**MANUAL TÉCNICO**

**Desarrollado por: Miguel Santiago Luna Villanueva y Eduar Rafael Lopez Zuñiga**

Este manual técnico va enfocado sobre cómo se desarrollo el juego Tetris para la clase de Herramientas Multimedia, la cual la imparte el M.S.I. Mario Humberto Rodríguez Chávez. Antes de comenzar a explicar el proceso del juego, se dictará el orden que tomaremos para explicar cada cosa que se hizo. Primero se empezará con el diseño del juego. Seguido de eso empezaríamos con la programación de este.

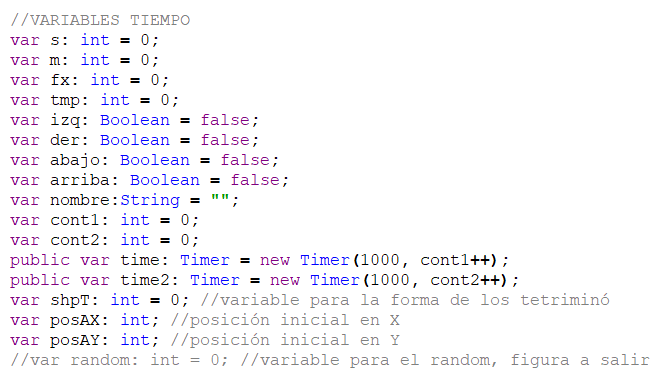
En el primer fotograma se encuentra la portada del juego, en esta se encuentra el título, que es Tetris. Un cuadro de texto, que nos menciona a los desarrolladores del juego. Contiene también un botón el cual dice “Play”, pulsando el botón.

En el siguiente fotograma se encuentran 3 botones, una para jugar al juego clásico de Tetris, el segundo para ir a un frame de opciones, y el tercero para regresar a la portada del proyecto.

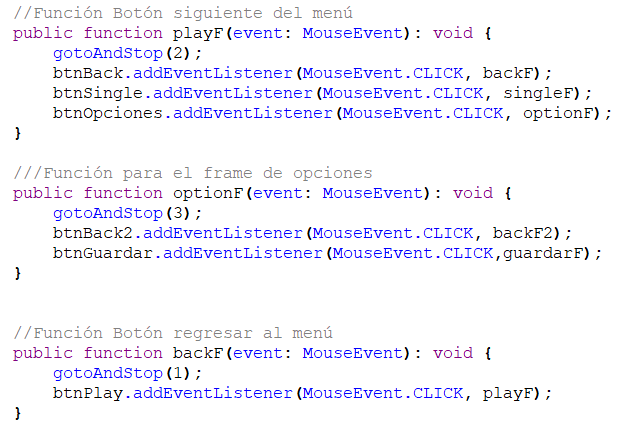
Seguimos al siguiente frame y está el menú de opciones, aquí puedes cambiar tu nombre de usuario.

En el cuarto frame, es donde y se encuentra el juego, contiene el escenario, un timer, dos botones, para pausar o salir del juego, y el nombre del jugador.

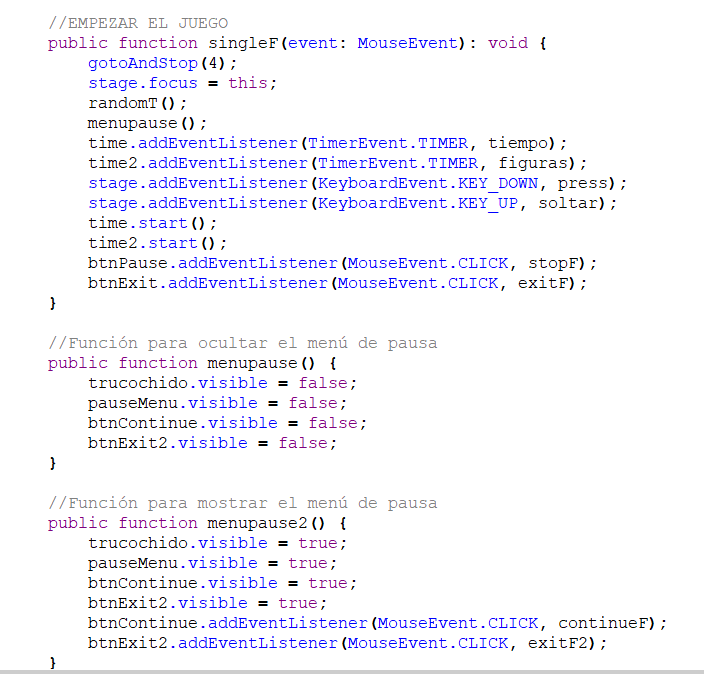
Pasando a lo que es el código sería lo siguiente:



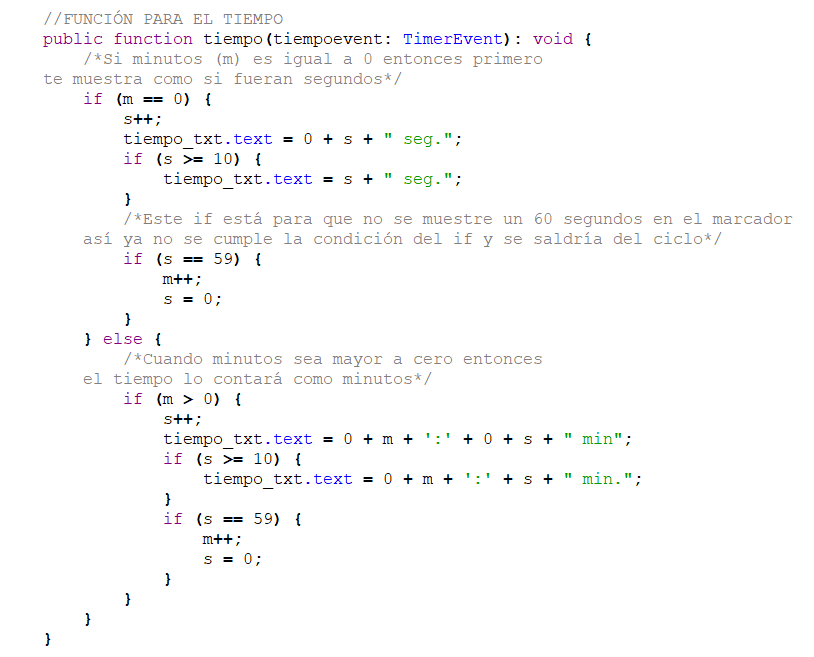
Aquí se encuentran la mayoría de las variables utilizadas para el juego, las primeras dos sirven para el timer, la primera de segundos y la segunda de minutos. La cuarta variable, es el tiempo para el segundo Timer (se explicará más adelante). Las cuatro variables booleanas son para las teclas del mouse, la variable nombre almacena el nombre del jugador, los contadores sirven para los Timer’s. La variable shpT nos sirve para saber en qué forma está nuestro tetriminó.

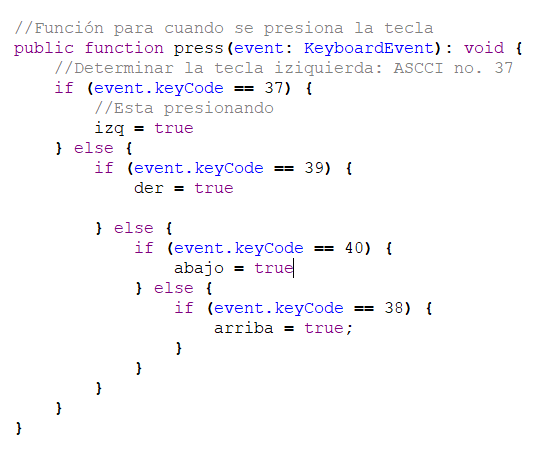


Las siguientes funciones, en orden descendente sirven para: la primera para avanzar al menú principal del juego. La segunda nos lleva al menú de opciones, y el tercero es para regresar a la portada del proyecto.

