

MANUAL TÉCNICO

“PUZZLE LA VIDA SECRETA DE MI MASCOTA”



Integrantes:

- Rodríguez Ramírez Kevin Guadalupe
Matrícula: 1730463
- Carrizales Piña Claudia Lizbeth
Matrícula: 1730048

Maestro: Mario Humberto Rodríguez Chávez

Materia: Herramientas Multimedia

Grupo: ITI 2-2

INDICE

<u>Introducción</u>	3
<u>Desarrollo</u>	4
• <u>Fotograma1</u>	4
• <u>Fotograma2</u>	6
• <u>Fotograma3</u>	7
• <u>Fotograma4</u>	
• <u>Fotograma8</u>	
<u>Conclusiones</u>	
• Conclusión Kevin	
• Conclusión Claudia	



INTRODUCCIÓN

En este juego que es un puzzle sobre la película de la vida secreta de mi mascota tiene como objetivo armar uno de los cuatro puzzles elegido que aparecerán después de haber ingresado el número de los jugadores y los nombres de los respectivos una vez elegido te llevará al fotograma donde inicializará el juego en donde te pondrá la imagen (puzzle) en 3*3 en random para que posteriormente el usuario en turno lo resuelva, ten en cuenta que tendrás uno de estos pedazos en vacío para que los puedas mover; también en la parte izquierda aparecerá en textos dinámicos el tiempo, el nombre del jugador en turno y el número de movimientos que haga para resolverlo, en el tiempo si tarda se convertirán en minutos depende de lo que tarde el jugador en resolverlo, también tendrá la opción debajo de “rendirse” para cuando el jugador no encuentre la forma de resolverlo aparezca ya armado el puzzle y es ahí cuando el tiempo y los movimientos se paran y nos muestra el botón de next player para que el siguiente jugador inicie su juego pero tendrá que elegir el puzzle de su preferencia al igual al mover el primer pedacito del puzzle empezará a correr el tiempo y el número de los movimientos y su respectivo nombre; ten en cuenta que este juego solo es para 2 o mayor de 2 jugadores; ya al terminar si se rindió te mostrará el botón de estadísticas esto para ir al fotograma 8 donde te mostrará el nombre el tiempo y el número de los movimientos pero en este caso como se rindió el usuario solo nos mostrara el nombre y con el texto de “rendido” para el tiempo y movimientos; o en caso contrario que los haya armado nos mandará directo al fotograma 8 y ahora si con sus respectivas estadísticas y en las estadísticas en ese fotograma tendrá para exportar al block de notas en este caso tenemos nuestra carpeta con el juego, con el block de notas en este caso que se llama “resultados.txt” solo sería cuestión de al darle clic al botón de guardar al block de notas nos mostrará el archivo de referencia en este caso solo sería de seleccionar el que ya mencionamos y reemplazarlo y después abrirlo esto para los resultados del texto dinámico aparezcan en el bloc de notas una vez ya haciendo esto y así sucesivamente .



DESARROLLO

Fotograma 1

En las acciones del fotograma uno lo que hicimos fue poner nuestras respectivas librerías, definir nuestras variables que vamos a ocupar en este caso “i” que controla el acceso al jugador en turno nuestra variable número como numbers, nombres como string y nuestros arrays donde guardaremos los nombres ingresados y el número de movimientos, pusimos los tweens para las partes de nuestra portada en este caso para el botón de ingresar, el logo de la up y el nombre de la universidad, también definimos nuestro contador de los movimientos los definimos como enteros y lo inicializamos en 0; pusimos la función para nuestro botón de ingresar y que nos mande al fotograma 2 el de las instrucciones.

```
1  import flash.events.MouseEvent;
2  import fl.transitions.Tween;
3  import fl.transitions.easing.*;
4  import fl.transitions.TweenEvent
5  stop();//para detenernos
6  var i: int = 0;//es en cntador que controla el jugador en turno
7
8  //DEFINIMOS NUESTRAS VARIABLES QUE VAMOS OCUPAR PARA ODO EL JUEGO
9  //LAS PONEMOS AQUI YA QUE SI LAS PONEMOS EN OTRO FRAME SE VAN A VOLVER A INICIALIZAR LOS VALOS
10 var numero: Number;
11 var nombres: String;
12 var NombreJugadores: Array = new Array(); //nuestro array para los nombre de los jugadores del puzzle
13 var Tiempo:Array = new Array(); //el array que va a guardar el tiempo de cada usuario
14 var Movimientos:Array = new Array();//Array para imprimir movimientos realizados
15 var contl: int = 0;
16 //nuestro tween para nuestro nombre de la universidad, nuestro logo y botón
17 var bounceEaseIn:Tween =new Tween (upv_mc, "x", Bounce.easeIn, 100, 41, 2, true);
18 var bounceEaseInOut:Tween =new Tween (logo_mc, "y", Bounce.easeInOut, 206, 75, 2, true);
19 var bounceEaseInOutl:Tween =new Tween (ingresar_btn, "x", Bounce.easeInOut, 65, 227, 2, true);
20 //función para ingresar al siguiente fotograma (función para el botón "ingresar")
21 function ingresar(event:MouseEvent):void{
22     gotoAndStop(2);//irnos al dos
23 }
24 ingresar_btn.addEventListener(MouseEvent.CLICK, ingresar);
25
26 //Contador de los movimientos del jugador
27 var contm:int=0;
28
```


Así es la portada y las acciones para esta.



Fotograma 2

En las acciones del fotograma dos lo que hicimos fue agregar nuestras librerías y ponerle tween a nuestra imagen en este caso al conejo la imagen y el titulo de instrucciones y nuestra función que al darle clic al botón de continuar nos mande al fotograma 3 que es en donde vamos a ingresar el numero de los jugadores y sus nombres; y al botón de regresar para regresarnos a la portada.

```
actions:2
1  import flash.events.MouseEvent;
2  import fl.transitions.Tween;
3  import fl.transitions.easing.*;
4  import fl.transitions.TweenEvent;
5  //nuestro tween para nuestro titulo de instrucciones y nuestra imagen
6  var bounceEaseIn1:Tween =new Tween (instrucciones_mc, "x", Bounce.easeIn, 12, 145, 2, true);
7  var bounceEaseInOut1:Tween =new Tween (imagen_mc, "y", Bounce.easeInOut, 60, 169, 2, true);
8
9  //función para el botón "continuar"
10 function continuar(event:MouseEvent):void{
11     gotoAndStop(3); //irnos al tres
12 }
13 continuar_btn.addEventListener(MouseEvent.CLICK, continuar);
14
15 //función para el botón regresar a la portada
16 function regresar(event:MouseEvent):void{
17     gotoAndStop(1); //irnos al uno
18 }
19 regresar_btn.addEventListener(MouseEvent.CLICK, regresar);
20
```


Las acciones son para las funciones del fotograma de las instrucciones.

INSTRUCCIONES

1. Después de leer las intrucciones les mandará al siguiente frame dando click al botón de continuar que se encuentra en la parte de abajo.
2. Ingresar tu nombre y el número de jugadores en el cuadro de texto.
3. Despues elegir el puzzle con la que va a querer jugar.
4. Una vez ya elegido el puzzle para jugar y haber ingresado el nombre antes, te mandará directo a la pantalla del puzzle seleccionado y empezará a correr el tiempo.
5. Ten en cuenta que se te contará cada movimiento que realices al cambiar de lugar cada parte del puzzle así mismo el tiempo y estos se mostrarán en la parte derecha al momento de estar jugando.
6. Una vez ya completo el puzzle te mandará a la pantalla donde aparecerá los resultados obtenidos del juego.

