

MAKING OF: BringRGBack

Bring RGBack es un juego desarrollado para Amstrad CPC 464 utilizando el emulador WinAPE 2.0 beta 2 y el framework CPCtelera. Es el primero juego desarrollado en lenguaje Ensamblador y para esta plataforma por el equipo Dead Pixel y por eso queremos explicar cómo ha sido el desarrollo del mismo.

El juego consta de diferentes zonas por las que deberás ascender realizando *scroll* vertical, evitando los diferentes obstáculos que te encontrarás en el camino. En cada nivel se introducen nuevas mecánicas.

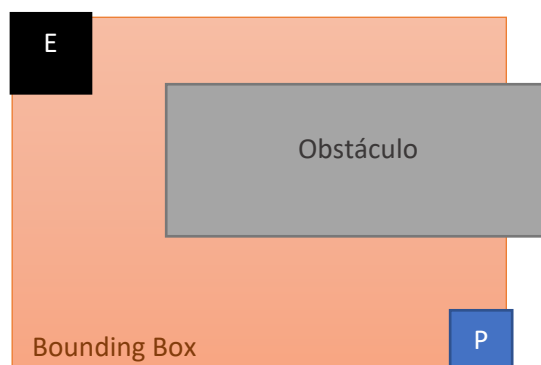
El *scroll* vertical primero se trataba de un *scroll* tradicional, moviendo todo el mapa y elementos que aparecían en pantalla, pero nos dimos cuenta de que suponía una considerable bajada de rendimiento que no queríamos sufrir en un juego ágil de disparos como el nuestro, y decidimos prescindir de un mapa de fondo y realizar un *scroll* personalizado que desplaza únicamente los elementos de pantalla sobre un fondo liso, consiguiendo así una mejora de en la jugabilidad bastante sustancial.

Esto, posteriormente, hizo que el juego fuera tan veloz, que tuviéramos que limitar el *framerate* máximo utilizando las interrupciones del sistema para que fuera más jugable.

En un primer momento, la generación de mapas a producirse de manera aleatoria, sin embargo, en un punto bastante avanzado del desarrollo nos dimos cuenta de que la aleatoriedad a veces puede jugar una mala pasada y haría que el juego a veces pudiera ser divertido y otras no. Por lo tanto, decidimos hacer un sprint para diseñar nuestros propios mapas preestablecidos que aseguren la diversión del jugador y que esperamos que disfrutéis y consigáis superar hasta el último de ellos.

Los enemigos era una parte muy importante de nuestro juego y hemos decidido emplear gran parte el tiempo de desarrollo en desarrollar nuestros NPC. Hay diferentes tipos de ellos:

Ojos Vigía: Son los que puedes encontrar desde el principio del juego, deambulan de un lado a otro de la pantalla vigilando sus alrededores. Si te detectan, se pondrán en modo ataque e irán a por ti. Su sistema de vigilancia ha sido desarrollado mediante el uso de *bounding boxes*. Si el jugador se encuentra a una distancia dentro del rango de visión del enemigo, este crea un *bounding box* desde su posición a la del personaje. Comprobando la colisión entre este *bounding box* y los diferentes obstáculos del mapa, podemos comprobar si el enemigo conseguirá ver al personaje, o este está oculto y el enemigo no puede verte.



Ojos Guardianes: Se encuentran a partir del nivel 3.1. Estos enemigos se desplazan de una parte de otra de la pantalla de manera implacable ya que no pueden ser destruidos.

Ojos Torreta: Aparecen en el juego si superas los dos primeros niveles, a partir del nivel 2.1. Disparan balas de manera constante para impedirte seguir avanzando. Pero no se inmutan con tu presencia, Sin embargo, si decides atacarles, irán directamente a por ti.

Ojos Relámpago: Se encuentran en el último nivel, y suponen un plus de dificultad. Los Ojos Relámpago te atacan desde la parte inferior de la pantalla si tardas demasiado tiempo en ascender en el nivel. Te impiden tomarte el nivel con calma y hacen que el último nivel sea frenético.

Al realizar el juego con solo 64K de memoria disponible, ha sido necesario que realicemos la compresión de los más de 100 Sprites con los que cuenta el juego, ya que estos varían en cada una de las 4 zonas disponibles. De manera que estos son descomprimidos al principio del nivel sustituyendo a los del nivel anterior y así ahorrar espacio.

En el último nivel, se encuentra el Sprite guiño al videojuego *Chicago's 30* de *Topo Soft*.



Para finalizar, queremos decir que ha sido un desarrollo relativamente complicado debido a que han surgido errores que, al ser nuestra primera vez desarrollando en este lenguaje, nos ha costado un gran número de horas corregir. Sin embargo, el proceso nos ha servido para aprender multitud de conocimientos que desconocíamos, para superarnos y realizar el mejor juego posible; que esperamos que os guste.

Dead Pixel.