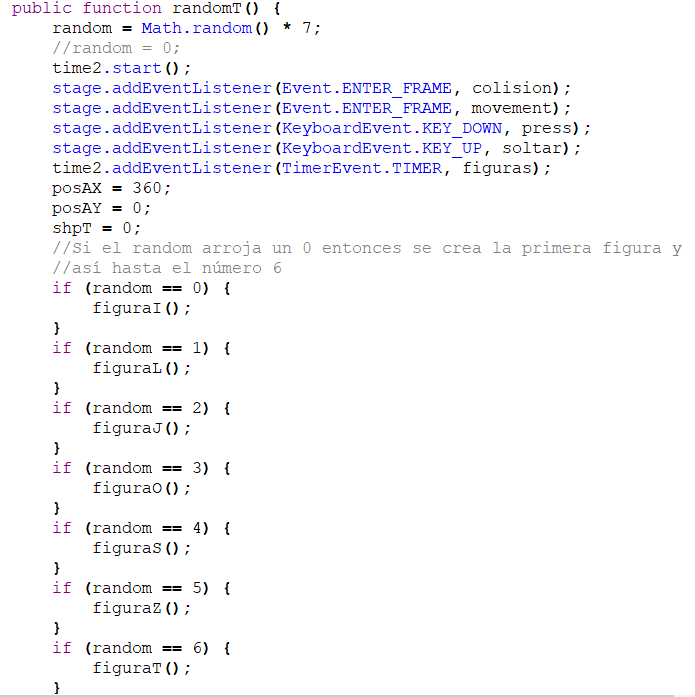


Las siguientes funciones son: la primera para ir al juego, se inician todas las funciones. La segunda y tercera son para el menú de pausa, la primera lo oculta, y la segunda lo muestra, ya si el usuario le presiona al botón de pausa pues manda llamar la función dentro de otra función. Y así con la tercera función.

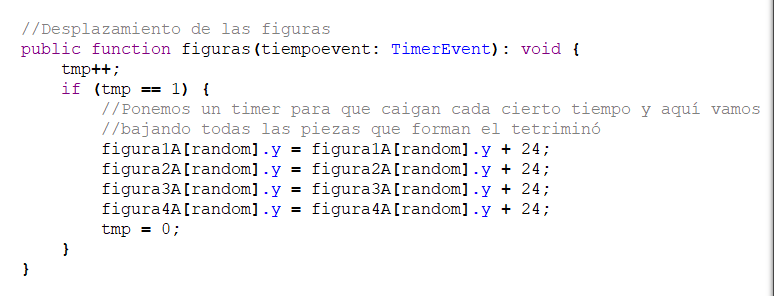




Esta función nos ayuda a saber si se están presionando las flechas y la de debajo de ella es lo mismo, sólo que para cuando no las presionan. Y con esto terminamos las funciones más fáciles de entender, a partir de ahora empiezan las funciones para el desarrollo de el juego Tetris en si.



Esta primera función es para el random, qué figura nos lanzará el juego, en qué posición spawneará, la forma que tendrá y ya, dependiendo del random será la figura a generar.

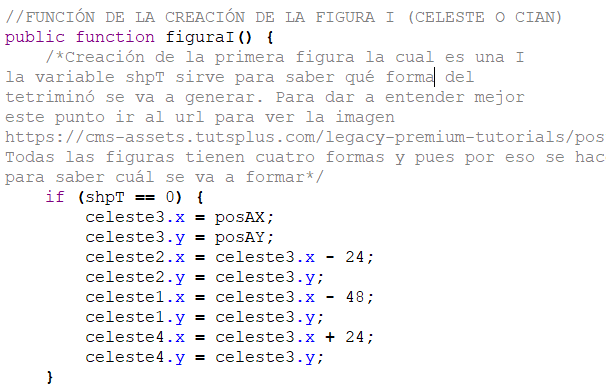


Esta función sirve para el movimiento en y del tetriminó.

Debido a que no es un movie clip junto, sino separados, tenemos que ir bajando todas las piezas juntas, y estas bajan cada segundo.

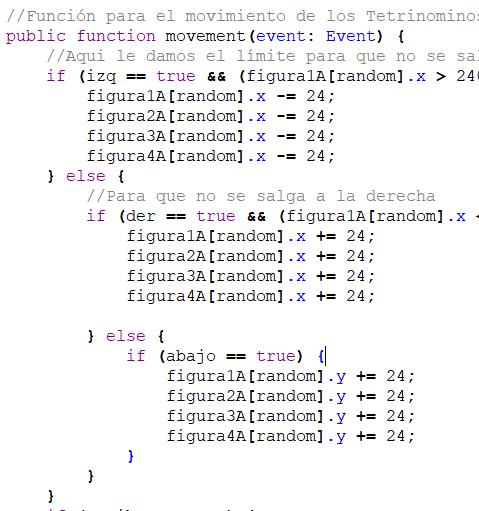
FUNCIONES PARA LA GENERACIÓN DE LOS TETRIMINÓS

