

MAKING OF: GRENADE'S CONQUEST

Introducción

El juego "Grenade's Conquest" ha sido programado con C++, usando las funciones propias de cpctelera para facilitar el desarrollo del mismo.

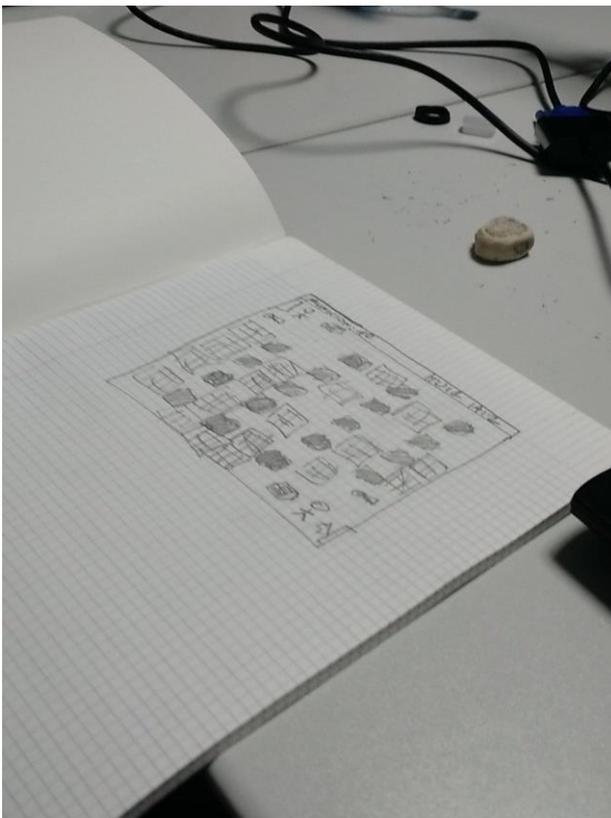
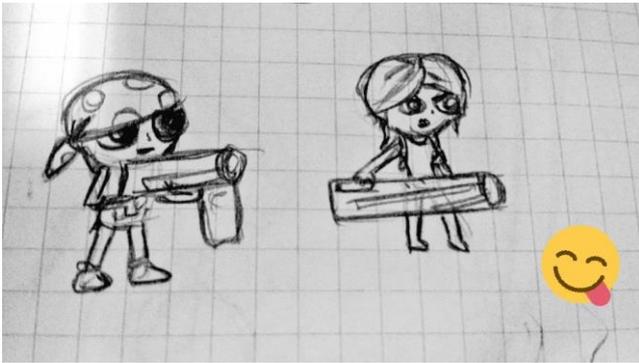
Hemos empleado netbeans y sublime text a la hora de realizar el código, probando cada cambio realizado al juego mediante el emulador WinApe, y ejecutándolo regularmente en un amstrad físico.



