En este documento iremos definiendo para qué sirve cada tipo de archivo dentro del Engine. En las notas, iremos colocando impresiones o suposiciones que puedan ser de utilidad.

Tipos de Archivo:

**(.lev)** Archivos de descripción del nivel

**(.stg)** El archivo del nivel

**(.sc4)** Los patrones utilizados para definir el nivel

**(.tgs)** Archivo de patrones de sprites.   
**(.tcs)** Datos de color del sprite, cargado junto al anterior. Debe de tener el mismo nombre.

**(.mst)** Atributos y posiciones de los enemigos

**(.mbm)** La musiquita Moonblaster 1.4.

**(.mbk)** Drumkits de Moonblaster.

**(.scr)** Archivo GFX en SC5. Utilizada para definir los textos ingame y la barra de marcadores.

**(.see)** Efectos de sonido

**(.fnt)** Imagen con fuentes en SC4. Incluye los 32 caracteres que son utilizados en el menú de inicio.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Archivo | Tipo | Direccion de inicio | Direccion final | Pagina del Mapper | Tamaño |
| VBIOSBIN.COM | Codigo | #C000 | #C480 | 0 | 1152 |
| VSLOAD.COM | Codigo | #100 | #980 | 3 | 2176 |
| VSCREEN.COM | Codigo | #FFD (#1000-3) | #7525 | 3 | 6528 |
| MBPLAY.COM | Codigo (replayer Moonblaster) | #4000 | #4E80 | 5 | 3712 |
| .lev | Archivo de nivel (Cargado desde VSLOAD.COM) | @leveldata |  |  | Max 255 |
| .tgs | Patrones de Sprite | #6500 |  | 6 | 2176 |
| .tcs | Colores de Sprite | #6500 + 3000 |  | 6 | 1088 |
| .scr | GFX del Score | #5037 |  | 6 | 5128 |
| .sc4 | Patrones del nivel | #4000 |  | 6 | 4128 |
| .stg | Datos del mapa | #4000 | #9FFF | 1 & 2 | Max 24576 |
| .mst | Datos de los enemigos | #A000 |  | 1 | Variable |
| MENU.COM |  | #1000 |  | 3 |  |
| VSCREEN.SEE | Efectos de sonido | #8000 | #9026 | 4 | 4134 |
| .mbk | Kit de Samples de Moonblaster | Audioram |  |  |  |
| .mbm | Archivo de melodia Moonblaster | #9500 |  | 4 | 4297 (max 11007) |

* **Efectos de sonido incluidos en Vscreen.see**

|  |  |
| --- | --- |
| Dirección | Vscreen.see |
| 0 | Salto |
| 1 | Coge bonus |
| 2 | Rompe bloque |
| 3 | N/A |
| 4 | Muerte protagonista |
| 5 | Muerte enemigo |
| 6 | PAUSA |
| 7 | Empezar juego / Salir(Menu) |
| 8 | N/A |
| 9 | Arriba / Abajo (Menu) |
| A | Cambiar valor (Menu) |
| B | Coge Power Up |
| C | N/A |
| D | Fin Power Up |

* **Archivo de descripción de nivel (.lev)**

Contiene los parámetros para definir el nivel.

La estructura es como sigue:

1 Bytes : Version del Engine

15 Bytes : Nombre del Nivel

4 bytes : Posiciones X e Y iniciales del protagonista

8 bytes: Nombre del archivo .stg (sin la extensión)

8 Bytes: Nombre del archivo .sc4 (sin la extensión)

8 bytes: Nombre del archivo .tgs/tcs (sin la extensión)

8 bytes: Nombre del archivo .mst (sin la extensión)

8 bytes: Nombre del archivo .mbm (sin la extensión)

8 bytes: Nombre del archivo .mbk (sin la extensión)

8 bytes: Nombre del archivo .scr (sin la extensión)

1 Byte: Número de Bonus necesarios para abrir la puerta de fin de nivel (opcional).

1 byte: Anchura del nivel (en pantallas)

1 byte: Altura del nivel (en pantallas)

1 byte: Velocidad del personaje jugador

1 byte: Fuerza de salto del jugador

1 byte: nivel de inercia del jugador

1 byte: Fuerza de los Trampolines

1 byte: Velocidad de caída del personaje (gravedad)

1 byte: Potencia de rebote (Cuando saltas sobre un enemigo)

1 byte: Puntos por coger un bonus

1 byte: Puntos por matar un enemigo

1 byte: Activar animacion off/on (1/2)

1 byte: Activar Parallax off/on (1/2)

1 byte: Número de salidas activas

Por cada salida activa::

2 bytes: Direccion offset de la salida en el archivo .stg

1 byte: Tipo de Salida.

