

Universidad Politécnica de Cd. Victoria

MANUAL TECNICO DE LA INTERFAZ



Alumno: Miravete Martínez Jesús Antonio

Matricula: 1730140

Grupo: HM - ITI-07116

Fecha: 03/04/2018

INDICE

Frame Inicio-----	3
Cambio de Frame-----	4
Frame Prácticas-----	10
Frame Videojuegos-----	12
Frame Multimedia-----	14
Función Okey If Verdadero-----	15
Frame Conclusiones-----	16
Interpolaciones de Botones-----	17

FRAME INICIO

En el frame de inicio realmente no hay botones propios del frame, los únicos botones que tiene son los del menú, que todos los frame tienen, "Inicio", "Practicas", "Videojuegos", "Multimedia" y "Conclusiones":



Cada Uno de los botones del menú tienen una función prácticamente idéntica, lo único que cambia en ellas es el nombre y el valor que se le da a la variable op:

```

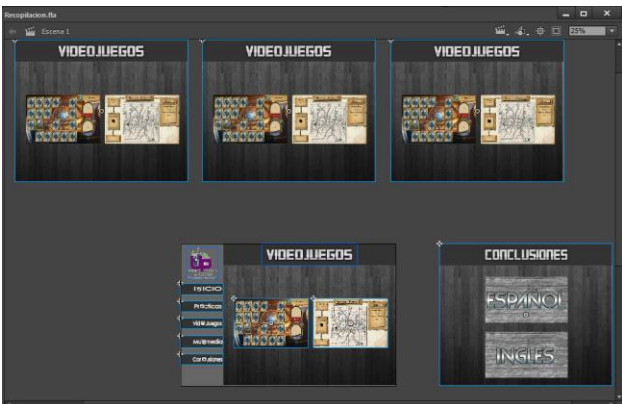
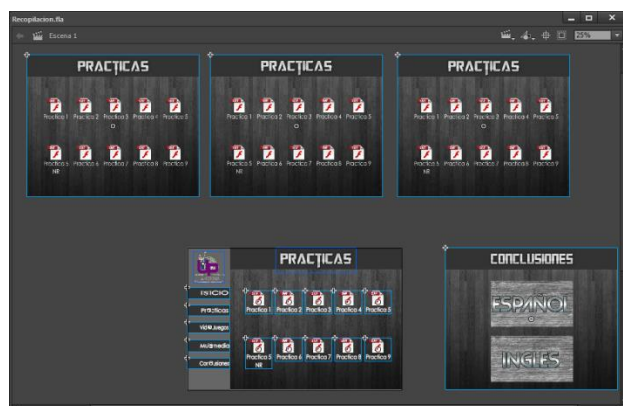
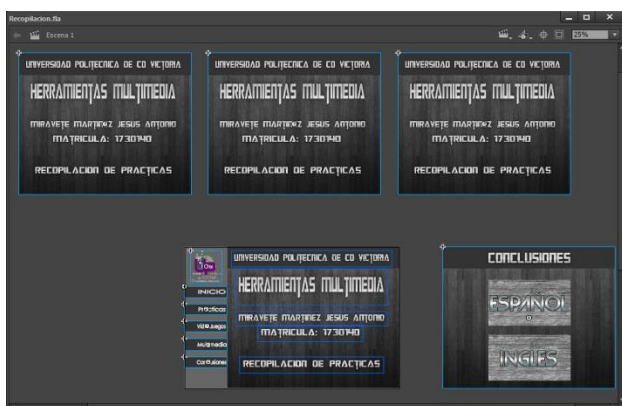
35  function irInicio(Event: MouseEvent): void {
36      op = 1;
37      okey();
38  }
39  //funcion para ir al segundo frame donde estan las practicas
40  function irPracticas(Event: MouseEvent): void {
41      op = 2;
42      okey();
43  }
44  //funcion para ir al tercer frame donde estan los videojuegos
45  function irvJuegos(Event: MouseEvent): void {
46      op = 3;
47      okey();
48  }
49  //funcion para ir al cuarto frame donde estan las galerias de videos e imagenes
50  function irMedia(Event: MouseEvent): void {
51      op = 4;
52      okey();
53  }
54  //funcion para ir al quinto frame donde estan los videos de conclusiones
55  function irConclusiones(Event: MouseEvent): void {
56      op = 5;
57      okey();
58  }

```

Cambio de Frame

Como se mencionó anteriormente, al seleccionar cualquier botón del menú se le asigna un valor u otro a la variable op, para que sirva esto?

