

Universidad Politécnica de Victoria

Manual Técnico



Materia: Herramientas Multimedia
Practica Recolectora.
Cd. Victoria, Tam. ITI 2-2

ALUMNO:
Ramiro Alejandro Cedillo
Rubalcava.

Índice

Introducción	2
Librerías	3
Desarrollo	3
Conclusión	6

Introducción

En este trabajo documento mostraremos lo que se llevó a cabo para la realización del trabajo de recolección de prácticas y la galería de 20 imágenes de la materia "Herramientas Multimedia", en el cual mostraremos las librerías utilizadas y las funciones que empleamos a su vez explicaremos cuál es su función y como es que esta ópera dentro del juego.

Tambien se mostrara captrua de cada fragmento de codigo como también la declaracion de las variables que se emplearon para que así la explicacion sea mas detallada.

Librerías

```
import flash.events.Event;
import flash.display.Loader;
import flash.text.TextField;
import flash.display.Sprite;
import flash.events.MouseEvent;
import flash.net.URLLoader;
import flash.net.URLRequest;
import flash.text.engine.TabAlignment;
import flash.display.Loader;
import flash.events.MouseEvent;
import flash.events.Event;
import fl.transitions.easing.*;
import fl.transitions.TweenEvent;
```

animaciones que empleamos y MouseEvent es para los eventos del ratón como lo son el Clic, doble clic o mantener presionado, etc.

La Librería Loader para cargar nuestras prácticas además de la URLRequest para solicitar la dirección donde se encuentra la práctica. La de Sprite con la que hice un contenedor para la galería.

La primera librería utilizada fue la de Tween para hacer uso de ese tipo de animaciones y agregarle algún efecto a nuestro trabajo.

La segunda importa los efectos del tween que podemos agregarle a nuestro texto o imagen, etc.

La tercera y cuarta son para los eventos, TweenEvent es para las

Desarrollo

```
var a:Archivos = new Archivos(); //crea un nuevo objeto archivos
var op:int=0; //opcion a elegir
var pra:String; //direccion de la practica
var numframe:int=1; //numero de frame en que se encuentra
//Agregamos el evento al boton
btn_practicas.addEventListener(MouseEvent.CLICK,ir);
btn_juegos.addEventListener(MouseEvent.CLICK,ir);
btn_conclusiones.addEventListener(MouseEvent.CLICK,ir);
btn_inicio.addEventListener(MouseEvent.CLICK,ir);
btn_multimedia.addEventListener(MouseEvent.CLICK,ir);

//pasa al fotograma correspondiente al boton clickeado.
function ir(e:MouseEvent):void{
    gotoAndStop(a.sig(e));
}
```

navegación.

Cree un nuevo objeto Archivos para poder utilizar sus métodos y después declare una variable para ver qué tipo de acomodo llevara y otra para identificar la dirección de la página y creamos una función en la que utiliza un método de nuestra clase Archivos para saber a qué frame me mandara.

Fotograma 1: Este fotograma es la portada, se muestra el título, el nombre del alumno e información adicional, además del menú con el que se navegara, este mismo se repite a lo largo de los frases para la

En el siguiente frame utilice 10 botones que al clicar ejecutar la función dependiendo de la práctica así mismo se guardaba la dirección de la práctica y la mandaba al frame 6 para usar el método obtenerCarga() para cargar la práctica correspondiente.

Aquí mismo se guardaba el número de frame para después volver a él.

```
var conectar:simpleFlv = new simpleFlv();
var video:String;
op=2;
numframe=5;
btn_ingles.addEventListener(MouseEvent.CLICK,irVideo);
btn_espanol.addEventListener(MouseEvent.CLICK,irVideo);
function irVideo(evt:MouseEvent){
    if(evt.target.name == "btn_ingles"){
        video = "Conclusiones/ingles.mp4";
        gotoAndStop(6);
    }
    if(evt.target.name == "btn_espanol"){
        video = "Conclusiones/español.mp4";
        gotoAndStop(6);
    }
}
```

Declare dos variables una para cargar la práctica y la otra para saber si un video ha sido pausado o no.

