvent;
3 //PARA CUADO LE DE NEXT PLAYER NOS MANDE PARA EL FRAME PARA QUE SELECCIONES EL JUGADOR
4 function nextPlayer(event:MouseEvent):void{
5     gotoAndStop(4); //NOS ANDA AL 4
6 }
7 siguienteJugador_btn.addEventListener(MouseEvent.CLICK, nextPlayer);
```

Conclusiones



Alberto

Mi conclusión es que en este proyecto se emplearon cosas nuevas, tales como la forma de emplear los saltos de nuestros personajes, tanto la manera en la que hacemos que nuestras plataformas entren con una altura random, cosas que parecen sencillas pero nos hacen pensar y analizar las cosas con calma, fue un proyecto tedioso pero en conjunto con mis compañeros logramos sacarlo adelante y con esto tener y lograr un gran resultado, a pesar de muchas veces ponernos a pensar para resolver algún error tanto de sintaxis como lógicos, llegamos a la conclusión de que fue un juego muy divertido de realizar ya que nos hizo trabajar en conjunto y ayudarnos entre todos para poder lograr hacer alguna cosa pero también fue un reto para nosotros. Reto que en conjunto logramos superar.



Cruz

El proyecto realizado fue de mucha ayuda y mejora del conocimiento de cada uno de nosotros, aunque ya habíamos trabajado con cosas similares, hubieron otras que fueron nuevas, tal es el caso del hitTestObject, el cual no ayudó mucho con el desarrollo del juego. Algo con lo que el juego nos permitió interactuar fue con la física y la programación unidas, tomar en cuenta la velocidad de desplazamiento y la gravedad para producir cada salto, además de crear una parábola para que saltara el personaje, fue una tarea que sin la ayuda de los conocimientos de la física nos hubiera sido más difícil desarrollar el

proyecto. Darle a elegir a ambos jugadores un personaje es algo que le da un poco más de competitividad al juego, pues pueden elegir personajes distintos y no solo el mismo, además de que cada personaje cuenta con un ambiente propio en donde se desarrolla la carrera. Fue algo divertido hacer este proyecto y ver el desarrollo desde que solo era un archivo en blanco hasta ser el juego que hoy es.



Claudia

Mi conclusión sobre este proyecto es que fue bueno trabajar en equipo ya que con ideas de cada uno de nosotros logramos elaborar lo que hoy es este juego, teníamos ideas diferentes y que con la ayuda de mis compañeros de equipo logramos implementarlas, uno de nuestros propósitos fue añadirle al juego que el usuario en turno eligiera el personaje con el que quisiera jugar, otra de las cosas que hicimos fue que aprendimos cosas



nuevas como el salto del personaje el random de los precipicios pero que fuera diferente la altura de cada una de ellas al salir y pues fue a la vez divertido en hacer el diseño, programar entre otras cosas que paso en el transcurso de la entrega.