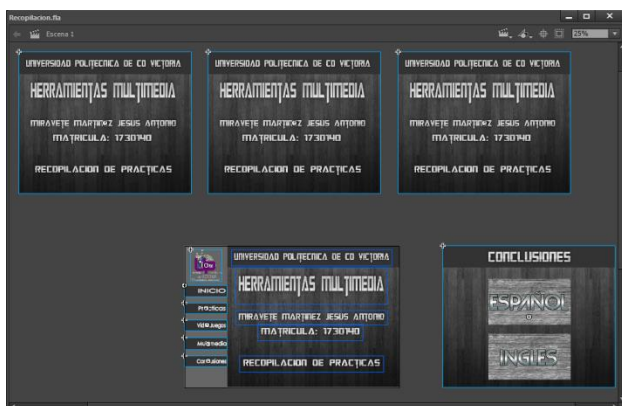
La recopilación tiene un pequeño detalle al momento de cambiar de frame, este detalle es que al ir de un frame a otro se ve una pequeña animación, como una transición entre el frame actual y al que se va, dichas transiciones son interpolaciones, las cuales fueron creadas con 10 frames dentro de diferentes movie clips en cada uno de los frames del proyecto:



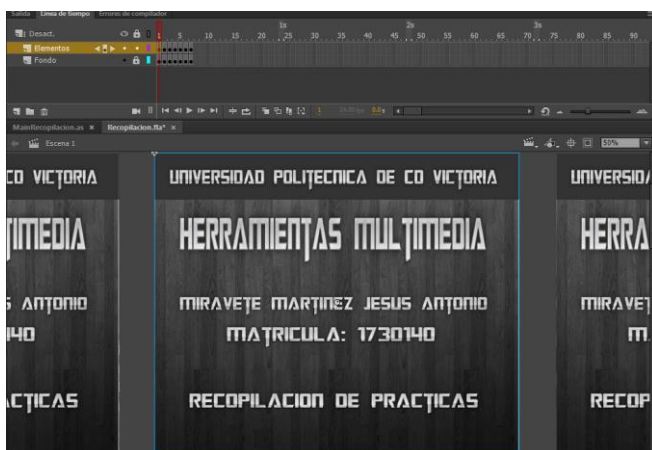
Como se puede ver en las 4 imágenes anteriores, los movie clips son los 3 elementos que parecen iguales, aunque realmente no lo son.

Cada uno de los 3 tiene principalmente la imagen de su frame, sin embargo, dentro de si como ya mencione tienen 10 frames, entre esos 10 frames es donde se hace un cambio entre la imagen que se muestra, y una imagen de uno de los otros 3 frames a los que puede ir el frame actual con una transición, para que se vea mejor mostrare un ejemplo.

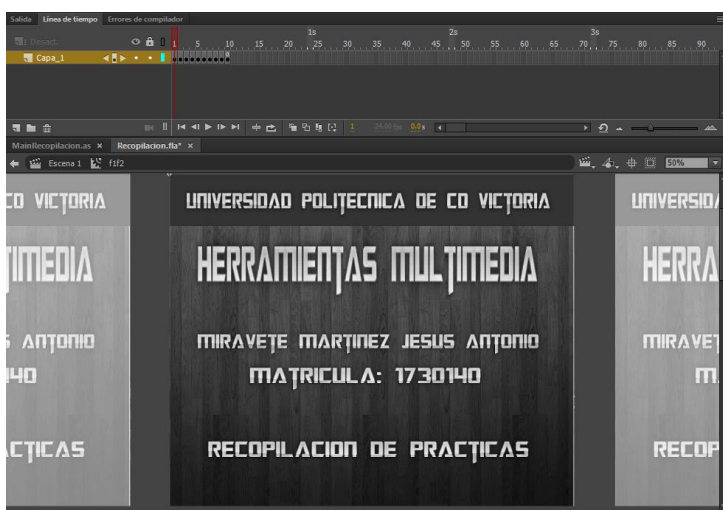
Enfoquémonos en el frame 1:



Enfoquémonos exactamente en este movie clip:



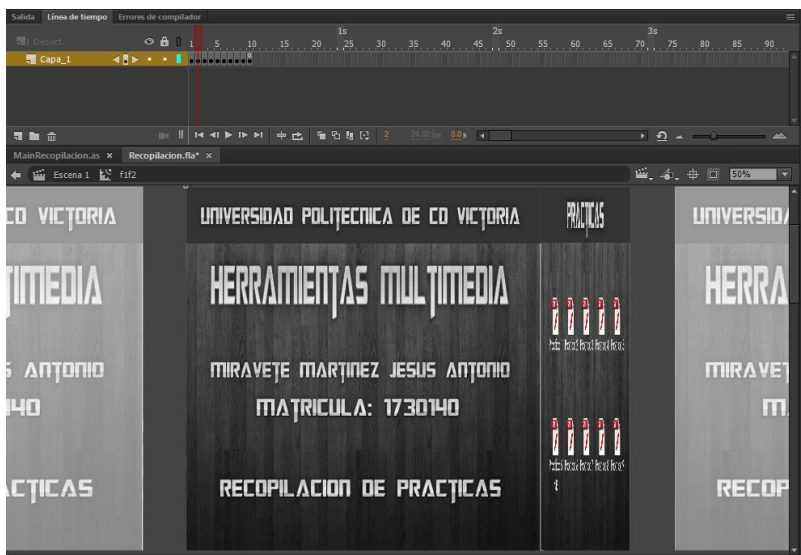
Si entramos a este movie clip podremos ver lo siguiente:



