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

[Introducción](#introduccion)………………………………3

[Desarrollo](#desarrollo)………………………………….4

* [Fotograma1](#fotograma1)……………………….4
* [Fotograma2](#fotograma2)……………………....6
* [Fotograma3](#fotograma3)……………………….7
* [Fotograma4](#fotograma4)………………………
* [Fotograma8](#fotograma8)………………………

[Conclusiones](#conclusiones)……………………………..

* Conclusión Kevin
* Conclusión Claudia



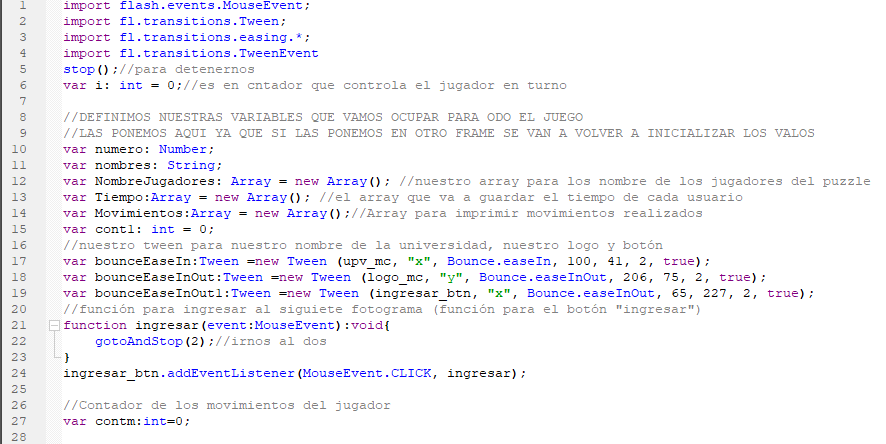
INTRODUCCIÓN

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 puzles 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 seria 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.

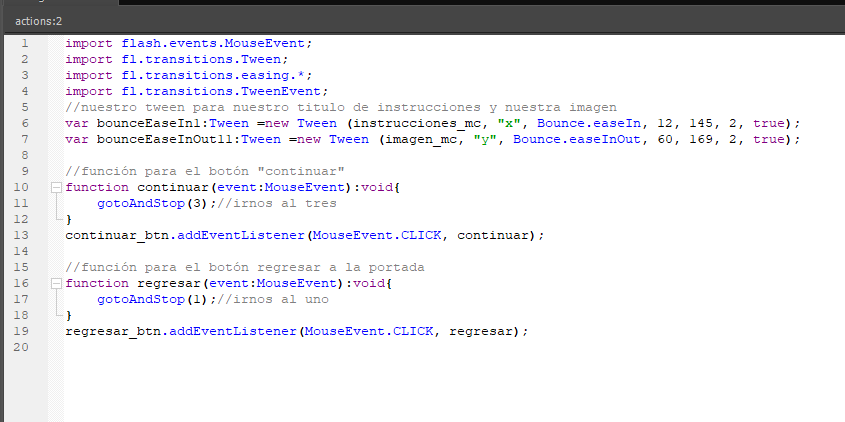


Así es la portada y las acciones para esta.

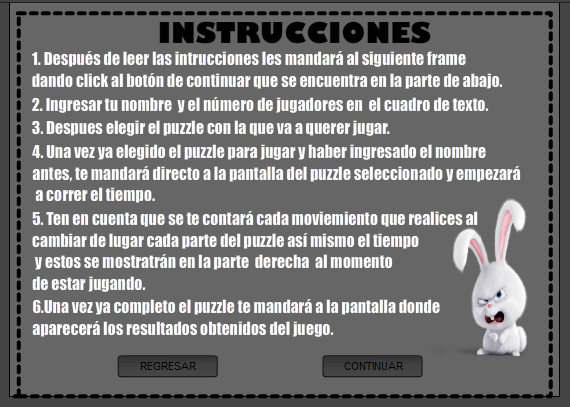


Fotograma 2

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.



