

TETRIS

Proyecto de Desarrollo de un Videojuego

César Tejero Manovel



Introducción

El proyecto consiste en la creación de un videojuego del tipo y complejidad que el alumno desee. También hay libertad para elegir los diferentes aspectos del videojuego, como su plataforma, motor usado, herramientas, etc.

Para este proyecto he optado por hacer un clon del *Tetris* usando el motor *Unity*.

Índice

Introducción	2
Índice	2
Descripción	3
Diseño artístico	4
Gráficos	4
Audio	4
Historia	4
Controles	5
Herramientas	6
Unity	6
Visual Studio	6
Aseprite	7
Audacity	7
Planificación y Temporalización Real	8
Valoración de los Resultados	10
Webgrafía	10

Descripción

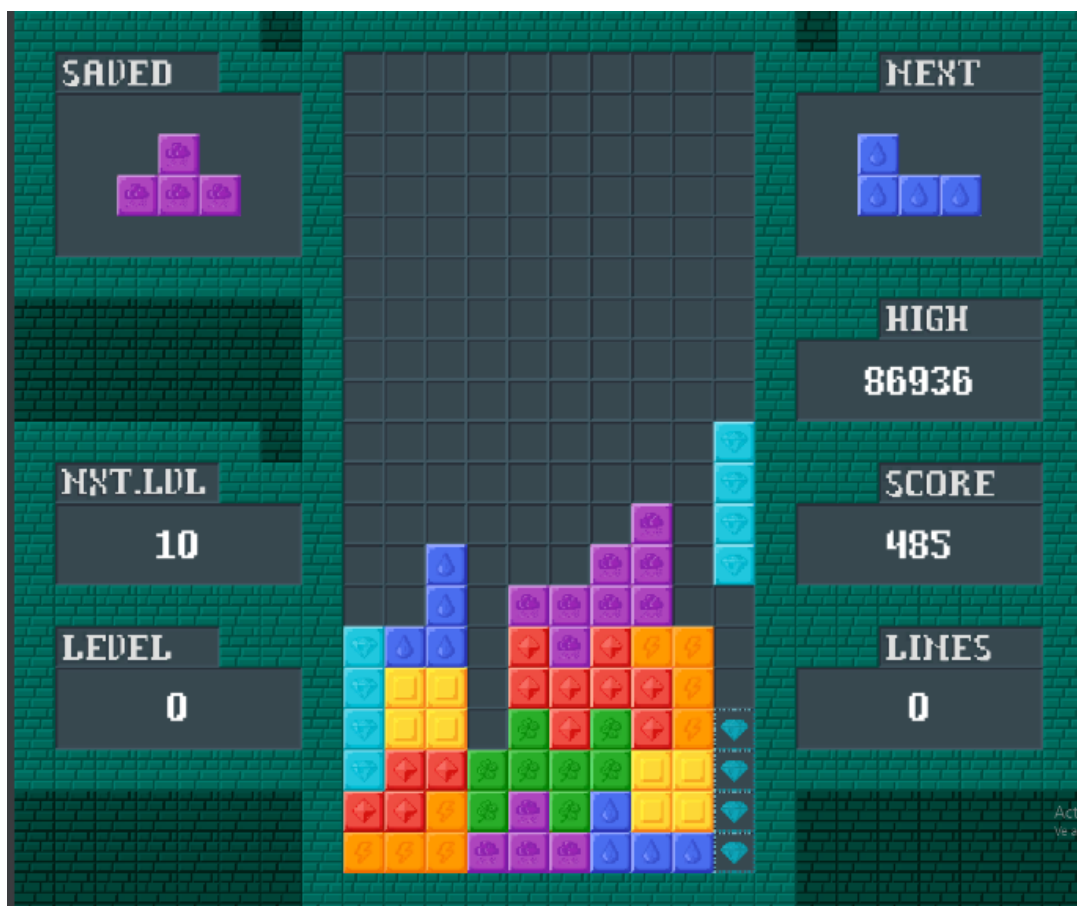
Tetris es un videojuego conformado por un tablero de 10 x 20 y 7 piezas llamadas tetriminos, las cuales están conformadas cada una de 4 bloques llamados minos. Este clon de Tetris ha sido hecho con el motor de Unity y es para la plataforma de Windows. Además, tanto el proyecto del juego como la build para probarlo estarán junto con este archivo.