Se puede ver como los demás elementos se tornan grises, esto ya que es la forma que tiene flas de decirnos cuales elementos no tomamos en cuenta, y como se puede ver, la línea de tiempo no es

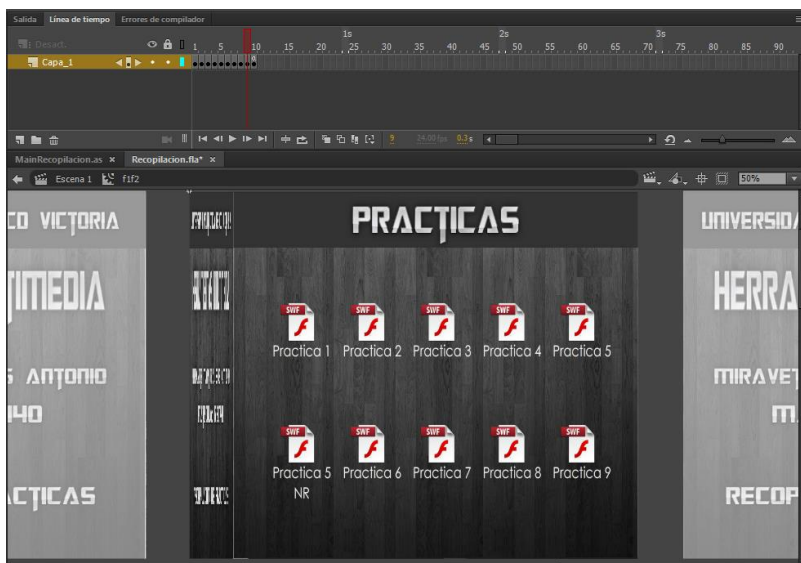
igual a la de antes. Esto pasa por que la nueva línea de tiempo es la línea de tiempo propia de este movie clip.

Actualmente nos encontramos en el frame 1, pero esto pasa si vamos al frame 2:



Se puede ver cómo hacemos menos ancha la imagen del frame inicio, y se empieza a hacer mas ancha la imagen del frame de prácticas, esto ya que esta interpolación, es la que nos manda desde el frame inicio al de prácticas.

Para no alargar tanto el ejemplo, si nos vamos al frame 9 se veria asi:



Se puede ver como el frame de prácticas está casi completo, asi como el de inicio está a punto de desaparecer.

Todas las interpolaciones del proyecto tienen la misma esencia, 10 frame en los que se va mostrando una imagen mientras se desaparece otra.

Una vez que esto se entiende, sigue explicar:

¿Cómo saber en qué momento se debe ejecutar una u otra interpolación?

Para esto creamos un algoritmo conformado por varias funciones, donde ya le daremos uso a la variable op mencionada en el tema de INICIO.

Si recuerdan, en el tema de inicio se pudo ver que las funciones de los botones del menú, además de darle valores a op, ejecutan la función okey();

Dicha función es prácticamente la encargada de hacer las interpolaciones, así como también se encarga de eliminar alguna práctica del escenario si es que ya se abrió alguna antes, pero esto se explicará en el siguiente tema.

Nuestra función okey lo único que tiene es un if, el cual para este tema no nos importa nada más que el else:

```
    } else { //si no, e
        timer.stop();
        cont = -1;
        timer.start();
    }
```

Como se puede ver, el else inicia deteniendo un timer, esto por si acaso algo está pasando y hacemos algo antes de que termine, así evitamos errores, vemos como también la variable cont se vuelve -1, este es su valor original, así que la estamos reseteando, esto con el mismo fin de detener el timer, por último, se puede ver que iniciamos el timer, este último es el que activa las interpolaciones.

Esto último hace que empiece a ejecutarse un timer, el cual como se puede ver incrementa a cont, y enseguida tiene un switch este switch es lo importante para saber que interpolación usar:

```
function tiempo(event: TimerEvent): void {
    cont++ //incrementamos cont
    switch (frame) { //comparamos a frame en
```

Este switch como pueden ver evalúa la variable frame, la cual solo puede tener 5 valores, los cuales son numéricos del 1 al 5, por ende, nuestro switch tiene 5 casos.

La variable frame nos ayuda a saber en que frame nos encontramos actualmente, al iniciar el programa frame empieza con el valor de 1, ya que al iniciar nos encontramos en el frame inicio, el cual es el número 1.

Más adelante se podrá ver donde cambia el valor de la variable frame.

Dentro de cada uno de los casos, a excepción del caso 5 lo primero que tenemos es un if, donde comparamos si cont es igual a 1, recordemos que cont es la variable que se incrementa cada que se ejecuta el timer, por lo tanto, aquí lo que queremos saber es si ya pasaron 2 ciclos del timer, ya que el primero no nos daría el tiempo que necesitamos.