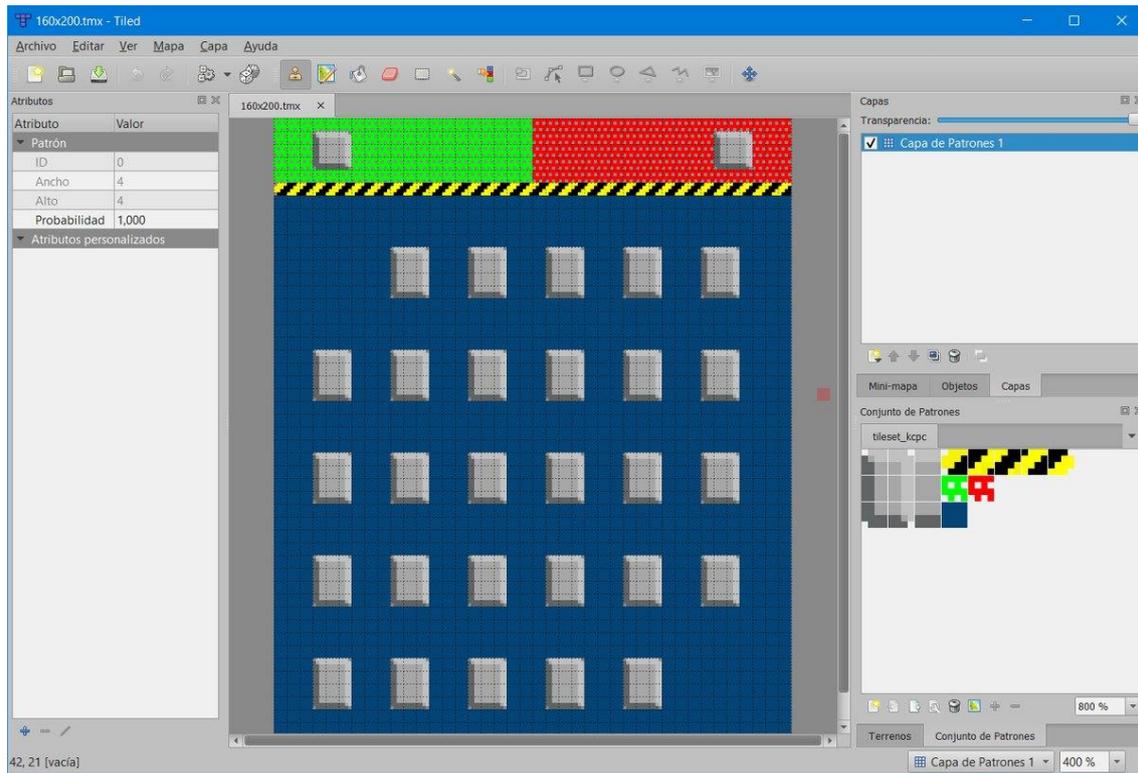
Desarrollo del juego

Al empezar, realizamos algunas pruebas básicas para familiarizarnos con la librería de cpctelera. Una vez aprendimos los conceptos básicos, empezamos con el desarrollo de nuestro videojuego.

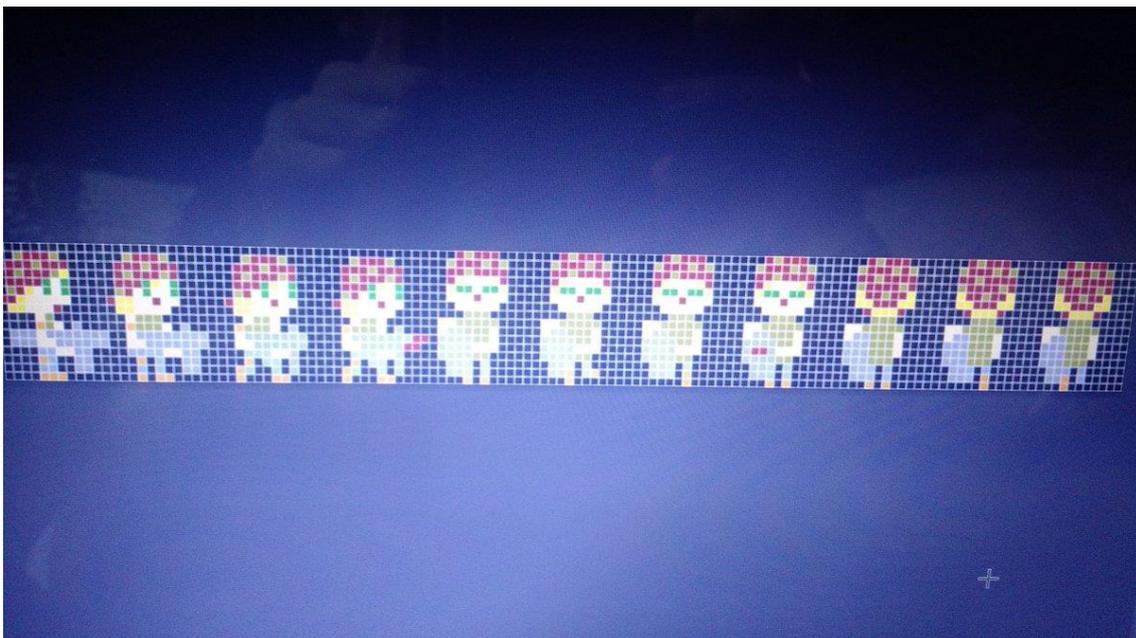
El primer paso fue crear los bocetos del mapa y de los personajes, basándonos en las ideas que habíamos desarrollado en papel previamente.