REGRESAR

CONTINUAR



Fotograma 3

En las acciones para nuestro fotograma 3 pusimos nuestra librería para al darle clic en cada función lo explicamos a continuación:

En este caso lo primero que hicimos fue convertir nuestras 4 imágenes del puzzle en botones y ponerle las siguientes funciones y como ya habíamos mencionados anteriormente que nuestra *i* es el que lleva el control del jugador en turno y como el juego es para dos o más jugadores cada que va terminando o rindiéndose se irá incrementando y mostrar el siguiente nombre del jugador en nuestro texto dinámico correspondiente esto lo hacemos con los 4 botones en este caso los puzzles para irnos al numero de frame respectivo.

```
actions:3
1  import flash.events.MouseEvent;
2
3  //función para irnos al puzzle 1
4  function puzzle1(event: MouseEvent): void {
5      gotoAndStop(4); //irnos al cuatro
6      i++; //aumenta el jugador en turno y se pone el nombre
7  }
8  puzzle1_btn.addEventListener(MouseEvent.CLICK, puzzle1);
9
10 //función para irnos al puzzle 2
11 function puzzle2(event: MouseEvent): void {
12     gotoAndStop(5); //irnos al cinco
13     i++; //aumenta el jugador en turno y se pone el nombre
14 }
15 puzzle2_btn.addEventListener(MouseEvent.CLICK, puzzle2);
16
17 //función para irnos al puzzle 3
18 function puzzle3(event: MouseEvent): void {
19     gotoAndStop(6); //irnos al seis
20     i++; //aumenta el jugador en turno y se pone el nombre
21 }
22 puzzle3_btn.addEventListener(MouseEvent.CLICK, puzzle3);
23
24 //función para irnos al puzzle 4
25 function puzzle4(event: MouseEvent): void {
26     gotoAndStop(7); //irnos al siete
27     i++; //aumenta el jugador en turno y se pone el nombre
28 }
29 puzzle4_btn.addEventListener(MouseEvent.CLICK, puzzle4);
30 guardarN_btn.visible = false; //para que el boton de guardar el nombre de los jugadores este en invisible
31 //FUNCION PARA GUARDAR LOS NUMEROS DE JUGADORES
```


Después vamos a validar que la información que ha ingresado sea correcta en caso contrario de no ser así nos mostrará un aviso en un texto dinámico que ingresamos, nuestra función tiene el nombre de guardar que se cumplirá cuando oprimimos el botón que solo se nos muestra el de guardar número de jugador, si ingresa una letra, solo un jugador o no ingresa nada nos mostrará el aviso que dice “SÓLO NÚMEROS Y MAS DE UN JUGADOR” si no, si todo está correcto que el número que a ingresado es mayor a dos se guardaría en número y se pondría en invisible el aviso y el guardar de números de jugadores y se pondría en visible en de guarda nombres, los puzzles siguen en invisibles.

Ahora explicaremos lo que hace la función de guardar nombres, hacemos lo mismo validamos que si nombre nuestra variable es un numero o nombre no hay nada en el textInput nos muestre un aviso de “SÓLO LETRAS PARS EL NOMBRE” si no, si cont1 es menor a número (de jugadores) se guardará en el array de NombreJugadores en la posición de cont 1 (que inició en 0 ya que el array inicia en esa posición) cada nombre que ingrese depende al número de jugadores ingresado.

```

28
29 puzzle4_btn.addEventListener(MouseEvent.CLICK, puzzle4);
30 guardarN_btn.visible = false; //para que el boton de guardar el nombre de los jugadores este en invisible
31 //FUNCION PARA GUARDAR LOS NUMEROS DE JUGADORES
32 function guardar(event: MouseEvent): void {
33     numero = Number(numeroJ_txt.text);
34     if ((isNaN(numero)) || (numero == 0) || (numero == 1)) {
35         aviso_txt.text = "SÓLO NÚMEROS Y SON MÁS DE UN JUGADOR";
36         //nos marcará que solo numeros y lo que salgas lo eliminamos para poner el umero correcto de los jugadores que quera
37     } else {
38         aviso_txt.visible = false; //para que se desaparezca el aviso
39         //para escribir nuestros numero de jugadores en nuestro textinput
40         guardar0_btn.visible = false; //para que despues de ingresar el numero delos jugadores se desaparezca el boton y pod
41         guardarN_btn.visible = true; //y ya cuando se haya cumplido que el numeo que ingresó este correcto se ponga visible
42     }
43 }
44 //ES PARA QUE AL MOMENTO DE DAR CLIC EN EL BOTON DE ENTER SE EJECUTE LA FUNCION
45 guardar0_btn.addEventListener(MouseEvent.CLICK, guardar);
46
47 //PARA QUE ESTEN EN INVISBLES LAS OPCIONES DE ROMPECABEZAS Y YA CUANDO TERMINE DE INGRESQAR EL # DE JUGADORES
48 //Y LOS NOMBRES APAREZCAN PERO ESO ESTARA MAS ABAJO DEL CÓDIGO
49 puzzle1_btn.visible = false;
50 puzzle2_btn.visible = false;
51 puzzle3_btn.visible = false;
52 puzzle4_btn.visible = false;
53
54 function guardarnombres(event: MouseEvent): void {
55     nombres = String(nombreJ_txt.text); //para que sea puro texto el que se escriba como nombre
56     if ((Number(nombres)) || (nombres == "")) {
57         aviso2_txt.text = "SÓLO LETRAS PARA EL NOMBRE";
58     } else
59     if (cont1 < numero) //el nmbre d elos jugadores
60     {
61         NombreJugadores[cont1] = nombres; //entre corchetes porque será la posición de cada iteración que de el contador
62         trace(NombreJugadores[cont1]);
63     }

```

Y para comprobarlo utilizamos el trace para imprimirlo en la salida y comprobar que íbamos bien y el cont1 irá aumentando hasta que sea igual al número ingresado de jugadores una vez ya igualado se desaparecerán los

avisos y se pondrán visible los puzzles para que el jugador en turno escoja el de su agrado para empezar a jugar .

```
61     NombreJugadores[cont1] = nombres; //entre corchetes porque será la posición de cada iteración que de el contador
62     trace(NombreJugadores[cont1]);
63     cont1++;
64     //irá aumentando de uno en uno
65     nombreJ_txt.text = ""; //para ir eliminando lo que ya se ingreso osea el nombre para poner otro y guardar
66 }
67 if (cont1 == numero) {
68     aviso2_txt.visible = false; //para que se desaparezca el aviso
69     guardarN_btn.visible = false; //para cuando se iguale el numero de jugadores con el contador se desaparezca el boton de
70     puzzle1_btn.visible = true;
71     puzzle2_btn.visible = true;
72     puzzle3_btn.visible = true;
73     puzzle4_btn.visible = true;
74 }
75 }
76 //ES PARA QUE AL MOMENTO DE DAR CLIC EN EL BOTON DE ENTER SE EJECUTE LA FUNCION
77 guardarN_btn.addEventListener(MouseEvent.CLICK, guardarnombres);
```

Aquí el fotograma de las acciones 3, podemos observar los 4 puzzles nuestros texinput el primero donde ingresará el número de jugadores nuestro aviso1 el botón de guardar, y nuestro segundo texinput de nombre nuestro aviso2 y nuestro botón para ir guardando los nombre en el array.