Nos dice que si el contador es igual a 1 entonces detiene el timer y además resetea cont, pero se puede ver como enseguida tenemos un nuevo switch, el cual evalúa a la variable op, recordemos nuevamente que op cambia su valor en las funciones de los botones del menú, y en si el objetivo de op, es decirnos hacia que frame debemos ir:

```
case 1: //caso 1 significa:
    if (cont == 1) { //cor
        timer.stop();
        cont = -1;
        switch (op) {
```

Al tener este switch dentro del switch anterior, estaríamos comparando, a que frame iremos, y desde que frame, por lo tanto:

```
case 2:
    frame = 2;
    gotoAndStop(2);
    asignacionesMenu();
    asignacionesF2();
    break;
case 3:
    frame = 3;
    gotoAndStop(3);
    asignacionesMenu();
    asignacionesF3();
    break;
case 4:
    frame = 4;
    gotoAndStop(4);
    asignacionesMenu();
    asignacionesF4();
    break;
case 5:
    frame = 5;
    gotoAndStop(5);
    asignacionesMenu();
    asignacionesF5();
    break;
```

Estos son los diferentes casos de frames a los que se puede ir, actualmente estamos en el switch interno del caso 1 del switch externo, por lo tanto, tenemos los casos 2, 3, 4 y 5, porque no tenemos el 1? Por qué el 1 es donde ya estamos, por lo tanto no podemos ir hacia él, entonces, estos casos se entienden como los posibles frames a los que se puede ir desde el frame 1.

Este mismo proceso se lleva a cabo en los demás casos del switch externo, donde solamente cambiarían los casos del switch interno.

En el caso 2 externo, los casos internos serian 1, 3, 4 y 5.

En el caso 3 externo, los casos internos serian 1, 2, 4 y 5.

En el caso 4 externo, los casos internos serian 1, 2, 3 y 5.

En el caso 5 externo, los casos internos serian 1, 2, 3 y 4.

Y en si lo que hace cada caso, es darle a frame el valor del caso, así como ir al fotograma con el número del caso, hacer las asignaciones de botones necesarias para cada frame.

asignacionesMenu(); es una función que almacena los eventListener de los botones del menú.

asignacionesFn(); son las funciones que almacenan las asignaciones para los botones de los diferentes frames

Después del switch interno, pero aun dentro del caso 1 del externo, nos encontraremos con un else, el else al if donde comparamos si cont es igual a 1:

```

    } else {
        switch (op) {
            case 2:
                flaf2.y = 0;
                flaf2.x = 270;
                flaf2.gotoAndPlay(2);
                break;
            case 3:
                flaf3.y = 0;
                flaf3.x = 270;
                flaf3.gotoAndPlay(2);
                break;
            case 4:
                flaf4.y = 0;
                flaf4.x = 270;
                flaf4.gotoAndPlay(2);
                break;
            case 5:
                var CT1:Tween = new Tween(FC,"x",Strong.easeOut,1366,270,1,true);
                break;
        }
    }
}

```

En este else se ejecuta un switch, donde comparamos otra vez a op y con los mismos casos que en caso de que si se cumpla el if, lo que cambia entre el switch del if y el del else, son las acciones que se realizan en cada caso, y en este caso viene lo importante.

En los casos del else, lo que hacemos es, por ejemplo, en el caso 2, vemos que posicionamos a f1af2 en y=0, y x=270, después a ese mismo f1af2 le damos un gotoAndPlay(2);

¿Qué hacemos exactamente aquí?

f1af2, así como f1af3 en el caso 2 y f1af4 en el caso 4, son las diferentes interpolaciones disponibles para el frame 1, entonces lo que hacemos es que dependiendo de que botón del menú se haya presionado, entra en uno o en otro caso para así ejecutar una u otra interpolación.

La posicionamos para que se coloque dónde debe ir, y hacemos el gotoAndPlay para que inicie el recorrido por los frames internos de la interpolación.

¿Por qué hice esto en un else? En un if donde comparamos si cont es igual a 1, dentro de un timer que incrementa a cont?

Esto lo hice para que al principio no se cumpla el if, por lo tanto se vaya al else, y se ejecute la interpolación correspondiente, y después de casi medio segundo que es lo que dura el ciclo del timer:

```
var timer: Timer = new Timer(450, tmp++);
var tmp: int = 0; //variable que increment
```

Se cumpla la condición, por lo tanto se haría el cambio de frame terminando la interpolación y se cree el efecto visual de interpolación de cambio de frames.

Así es cómo funcionan las interpolaciones para los cambios de frame con efecto de cubo.

Cabe mencionar, que en los casos 5, lo único que se realiza es un tween, ya que este frame no se incluyó en el efecto de cubo.

