

Universidad Politécnica de Victoria

PROYECTO FINAL

ABRIL 16, 2017

Manual Técnico

Paola Guadalupe Herrera Quiroz 1730413

Oscar Francisco Flores Gallegos 1730138

TABLA DE CONTENIDO

Contenido

Introducción	1
Interfaz	2
Portada	5
Multijugador	6
Creación de las piezas	8
Colisiones	11
Resultados	12
Conclusión	13

Introducción

Este es el manual técnico referente al proyecto final de la unidad número 2, de la materia de Herramientas Multimedia, impartida por el maestro Mario Humberto Rodríguez Chávez. El proyecto consta de 4 partes, elaborar el juego de Tetris sobre la herramienta que hemos estado utilizando que es Action Script de Flash, un diagrama de flujo en el cual se organicen los tiempos dado a cada parte de la elaboración del juego, agregar el juego a la recopilación de evidencias que habíamos elaborado anteriormente, y este mismo manual técnico, que es acerca del desarrollo del juego, de cómo se programó y cómo se hace funcionar.

Interfaz

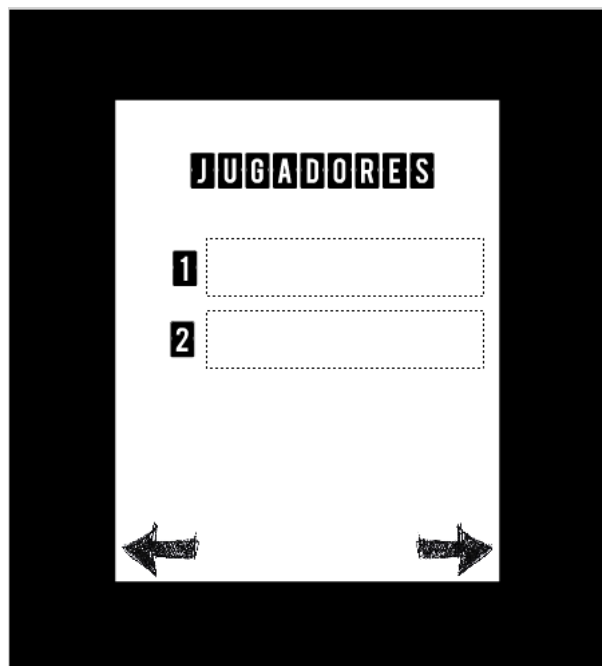


La interfaz de este juego no tiene una temática en específico, por lo cual el diseño ha provenido de ideas que fueron surgiendo a lo largo de los días de trabajo. Se ocuparon imágenes de vectores de lo que quisiéramos. En este caso, por ser el juego del tetris usamos figuras más cuadradas, es decir como cuadrados y rectángulos. También descargamos diferentes fonts para poder darle un mejor diseño.



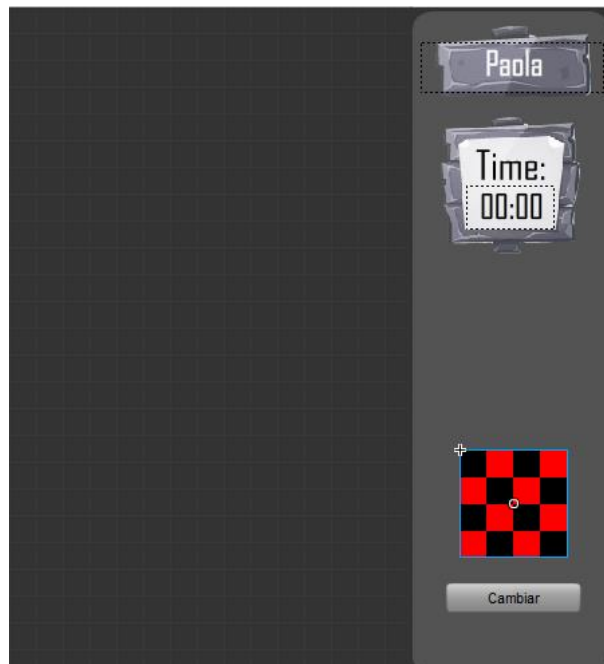
PROYECTO FINAL

Al iniciar el juego, se tiene una ventana en donde se encuentra un espacio para ingresar texto, en el cual serían los nombres; y dos botones para el primer y segundo jugador respectivamente, el segundo está inhabilitado hasta que se presione el del primero jugador.