Fotograma 4

Nuestras acciones del fotograma 4 agregamos nuestras librerías que ocuparemos, y en nuestro texto dinámico que se llama “Njugadorenturno_txt” le asignaremos el array de NombreJugadores en la posición de “i” y posteriormente definimos nuestras variables que utilizaremos y nuestro array de tiempo1 que utilizaremos para este puzzle

```

1 import flash.events.MouseEvent;
2
3 //para que aparezca el nombre en el texto dinamico del jugador en turno
4 Njugadorentorno_txt.text = NombreJugadores[i];
5 //DEFINIMOS LAS VARIABLES QUE UTILIZAREMOS EN LOS 4 TIMER QUE UTILIZAREMOS PARA NUESTROS FRAMES
6 //variables para el timer1
7 var tiempo1: int = 0; //es una variable y la inicializamos para que empiece en 0
8 var cont2: int;
9 var minutos: int=0;
10 var timer1: Timer = new Timer(1000, cont2++); //lleva la contabilidad de 1 en 1
11 function tiempo(tiempoevent: TimerEvent): void {
12     tiempo1++;
13     if(tiempo1<10){
14         tiempo1_txt.text = minutos + ":" + "0" + tiempo1 + "";
15     }else{
16         tiempo1_txt.text = minutos + ":" + tiempo1 + "";
17     }
18     if(tiempo1==59){
19         minutos++;
20         tiempo1=0;
21     }
22     if ((cuadril1_mc.x == 96 && cuadril1_mc.y == 65) && (cuadril2_mc.x == 181 && cuadril2_mc.y == 65) && (cuadril3_mc.x == 266 && cuadril3_mc.y == 65) && (cuadril4_mc.x == 96 && cuadril4_mc.y == 65)) {
23         Tiempo[i-1]=tiempo1_txt.text; //el numero que contenga el texto dinamico se guardará en el array
24         Movimientos[i-1]=movim_txt.text; //el numero de movimientos que tenga nuestro texto dinamico se guardará en el array
25         if (numero >= 2) {
26             nextPlayer1_btn.visible = true; //si el numero de jugadores es mayor a uno que aparezca el
27             juegoTerminado1_mc.visible = true;
28             cuadril1_mc.visible = false;

```

Implementacion del timer con minutos y segundos

Empezamos definiendo nuestras variables en para el timer y nuestro respectivo timer con su parámetro de 1000 milisegundos. Después de crear nuestra función para el evento del timer aumentamos el contador del timer en uno y entramos a la condición if en esta condición lo que hacemos es preguntar si nuestro contador es menor a 10 segundos (ósea que valga 1, 2, 3, 4, 5, 6, 7, 8, 9) nuestro texto dinámico valdrá lo que tenga minutos más dos puntos para hacerlo parecer un reloj más otro 0 y por ultimo más nuestro contador de tiempo por lo que quedaría algo así "0:06" esto lo hará cuando nuestro contador tiempo 1 sea menor a 10 después tenemos un else que sería en caso de que nuestro contador sea igual o mayor a 10 segundos nuestro texto dinámico valdrá lo que tenga minutos más dos puntos más nuestro contador de tiempo1 por lo que quedaría algo así "0:10", por ultimo tenemos una condición la cual pregunta si el contador de tiempo1 alcanza el valor de 60 (tiempo1==60) nuestra variable minutos se aumentara en 1 y

nuestra variable tiempo1 valdrá 0 nuevamente esto para poner minutos en nuestro timer por lo que pasaría de esto “0:60” a esto “1:01” y así sucesivamente lo estará haciendo hasta que el jugador termine de jugar.

```

6 //DEFINIMOS LAS VARIABLES QUE UTILIZAREMOS EN LOS 4 TIMER QUE UTILIZAREMOS PARA NUESTROS FRAMES
7 //variables para el timer1
8 var tiempo1: int = 0; //es una variable y la inicializamos para que empiece en 0
9 var cont2: int;
10 var minutos: int = 0;
11 var timer1: Timer = new Timer(1000, cont2++); //lleva la contabilidad de 1 en 1
12 function tiempo(tiempoevent: TimerEvent): void {
13     tiempo1++;
14     if(tiempo1<10){
15         tiempo1_txt.text = minutos + ":" + "0" + tiempo1+ "";
16     }else{
17         tiempo1_txt.text = minutos + ":" + tiempo1 + "";
18     }
19     if(tiempo1==60){
20         minutos++;
21         tiempo1=0;
22     }

```

Si los 8 cuadritos más el vacío están en la posición correspondiente se indicara que ganó y como el juego es multijugador de dos o más se pondrá en visible el botón de next player y la imagen de juego terminado como veremos más adelante y los cuadritos y el vacío se pondrá invisible el timer1 se detendrá y así hasta que “i” iguale al número de jugadores ingresados y nos llevará directo al frame de las estadísticas, un ejemplo es que haya puesto dos jugadores y este los tuvo que haber armado todos o el ultimo para que nos lleve directo a los resultados.

En la línea de código 24 y 25 que son nuestros array donde se guardará el tiempo de cada jugador y el número de movimientos en este caso lo que contenga los textos dinámicos en este caso que solo nos que se llaman tiempo1_txt lo que contenga este se guardara en el array de tiempo el se guardará en el array en la posición i-1 y así mismo con el array de los movimientos.

```

22 if ((cuadrito1_mc.x == 96 && cuadrito1_mc.y == 65) && (cuadrito2_mc.x == 181 && cuadrito2_mc.y == 65)) {
23     //&& (cuadrito3_mc.x == 266 && cuadrito3_mc.y == 65) && (cuadrito4_mc.x == 96 && cuadrito4_mc.y == 150) && (cuadrito5_mc.x == 181 && cuadrito5_mc.y == 150)
24     Tiempo[i-1]=tiempo1_txt.text; //el numero que contenga e texto dinamica se guardará en el array en la posicion menos 1
25     Movimientos[i-1]=movim_txt.text; //el numero de movimientos que tenga nuestro texto dinamico se guardará en el array de movimientos tambien
26     if (numero >= 2) {
27         nextPlayer1_btn.visible = true; //si el numero de jugadores es mayor a uno que aparezca el boto de sig jugador
28         juegoTerminado1_mc.visible = true;
29         cuadrito1_mc.visible = false;
30         cuadrito2_mc.visible = false;
31         cuadrito3_mc.visible = false;
32         cuadrito4_mc.visible = false;
33         cuadrito5_mc.visible = false;
34         cuadrito6_mc.visible = false;
35         cuadrito7_mc.visible = false;
36         cuadrito8_mc.visible = false;
37         vacio_mc.visible = false;
38         timer1.stop(); //para que el tiempo se detenga cuando gana
39         rendirse1_btn.visible = false; //para que el boton de rendirse se ponga invisible cuando lo haya terminado
40     }
41     if (i == numero) {
42         timer1.stop(); //si se va a este ya no entra al de abajo por eso el otro stop
43         gotoAndStop(8);
44     }
45 }
46 }
47 timer1.addEventListener(TimerEvent.TIMER, tiempo);
48

```

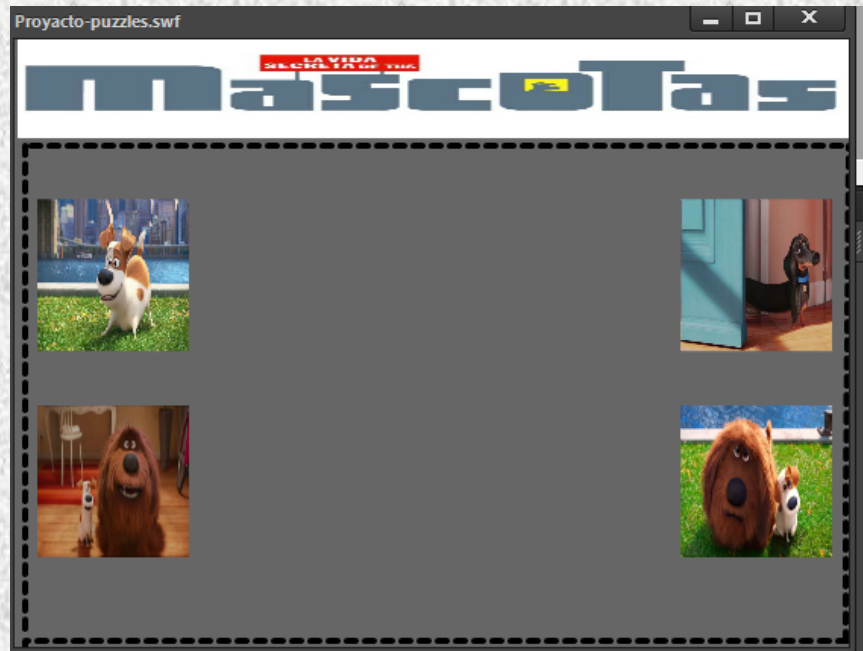
Para eso en cada entrada al puzzle nuestra "i" irá aumentando 1 en 1.

```
1 import flash.events.MouseEvent;
2
3 //función para irnos al puzzle 1
4 function puzzle1(event: MouseEvent): void {
5     gotoAndStop(4); //irnos al cuatro
6     i++; //aumenta el jugador en turno y se pone el nombre
7 }
8 puzzle1_btn.addEventListener(MouseEvent.CLICK, puzzle1);
9
10 //función para irnos al puzzle 2
11 function puzzle2(event: MouseEvent): void {
12     gotoAndStop(5); //irnos al cinco
13     i++; //aumenta el jugador en turno y se pone el nombre
14 }
15 puzzle2_btn.addEventListener(MouseEvent.CLICK, puzzle2);
16
17 //función para irnos al puzzle 3
18 function puzzle3(event: MouseEvent): void {
19     gotoAndStop(6); //irnos al seis
20     i++; //aumenta el jugador en turno y se pone el nombre
21 }
22 puzzle3_btn.addEventListener(MouseEvent.CLICK, puzzle3);
23
24 //función para irnos al puzzle 4
25 function puzzle4(event: MouseEvent): void {
26     gotoAndStop(7); //irnos al siete
27     i++; //aumenta el jugador en turno y se pone el nombre
28 }
29 puzzle4_btn.addEventListener(MouseEvent.CLICK, puzzle4);
```