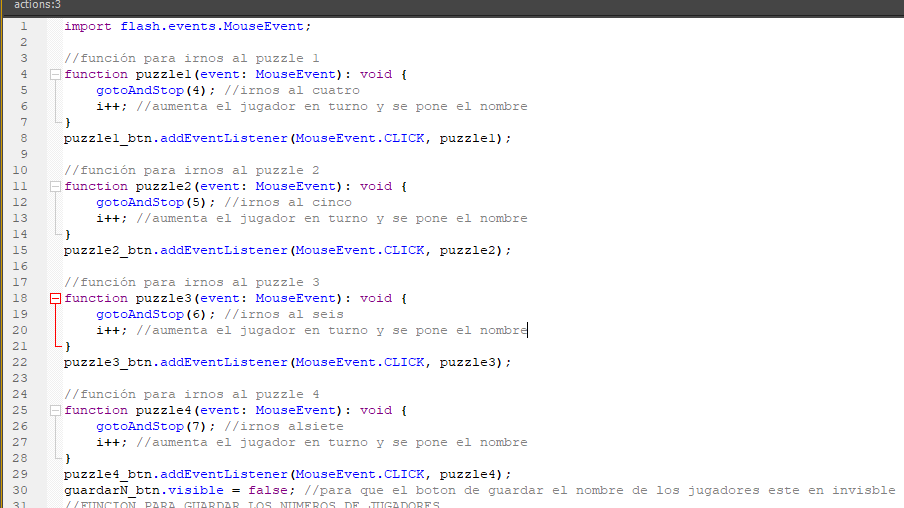
Las acciones son para las funciones del fotograma de las instrucciones.



Fotograma 3

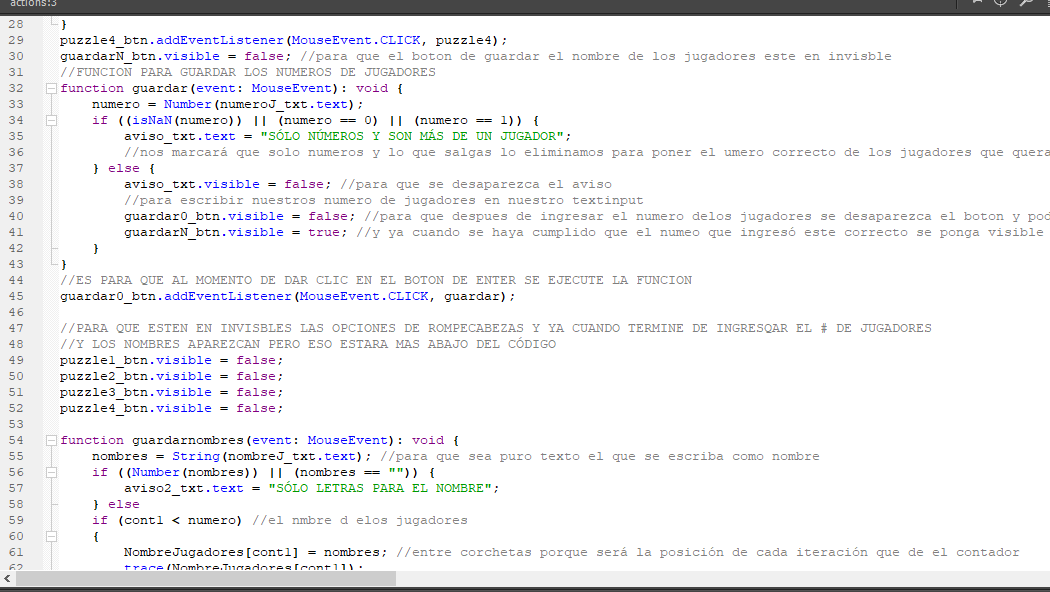
En las acciones para nuestro fotograma 3 pusimos nuestra librería para al darle clic en cada función lo explicamos a continuación:

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 puzles para irnos al numero de frame respectivo.

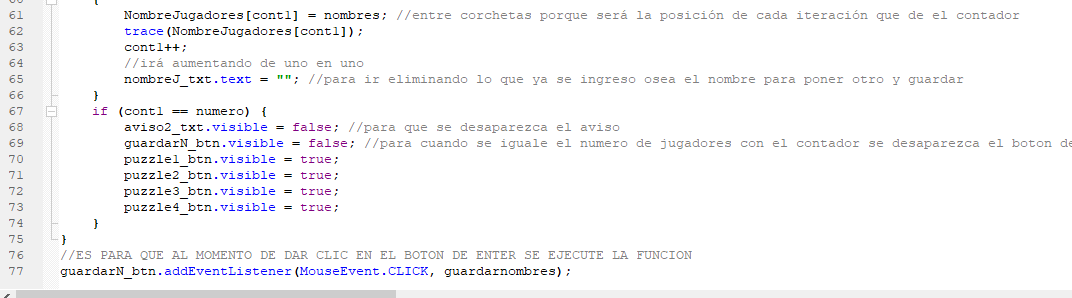


Después vamos a validar que la información que ha ingresado sea correcta en caso contrario de no ser así nos mostrará un aviso en un texto dinámico que ingresamos, nuestra función tiene el nombre de guardar que se cumplirá cuando oprimimos el botón que solo se nos muestra el de guardar número de jugador, 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 puzles siguen en invisibles.