Una vez que se tienen registrados los nombres de los dos jugadores, pasa a una siguiente ventana, que se encuentra en el mismo fotograma, pero en realidad es solo un recuadro en forma de movie clip, que aparece indicando la relación del número de jugador con su respectivo nombre, en esa ventana se ponen visibles dos botones, uno para regresar a la parte del registro de los jugadores, y el otro para iniciar el juego.

PROYECTO FINAL



El tercer y último fotograma es en donde se ejecuta el juego. En él, hay una cinta en la parte derecha, en el cual está un apartado con una etiqueta o texto dinámico que se actualiza con el nombre de los jugadores, debajo de eso está uno para el tiempo, el cual va avanzando y con base a ese tiempo se darán los resultados finales del ganador y el perdedor. La matriz que aparece en rojo es para mostrar la figura que va a caer después.

Portada

```
1  stop();
2  import fl.transitions.Tween;
3  import fl.transitions.easing.*;
4
5
6  var inicio:Tween=new Tween(inicio_btn,"y",Bounce.easeIn,1100,310,3,true);
7  var paola:Tween=new Tween(paola_mc,"x",Bounce.easeOut,1100,70,3,true);
8  var oscar:Tween=new Tween(oscar_mc,"x",Bounce.easeOut,-1100,65,3,true);
9  var iti:Tween=new Tween(iti_mc,"x",Bounce.easeOut,1100,175,3,true);
10 var c1:Tween=new Tween(unos,"rotationY",Bounce.easeIn,180,0,3,true);
11 var c2:Tween=new Tween(dos,"rotationX",Elastic.easeIn,270,0,3,true);
12 var c3:Tween=new Tween(tres,"rotationZ",Elastic.easeIn,360,0,3,true);
13 var c4:Tween=new Tween(cuatro,"rotationY",Bounce.easeIn,180,0,3,true);
14 var c5:Tween=new Tween(cinco,"rotationX",Elastic.easeIn,270,0,3,true);
15 var c6:Tween=new Tween(seis,"rotationZ",Elastic.easeIn,360,0,3,true);
```

Para poder lograr las animaciones de la portada, primero se separó la palabra “tetris” para poder controlar lo que hace cada una, se hizo de cada letra un movie clip diferente con las instancias de uno, dos, tres, cuatro, cinco y seis, a las cuales se les dio un movimiento de rotación intercalada en X, en Y y en Z. Es decir, la primera letra tuvo una rotación en el eje Y empezando en 180°, la segunda una en X empezando en el grado 270, la tercera la tuvo en Z empezando en el grado 360°; y así sucesivamente repitiéndose el patrón hasta terminar las letras. Todas terminan en el grado 0, o sea la posición inicial de la figura.

Multijugador

```
function Funo(event:MouseEvent):void{
    if (nombres_txt.text==" " || nombres_txt.text=="Ingresa nombre"){
        nombres_txt.text="Ingresa nombre";
    }else{
        nombresArray.push(nombres_txt.text);
        trace(nombresArray);
        uno_btn.visible=false;
        nombres_txt.text="";
        dos_btn.enabled=true;
    }
}
```

Esta es la función del primer jugador, es un evento del mouse ya que se realiza solo cuando se presiona ese botón. Primero estamos validando que tenga un texto escrito con una condición if, en caso de no ser así, se pone el mensaje de “Ingresa nombre”, texto que tampoco se admite como nombre, por lo que en la condición entra que no esté en blanco ni que tenga ese mensaje. En caso de que no se cumpla esa condición, el nombre en el inputText se guarda en el array nombresArray con un push, se pone invisible el botón del jugador uno, se limpia la caja de texto y se habilita el botón del segundo jugador, ya que antes de esto esta deshabilitado.

```
//2° JUGADORES
function Fdos(event:MouseEvent):void{
    if (nombres_txt.text==" " || nombres_txt.text=="Ingresa nombre"){
        nombres_txt.text="Ingresa nombre";
    }else{
        nombresArray.push(nombres_txt.text);
        trace(nombresArray);
        dos_btn.visible=false;
        nombres_txt.visible=false;
        jugar_btn.visible=true;
        regresar_btn.visible=true;
        ingresa.visible=false;
        jugadores.visible=true;
        j1.visible=true;
        j2.visible=true;
        j1.text=String(nombresArray[0]);
        j2.text=String(nombresArray[1]);
    }
}
```