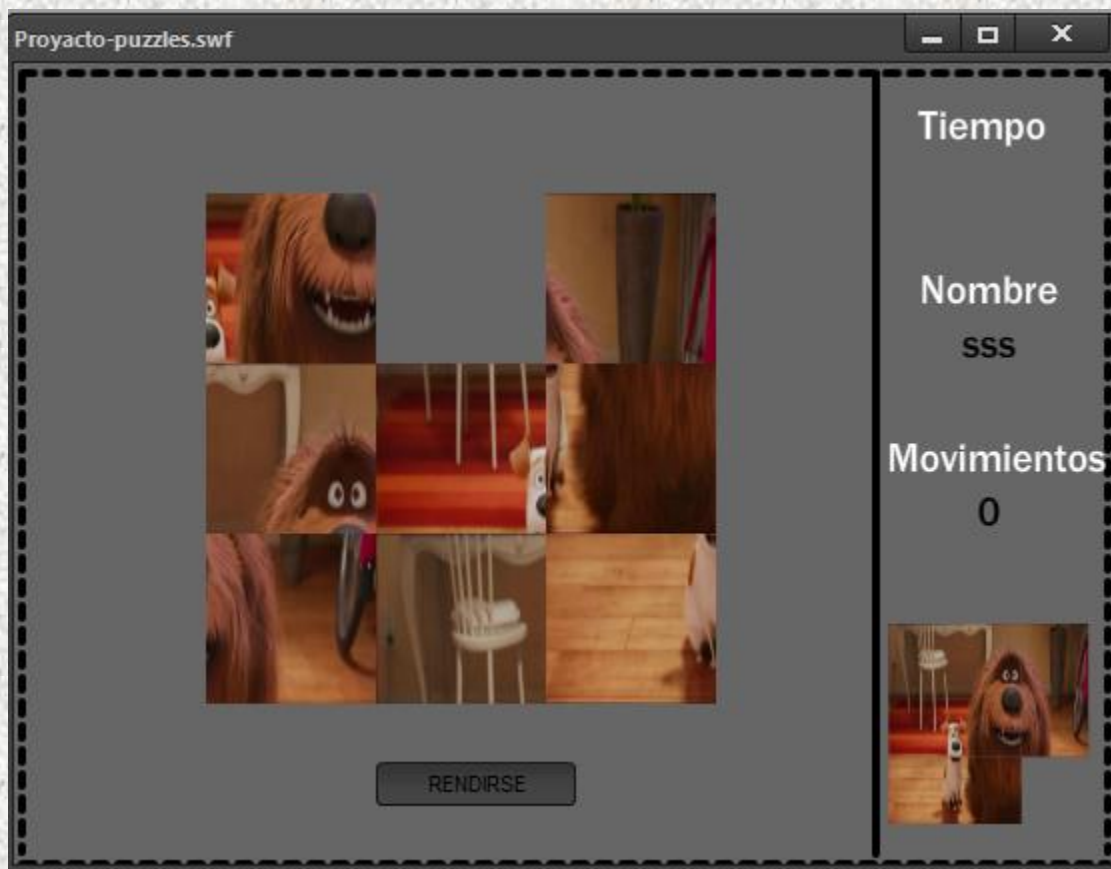
Así nos mostrará cuando se haya armado completo para pasar al siguiente jugador y que elija su puzzle y como vemos a lado izquierdo nuestro nombre del jugador en turno con su respectivo tiempo y numero de movimientos con el que quedó.



Para posterior ir a elegir el otro puzzle del siguiente jugador.



Y vemos que nuestro “i” se aumentó al siguiente jugador en turno en este caso le pusimos como nombre “sss”



Y cuando se iguala i al número en este caso pusimos dos jugadores nos mandará al fotograma de las estadísticas.

Acomodo de los pedazos de imagen en Aleatorio (random)

Definición de variables

Lo que hacemos aquí es definir las variables que ocuparemos para guardar el número random de cada uno de los pedazos incluyendo el vacío, definimos también las variables que nos servirán para guardar su posición en el eje "X" y el eje "Y" y por ultimo definimos el Array que nos ayudara a guardar las posiciones originales de nuestras imágenes para así extraer aleatoriamente sus ejes y acomodarlas.

```
48 //Definimos las variables que almacenaran nuestro numero random de cada cuadro de la imagen
49 var numrandom: int;
50 var numrandom2: int;
51 var numrandom3: int;
52 var numrandom4: int;
53 var numrandom5: int;
54 var numrandom6: int;
55 var numrandom7: int;
56 var numrandom8: int;
57 var numrandom9: int;
58
59 //Definimos las variables que almacenaran nuestras posiciones en X y Y de cada cuadro
60 //de nuestra imagen
61 var posx: int;
62 var posy: int;
63 var posx2: int;
64 var posy2: int;
65 var posx3: int;
66 var posy3: int;
67 var posx4: int;
68 var posy4: int;
69 var posx5: int;
70 var posy5: int;
71 var posx6: int;
72 var posy6: int;
73 var posx7: int;
74 var posy7: int;
75 var posx8: int;
76 var posy8: int;
77 var posx9: int;
78 var posy9: int;
79 contm = 0;
80
81 //Definimos el Array que usaremos para tomar las posiciones de nuestros cuadritos
82 var Arraypos: Array = new Array();
```

Asignación necesaria para el acomodo

Como podemos ver en las siguientes imágenes primeramente ponemos nuestros cuadritos originales y nuestras nuestro vacío en invisible como segundo ponemos nuestras copias de los cuadros y la copia del vacío en invisible (se crearon copias para evitar empalme de imágenes por lo que creamos imágenes idénticas a nuestros cuadritos y las pusimos en la misma posición para que siempre estén inmovibles e invisibles, solo nos sirven para la extracción de la posición). Por lo tanto le asignamos las copias a nuestro Array de extracción de posiciones.


```

81 //Definimos el Array que usaremos para tomar las posiciones de nuestros cuadritos
82 var Arraypos: Array = new Array();
83
84 //Ponemos nuestros cuadritos en falso para despues meterlos aleatoriamente(revolverlos)
85 //y volver a ponerlos en visible
86 cuadrito1_mc.visible = false;
87 cuadrito2_mc.visible = false;
88 cuadrito3_mc.visible = false;
89 cuadrito4_mc.visible = false;
90 cuadrito5_mc.visible = false;
91 cuadrito6_mc.visible = false;
92 cuadrito7_mc.visible = false;
93 cuadrito8_mc.visible = false;
94 vacio_mc.visible = false;
95 estadisticas_btn.visible=false;//boton para ir a estadisticas en invisible
96 //Estas son unas copias de cada uno de nuestro cuadrito y estan en la misma posicion de
97 // que los cuadritos originales y seran usadas para tomar su posicion y meterlos en el
98 //array
99 cuadritocopia1_mc.visible = false;
100 cuadritocopia2_mc.visible = false;
101 cuadritocopia3_mc.visible = false;
102 cuadritocopia4_mc.visible = false;
103 cuadritocopia5_mc.visible = false;
104 cuadritocopia6_mc.visible = false;
105 cuadritocopia7_mc.visible = false;
106 cuadritocopia8_mc.visible = false;
107 copiavacio_mc.visible = false;
108
109
111 //Le asignamos el cuadrito copia a cada posicion del array para despues extraer sus
112 //Posiciones en X y Y
113 Arraypos[0] = cuadritocopia1_mc;
114 Arraypos[1] = cuadritocopia2_mc;
115 Arraypos[2] = cuadritocopia3_mc;
116 Arraypos[3] = cuadritocopia4_mc;
117 Arraypos[4] = cuadritocopia5_mc;
118 Arraypos[5] = cuadritocopia6_mc;
119 Arraypos[6] = cuadritocopia7_mc;
120 Arraypos[7] = cuadritocopia8_mc;
121 Arraypos[8] = copiavacio_mc;