8 bytes: Nombre del archivo .lev enlazado a la salida

* **Graficos de Patrón (.sc4)**

El archivo .SC4 los tiles del nivel, se puede crear usando el programa [**GE5 to SC4 graphic converter**](http://abyssmsx.com/tools.htm)

Los archivos .sc4 tienen la siguiente estructura:

* 32 Bytes para la paleta de color.
* 2048 bytes para la tabla de patrones (256 tiles x 8 bytes).
* 2048 bytes para la tabla de colores de patrón (256 tiles x 8 bytes).
* **Graficos de Sprites (.tgs)**

El archivo .tgs contiene los sprites.

El archivo de los patrones gráficos de sprites ocupa 2944 bytes y corresponde a 96 de 16\*16.

Los Sprites de Vscreen siempre están compuestos por dos capas (3 colores por línea con OR). Por lo que tenemos en total 48 sprites diferentes en el archivo .tgs. Los archivos TGS y TCS se pueden crear con la aplicación [**MSX2 Sprites Converter**](http://abyssmsx.com/tools.htm)

* **Colores de Sprites (.tcs)**

El archivo .tcs contiene los colores de los sprites.

El archivo ocupa 1472 bytes y corresponde a 96 sprites de 16 filas.

Hay dos capas de color por Sprite.

El archivo .tcs debe tener el mismo nombre que el archivo .tgs.

* **Archivo de mapa del nivel (.stg)**

Contiene el mapeado del nivel. Ocupa 24384 bytes y corresponde a 256 bloques (ancho del nivel) por 96 (alto del nivel).

Este archivo no tiene ninguna cabecera.

* **GFX Score Board (.scr)**

El archivo .scr contiene una cabecera de 7 bytes (Cabecera de "bload" de MSX basic).

El archivo .scr contiene la imagen del marcador, así como las fuentes gráficas utilizadas durante el juego. El archivo está compuesto por un total de 40 líneas de datos de Screen 5.



* **Archivos de música y Samples (.mbm y .mbk)**

Los archivos .mbm y .mbk son archivos de música en formato Moonblaster 1.4.

* **Archivo de efectos de sonido (vscreen.see)**

El archivo .see contiene los efectos de sonido PSG en formato SEE 3.0

* **Archivos de posicionamiento de enemigos (.mst)**

El archivo .mst contiene las posiciones y atributos de movimiento de los enemigos..

(si necesitas espacio, empuja este capítulo hacia abajo xD)

**Rutinas y Subrutinas en ensamblador.**

En este capítulo van las rutinas y subrutinas que hay en el disco, hechas con ensamblador directamente en MSX que he ido pasando para poder ensamblar desde un pc. Tanto macro.gen como macro.asm es lo mismo, solo que \*.gen es como viene en el original y \*.asm es una vez arreglado y corregido para que ensamble con el Glass Assembler en el OpCode IDE.

\*\* Las que figuran como \*.gen es que aun no las he pasado a \*.asm

**VSLOAD.COM**

Es el cargador de niveles (entiendo que la rutina principal de VScreen) y hace llamadas a:

macro.asm

Loader.asm

Mbloader.asm

Vsbios.asm

.- Ficheros que llama VSLOAD (Revisar cuales necesitan convertirse GEN ->ASM)

**fbios**: db 0,"vbiosbincom" ; vscreen bios

(necesita estar ensamblada por separado?)