La mecánica principal del juego consiste en que este genera un tetrimino aleatorio en la parte superior del tablero y el tetrimino va bajando a una velocidad determinada hasta la parte inferior del tablero. Una vez llega a la parte inferior, y pasados unos segundos de gracia, el tetrimino se bloquea y el juego genera otro tetrimino aleatorio.

Durante este ciclo, el jugador puede mover horizontalmente la pieza, rotarla, guardarla y que el tetrimino caiga más rápido o de manera instantánea a la parte inferior del tablero.

Si una línea del tablero está completamente llena de minos, estos son eliminados del tablero sumando puntos a la puntuación del jugador y bajando una línea todas las líneas superiores a la borrada.

De esta manera el objetivo del jugador es ir colocando los tetriminos para hacer líneas, y sumar puntos, y así evitar que los tetriminos lleguen a la parte superior del tablero, momento en el que el juego es finalizado y se lanza la pantalla de “Game Over”.



Diseño artistico

Gráficos:

Para este juego he optado por un diseño pixel art de 36 x 36 píxeles.

Los minos están formados por un bloque base y una figura central que es diferente en cada tipo de mino.

Los colores de los minos han sido determinados por la propia Tetris Company.

El background y hud del juego está conformado por tres bloques. Los dos primeros son bloques de ladrillos azules para hacer el contorno del hud y el último es para hacer el grid del tablero del juego.



Audio:

Todas las músicas usadas en este juego son remixes de la canción original del Tetris. Estas se encuentran al final de este documento en la webgrafía donde también están sus creadores.

Historia:

No tiene historia este juego de puzzle, es solo para divertirse.

Es verano de 1915. Nicolás II decide cerrar la Duma tras el avance de las tropas alemanas y que un solo año antes en torno a 2 millones de soldados rusos fueran masacrados. Aún así los nazis llegan a la Duma y lo único que les bloquea el paso es un agujero de 20 metros que rodea toda la asamblea. Para poder pasarlo deciden tirar diferentes piezas desde la parte superior al fondo para que caigan y creen un puente para poder pasar. Los generales rusos tienen los días contados y es por esto que te encargan a ti, junto a un equipo de limpieza, evitar que cumplan su objetivo. Tu misión es colocar estratégicamente las piezas para que los equipos de limpieza puedan limpiarlas de manera eficaz y efectiva para evitar que los nazis entren, pero no se van a dar por vencidos tan fácilmente.

Controles:

Para el Tetris he optado por el uso exclusivo del teclado a dos manos.

Para el movimiento horizontal se usan las teclas “A” y “D”.

Para las rotaciones se usan las flechas izquierda y derecha.

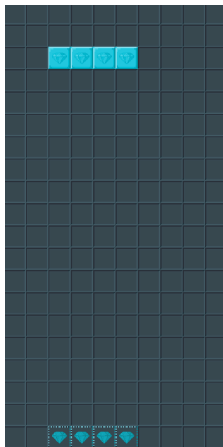
Para bajar rápidamente o de manera instantánea se usan las teclas “S” y “W” respectivamente.

Para guardar una pieza o intercambiar la pieza guardada con la que hay en el tablero se pulsa la barra espaciadora.

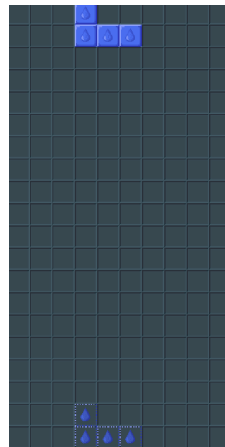
*Si no se pueden ver los gifs los dejo en una carpeta a parte llamada “controles”



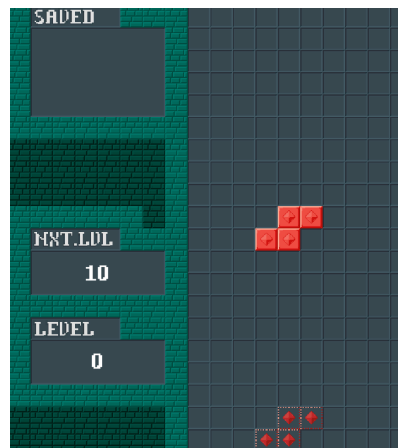
Ejemplo de movimiento horizontal.



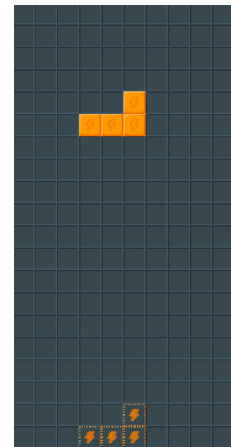
Ejemplo de bajada rápida.