```

Acomodo en random del pedazo de imagen

Para el acomodo de nuestro primer pedazo de imagen lo que hacemos es asignarle un número random del 0 al 9 a nuestra variable “numrandom”, le asignamos a la variable “posx” lo que tenga nuestro Arraypos en la posición del número random que nos salió y “.x” para recibir el eje de las X que tenga lo que está guardado ahí por lo que nos devolverá un numero lo mismo hacemos pero esta vez para el eje de las Y los números que nos haya devuelto se los asignamos a nuestra imagen original en sus respectivos ejes y le indicamos que ya que se reacomodo en una posición aleatoria se ponga en visible nuevamente

Explicación del “do While”

Lo que hacemos en el do While es lo mismo que en el primer acomodo random solo que aquí introducimos una condición la cual nos dice que si el número random que me dio es distinto al número random que me había

dado anteriormente entonces que haga todo el proceso de acomodo si no se cumple la condición no hace la asignación por lo que se podría leer nuestro While de la siguiente forma “Hacer lo siguiente, Mientras el número random que te estoy pidiendo sea igual a algún número random que me hayas dado anteriormente” por lo que se repetirá hasta que me dé un numero diferente y así tomaremos posiciones diferentes en cada una de nuestras imágenes, esto mismo se hace 7 veces más para los pedazos de imágenes que nos faltan solo se va aumentando la condición y comparando el numrandom a pedir con cada uno de los anteriores para el primer número random que pedimos no hacemos el do While ya que como es el primero que solicitamos nos dará uno diferente por lo que no ocupamos condicionar la primera vez.

```

124 //Le asignamos un numero random del 0 al 7 a nuestra variable
125 numrandom = Math.random() * 9;
126 posx = Arraypos[numrandom].x; //Le asignamos la posicion de x que tiene la imagen almacenada en la posicion numrandom
127 posy = Arraypos[numrandom].y; //Le asignamos la posicion de y que tiene la imagen almacenada en la posicion numrandom
128 cuadrito1_mc.x = posx; //Le asignamos lo que salio del random de x a nuestro eje x de la imagen
129 cuadrito1_mc.y = posy; //Le asignamos lo que salio del random de y a nuestro eje y de la imagen
130 cuadrito1_mc.visible = true; //Volvemos a poner nuestra imagen en visible ya en su nueva posicion random
131 trace(numrandom); //Aqui imprimi mi numero random en consola para ver que numero me daba y solucionar errores
132
133 //Hacemos un "Hacer mientras" para que repita el numero random hasta que me de uno
134 //diferente si es que me dio el mismo que el anterior
135 do {
136     //Le asignamos un numero random del 0 al 7 a nuestra variable
137     numrandom2 = Math.random() * 9;
138     //Condicionamos que si el numero random que nos dio si es diferente se hace lo mismo que
139     //anteriormente y pone la imagen en una nueva posicion random
140     if (numrandom2 != numrandom) {
141         posx2 = Arraypos[numrandom2].x; //Le asignamos la posicion de x que tiene la imagen almacenada en la posicion
142         posy2 = Arraypos[numrandom2].y; //Le asignamos la posicion de y que tiene la imagen almacenada en la posicion
143         cuadrito2_mc.x = posx2; //Le asignamos lo que salio del random de x a nuestro eje x de la imagen
144         cuadrito2_mc.y = posy2; //Le asignamos lo que salio del random de y a nuestro eje y de la imagen
145         cuadrito2_mc.visible = true; //Volvemos a poner nuestra imagen en visible ya en su nueva posicion random
146     }
147     //Aqui ponemos el mientras que le digo que lo haga mientras el numero random sea igual a alguno de los anteriores
148 } while (numrandom2 == numrandom);
149 trace(numrandom2);

```

```

//Hacemos un "Hacer mientras" para que repita el numero random hasta que me de uno
//diferente si es que me dio el mismo que alguno de los anteriores
do {
    //Le asignamos un numero random del 0 al 7 a nuestra variable
    numrandom3 = Math.random() * 9;
    //Condicionamos que si el numero random que nos dio si es diferente a los demas se hace lo mismo que
    //anteriormente y pone la imagen en una nueva posicion random
    if (numrandom3 != numrandom && numrandom3 != numrandom2) {
        posx3 = Arraypos[numrandom3].x;
        posy3 = Arraypos[numrandom3].y;
        cuadrito3_mc.x = posx3;
        cuadrito3_mc.y = posy3;
        cuadrito3_mc.visible = true;
    }
    //Aqui ponemos el mientras que le digo que lo haga mientras el numero random sea igual a alguno de los anteriores
} while (numrandom3 == numrandom || numrandom3 == numrandom2);
trace(numrandom3);

//Hacemos un "Hacer mientras" para que repita el numero random hasta que me de uno
//diferente si es que me dio el mismo que alguno de los anteriores
do {
    //Le asignamos un numero random del 0 al 7 a nuestra variable
    numrandom4 = Math.random() * 9;
    //Condicionamos que si el numero random que nos dio si es diferente a los demas se hace lo mismo que
    //anteriormente y pone la imagen en una nueva posicion random
    if (numrandom4 != numrandom && numrandom4 != numrandom2 && numrandom4 != numrandom3) {
        posx4 = Arraypos[numrandom4].x;
        posy4 = Arraypos[numrandom4].y;
        cuadrito4_mc.x = posx4;
        cuadrito4_mc.y = posy4;
        cuadrito4_mc.visible = true;
    }
    //Aqui ponemos el mientras que le digo que lo haga mientras el numero random sea igual a alguno de los anteriores
} while (numrandom4 == numrandom || numrandom4 == numrandom2 || numrandom4 == numrandom3);
trace(numrandom4);

```

Movilidad con Tween

Movimiento del cuadrito (imagen/botón) hacia la derecha

Ya que tenemos nuestros pedazos de imagen acomodados y convertidos en botón crearemos una función para cada uno de ellos, lo que hicimos a continuación fue condicionar Si el eje X de nuestro cuadrito (cuadrito8_mc.x) mas 85 que es lo que mide nuestro lado del cuadrito es igual al eje X de nuestro vacío (vacío_mc.x) y además nuestro cuadrito en el eje de las Y es igual al eje de las Y de nuestro vacío (&& cuadrito8_mc.y == vacío_mc.y) esto para evitar movimientos en diagonal, aquí lo terrenalmente lo que hacemos es preguntarle “Si me recorro 85 pixeles a la derecha y al hacer esto estoy en el mismo eje de las “X” que el vacío y preguntamos si ambos estamos en el mismo eje de las “Y” quiere decir que estoy a la izquierda del vacío” si se cumple todo esto que condicionamos lo que haremos es crear un tween para nuestro cuadrito y otro para nuestro vacío y de parámetros le pondremos a cada uno que empiecen en la misma posición en la que están y terminen en el eje donde estaba el otro por lo que mediante este tween haremos un intercambio de posiciones con el vacío y aumentamos “contM (nuestro contador de movimientos)” en uno ya que cada que se haga un intercambio se contara como un movimiento en pantalla.

```

275 //Creamos la funcion Pieza8Tween que es la funcion de nuestro boton 8 para que pueda
276 //moverse hacia las 4 direcciones posibles (arriba, abajo, derecha, izquierda)
277 function Pieza8Tween(event: MouseEvent): void {
278     //Condicionamos que si el eje de las X de nuestra imagen mas 85 (que es lo que mide de longitud
279     //nuestra imagen) es igual a el eje de las X de nuestro cuadro vacío y tambien checamos que
280     //estén en la misma posición del eje de las Y para evitar que haga movimientos diagonales
281     if (cuadrito8_mc.x + 85 == vacío_mc.x && cuadrito8_mc.y == vacío_mc.y) {
282         //Si se cumple la condicion le decimos que intercambien de posiciones el cuadro y
283         //el vacío mediante un tween animado
284         var masx8: Tween = new Tween(cuadrito8_mc, "x", Strong.easeOut, cuadrito8_mc.x, vacío_mc.x, 1, true);
285         var vaciomax8: Tween = new Tween(vacío_mc, "x", Strong.easeOut, vacío_mc.x, cuadrito8_mc.x, 1, true);
286         //Aumentamos el contador de movimientos cada que se cumpla alguna condicion (un movimiento realizado)
287         contM++;
288         movim_txt.text = String(contM);
289     }
290 }