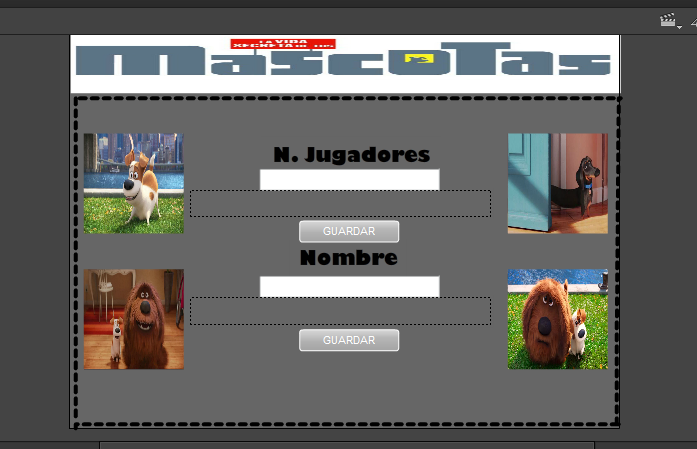
Ahora explicaremos lo que hace la función de guardar nombres, hacemos lo mismo validamos que si nombre nuestra variable es un numero o nombre no hay nada en el textInput nos muestre un aviso de “SÓLO LETRAS PARS EL NOMBRE” si no, si cont1 es menor a número (de jugadores) se guardará en el array de NombreJugadores 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.



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 puzles para que el jugador en turno escoja el de su agrado para empezar a jugar .

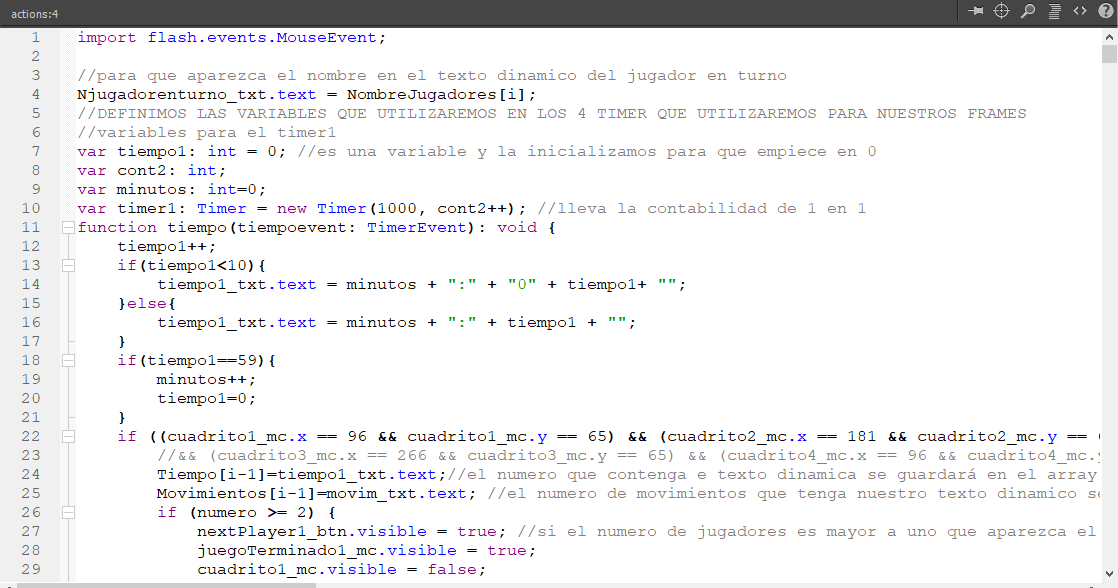


Aquí el fotograma de las acciones 3, podemos observar los 4 puzzles nuestros texinput el primero donde ingresará el número de jugadores nuestro aviso1 el botón de guardar, y nuestro segundo texinput de nombre nuestro aviso2 y nuestro botón para ir guardarndo los nombre en el array.