Ejemplo de bajada instantánea.



Ejemplo guardar tetrimino e intercambiar tetrimino.



Ejemplo de rotación.

Herramientas

Para este juego he usado el motor gráfico Unity que autogestiona elementos críticos del juego como el refresco de pantalla o las físicas del juego. Para toda la programación de las mecánicas del juego he usado el entorno de desarrollo Visual Studio. Para el apartado artístico he usado Aseprite. Por último, para la edición de video he usado Audacity.

Unity:

He decidido usar Unity por ser el motor gratuito más potente del mercado. Este permite tanto gestionar modelos 2D como 3D, gestionar sus físicas y colisiones, animaciones de los objetos en pantalla y gestionar elementos gráficos y de audio. Además Unity posee un sistema de rayos que sirve para detectar otros elementos que ha sido indispensable para desarrollar toda la lógica interna de los diferentes movimientos posibles de las fichas.



Visual Studio:

Visual Studio es el IDE por excelencia para Unity y que viene por predeterminado con este. Visual Studio permite escribir en lenguaje C# para todos los scripts que usa el juego y luego poder depurarlos fácilmente con sus herramientas que permiten usarlas mientras se juega y que lo paran automáticamente cuando se llega a un punto específico del código, y si se cumplen ciertos requisitos.



Aseprite:

El aseprite es un programa de dibujo que he usado debido a que esta enfocado en el apartado pixel art y en las animaciones de estos. Dado que mi juego es pixel art y que mis dotes de dibujado son bajas, me ha venido bien este programa para hacer los sprites del juego.



Audacity:

Audacity es el editor de sonido más famoso debido a, principalmente, ser opensource y tener un buen surtido de herramientas para editar tus audios. Para mi juego la mayoría de canciones y sonidos que tiene el juego son descargadas de Internet, por lo que Audacity me vino bien para cortarlos y hacer bien los loops de música.



Planificación y Temporalización Real

Cuando empecé el proyecto tenía bastante claro los pasos a seguir para hacerlo.

Primero haré unos sprites placeholder¹ que me servirán como punto de partida para hacer los scripts.

Luego de tener los sprites empezaré a hacer el script de los tetriminos. Sabía que tendría que hacer también el script de los minos pero primero quería hacer cosas básicas como el movimiento.

Una vez tuviese el movimiento y la bajada de los tetriminos haré los prefabs² de estos para poder añadirles el script y testarlo.

Una vez que tenga el movimiento horizontal y el vertical me dirigiré a hacer las colisiones. Para ello necesitaría el script de los minos que serán los que se encargan de detectar los alrededores del tetriminos.

Una vez hechas las colisiones, insertaré la rotación del tetrimino y sus respectivas comprobaciones para evitar que se salgan del tablero.

Una vez tenemos completado el script del tetrimino, en su forma base, sería empezar con el Game Ctrl. Además llegados a este punto sería conveniente ir empezando a hacer los sprites y el HUD³.

Para el Game Ctrl lo principal es tener la generación inicial aleatoria de los tetriminos y tras colocar cada pieza.

Posteriormente una vez tenemos el loop jugable hay que añadir la eliminación de líneas.

Por último hay que hacer el "game over" del juego, una fase temprana que reinicie el juego al menos.

Llegados a este punto tendríamos planificada toda la base del juego, solo faltaría añadir los sprites como he comentado antes si no se han añadido todavía. Después de esto solo hay que añadir los detalles, que también son importantes y que conforman parte fundamental del juego.

Al HUD hay que añadirle los marcadores, nivel, líneas, puntuación, etc. Además tienen que estar funcionales dentro del Game Ctrl, el sistema de puntuación, la velocidad del juego a base de niveles, etc.

Luego insertaré el sistema de guardado de piezas.

Hay que añadir las animaciones, de completar línea, del game over, etc.

Los sonidos y los efectos de sonido puede ser buen momento para ir preparándolos y añadiéndolos para que funcionen con sus respectivas acciones.

Hay que añadir el tetrimino fantasma⁴ a cada tetrimino.

Por último, añadiré una pantalla de verdad para el game over, en la que puedas cerrar o reintentar.

Placeholder: elemento temporal mientras el elemento que se va a usar no está disponible o no se ha creado.