```

Movimiento del cuadrito (imagen/botón) hacia la izquierda

Para este movimiento es el mismo proceso de la parte de arriba solo que en lugar de condicionar si el cuadrito se recorre 85 a la derecha (sumarle 85) recorreremos 85 hacia la izquierda (restarle 85) y preguntar si estamos en el mismo eje de las “Y” los dos quiere decir que estamos a la derecha del vacío por lo que haremos el intercambio.

```

291 //Condicionamos que si el eje de las X de nuestra imagen menos 85 (que es lo que mide de longitud
292 //nuestra imagen) es igual a el eje de las X de nuestro cuadro vacio y tambien checamos que
293 //esten en la misma posicion del eje de las Y para evitar que haga movimientos diagonales
294 if (cuadrito8_mc.x - 85 == vacio_mc.x && cuadrito8_mc.y == vacio_mc.y) {
295     //Si se cumple la condicion le decimos que intercambien de posiciones el cuadro y
296     //el vacio mediante un tween animado
297     var menosx8: Tween = new Tween(cuadrito8_mc, "x", Strong.easeOut, cuadrito8_mc.x, vacio_mc.x, 1, true);
298     var vaciomex8: Tween = new Tween(vacio_mc, "x", Strong.easeOut, vacio_mc.x, cuadrito8_mc.x, 1, true);
299     //Aumentamos el contador de movimientos cada que se cumpla alguna condicion (un movimiento realizado)
300     contm++;
301     movim_txt.text = String(contm);
302 }
303

```

Movimiento del cuadrito (imagen/botón) hacia abajo

Para realizar este movimiento es lo mismo que hicimos para moverlo a la derecha o la izquierda solo estamos jugando con los ejes "X" y "Y", condicionamos si nuestro cuadrito (cuadrito8_mc.y) mas 85 es igual al eje de las "Y" de nuestro vacío (vacio_mc.y) y además los dos están en el mismo eje de las "X" (&& cuadrito8_mc.x == vacio_mc.x) esto para evitar movimientos en diagonal, si esto se cumple quiere decir que el vacío esta debajo de mi por lo que haremos el intercambio igual mediante el tween y tendrán los mismos parámetros que anteriormente les habíamos puesto (empiezan en el mismo eje donde están y terminan en el eje del otro) .

```

304 //Condicionamos que si el eje de las Y de nuestra imagen mas 85 (que es lo que mide de longitud
305 //nuestra imagen) es igual a el eje de las Y de nuestro cuadro vacio y tambien checamos que
306 //esten en la misma posicion del eje de las X para evitar que haga movimientos diagonales
307 if (cuadrito8_mc.y + 85 == vacio_mc.y && cuadrito8_mc.x == vacio_mc.x) {
308     //Si se cumple la condicion le decimos que intercambien de posiciones el cuadro y
309     //el vacio mediante un tween animado
310     var mayy8: Tween = new Tween(cuadrito8_mc, "y", Strong.easeOut, cuadrito8_mc.y, vacio_mc.y, 1, true);
311     var vaciomay8: Tween = new Tween(vacio_mc, "y", Strong.easeOut, vacio_mc.y, cuadrito8_mc.y, 1, true);
312     //Aumentamos el contador de movimientos cada que se cumpla alguna condicion (un movimiento realizado)
313     contm++;
314     movim_txt.text = String(contm);
315 }
316

```

Movimiento del cuadrito (imagen/botón) hacia abajo

Para realizar este movimiento es lo mismo que hicimos para moverlo a la derecha o la izquierda solo estamos jugando con los ejes "X" y "Y", condicionamos si nuestro cuadrito (cuadrito8_mc.y) menos 85 es igual al eje de las "Y" de nuestro vacío (vacio_mc.y) y además los dos están en el mismo eje de las "X" (&& cuadrito8_mc.x == vacio_mc.x) esto para evitar movimientos en diagonal, si esto se cumple quiere decir que el vacío esta encima de mi por lo que haremos el intercambio igual mediante el tween y tendrán los mismos parámetros que anteriormente les habíamos puesto (empiezan en el mismo eje donde están y terminan en el eje del otro) .


```

317 //Condicionamos que si el eje de las Y de nuestra imagen menos 85 (que es lo que mide de longitud
318 //nuestra imagen) es igual a el eje de las Y de nuestro cuadro vacio y tambien checamos que
319 //estén en la misma posición del eje de las X para evitar que haga movimientos diagonales
320 if (cuadrito8_mc.y - 85 == vacio_mc.y && cuadrito8_mc.x == vacio_mc.x) {
321     //Si se cumple la condicion le decimos que intercambien de posiciones el cuadro y
322     //el vacio mediante un tween animado
323     var menosy8: Tween = new Tween(cuadrito8_mc, "y", Strong.easeOut, cuadrito8_mc.y, vacio_mc.y, 1, true);
324     var vaciomey8: Tween = new Tween(vacio_mc, "y", Strong.easeOut, vacio_mc.y, cuadrito8_mc.y, 1, true);
325     //Aumentamos el contador de movimientos cada que se cumpla alguna condicion (un movimiento realizado)
326     contm++;
327     movim_txt.text = String(contm);
328
329 }
330 //arrancar el tiempo del timer1
331 timer1.start();
332 }
333 cuadrito8_mc.addEventListener(MouseEvent.CLICK, PiezasTween);

```

Después de condicionar cada una de las direcciones posibles, con “timer1.start()” lo que hacemos es comenzar el timer cuando se presione el botón que es cuando el jugador realiza su primer movimiento y es cuando empieza el juego, al final mandamos llamar a la función mediante el nuestro cuadrito imagen que es un botón todo este proceso que acabamos de ver se realiza exactamente igual con todos los cuadritos que nos restan (7, 6, 5, 4, 3, 2, 1) con sus respectivas funciones y así es como realizamos el movimiento con tween .

El contm que es el contador que irá aumentando cada vez que movamos los cuadritos de lugar a esto se le llamaría número de movimientos para realizar el puzzle.