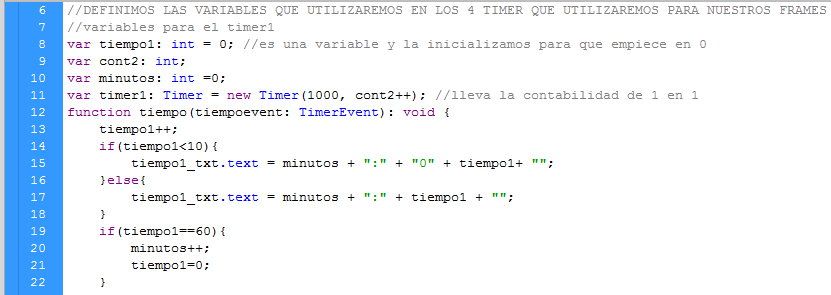
Fotograma 4

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

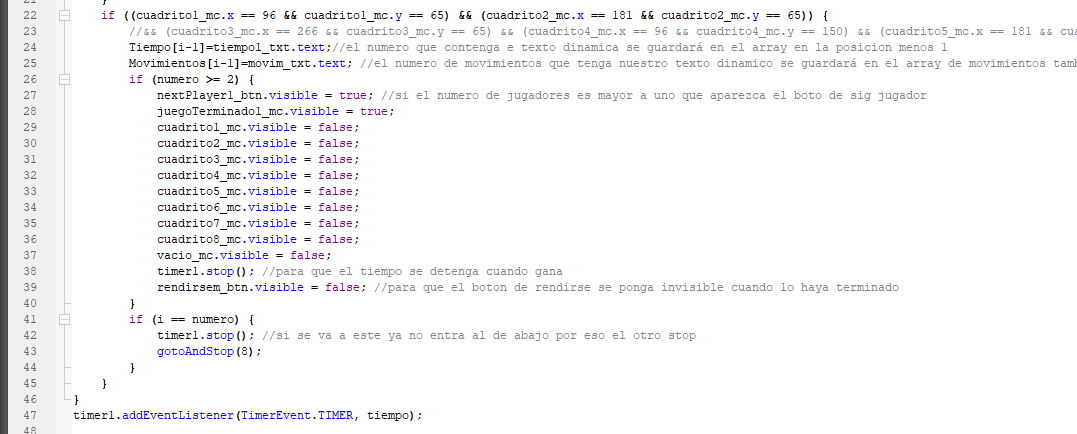


**Implementacion del timer con minutos y segundos**

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.



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. 

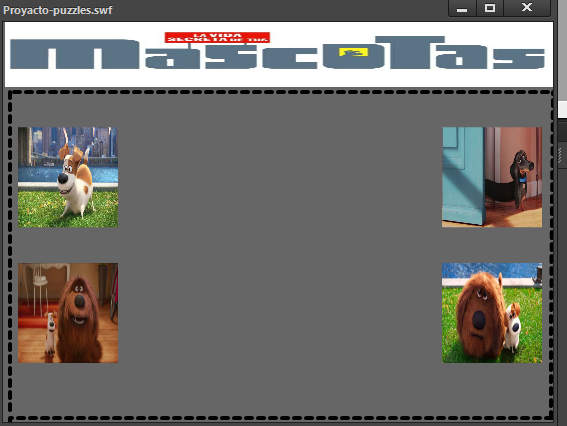
Para eso en cada entrada al puzzle nuestra “i” irá aumentando 1 en 1.



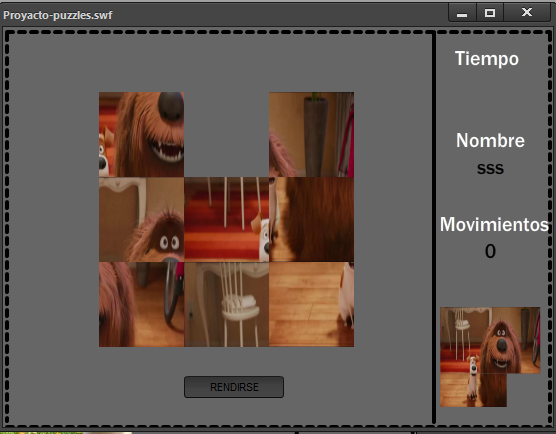
Así nos mostrará cuando se haya armado completo para pasar al siguiente jugador y que elija su puzzle y como vemos a lado izquierdo nuestro nombre del jugador en turno con su respectivo tiempo y numero de movimientos con el que quedó.