PROYECTO FINAL

Para el segundo jugador también se valida que se haya ingresado un nombre, una vez que se tiene registrado, se le añade al array de los nombres con un push, el botón del segundo jugador se pone invisible al igual que el label de “Ingresa nombre” y la caja de texto en donde se colocan. Por el contrario, se ponen en visible el recuadro de la lista de jugadores, los botones de jugar y regresar, y las etiquetas de los nombres de los jugadores (j1 y j2), a los cuales se les asigna el valor del array en la posición 0 y 1 respectivamente, que fue el orden en que se ingresaron.

```
//F3, JUEGO
function Fjugar(event:MouseEvent):void{
    //Iniciar el tiempo del juego
    gotoAndPlay(3);
    timer.start();
    timer.addEventListener(TimerEvent.TIMER,Ftiempo);
    resultados.visible=false;
    resultados_mc.visible=false;
    perdiste.visible=false;
    n1.visible=false;
    n2.visible=false;
    r1.visible=false;
    r2.visible=false;
    //Crear el Sprite del escenario
    creadorS.graphics.drawRect(0, 0, 300, 500);
    addChild(creadorS);

    //Iniciar creacion de la pieza
    mostrar = Math.random() * 7 + 1;
    Pieza();

    stage.addEventListener(Event.ENTER_FRAME, Movimeinto);
    stage.addEventListener(KeyboardEvent.KEY_DOWN, Presionar);
    stage.addEventListener(KeyboardEvent.KEY_UP, Soltar);
}
```

El botón de jugar te lleva al siguiente fotograma, inicia el timer de juego y manda a llamar la función de ese mismo timer. También pone invisible el recuadro de los resultados, el mensaje de que ya perdiste y el botón que te muestra los resultados. Añade el “creadorS” al escenario.

Creación de las piezas

```
//GENERADOR DE PIEZAS
private function Pieza(){
    //Generar un numero aleatorio
    select = mostrar;
    trace("seleccion: " + select + ", " + mostrar);

    //En base al numero, generar la pieza correspondiente
    switch(select){
        case 1: v1y = 0; v1x = 7;
                v2y = 1; v2x = 7;
                v3y = 2; v3x = 7;
                v4y = 2; v4x = 8;
                limite = 21;
                break;
```

Lo primero en la creación de la pieza es seleccionar cual pieza se creara, para eso se usan dos variables, una toma un valor aleatorio en 1 y 7 y muestra la figura que se va a generar, mientras que la otra toma ese valor y se usa para indicar que figura está en juego. El uso de dos variables permite tener una pieza en juego y mostrar la que sigue.

Luego se entra en un switch en donde según el número, será la figura que se “trazara” en la matriz, esto se hace en base a las posiciones en donde se ubicara la pieza inicialmente en la matriz del escenario. Se usan variables para indicar las posiciones para futuras manipulaciones de las mismas. También se indica en que fila se debe parar la figura para tocar la base, esto según su posición original, aunque cambia al girar la pieza en la mayoría de los casos.

PROYECTO FINAL

Una vez seleccionada la pieza se revisa si ya cayó la primera pieza, si ya paso la primera pieza, se dibuja en la matriz, actualizando en el lugar en la que cayó la pieza anterior. Después se indica con un 1 las posiciones correspondientes el trazo de la pieza. Para terminar se reinician las variables de control de las demás funciones (control del giro, movimiento, etc.), se indica la siguiente pieza que se generará y se pasa a la función para dibujar la pieza en el escenario.

```
if(numPiezas > 1){
    //Se dibuja la figura en la matriz, en base en donde cayo
    generador[fly][flx] = 1;
    generador[f2y][f2x] = 1;
    generador[f3y][f3x] = 1;
    generador[f4y][f4x] = 1;

    //Muestra la matriz en consola
    for(var ed:int = 0; ed < 25; ed++){
        trace(generador[ed]);
    }
    trace("\n");
}

generador[vly][vlx] = 1;
generador[v2y][v2x] = 1;
generador[v3y][v3x] = 1;
generador[v4y][v4x] = 1;

indX = 0;
indY = 0;
giro = 0;
block = false;

//Generar el numero aleatorio
mostrar = Math.random() * 7 + 1;
```

PROYECTO FINAL

```
//DIBUJAR LA PIEZA EN EL ESCENARIO
private function Dibujar(){
    posX = 0;
    posY = 0;
    ind = 0;

    //LEER FILAS
    for(var a:int = 0; a < 4; a++){
        //LEER COLUMNAS
        for(var b:int = 0; b < 15; b++){

            //Selección del color de la pieza
            if(select == 1 || select == 3 || select == 5 || select == 7){
                atomoM = atomo1 = new Atomo1;
            }
            else{
                atomoM = atomo2 = new Atomo2;
            }

            //DIBUJAR SOBRE EL ESCENARIO
            if(generator[a][b] == 1){
                atomoM.x = (posX * 20);
                atomoM.y = (posY * 20);
                creadorS.addChild(atomoM);
                creadorS.addChildAt(atomoM, ind);
                ind++;
            }
        }
    }
}
```