FRAME PRACTICAS

En el frame de prácticas, a diferencia del frame de inicio no solo tenemos los botones del menú, en este frame tenemos los botones que ejecutarán las diferentes prácticas que se han realizado en el cuatrimestre:



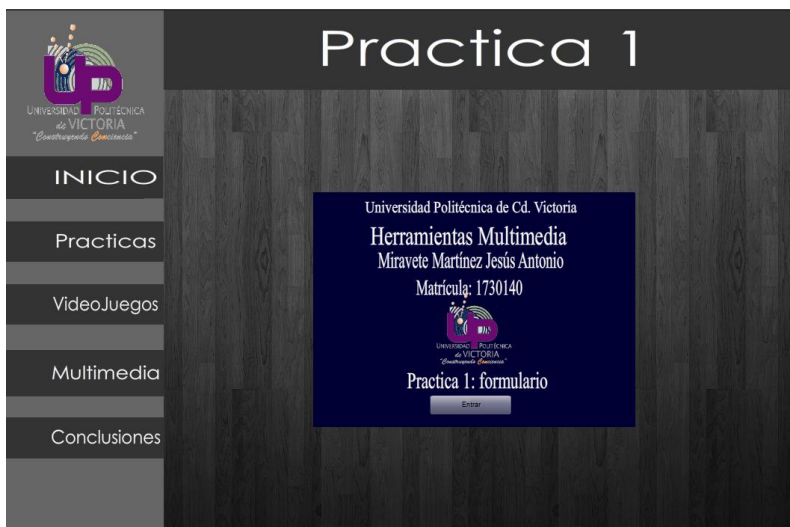
Anteriormente se explicó donde se asignan las funciones a los botones de cada frame, sin embargo, en el caso del frame de prácticas, debemos explicar cómo es que los botones ejecutan los swf de otras prácticas.

Es bastante sencillo, basta con tener el archivo swf de la practica junto a la recopilación, y utilizar la siguiente función:

```
//funcion para abrir la practica 1
function p1(event: MouseEvent): void {
    ok = true;
    gotoAndStop(6);
    numPractica.text = "Practica 1";
    fondoP1.x = 528;
    fondoP1.y = 313;
    var practicaSWF: URLRequest = new URLRequest("Practica 1.swf");
    swf.load(practicaSWF);
    stage.addChild(swf);
    swf.x = 528;
    swf.y = 313;
}
```

La función anterior es la que se encarga de ejecutar la primer practica de todas, como pueden ver a la variable booleana ok le damos el valor de true, esto nos ayudara a saber que ya se ejecutara, o más bien que ya se hizo un addChild, lo cual se puede ver un poco más abajo en la misma función.

Después de lo anterior hacemos un gotoAndStop(6), esto para que la práctica se ejecute en el frame 6, el cual contiene el menú, y un título el cual va cambiando dependiendo de que practica se abra, ya que como se ve a continuación, hacemos que numPractica.text sea igual a "Practica 1", esto hará que el título al presionar la practica uno sea el de la practica 1:



Después se puede ver como posiciono fondoP1 en x y en y, esto ya que este método no se trae el fondo del proyecto si este es directamente el escenario.

Después viene la función que llama al archivo swf, declaramos la variable practicaSWF de tipo URLRequest, y le especificamos el nombre de la practica deseada, en este caso Practica 1.swf.

Una vez llamado el archivo, debemos cargarlo, por ello hacemos swf.load(practicaSWF), ya que swf es una variable de tipo Loader. Después de esto hacemos el addChild anteriormente mencionado, al escenario y lo que le agregamos es swf que contiene la práctica ya cargada, por lo tanto esto sería lo que lo reproduzca.

Por ultimo posicionamos la práctica, en las mismas coordenadas que el fondo.

Y básicamente esta función se hace para todos los demás botones de prácticas, lo único que cambia es el nombre de la función, el nombre de la practica práctica, si es necesario, el fondo, porque hay algunas prácticas que no necesitan un fondo, como por ejemplo la 9:

```

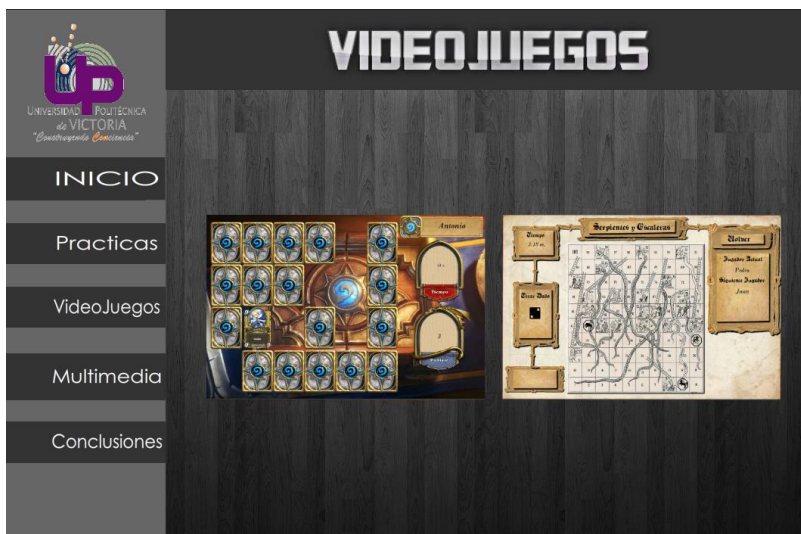
function p9(event: MouseEvent): void {
    ok = true;
    gotoAndStop(6);
    numPractica.text = "Practica 9";
    var practicaSWF: URLRequest = new URLRequest("Practica 9/Practica 9.swf");
    swf.load(practicaSWF);
    stage.addChild(swf);
    swf.x = 528;
    swf.y = 313;
}
  
```

Que se puede ver como no posiciona ningún fondo.