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)**

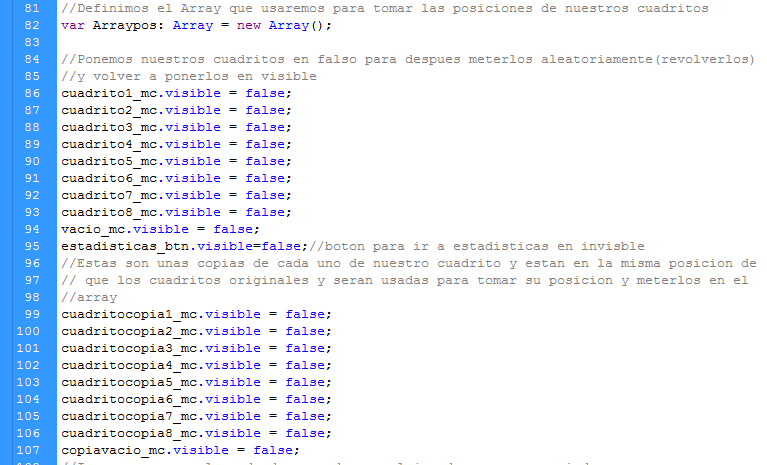
**Definición de variables**

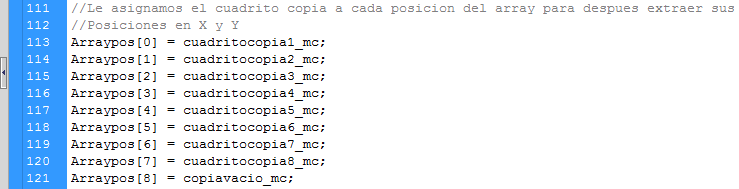
Lo que hacemos aquí es definir las variables que ocuparemos para guardar el número random de cada uno de los pedazos incluyendo el vacío, definimos también las variables que nos servirán para guardar su posición en el eje “X” y el eje ” Y” y por ultimo definimos el Array que nos ayudara a guardar las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



**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.



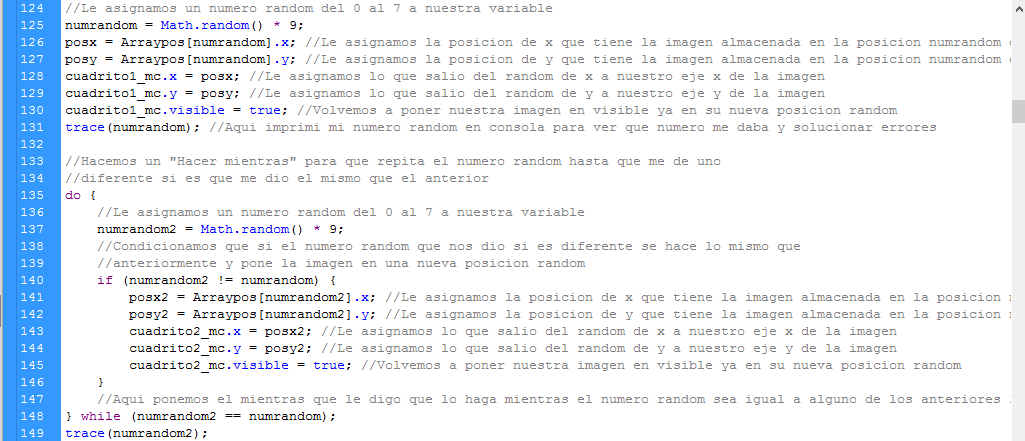
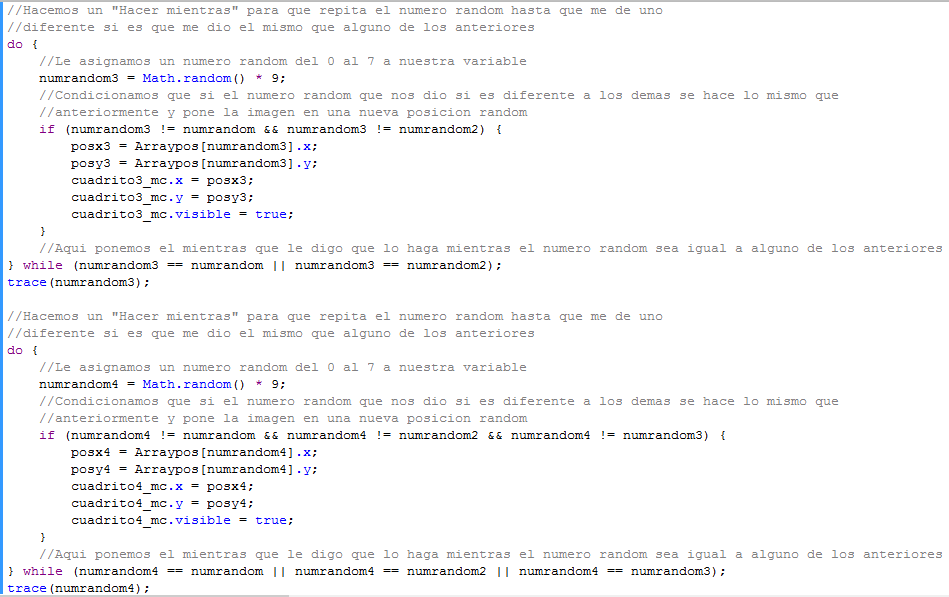