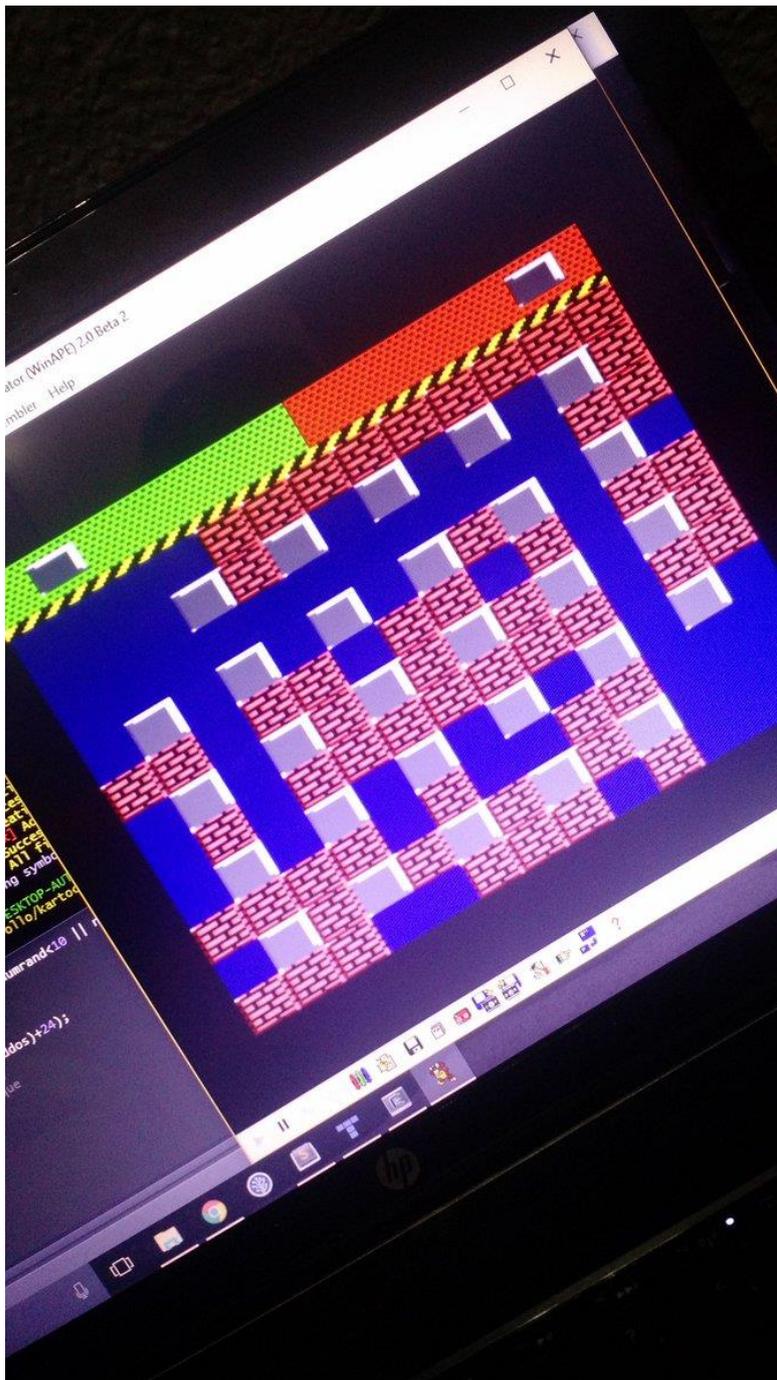
El segundo paso consistía en realizar lo que sería el mapa de nuestro juego, usando la herramienta Tiled.



Una vez creado el mapa, insertamos a nuestro personaje principal, Jimmy, en el juego. Realizamos todas sus animaciones en movimiento, así como las de ataque. Realizamos los métodos necesarios para su movimiento, y eliminamos el efecto de parpadeo.



Una vez teníamos el mapa y al personaje creados, realizamos las colisiones del personaje con los bloques que habíamos dibujado en el fondo, los no destruibles. Realizamos varios métodos, pero acabamos por decantarnos con el método de mirar el color de las casillas. Si era color de fondo, podía atravesarlo, si no, chocaba. Esto nos vino muy bien a la hora de chocar con los bloques destruibles, ya que tienen colores distintos al fondo. Estos bloques aparecen en zonas aleatorias usando funciones de `cpctelera` junto a nuestros propios cálculos para crear números pseudo-aleatorios, comprobando que siempre se sitúen sobre el fondo azul, y no sobre otras casillas.



Cuando terminamos con estos pasos, pasamos a la realización de los disparos. Creamos otro sprite para el mismo, y realizamos el movimiento del mismo en las cuatro direcciones: arriba, abajo, izquierda y derecha. Luego pasamos a realizar las colisiones de estos con los bloques no destructibles, mirando también el color de las casillas por las que pasan los disparos, método que empleamos también para comprobar la colisión con los bloques destructibles que, en caso de encontrar uno, redibujará el fondo para borrar el mismo, y para la comprobación de choque con el otro personaje, aunque esto último fue realizado más tarde, cuando creamos al segundo personaje.