Cabe mencionar, que algunas prácticas, como la practica 9, utilizan varios archivos, y estos fueron almacenados junto con la practica en una carpeta, por ello, algunas funciones, como la de la practica 9 en el URLRequest colocan "El nombre de la carpeta"/"El nombre de la practica.swf".

FRAME VIDEOJUEGOS

El frame de videojuegos, al igual que el de prácticas, tiene los botones del menú y sus botones propios, este frame es el que almacena los 2 juegos que se realizaron en equipo, el memorama y el serpientes y escaleras:

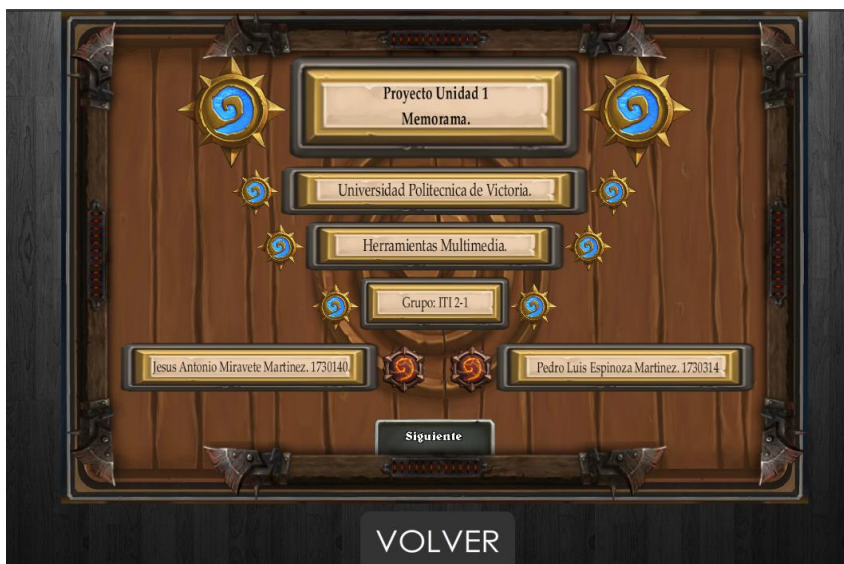


Estos 2 botones utilizan la misma función que los botones de prácticas, como se puede ver con el de memorama:

```

//funcion para abrir el juego del memorama
function Jmemo(event: MouseEvent): void {
    ok = true;|
    gotoAndStop(7);
    volver_btn.addEventListener(MouseEvent.CLICK, volverf3);
    var practicaSWF: URLRequest = new URLRequest("Proyecto Unidad 1.swf");
    swf.load(practicaSWF);
    stage.addChild(swf);
    swf.x = 90;
    swf.y = 0;
}
    
```

La única diferencia, es que tanto el memorama como el serpientes y escaleras, en lugar de ir al frame 6 como las prácticas, van al frame 7, el cual no contiene nada más que el espacio donde se mostrara la práctica, y un botón para volver al frame de videojuegos, el cual almenos en estos 2 casos se le asigna la función volver 3:



```

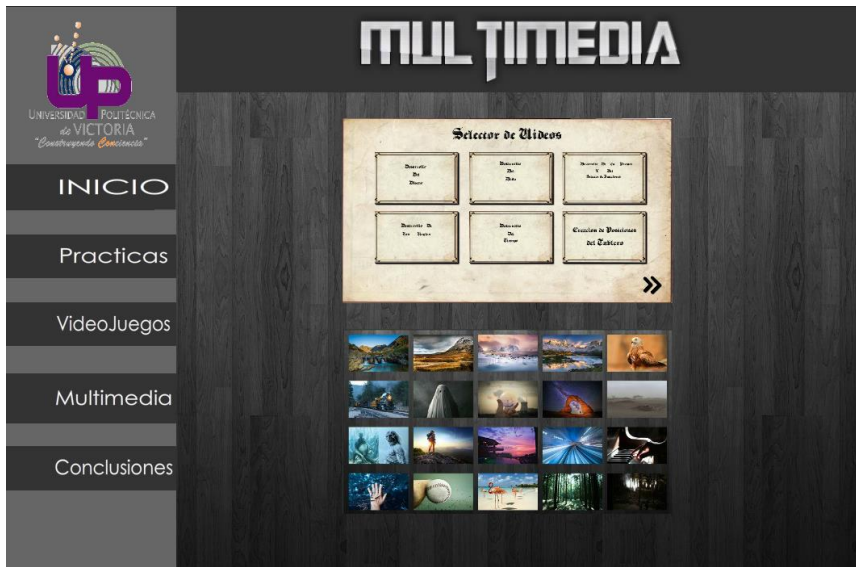
function volverf3(event: MouseEvent): void {
    okey();
    gotoAndStop(3);
    asignacionesF3();
    asignacionesMenu();
}
    
```

Dicha función lo que hace es ejecutar okey(), volver al frame 3 que corresponde al de videojuegos, y volver a asignar los botones correspondientes.