**Acomodo en random del pedazo de imagen**

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”**

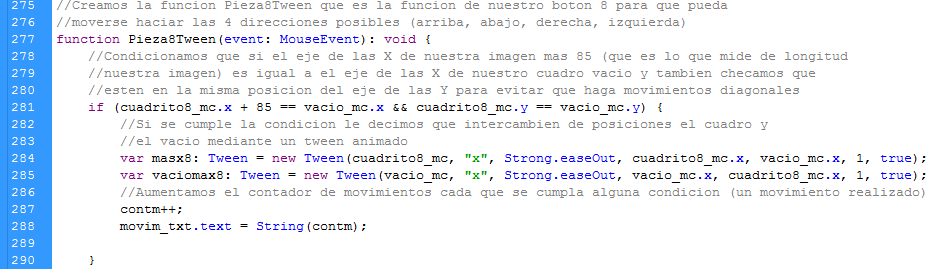
Lo que hacemos en el do While es lo mismo que en el primer acomodo random solo que aquí introducimos una condición la cual nos dice que si el número random que me dio es distinto al número random que me había dado anteriormente entonces que haga todo el proceso de acomodo si no se cumple la condición no hace la asignación por lo que se podría leer nuestro While de la siguente forma “Hacer lo siguente, 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.



**Movilidad con Tween**

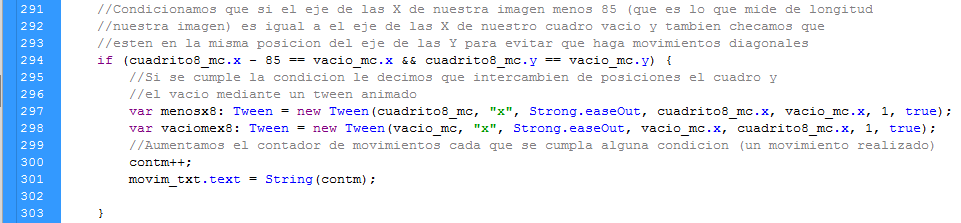
**Movimiento del cuadrito (imagen/botón) hacia la derecha**

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 vacio (vacio\_mc.x) y además nuestro cuadrito en el eje de las Y es igual al eje de las Y de nuestro vacio (&& cuadrito8\_mc.y == vacio\_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.



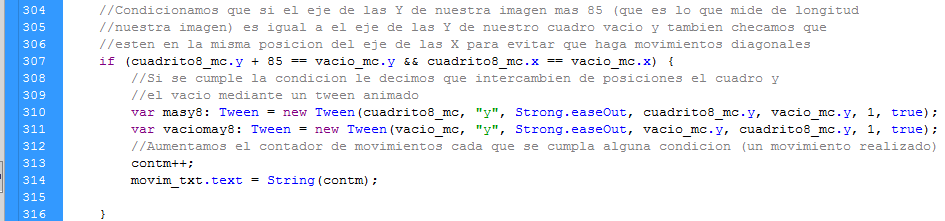
**Movimiento del cuadrito (imagen/botón) hacia la izquierda**

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.



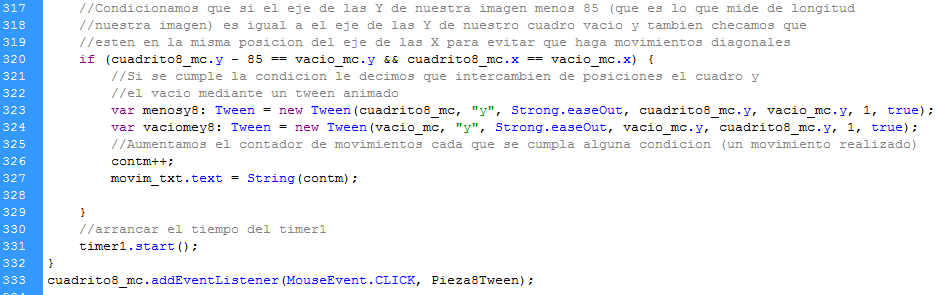
**Movimiento del cuadrito (imagen/botón) hacia abajo**

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) .



**Movimiento del cuadrito (imagen/botón) hacia abajo**

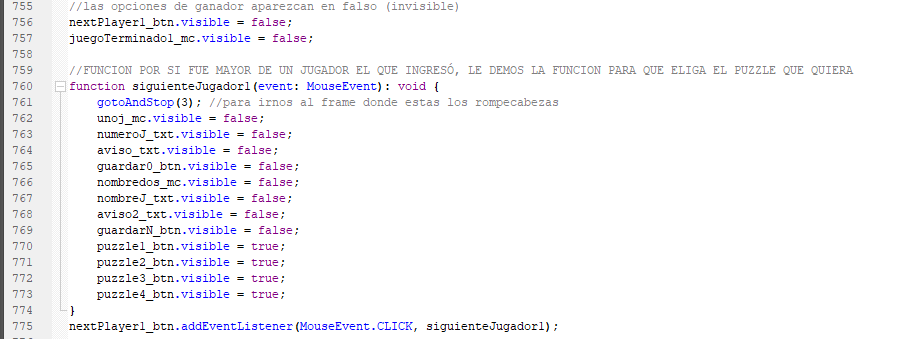
Para realizar este movimiento es lo mismo que hicimos para moverlo a la derecha o la izquierda solo estamos jugando con los ejes “X” y “Y”, condicionamos si nuestro cuadrito (cuadrito8\_mc.y) menos 85 es igual al eje de las “Y” de nuestro vacío (vacio\_mc.y) y además los dos están en el mismo eje de 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) .