Y para cuando se gane y tiene que ir a elegir el siguiente puzzle el botón de next player y juego terminado deberán estar invisibles, ahora explicaremos cuando se gane tendrá que irse al fotograma 3 donde elige el puzzle pero solo pondremos visibles los 4 puzzle y lo de ingresar datos en invisible esto hará nuestro botón de next player 1 que es el botón cuando se armó completo.

```

755 //las opciones de ganador aparezcan en falso (invisible)
756 nextPlayer1_btn.visible = false;
757 juegoTerminado1_mc.visible = false;
758
759 //FUNCION POR SI FUE MAYOR DE UN JUGADOR EL QUE INGRESÓ, LE DEMOS LA FUNCION PARA QUE ELIGA EL PUZZLE QUE QUIERA
760 function siguienteJugador1(event: MouseEvent): void {
761     gotoAndStop(3); //para irnos al frame donde estas los rompecabezas
762     unoj_mc.visible = false;
763     numeroJ_txt.visible = false;
764     aviso_txt.visible = false;
765     guardar0_btn.visible = false;
766     nombredos_mc.visible = false;
767     nombreJ_txt.visible = false;
768     aviso2_txt.visible = false;
769     guardarN_btn.visible = false;
770     puzzle1_btn.visible = true;
771     puzzle2_btn.visible = true;
772     puzzle3_btn.visible = true;
773     puzzle4_btn.visible = true;
774 }
775 nextPlayer1_btn.addEventListener(MouseEvent.CLICK, siguienteJugador1);

```

También si el usuario está jugando y no encuentra la manera de solucionarlo tendrá el botón de rendirse que al iniciarlo estará en invisible pero si ingresó más o igual de dos jugadores se nos pondrá visible el botón de rendirse y este lo que hará al darle click es mostrarnos el puzzle completo realizado y nos pondrá los 8 cuadritos y el vacío en invisibles al igual que el botón de estadísticas ; si “i” es igual al número de jugadores ingresó se nos pondrá nuestro botón de estadísticas visible y el botón de nextplayer invisibles y el tiempo se detendrá.

Por ejemplo a lo de armar solo el puzzle y rendirse hay una diferencia cuando se arma el puzzle completo nos manda directo al fotograma de las estadísticas pero supongamos que ponemos dos jugadores y a los dos les doy rendir al final cuando la “i” en este caso sea igual a numero nos mostrará el botón de estadísticas que al darle click nos mandará al fotograma 8.

```

777 //función para que muestre la imagen resuelta al rendirnos
778 nextPlayer11.btn.visible = false; //para que este en invisible y al darle rendir...(1)
779 function Frendersem(event: MouseEvent): void {
780     rendersem_btn.visible = false;
781     Tiempo[i-1]="RENDIDO"; //POR SI SE RINDIO NO MARQUE EL TIEMPO Y SLO ESTA PALABRA
782     Movimientos[i-1]="RENDIDO"; //POR SI SE RINDIO NO MARQUE EL NUMERO DE MOVIMIENTO SOLO ESTA PALABRA
783     if (numero >= 2) {
784         timer1.stop(); //para el tiempo se detenga cuando se rinde
785         nextPlayer11.btn.visible = true; //..(1) se ponga visible//por si se rinde el jugador y puso mas de uno se pone en visible este
786         //Ponemos en invisible nuestros pedazos de imagen junto con el vacío
787         cuadrito1_mc.visible = false;
788         cuadrito2_mc.visible = false;
789         cuadrito3_mc.visible = false;
790         cuadrito4_mc.visible = false;
791         cuadrito5_mc.visible = false;
792         cuadrito6_mc.visible = false;
793         cuadrito7_mc.visible = false;
794         cuadrito8_mc.visible = false;
795         vacio_mc.visible = false;
796         estadisticas_btn.visible=false;
797         //Ponemos en visible nuestra imagen ya resuelta
798         rendersemMax_mc.visible = true;
799     }
800     if (i == numero) {
801         estadisticas_btn.visible=true;
802         nextPlayer11.btn.visible = false; //para que se ponga en invisible el boton de next player
803         timer1.stop(); //si se va a este ya no entra al de abajo por eso el otro stop
804     }
805 }
806 rendersem_btn.addEventListener(MouseEvent.CLICK, Frendersem);
807

```

Y como vemos en las líneas 781 y 782 como se rindió en las estadísticas no mostrará el tiempo ni los movimientos solo aparecerá el nombre pero con el texto “RENDIDO” tanto como para tiempo y numero de movimientos.

aa	RENDIDO	RENDIDO
ss	RENDIDO	RENDIDO

Como vemos la función de nuestro botón de estadísticas que aparece cuando se muestra al final si te rendiste en el último o en todos los turnos de juego al darle click nos manda al fotograma 8.

Nuestra función de siguiente jugador11 es el que sale al elegir cuando se rinden la primera vez y su función es mandarnos al frame para elegir el puzzle y se pongan en invisible los campos para ingresar numero o nombres de jugadores y solo se pongan visibles los 4 puzzles.

```
//FUNCIO PARA CUANDO SE IGUALE EL NUMERO DE JUGADOR EN TURNO CON EL NUMERO DE JUGADORES QUE INGRESÓ
function estadisticas(event:MouseEvent):void{
    gotoAndStop(8);
}
estadisticas_btn.addEventListener(MouseEvent.CLICK, estadisticas);
//FUNCION SI LE DIO RENDIR E INGRESO MAS DE UN JUGADOR, LE DEMOS LA FUNCION PARA QUE ELIGA EL PUZZLE QUE QUIERA
function siguienteJugador11(event: MouseEvent): void {
    gotoAndStop(3); //para irnos al frame donde estas los rompecabezas
    //PARA LO DE INGRESAR YA NO APAREZCA
    unoj_mc.visible = false;
    numeroJ_txt.visible = false;
    aviso_txt.visible = false;
    guardar0_btn.visible = false;
    nombredos_mc.visible = false;
    nombreJ_txt.visible = false;
    aviso2_txt.visible = false;
    guardarN_btn.visible = false;
    //PERO LOS PUZZLES A ELEGIR SI APAREZCAN
    puzzle1_btn.visible = true;
    puzzle2_btn.visible = true;
    puzzle3_btn.visible = true;
    puzzle4_btn.visible = true;
}
nextPlayer11_btn.addEventListener(MouseEvent.CLICK, siguienteJugador11);
```

Fotograma 8

Ponemos nuestras respectivas librerías que vamos a usar en este caso para las de mouse event y para las animaciones que utilice para los globos del fotograma de estadísticas y para el título de resultados.

Añadimos nuestro contador u que inicializa en 0 y posteriormente agregamos un ciclo para que inicia en 0 hasta u menor a número y se irá aumentando uno en uno en cada iteración.

```
actions:8
1  import flash.events.MouseEvent;
2  import fl.transitions.Tween;
3  import fl.transitions.easing.*;
4  import fl.transitions.TweenEvent;
5
6  //nuestro tween para nuestro titulo de resultados
7  var noneInOut:Tween =new Tween (resultados_mc, "x", None.easeIn, 6, 169, 2, true);
8  var noneInOut1:Tween =new Tween (globol_mc, "y", None.easeIn, 6, 250, 2, true);
9  var noneInOut2:Tween =new Tween (globo2_mc, "y", None.easeIn, 6, 250, 2, true);
10
11  var u:int=0;//nuestro contador
12  for(u=0;u<numero;u++) //inicializa en 0, hasta que sea menor a numero de jugadores el que ingreso, y en cada iteracion va aumentando
13  {
14      estadisticas_txt.text+=NombreJugadores[u]+"*Movimientos[u]+"
15  }
16
17  function notas(event:MouseEvent):void{
18      var resultados:String;
19      resultados=estadisticas_txt.text;
20      var notas:FileReference = new FileReference();
21      notas.save(resultados, "resultados.txt");
22  }
23  notas_btn.addEventListener (MouseEvent.CLICK, notas);
24
25
```

