

TETRIS UN POCO DE SU HISTORIA

Es un videojuego de rompecabezas que combina azulejos, diseñado y programado originalmente por el diseñador de juegos ruso Alexey Pajitnov. Fue lanzado el 6 de junio de 1984, mientras trabajaba para el Centro de Computación Dorodnitsyn de la Academia de Ciencias de la Unión Soviética en Moscú. Derivó su nombre del prefijo numérico griego tetra- (todas las piezas del juego contienen cuatro segmentos) y del tenis, el deporte favorito de Pajitnov. El juego, o una de sus muchas variantes, está disponible para casi todas las consolas de videojuegos y sistemas operativos informáticos, así como en dispositivos como calculadoras gráficas, teléfonos móviles, reproductores multimedia portátiles, PDA, reproductores de música en red y como huevo de Pascua en productos no multimedia como osciloscopios.

EL DESARROLLO DEL TRABAJO

EXPLICACIÓN

Este juego de bloques está programado con JavaScript y utiliza **canvas** para pintar el juego. También utilizo Bootstrap para el diseño de los botones y la página en general, con un poco de Sweet alert para las ventanas emergentes que permiten reiniciar e iniciar el juego.

El juego cuenta con sonidos, los cuales se utilizan para la música de fondo, sonido cuando la pieza no se puede rotar, cuando se hace una fila completa y cuando el tetrominó toca el final. Por otro lado, cuenta con **rotaciones**, donde las piezas se pueden rotar para acomodarlas y acumular puntos. Además, el código permite **que** el juego se puede pausar o reanudar en cualquier momento.

Algoritmo general

El algoritmo es sencillo. Tenemos un plano cartesiano en donde existen las coordenadas **X** e **Y**. Tenemos 3 cosas:

1. Tablero de juego: el tablero en donde se dibujará o mostrará todo.
2. Piezas ya existentes: los puntos o piezas que ya han caído antes; es decir, las figuras que se quedaron ahí.
3. Pieza actual: la pieza que va bajando actualmente y que el jugador puede mover o rotar.

Para realizar el juego dibuje el tablero del juego (vacío), luego sobrepuse las piezas existentes y finalmente coloque la pieza actual (la que va bajando). En cada movimiento o intento de rotación se verifica si la pieza no colapsa con la pared o con otra figura abajo. Cuando la pieza se rota, se hace una simulación para ver si los puntos, después de haber sido rotados, no colapsarán con otra pieza. Además, cuando se detecta que la pieza ha llegado al suelo, se inicia un temporizador que bajará la siguiente pieza en determinados milisegundos (de este modo el jugador tiene tiempo para mover la pieza). Antes de mostrar otra figura, se mueven los puntos de la figura actual a las piezas existentes.

El resto de elementos son colisiones y trabajos con arreglos. Por ejemplo, para verificar si una fila está llena recorreremos cada punto del arreglo en determina posición de **Y** y verificamos si está tomado.

Clases

Punto: un punto tiene dos cosas: coordenadas X, Y. Y un Tetrominó tiene varios puntos que lo conforman.

Tetrominó: Las rotaciones se definen igualmente como un Tetrominó, y se intercambian al rotar, Las rotaciones no son más que un arreglo que tiene varios Tetrominós, mismo que representa todas las rotaciones posibles.

Funciones útiles: Entre ellas tenemos la función que carga el sonido, la que elige un color aleatorio o la que elige un número aleatorio para saber cuál figura elegir.

Controles para jugar

El juego de bloques o tetris en JavaScript que he programado se puede jugar con el teclado o con los botones. Cada botón tiene un id, que luego recupero dentro del juego usando querySelector.

Función init

Todo comienza en la función INIT, pero también es importante el constructor en donde se definen varias cosas. El juego recibe el id del canvas en donde se va a dibujar.

Obtener figura aleatoria

La función en donde definimos las figuras y sus rotaciones. Este método devolverá un Tetrominó aleatorio en cada invocación:

Loop: ciclo del juego

Según el intervalo en milisegundos definido anteriormente va bajando la pieza, mientras se pueda. En caso de que ya no se pueda mover la pieza, se inicia un temporizador para darle oportunidad al jugador de mover la pieza. Cuando se acaba el temporizador, si la pieza sigue sin poder moverse, se agrega la pieza actual a las piezas existentes y luego se selecciona otra figura.