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 realizo 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 .

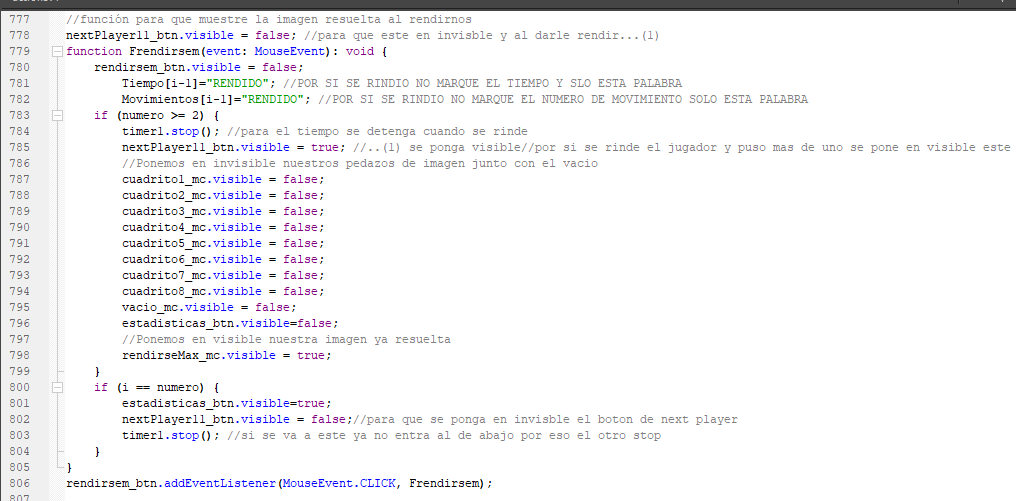
El contm que es el contador que irá aumentando cada vez que movamos los cuadritos de lugar a esto se le llamaría número de movimientos para realizar el puzzle.

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.

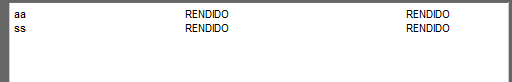


También si el usuario está jugando y no encuentra la manera de solucionarlo tendrá el botón de rendirse que al iniciarlo estará en invisible pero si ingresó más o igual de dos jugadores se nos pondrá visible el botón de rendirse y este lo que hará al darle click es mostrarnos el puzzle completo realizado 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.

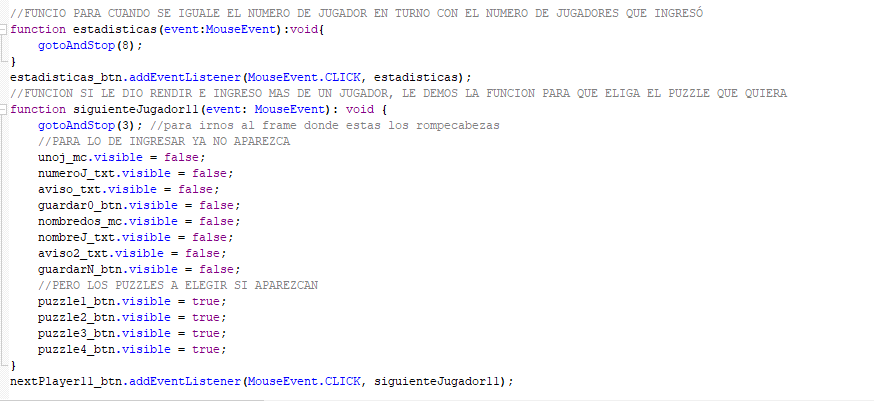


Y como vemos en las líneas 781 y 782 como se rindió en las estadísticas no mostrará el tiempo ni los movimientos solo aparecerá el nombre pero con el texto “RENDIDO” tanto como para tiempo y numero de movimientos.



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.