Después compare el tipo de ordenamiento para las prácticas y el video. Ahí mismo agregaba con un addChild() lo que quería que se viera en el escenario.

```
op=1; //opcion de carga
numframe=2; //numero de frame en el que se encuentra

//funciones que mandan al ultimo fotograma donde se mostrara
function p1(event:MouseEvent):void{
    pra="Practicas/Practica 1.swf"; //direccion de la practica
    gotoAndStop(6);
}
function p2(event:MouseEvent):void{
    pra="Practicas/Practica 2.swf"; //direccion de la practica
    gotoAndStop(6);
}
function p3(event:MouseEvent):void{
    pra="Practicas/Practica 3.swf"; //direccion de la practica
    gotoAndStop(6);
}
function p4(event:MouseEvent):void{
    pra="Practicas/Practica 4.swf"; //direccion de la practica
    gotoAndStop(6);
}
function p5(event:MouseEvent):void{
    pra="Practicas/Practica 5.swf"; //direccion de la practica
    gotoAndStop(6);
}
function p6(event:MouseEvent):void{
    pra="Practicas/Practica 6.swf"; //direccion de la practica
    gotoAndStop(6);
}
function p7(event:MouseEvent):void{
    pra7=1; //indicador de acomodo distinto
    pra="Practica 7.swf"; //direccion de la practica
    gotoAndStop(6);
}
function p8(event:MouseEvent):void{
    pra="Practicas/Practica 8/Practica 8.swf"; //direccion de la practica
    gotoAndStop(6);
}
```

Después utilicé mi clase simpleFlv para reproducir los videos de las conclusiones. Declare una variable tipo String para guardar la dirección del video. Y para verificar cual botón se presionó compare el nombre de instancia del evento con su respectivo nombre.

```
var aux:Loader; //carga la practica
var bandera:Boolean=false;

if(op==1){
    a.obtenerCarga(pra); //Se invoca el metodo para cargar
    aux=a.regresarCarga(); //obtengo lo que se cargo
    addChild(aux); //agregamos la practica al escenario

    if(pra7!=1){ //se requiere un acomodo distinto
        aux.x=250;
        aux.y=150;
    }
} else if ( op==2){
    conectar.playMyFlv(video);
    conectar.y=77;
    conectar.x=55;
    addChild(conectar);
    conectar.addEventListener(MouseEvent.CLICK,pause);
}

function pause(event:MouseEvent){
    if(bandera == false){
        conectar._stream.pause();
        bandera = true;
    } else {
        conectar._stream.resume();
        bandera = false;
    }
}
```

```

public class Archivos extends MovieClip{
    var cargador:Loader;
    var url:URLRequest;
    var practica:String;
    public function Archivos(){
        // constructor code
    }

    public function obtenerCarga(p:String) {
        url = new URLRequest(p);
        cargador = new Loader()
        cargador.load(url);
    }

    public function regresarCarga():Loader{
        return cargador;
    }

    public function sig(evt:MouseEvent):int{
        var framemenu:int=0;

        //dependiendo del nombre de instancia indica al frame al cual ir
        if(evt.target.name == "btn_inicio"){
            framemenu=1;
        }
        if(evt.target.name == "btn practicas"){

```

En mi clase Archivos, tengo la funcion obtenerCarga() que recibe un parámetro string (que es la dirección de la practica) en la cual obtengo con un Loader y un URLRequest la práctica para luego devolverla a la practica con el método regresarCarga().

Otro método es el sig() que recibe un parámetro de tipo MouseEvent para saber a qué frame me mandara el botón del menú. Retorna un valor de tipo int que es el numero de frame.

Conclusión

Con este trabajo aprendí a darle un mejor uso a las clases en actionscript, también a usar algunos métodos que desconocía hasta el momento como `él.target.name` que hace referencia al nombre de instancia, lo cual me ayudo a reducir el código de una manera un poco significativa, también aprendí a como mandar llamar un archivo swf a otro swf al principio creí que sería un poco complicado sin embargo, después de buscar información sobre ello eso se disipo.

Aquí se puede ver la clase que use para el llamado de los videos utilice una variable del tipo `NetStream` y un `Sprite` para agregarlo al escenario.

Además otra clase que utilice fue la de `Loader` para cargar el archivo swf y la clase `URLRequest` para obtener la dirección del video.

También se agregó una galería como la que sé que se realizó en la practica 7, la diferencia entre ambas es que la practica 7 tenía menos imágenes y no tenía una función para agrandar el tamaño de la imagen. Como pueden ver en la pantalla, también pensé, al principio que podía ser complicado pero después de intentarlo, no lo fue tanto, tenía su grado de dificultado pero al final quedo hecho.

Aquí se puede ver como utilice una función para saber a qué frame es al que debe mandar el botón que se acaba de presionar así solo utilizo una sola función para cada botón, esta función recibe un parámetro de tipo `MouseEvent` y devuelve un valor de tipo `int` que es el que necesitamos.