Existen varias funciones volver para diferentes frames, donde lo que cambia es al frame al que se va y las asignaciones que se hacen.

FRAME MULTIMEDIA

Multimedia al igual que videojuegos contiene los botones del menú y 2 botones propios, los botones propios de multimedia son los encargados de ejecutar la galería de videos de la práctica 10, y la galería de imágenes que pueden mostrarse en tamaño original:



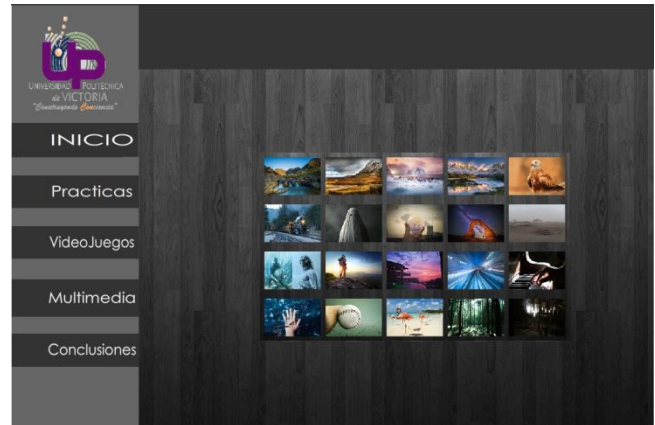
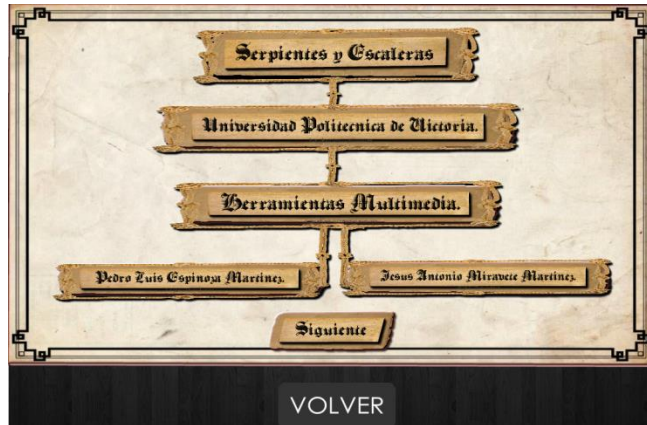
Estos botones utilizan la misma función que los de prácticas y videojuegos, solo que estos varían un poco entre si

```

function gVideos(event: MouseEvent): void {
    ok = true;
    gotoAndStop(7);
    volver_btn.addEventListener(MouseEvent.CLICK, volverf4);
    var practicaSWF: URLRequest = new URLRequest("Practica 10/Practica 10.swf");
    swf.load(practicaSWF);
    stage.addChild(swf);
    swf.x = 0;
    swf.y = 0;
}

//funcion para abrir la galeria de imagenes que se muestran en tamaño original
function gTO(event: MouseEvent): void {
    ok = true;
    gotoAndStop(6);
    var practicaSWF: URLRequest = new URLRequest("GaleriaTO/GaleriaTO.swf");
    swf.load(practicaSWF);
    stage.addChild(swf);
    swf.x = 528;
    swf.y = 313;
}
  
```

Como pueden ver la de videos manda al fotograma 7, y la galería de imágenes manda al fotograma 6, esto ya que la galería de videos es muy grande para la 6, sin embargo la galería de imágenes cabe a la perfección:



Como se mencionó anteriormente, algunas prácticas se encuentran en carpetas por tener varios archivos, estas 2 son así, y como se puede ver se especificó la carpeta donde están y el nombre del swf.

FUNCION OKEY IF VERDADERO

```

function okey() {
  if (ok == true) { //si se cumple
    stage.removeChild(swf); //quitamos el swf
    ok = false; //reseteamos la variable
    switch (op) { //switch que cambia de frame
      case 1:
        frame = 1; //damos un frame
        gotoAndStop(1); //var
        //hacemos las asignaciones
        asignacionesMenu();
        break;
      case 2:
        frame = 2;
        gotoAndStop(2);
        asignacionesMenu();
        asignacionesF2();
        break;
      case 3:
        frame = 3;
        gotoAndStop(3);
        asignacionesMenu();
        asignacionesF3();
        break;
      case 4:
        frame = 4;
        gotoAndStop(4);
    }
  }
}
  
```

En esta función como ya sabemos si la condición no se cumple entonces se activa el timer de cambios de frame, sin embargo, si se cumple la condición de `ok==true`, significa que una práctica, videojuego o elemento multimedia se ha ejecutado, entonces, si le damos click a cualquier botón del menú, no solo deberíamos ir a ese frame, sino que también deberíamos remover el proyecto ejecutado, por lo tanto lo primero que hacemos es un `removeChild`, donde removemos ese proyecto ejecutado.

Después iniciamos un `switch` donde comparamos `op`, para saber hacia que frame ir, en estos casos, el cambio de frame se haría sin interpolación.