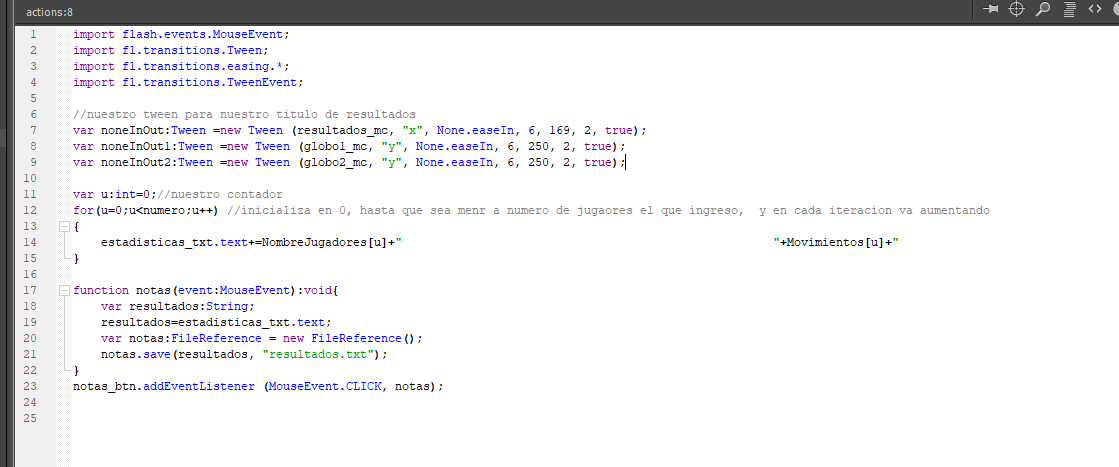
Nuestra función de siguiente jugador11 es el que sale al elegir cuando se rinden la primera vez y su función es mandarnos al frame para elegir el puzzle y se pongan en invisible los campos para ingresar numero o nombres de jugadores y solo se pongan visibles los 4 puzzles.



Fotograma 8

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

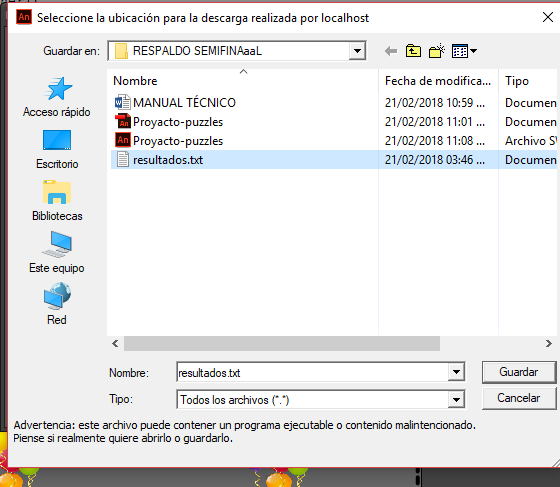
Añadimos nuestro contador u que inicializa en 0 y posteriormente agregamos un ciclo para que inicia en 0 hasta u menor a número y se irá aumentando uno en uno en cada iteración.



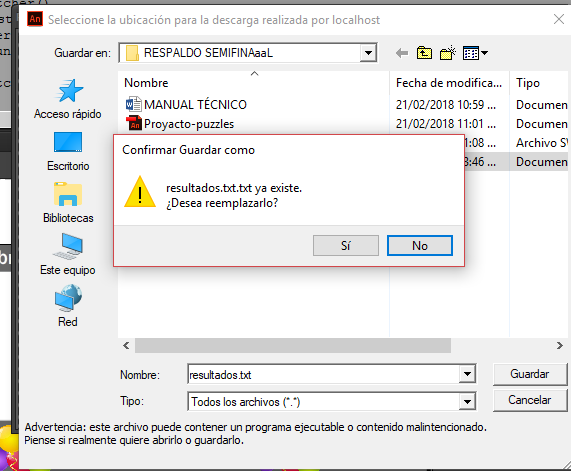
Explicaremos ahora de nuestra función para exportar en bloc de notas nuestro resultados (estadísticas) en este caso definimos resultados como string y a este resultado le vamos a sumar lo que se encuentra en nuestro text área y después notas.save para guarda lo guardaremos resultados lo que contiene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



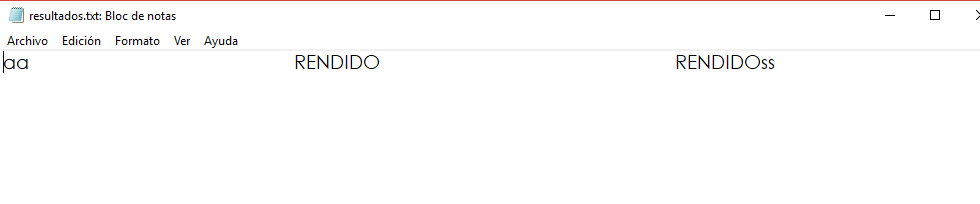
Asi es como aparecen nuestro resultados al finalizar y observamos que el primer jugador se rindió.

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

Le damos en reeplazar.



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



Y asi fuimos haciendo los otros puzles 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 Tweens.
* 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.