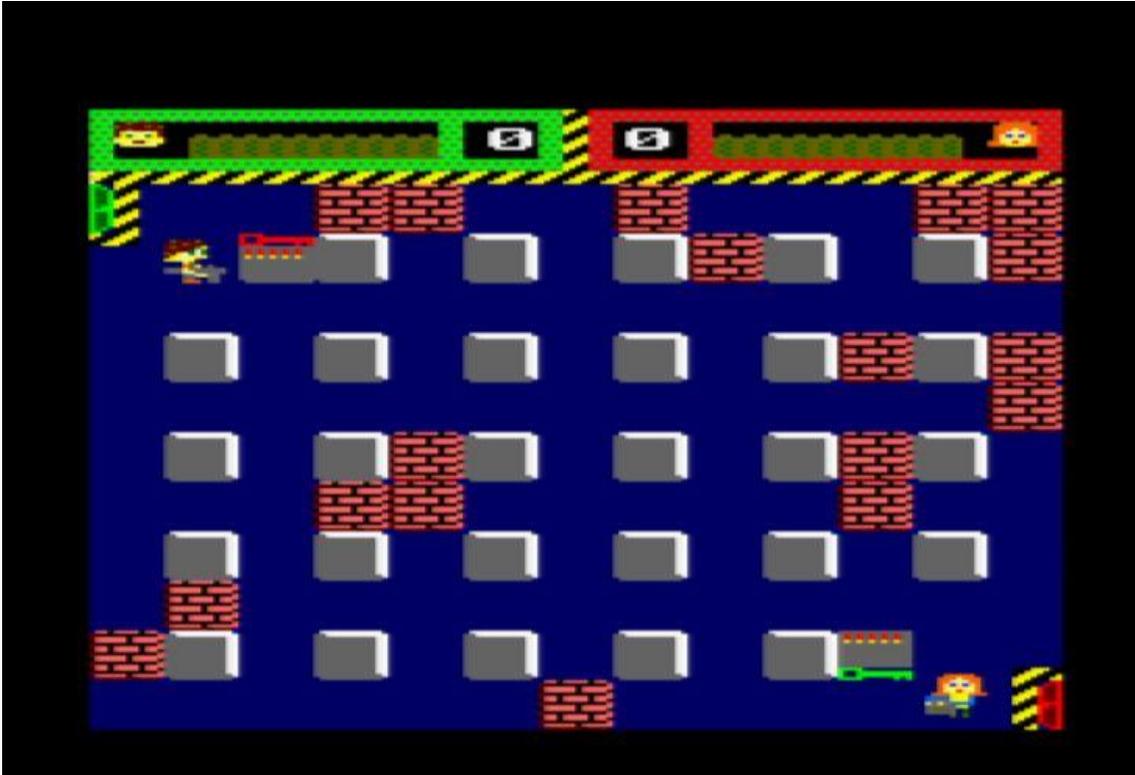
Una vez realizados los disparos, pasamos a la creación de Obdulia, el segundo personaje del juego. Realizamos sus animaciones de movimiento y de ataque, y aprovechamos los métodos creados para Jimmy tanto de movimiento, colisiones como disparos, para implementárselos a ella, obteniendo resultados mucho más rápidamente de lo que nos costó realizarlos para Jimmy.



A continuación, creamos los métodos necesarios para que, cuando un disparo chocara contra el personaje enemigo, consiguiéramos un punto y se reseteara la posición de los jugadores. Creamos un contador de puntuaciones y que, cuando uno de los dos llegaba a cinco puntos, la partida terminaba y se reseteaba el juego.

Posteriormente añadimos un número límite de balas. Al disparar, se restará una bala (añadimos sprites de balas a la parte superior de la pantalla, uno por cada bala restante), y estas se verán incrementadas cuando vayamos a la base, junto a la munición.

También añadimos dos sprites para las llaves, y otros para las puertas. Realizamos las comprobaciones para que, cuando un personaje pasara cerca de su llave, la recogiera y se abriera directamente la puerta de su base, y si se acercaba a esta teniendo la llave en su poder, gana un punto.



Finalmente, creamos pantallas de carga, un menú donde poder acceder a distintas funcionalidades del juego y creamos e insertamos nuestra propia música en el juego.