FIGURA I



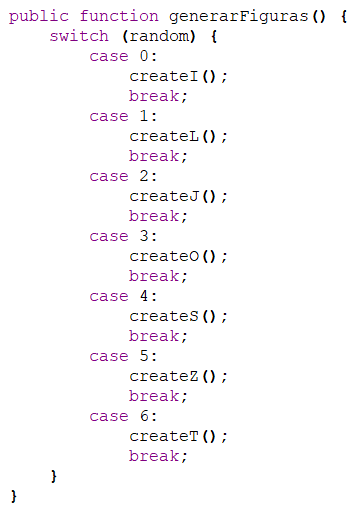
Creación de la primera figura la cual es una I la variable shpT sirve para saber qué forma del tetriminó se va a generar. Para dar a entender mejor este punto ir al url para ver la imagen https://cms-assets.tutsplus.com/legacy-premium-tutorials/posts/13778/images/13778\_402cf36c3b2daeef3aeb3d6e26ca2223.png

Todas las figuras tienen cuatro formas y pues por eso se hace la condición para saber cuál se va a formar. Sólo es necesario mostrar esto debido a que se repite la función, lo único que cambia es el MovieClip a utilizar y que la figura O no cambia.



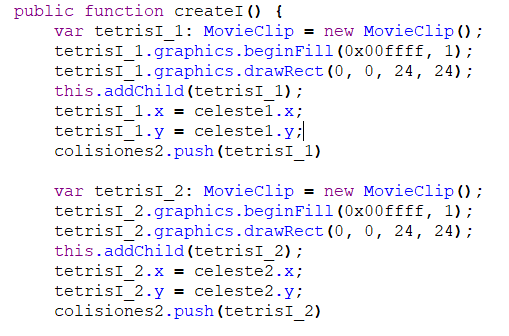
Función para darle movimiento en x a la figura y el cambio de forma

La siguiente función sirve para saber después de la colisión, qué figura se generará en el escenario.

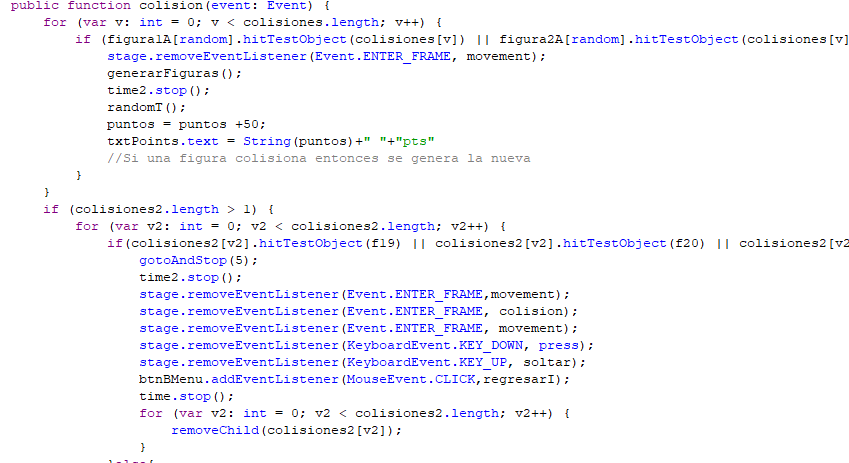


Después de esto se mostrará cómo generarla.

Las siguientes funciones sirven para crear nuevos MovieClip a partir de los ya creados (los tetriminos que caen). Se crea una nueva variable, se le da color y dimensiones, después la agregamos al escenario con addChild, le damos las coordenadas, y por último la metemos en un Array con un push, esto para cada cuadro del tetriminó y así para cada tetriminó. Esta primera función es para la figura I.



FUNCION PARA LAS COLISIONES



Esta función, la primera vez que entra al for, comprueba si la primera figura generada choca con el fondo de la matriz de cuadros. Esto con una propiedad llamada hiTestObject, que lo que hace es verificar si una imagen, cuadro o alguna propiedad en x y y esta chocando con otra. Entonces lo que hacemos es verificar cada cuadro con diferentes opciones y asi es como vemos si entra al final. Después si esta colisión entra detiene el tiempo que hace mover a las figuras y genera una nueva con el código de arriba.

Aquí mismo se encuentra la manera en que el usuario va a perder. De igual manera verifica que un hiTestObjet suceda, pero esto directamente en la parte donde salen las figuras, es decir saber si el usuario ya tiene una pila grande de figuras formadas.