Prefabs: Objetos de Unity ya preparados con sus componentes y variables listas para poder insertar en escena rápidamente. Se usa comúnmente cuando se requiere de un elemento múltiples veces, ejemplo: Balas.

HUD: Elemento que rodea la visión del jugador que ofrece información extra durante toda la partida.

Tetrimino fantasma: pieza que se encuentra en la parte inferior del tablero marcando la posición donde va a caer el tetrimino.

Temporalización real:

Con respecto a lo que realice en realidad puede ser marcado como algo más caótico.

Nada más empezar hice los sprites de los tetriminos, más o menos un día tarde.

Seguido de esto hice el proyecto en Unity y cree la escena para el juego.

En esta escena añadí el HUD y creé los prefabs para los tetriminos, sin los scripts. Esta parte me llevó bastante dado que tuve problemas para ajustar el tamaño del HUD a los tetriminos ya que ambos eran objetos a diferentes distancias de la cámara¹ y por lo tanto las escalas estaban mal. Al final arregle esto al mover el HUD a un canvas² de Unity que se ajusta a la cámara del juego.

Luego seguí con los movimientos del tetrimino. Para el movimiento vertical lo fui cambiando varias veces hasta el que tengo actualmente, aunque todos funcionaban de similar manera pero a velocidades diferentes.

También hice el script de los minos para evitar que los tetriminos se salgan del tablero o se superpongan. Junto con los movimientos horizontales y verticales, el conjunto me costó toda una mañana.

Con la rotación empecé con un modelo sencillo que al girar se salía de los bordes. Sin embargo posteriormente hice el sistema de rotaciones completo. Todo el sistema me costó varios días dada su complejidad y todas posiciones a comprobar. Intente con varios sistemas similares al de ahora y con varios días de receso para hacer otras cosas del juego. Mientras tanto empecé con el Game Ctrl al cual lo primero que hice fue añadirle los textos. Me costaron toda una mañana debido a que estos se desajustaban al cambiar la resolución de la pantalla.

Luego, añadí el generador aleatorio de tetriminos. Tarde un par de horas en integrarlo y otra mañana extra para hacer la bolsa de tetriminos³ para la generación aleatoria.

Posteriormente hice el sistema de borrado de líneas. Este me costó un día hacerlo y otro más para rehacerlo dado que el primero no me permitía añadirle las animaciones de borrado.

Hice el menú de game over, con sus respectivas animaciones. Tarde varios días con esto. Por último, inserte la música, sus efectos de sonido y su gestión. Elemento que me llevó un día entero recolectar las músicas y efectos y otro día añadirlas.

Como detalle, durante todo este proceso limpié varias veces el código, cosa que dedicaba un día para ello.

Cámara: Elemento de Unity que muestra por pantalla los elementos que graba.

Canvas: Elemento específico para HUDs que siempre están en pantalla, siempre y cuando haya una cámara.

Bolsa de Tetriminos: Elemento añadido en los juegos de Tetris más nuevo para gestionar la aleatoriedad del juego. En vez de generar un tetrimino aleatorio se rellena la bolsa con uno de cada tetrimino y a cada ronda se saca un tetrimino aleatorio de la bolsa. Hasta que esta no se vacía, no se rellena permitiendo que cadenas de cinco o más de uno o dos tetriminos no puedan suceder nunca.

Valoración de los Resultados

Creo que los resultados obtenidos con este proyecto son bastante interesantes a lo que esperaba conseguir. Sin embargo creo que me he dejado bastantes cosas en el tintero que no he podido implementar, como por ejemplo un menú de inicio o un menú de pausa. Con todo creo que el Tetris que he hecho está bastante bien y llega a cumplir las bases de lo que un Tetris tiene que ser.

Webgrafía

Documentación sacada de Unity y c#:

- **Header:** [Atributo Header](#).
- **RaycastAll:** [Elemento RaycastAll](#).
- **Array:** [Funciones de Array](#).

Scripts sacados de internet:

**No es completamente una copia pero sí he sacado algunos elementos y los he modificado a lo que necesitaba.*

- **Movimiento vertical y rotación:** [Video Youtube](#).
- **Wallkick:** [Wallkick Script](#).

Músicas:

- **Música tetris piano:** [Musica 1](#).
- **Música retro y tecno1:** [Musicas 2 y 3](#).
- **Música tecno 2:** [Musica 4](#).

Efectos de sonido: *Me los pasó mi padre del Tetris que hizo él y no se acuerda de dónde los saco, perdón.*