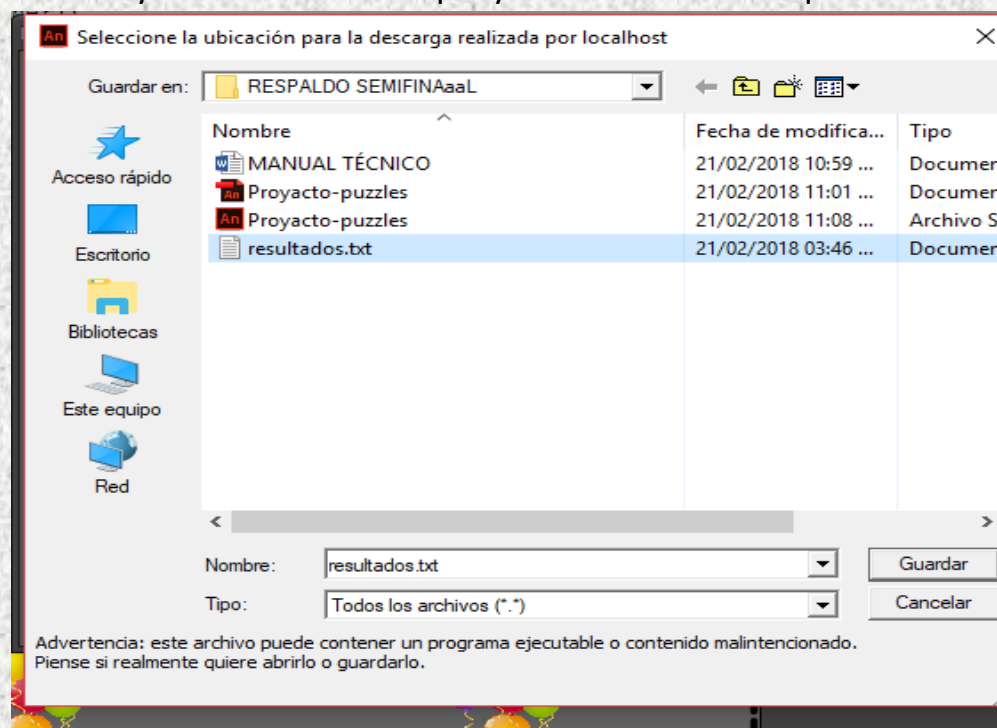
Explicaremos ahora de nuestra función para exportar en bloc de notas nuestro resultados (estadísticas) en este caso definimos resultados como string y a este resultado le vamos a sumar lo que se encuentra en nuestro text área y después notas.save para guarda lo guardaremos resultados lo que contiene en el archivo de referencia en este caso tenemos que seleccionar nuestro archivo de referencia y como ya lo tenemos en nuestra carpeta que es el siguiente:



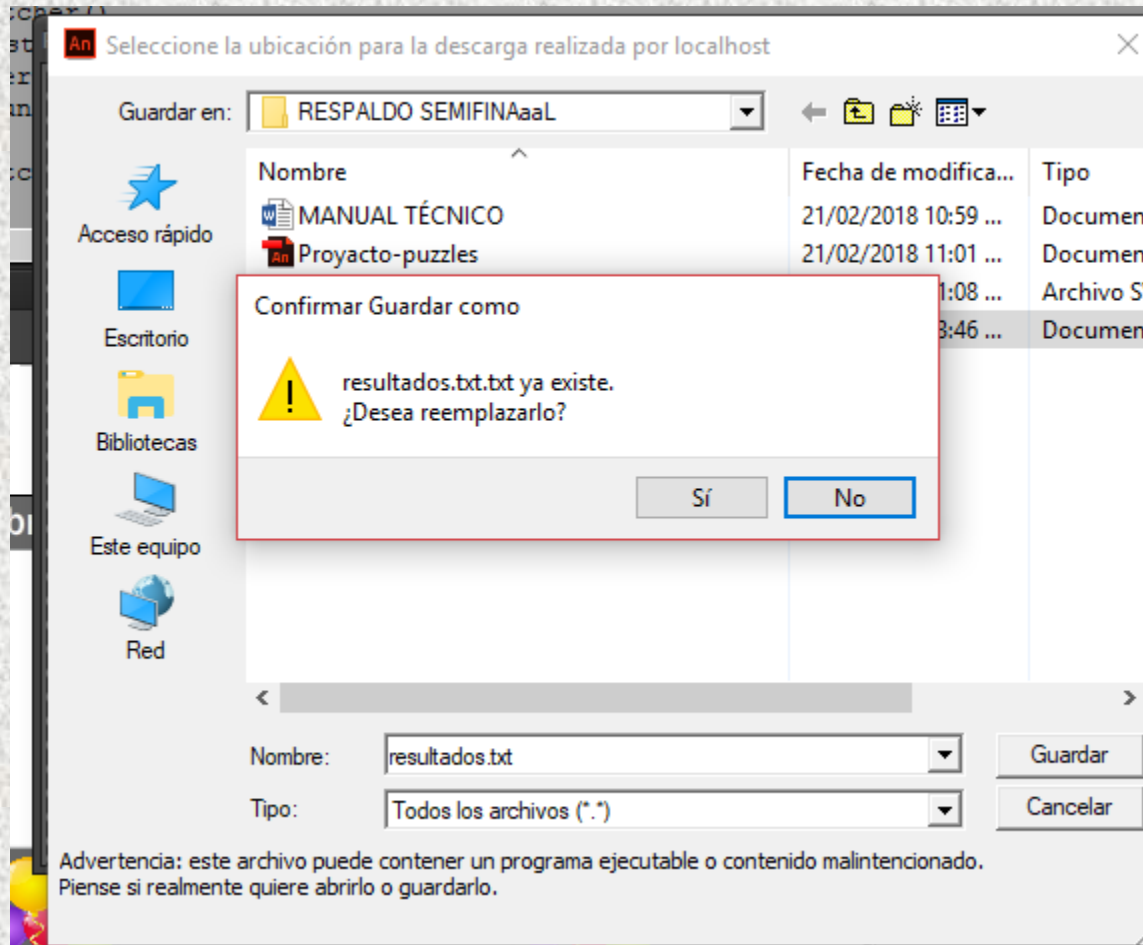
Así es como aparecen nuestros resultados al finalizar y observamos que el primer jugador se rindió.



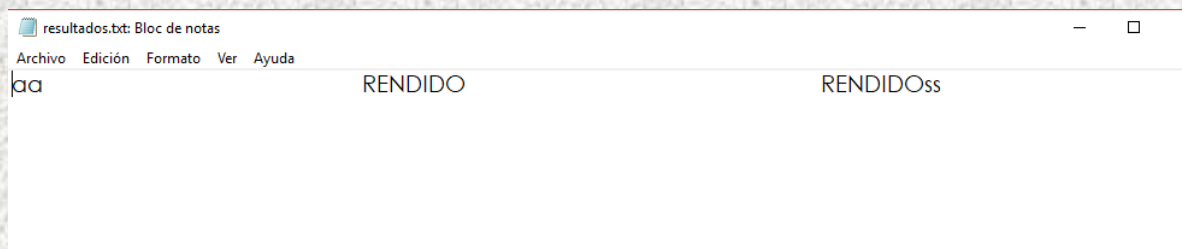
Al darle click en “GUARDAR EN BLOC DE NOTAS” nos aparecerá lo siguiente y seleccionamos el que ya tenemos de la carpeta.



Le damos en reemplazar.



Y es como nos muestra ya en el bloc de notas nuestro resultado del text área.



Y así fuimos haciendo los otros puzzles pero con diferentes nombre de instancias.

CONCLUSIONES

Conclusión personal (Kevin)

Al finalizar el juego, mis aprendizajes fueron:

- Aprendí a manejar Arrays.
- Aprendí a manejar funciones y Tween.
- Aprendí a manejar herramientas del flash como por ejemplo el separar para cortar la imagen en partes iguales y así convertir cada una en un botón diferente.
- Aprendí manejo de variables en otros fotogramas (acumuladores, auxiliares, en ciclos, etc...).

Era algo que ya habíamos implementado en ejercicios pasados, pero yo no me había familiarizado con ellos y, al elaborar este proyecto, tuve que esforzarme por entender la lógica que esto conllevaba.

Además, aprendí a trabajar en equipo y saber repartir el trabajo; tuve que buscar por mi cuenta muchas cosas que tal vez no entendía y después explicárselas a mi compañera de equipo.

Conclusión personal (Claudia)

Tuve muchas experiencia y sabiduría al terminar el juego ya que aprendí a cómo crear un juego de puzles, a como poner el timer en minutos y aprendi otras cosas entre ellas como exportar resultados en este caso a un bloc de notas y me sirvió para reforzar lo que ya habíamos aprendido y a usar mejor el math random y validar ciertas cosas, trabajar en equipo con compañeros que nunca había compartido algún trabajo juntos y tener una buena colaboración para apoyarnos en lo que se nos dificultaba y en este caso pudimos trabajar bien ya que a los dos nos toco cosas por igual y entre los dos las íbamos haciendo para reforzar os temas ya vistos y nuestra sabiduría aplicarla.