

LPrint Runaway

Por @ignacobo

<http://ignacobo.itch.io>

La maquina original LCD salió al mercado en 1983, creada por la casa japonesa GAKKEN.



Las instrucciones decían:

“Escapa de la cárcel. Avanza hasta la pared sin que te detecte el rayo de luz ni te descubran los guardias. Cuando te pillen 3 veces, se acabó el juego.”

Esta es la conversión al ZX Spectrum, 40 años después, programada en Basic puro para el 48kb, y presentada al concurso Bytemaniacos Basic2023, 20 años, de Radastan.



Agradecimientos:

A IvanBasic (@_IvanBasic_) por explicar los principios de la técnica del LPrint, que empleó con maestría en su juego Brain8

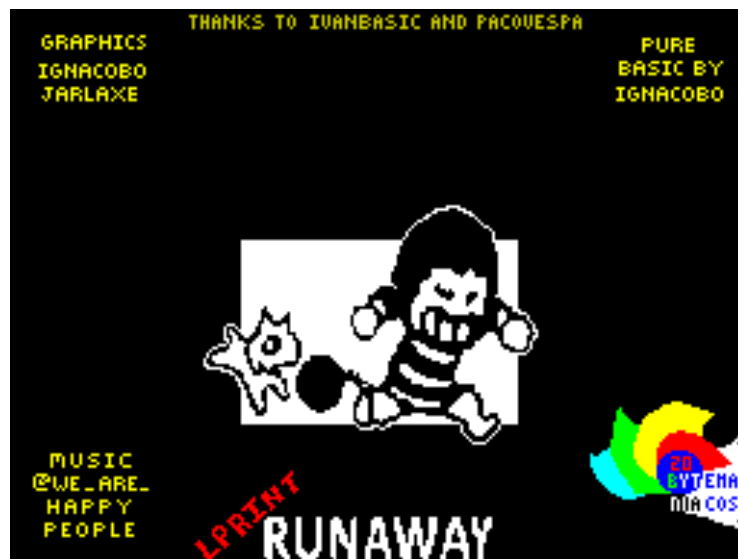
A Jarlaxe (@JarlaxeRetroGa1) por orientarme y aconsejarme en el aspecto grafico del juego. Los sprites de runaway llevan su sello

A @we_are_happypeople por la melodía creada para el juego

A Pacovespa (@Pacofranquelo) por hacer de betatester de manera tan dedicada y detallista.

A AnbJørn Myren por su ayuda con la mecánica del juego original

A Radastan (@Bytemaniacos), por su apoyo al desarrollo de proyectos en Basic y al retro en general, a través de sus concursos.



Teclas:

Las teclas de control son

O para mover el prisionero hacia la izquierda,

P para moverlo a la derecha. La tecla

M activa y desactiva la "música"

Parte técnica:

El programa usa la técnica del LPrint en zona de atributos para cambiar los colores de la pantalla de manera rápida. En el código solo se usa una instrucción Print para manejar la puntuación. Todos los movimientos de los personajes y del rayo de luz se hace a base de cambiar la tinta y el papel de los caracteres, pero el bitmap de la pantalla es siempre el mismo. Más detalle de esta técnica en el blog:

http://blog.jafma.net/2020/08/29/efficient-basic-coding-for-the-zx-spectrum-v/#sp_5

Lo más costoso del desarrollo fue diseñar cada grupo de 8bytes con el color de tinta y papel necesario en cada parte de la pantalla. En total necesité los 21 UDG y un charset entero (21+96 caracteres!) para completar todas las posibilidades de color necesarias.

Cada 16 frames suena una nota Beep de la música in-game.

Con respecto al código:

Se ejecuta con RUN 9000, donde se inicializan las variables. Los ficheros auxiliares están ya en memoria, cargados con el programa de la Intro.

Después de un BREAK, hacer RUN 9999 para poder leer el listado.

Utilizo DefAdd para dibujar rápidamente los colores con la pantalla inicial (768bytes) cada vez que perdemos una vida.

En el bucle principal, desde 0 a 20,

[0] imprimir puntuación. Única instrucción Print del programa!

[4] salta a pintarRayo, pintarGuardialzquierda o pintarGuardiaDerecha según corresponda al tiempo T.

[6]Toca una nota musical de una duración determinada. La gestión de la música se hace en una sola línea!

[8] Acelera a los malos cada 10 segundos, incrementa la velocidad y dificultad. También depende de la puntuación actual.

[10-11]Lee el teclado y salta a pintarBuenolzquierda o pintarBuenoDerecha si es necesario.

Cada una de estas subrutinas son una ristra de pokeos indicando en que zona de atributos pintar, y uno o varios LPrint para colorear.

[30-60] pintarBuenoADerecha

[66-96] pintarBuenoAlzquierda

[110-180] pintarRayo

[200-241] pintarGuardiaDerecha

[300-361] pintarGuardialzquierda

[1200-1350] pintarBuenoEnLaPared

A partir de [6000-6100] es pérdida de una vida, dibujar preso en la celda, y comprobar fin de la partida y HighScore.

[9000-9100] Inicio del programa, inicialización de variables y reseteo de partida nueva.

Ficheros:

DefAdd, datos de 4 variables usadas para mover colores a la zona de atributos de forma rápida

Nums64cx, números de 4 pixels de ancho, para pintar la puntuación

ColOscuro, atributos de pantalla negra con recuadro blanco en el centro, para la intro, con efecto chulo si se carga desde cinta

Music y durMusic, notas musicales y su duración

Gak.bin, bitmap de la pantalla de presentación

Timing, ciclicidad de movimientos del rayo(1cx), del guardia izquierdo(3x) y del guardia derecho(4x)

ColoresIni, colores de inicio de partida o después de perder una vida.

lcdInv4.bin, bitmap de la pantalla de juego principal

Charset y UDG, patrones de bytes conteniendo las combinaciones de papel y tinta necesarios para los colores de cada carácter, en total necesité 96+21!!

RunAway.bas, código basic del juego