**fvscreen**: db 0,"vscreen com" ; vscreen main

Está ensamblada con todas las llamadas

**fmbplay**: db 0,"mbplay com" ; Moonblaster replayer

Ensamblada por separado (mbplay.gen)

**flevel**: db 0," lev" ; level file

**fmbsample**: db 0," mbk" ; sample kit file

**ftgs**: db 0," tgs" ; sprite gfx

**ftcs**: db 0," tcs" ; sprite color

**fms**t: db 0," mst" ; enemies position and movements

**fsee**: db 0,"vscreen see" ; sound effects

**fscr**: db 0," scr" ; score panel gfx

**fmbm**: db 0," mbm" ; music

**fsc4**: db 0," sc4" ; gfx of the tiles

**fstg**: db 0," stg" ; stage data

Media: Para todos estos, mirar capitulo de Jorge

**fmenu**: db 0,"vsmenu com" ; menu

Menú de opciones justo antes de empezar a jugar con opciones de teclado o joystick, PAL o NTSC, Música y/o EFX y SALIR

**fmenutc**: db 0,"vsmenu tc "

Colores ¿?¿?

**fmenufnt**: db 0,"vsmenu fnt"

Fuente de letra que usa el Menú.

**VSCREEN.COM**

Es el motor general del juego, hace llamadas a:

Msxdos.asm ;esta no es original del disco vscreen

macro.asm

vsbios.asm

Enmtype2.asm

enmtype3.asm

ctrlhero.asm

obstacle.asm

special.asm

screen.asm

manette.asm

scorebar.asm

\* (Inicio de nuevo capítulo. Ve añadiendo los espacios que creas conveniente Jose. :D)

**Personalizando tu juego en VScreen**

Una vez tienes creados tus propios gráficos, efectos de sonido e imágenes, es conveniente que realices algunos cambios en el proyecto para darle un empaque más profesional.  
  
También es posible que quieras añadir variantes en las mecánicas del juego, nuevos ítems, o comportamientos personalizados para tus enemigos. En este capítulo te ofrecemos algunas nociones de como hacerlo.

En primer lugar, debes saber que necesitas unas nociones mínimas de ASM, por lo que si no las tienes, deberás hacer equipo con alguien que posea estos conocimientos, o adquirirlos por tu cuenta. ¡Ánimo!

**Eliminando o modificando los textos de carga**

VScreen por defecto muestra todo el proceso de carga de los archivos que componen cada nivel. Esta información es muy útil durante el desarrollo de tu juego, pero una vez completado, es fácil que estos textos estropeen la inmersión del jugador.

Para editarlos o eliminarlos, deberás editar el archivo vscreen.gen.

Dentro del archivo, localiza la rutina que llama a “start\_text”

Ld hl, start\_text ; Call inittext

Basta con borrarla o modificar el texto cargado en “start\_text” para mostrar cualquier otro texto durante las pantallas de carga.

**Modificando el menú de inicio**

En nuestro juego, también será preciso crear nuestro propio menú. No queremos que todos nuestros juegos se vean iguales, ¿verdad?  
  
Para realizar esta tarea, es preciso modificar el archivo vsmenu.gen

**Definiendo nuevos comportamientos para nuestros enemigos**

VScreen por defecto incluye algunas rutinas de comportamiento para nuestros enemigos, cuyos valores pueden ser modificados. Esto nos permite crear una gran variedad de combinaciones. Sin embargo, es muy probable que para nuestro juego queramos añadir nuevas rutinas y comportamientos. Para hacerlo es necesario conocer los siguientes detalles:

Cada enemigo es gestionado por una tabla de 15 bytes. El segundo Byte (ix+1) define su comportamiento. (Seguir un patrón adelante-atrás, parar e invertir movimiento al final de cada plataforma o dejarse caer al final de la plataforma). Existe un cuarto tipo (seguir una trayectoria vectorial) que está codificado, pero no testeado.

El código de estos comportamientos, se encuentra ubicado en los archivos enmtype2.gen y enmtype3.gen. Por lo que será en estos archivos donde necesitarás realizar los cambios pertinentes.

**Definiendo y utilizando Tiles especiales**

VScreen tambien te permite crear Tiles especiales.

Al principio del archivo “vscreen.gen” encontrarás una tabla “Tiletlb” con la dirección de la acción a realizar para cada Tile con un número superior a 208. “Noact” significa “sin acción”, y apunta a una rutina que no hace nada. En esta tabla es donde podrás añadir tus propios Tiles especiales.