Arkanoid, Back to Basic!
por ignacobo

1- INSTRUCCIONES

Tributo al clásico juego Arkanoid Revenge of Doh. Está realizado en BASIC puro para ZXSpectrum 48K, presentado al concurso de Radastan Bytemaniacos de Febrero 2022.

Misión

Tienes que guiar la nave en forma de pala a través de 17 fases, tratando de destruir los ladrillos de cada pantalla, utilizando una bola que los golpee rebotando en la pala. Si la bola se va por la línea inferior perderás una vida.
Usa los extremos de la pala para rebotes pronunciados y el centro de la pala para rebote normal.
Consigue más puntos terminando la fase en el menor tiempo posible, y cogiendo PowerUp

Tipos de ladrillos

Malla de color: Básico, se destruyen con un solo golpe. Algunos esconden un Power Up que caerá
Relieve blanco y negro: Estos necesitan dos toques para eliminarlos
Relieve amarillo y rojo: Ladrillo irrompible. Provocan un gran ángulo de rebote de la bola
Malla blanca y negra: Este tipo de ladrillo desaparece después de dos impactos. Al cabo de unos 10 segundos reaparecerán de nuevo en pantalla

Power Up

Hay 7 tipos de ayudas, que aparecen tras golpear determinados ladrillos
GLUE: color azul. La bola se quedara pegada a la pala durante unos segundos, o hasta que pulses Spc.
WIDE: color rojo. Aumenta el tamaño de la pala, pero acelera la velocidad de la bola
SHORT: color magenta. Reduce el tamaño de la pala y reduce la velocidad de la bola
LASER: color verde. Pulsando Spc, la pala disparara un rayo laser que destruye ladrillos
1 UP: color cyan. Vida extra, hasta un máximo de 8
3BALLS: color amarillo. Maneja 3 bolas a la vez. Puedes perder hasta dos de esas bolas sin que pierdas una vida.
DRILL: color blanco. La bola atravesara todos los tipos de ladrillos al primer toque y sin rebotar.

Menú principal.

Podemos elegir el Nivel de dificultad pulsando N. Hay tres distintos, Difícil, Normal y Fácil. Se diferencian en el ancho de la pala, el numero de vidas inicial y cuántos PowerUp se esconden en cada pantalla.

Pulsando A empieza la partida.

Accedes al menú de Músicas con la tecla M, donde podrás escuchar las melodías y efectos de sonido del juego, con las teclas 1 al 6

Relee las instrucciones apretando I. Recuerda, mueve la pala a la izquierda con O y a la derecha con P. Suelta la bola pegada o dispara el laser con la tecla Spc.

Para entrar al Editor pulsa E. Podrás crear tu propio diseño de pantalla. Muévete con OPQA. Para los ofendiditos, también se permite usar QAOP, aunque va mas lento ;-)
Cambia el brillo, la tinta y el fondo del ladrillo, con B, T y F. Selecciona el tipo de ladrillo con L.
Pulsando Spc pones o quitas el ladrillo.
Cuando termines tu diseño podrás jugarlo pulsando J.

2- DATOS TÉCNICOS

Lo más destacable del diseño es el movimiento de la bola de 4 en 4 pixels. Tanto horizontal como verticalmente, la bola se mueve a medio carácter en velocidad lenta, o a carácter en velocidad alta. Esto nos permite gran variedad de ángulos de rebote: 30º, 45º y 60º

Para lograrlo, he necesitado nada menos que 9 UDGs para el sprite de la bola:

También hay que reseñar que los pseudo sprites de la bola y de los PowerUp no borran el fondo por el que se desplazan.

Hay 4 tipos de fondos diferentes, para ello he necesitado 4 bancos de UDGs para simular el efecto de pasar sobre el fondo sin borrarlo.

El juego se mueve a unos 7 FPS, que para un juego en BASIC es un gran reto. Para arañar frames en la ejecución he utilizado estas técnicas:

*El bucle principal empieza en la primera línea del código.

*Todos los PRINT terminan con un ; que hace que no se envíe RetornoDeCarro, ahorrando algunos frames.

*La mayoría de las condiciones de los IFs son Var o NOT Var, para que el interprete tenga que leer menos caracteres.

*Incluir los códigos de color dentro de las cadenas a imprimir, ya que INK, PAPER, etc son algo más lentos. Las fases están almacenadas de esta manera, siendo una cadena larga que además incluye los AT x,y incrustados, de modo que con un solo PRINT se dibuja cada fase.