En esta función es en donde, en base al trazado en la matriz, se dibuja la pieza sobre el escenario. Lo primero es inicializar las variables de posición en 0 y el índice de la pieza en 0. Después con dos ciclos for se leen las primeras 4 filas de la matriz, columna por columna, y en base a un valor se determina de qué color será la figura. Luego se pasa a la condición en donde sí se encuentra un 1 coloca un cuadro y le otorga un índice para identificarlo después. Durante cada ciclo se aumenta las variables de posición para ir generando la pieza.

Colisiones

```
//GENERADOR DE FISICA DEL JUEGO
private function Collision(event:Event){
    //v1
    var lecturalY:Number = Number((creadorS.getChildAt(0).y / 20) + 1);
    var lecturalX:Number = Number((creadorS.getChildAt(0).x / 20));

    var fisicalY:Number = Number((creadorS.getChildAt(0).y / 20));
    var derlX:Number = Number((creadorS.getChildAt(0).x / 20) + 1);
    var izqlX:Number = Number((creadorS.getChildAt(0).x / 20) - 1);
    //v2
    var lectura2Y:Number = Number((creadorS.getChildAt(1).y / 20) + 1);
    var lectura2X:Number = Number((creadorS.getChildAt(1).x / 20));
```

Para realizar las colisiones lo que se hace es que durante la caída se leen las posiciones de la figura en base a la matriz. Para hacer esto se toma su posición en Y, y en X, y se dividen entre 20, con esto se determina la posición en la que está en la matriz, por lo que a Y se le suma 1 para ver la posición de la siguiente fila. Con una serie de switches se verifica que, si la posición siguiente es 1, se detiene la caída. Esto se hace para cada cuadro involucrado puede tener contacto con otra pieza durante la caída.

Resultados

```
1  stop();
2  import flash.net.LocalConnection;
3  import flash.events.MouseEvent;
4
5  stage.focus = this;
6  nombre.text=String(nombresArray[0]);
7  resultados.addEventListener(MouseEvent.CLICK, Send2);
8
9  var lc2:LocalConnection = new LocalConnection();
10 lc2.connect("res2");
11 lc2.client = this;
12
13
14 function Send2 (event:MouseEvent):void{
15     lc.send("res", "Fres", tiempo_txt.text);
16     perdiste.visible=false;
17     resultados.visible=false;
18     resultados_mc.visible=true;
19     n1.visible=true;
20     n2.visible=true;
21     r1.visible=true;
22     r2.visible=true;
23 }
24
25 function Fres2 (tiempo2:String):void{
26     trace(tiempo2);
27     r1.text=tiempo_txt.text+" s";
28     r2.text=tiempo2+" s";
29 }
```

Para transferir los datos de los dos jugadores se utilizó una conexión local en donde ambos archivos fungen de cliente y servidor a la vez, primero se debe declarar la variable, en este caso es lc2, después se determina que se va a conectar a “res2” que es el atributo de la función Fres2, en donde se le da un nombre al dato recibido (tiempo2), y se le asigna a un texto dinámico (r2.text). El dato se envía mediante una función agregada en el otro archivo con el nombre de la variable (lc2) con extensión .send, en la cual se tienen como parámetros el nombre de la conexión (res2), la función en donde se le da el nombre (Fres2) y el texto, label, input text, etc, de donde se quiere extraer el dato.

Conclusión

A lo largo del plazo que se nos dio para la creación, diseño y programación de este proyecto, fuimos descubriendo y aprendiendo cosas que previamente desconocíamos. Probablemente se habían escuchado algunos términos que necesitamos, pero en realidad no estábamos familiarizados con muchos, algunos o la mayoría de ellos. Se trató de investigar por nuestra cuenta, trabajar en equipo y no dejarlo todo para los últimos días, lo cual no era posible ya que se realizaron videos documentando el avance diario y el tiempo invertido al juego. El manual técnico es de gran ayuda para poder entender el cómo de nuestras líneas de código, en él se va explicando el desarrollo de una manera más general, ya que dentro del mismo código se fueron documentando cada una de las funciones y variables utilizadas. Todo esto para el posible mejoramiento del mismo.