FRAME CONCLUSIONES

Este frame, podría decirse que es el último, aunque realmente no es así por el frame 5 y 6, es el último importante por así decirlo, en el tenemos nuevamente los botones del menú, y 2 botones propios, los cuales se encargan de reproducir los videos de conclusiones en inglés y español.



Pero a diferencia de todos los botones propios anteriores, estos utilizan una función diferente:

```
function video1(Event:MouseEvent):void{
    var Videos:URLRequest = new URLRequest("conclusionesEspañol.mp4");
    navigateToURL(Videos,"self");
}
function video2(Event:MouseEvent):void{
    var Videos:URLRequest = new URLRequest("conclusionesIngles.mp4");
    navigateToURL(Videos,"self");
}
```

Esta función puede parecer que es igual, ya que hacemos un URLRequest al que le asignamos el nombre del archivo, pero como pueden ver estos son archivos mp4, y el leader no reconoce este tipo de archivos, por lo tanto utilice el método navigateToURL, para que los videos no se ejecuten en el frame, si no, que abra una página en el navegador donde se reproduzca el video.

Y esto sería todo por este frame, las únicas diferencias son el nombre de la función y del archivo.

INTERPOLACIONES DE BOTONES

En mi recopilación, absolutamente todos los botones ya sean propios de frame o los del menú, tienen interpolaciones para cuando el cursor esta encima de ellos, como por ejemplo:

Botones del menú: Se mueve y la letra cambia de color



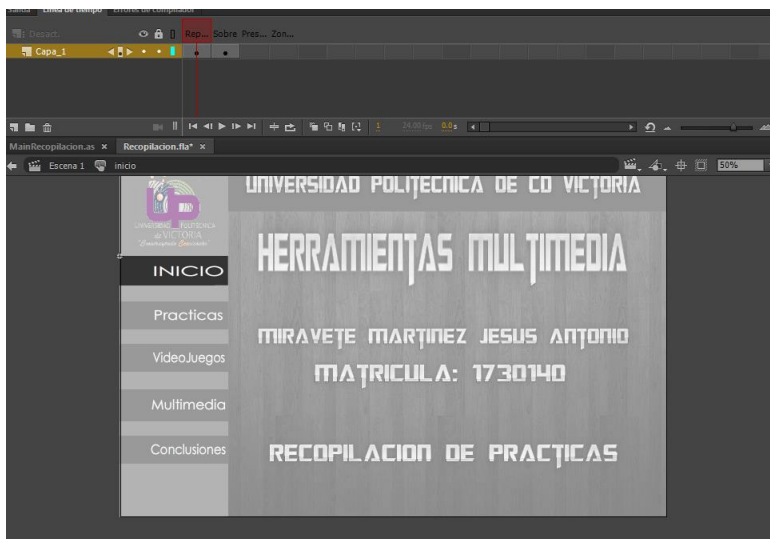
Botones de Practicas, Videojuegos, Multimedia y Conclusiones: Se coloca un cuadro negro medianamente transparente sobre el botón, el cual también contiene una pequeña descripción de lo que abrirá ese botón:



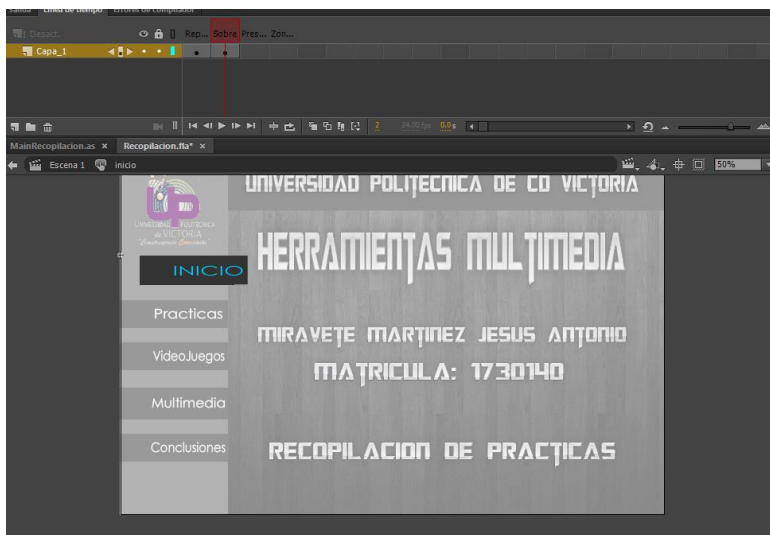
Estas interpolaciones son muy sencillas de hacer, absolutamente todas se hacen de la misma manera.

Lo que hacemos es abrir el clip de película para ver su interior, y siempre utilizaremos 2 frames.

En el primero se encuentra lo que se suele ver sin colocar el cursor encima del boton:



Y en el segundo frame se encuentra la imagen que se mostrara al colocar el cursor sobre el botón:



No hay que programar nada para que se haya el cambio al segundo frame, flash hace que al momento de colocar el cursor sobre un botón este cambie su forma.