*La máquina de estados del engine tiene unos 24 estados. se puede implementar con unas 6 líneas y 2 prints, pero en cada iteración realiza muchos cálculos y el print estaría lleno de AND, OR, y otras condiciones que ralentizan mucho. Entonces he desplegado cada uno de los posibles estados y hay un print para cada estado, sin necesidad de cálculos redundantes, ganando mucha velocidad, pero a costa de gastar mucha memoria, unas 52 líneas y 40 prints para el mismo engine. En función del desplazamiento a izq o a drcha, arriba o a abajo, si son 4 u 8 pixels y el frame de animación horizontal y el vertical se accede a una tabla de saltos y hace GOTO a esa línea.

*Repito mucho código en diferentes líneas, gastando mucha memoria, para evitar saltos GoSub que ahorrarían bytes, a costa de velocidad.

*Declarar las primeras las variables a las que se acceda con mayor frecuencia.

*Tras muchas pruebas he comprobado que a partir de la variable numero 13 o 14, la búsqueda de la variable es más lento que el acceso a una posición de memoria con POKE y PEEK, por lo que he sustituido las variables Byte a partir de la 13 por POKEs. El gran inconveniente de esto es que la casi imposibilita la legibilidad y depuración del código, así que solo recomiendo hacer esto al final del desarrollo del proyecto. Además, este cambio ocupa mucha más memoria.

*En el bucle principal primero veo si hay que redibujar algún ladrillo de los que reaparecen, línea 4

Después comprobación de tecla izq, drch o disparo, líneas 5,6 y 7

Maquina de estados desde la línea 11 a la 81.

*Para el PowerUp de 3 bolas, utilizo DefAdd para hacer una copia rápida de las 11 variables necesarias para mover la bola, actualizarlas, guardarlas en memoria, y en cada iteración, leer las otras 11 a la zona de variables del Basic activas, volver al engine, y etc etc. Esto es mucho más rápido que tener una matriz Dim(11,3) para las 11 variables de cada bola.

3-CONTENIDO DEL FICHERO ARKAB2B.TAP

arkab2b.bas

Código BASIC con la lógica del juego, y desde el que se cargan los ficheros auxiliares

fase0

Pantalla con el marco de juego y sin ladrillos

fases1a17

Cadenas de texto con los ladrillos para imprimir cada una de las 7 fases del juego

music
notas y duración para las melodías y efectos

UDGs
4 bancos de gráficos, se diferencian entre si por el diseño del fondo por el que pasa la bola. En cada banco hay 9 UDG dedicados al sprite de la bola desplazado 4pixels en cada dirección, para simular el movimiento suave de 4pixels

gotoQ
Tabla de saltos GoTo para las colisiones con los ladrillos cuando coges el PowerUp Drill

tablas
Conjunto de 7 tablas unidas en un solo fichero:
gotoP: tabla de saltos GoTo para las colisiones con ladrillos
defadd: declaración estructura defadd para la copia rápida de datos
R(32): manera rápida de ver si un número es par o impar
K(17): datos para dibujar el rayo laser con Draw
Hpow(22): animación del PowerUp mientras cae
laser: más datos para dibujar el rayo laser con Draw
rnd(Pow8): para seleccionar qué tipo de PowerUp va a salir

backgr
Array de cadena de caracteres, guarda el diseño de los ladrillos por los que va pasando un PowerUp mientras cae, para poder restaurar su valor después, y que el Pow no deje rastro

laserDir
Number array, guarda las direcciones de memoria de cada inicio de fila, para ahorrar cálculos en dibujo del laser

arkaDATA.bas
Código BASIC utilizado para generar los ficheros anteriores. Es necesario hacerlo en archivo .bas separado del principal para ahorrar muuuchos bytes y velocidad de carga de los datos en memoria

1- VERSIONES

1.2 *Arreglado bug de la fase 4, que en determinadas ocasiones no pasaba a la siguiente fase a pesar de no quedar ladrillos
* Mejora de la velocidad del Pow 3bolas en un 40%! usando varias direcciones de la variable del sistema VARS
* Mejora de la velocidad del Pow Laser cambiando la variable del sistema PROG
* Los ladrillos que reaparecen ahora suman puntos

1.1 *Corregido el bug de la linea 228 que te sacaba al Basic

1.0 *Version inicial

Bugs conocidos:

Si la bola rebota en el Pow blanco Drill se crea un ladrillo de los que reaparecen
Si la bola rebota en el Pow azul Glue se quedó parada en el aire unos segundos
El láser no alcanza a los ladrillos de la última columna