



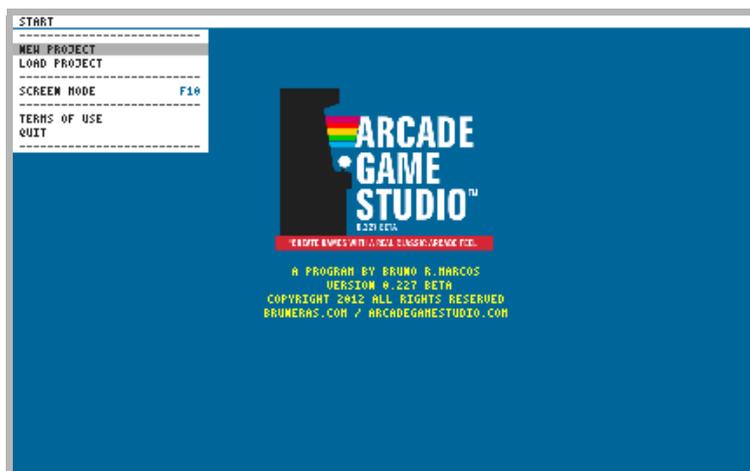
Desarrollador Independiente
de Videojuegos

[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

[english version](#)

Pantalla Start y géneros ayuda

La primera pantalla que aparece cuando cargas **Arcade Game Studio** es la pantalla **START**.

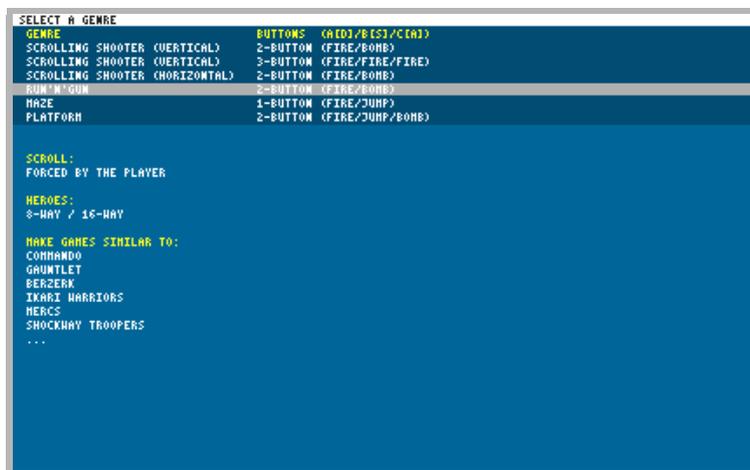


Pulsa [F10] para cambiar la resolución de la pantalla, todos los juegos usan una resolución de 256*256 pixels por lo que es importante mantener el aspecto del entorno de trabajo, **ARGS** se ajustará a esta resolución así que pueden aparecer unas franjas negras arriba y abajo de la pantalla. Pulsa [F11] para cambiar de ventana a pantalla completa y viceversa. Pulsa [F9] para guardar una captura de pantalla.

Algunos modos de resolución de pantalla parecerán iguales, esto es debido a que hay casos que en la pantalla de **STAGES** pueden aparecer líneas molestas entre las cuadrículas del fondo de fase, si esto ocurre probar distintos modos para seleccionar uno en que no aparezcan.

Selecciona **NEW PROJECT** para comenzar tu primer juego, un asistente aparecerá para guardar el fichero (puedes usar un nombre diferente que el del juego), recomendando crear una carpeta para el proyecto ya que cada proyecto creará las siguientes subcarpetas: actors, sounds, sprites, stages and tiles, aparte de los ficheros generales en el directorio principal.

Una vez hayas guardado tu proyecto debes elegir que género / configuración de botones quieres crear.



Arcade Game Studio actualmente soporta varios géneros, cada uno diseñado con unas mecánicas específicas, pero con la suficiente libertad como para crear juegos totalmente diferentes y originales.

Veamos estos generos mas de cerca, también toma nota que esta es mi visión particular de los géneros arcade:

Scrolling Shooter (shooters)

Género donde el jugador se desplaza hacia arriba (shooter vertical) o hacia la derecha (shooter horizontal), el desplazamiento de la pantalla (scroll) es automático y el objetivo usualmente es llegar al jefe final (boss) que espera al final de la fase.

Selecciona este género si quieres hacer juegos similares a *1942*, *Xevious*, *Flying Shark*, *Raiden*, *Gyrodine*, *Gunsmoke* (verticales), o *Scramble*, *R-Type*, *Side Arms*, *Gradius* (horizontales).

Si el scroll está deshabilitado entonces podemos llegar a hacer clásicos mata-marcianos similares a *Space Invaders*, *Galaxian* o *Galaga*.

Con algún truco también podrías llegar a hacer juegos del tipo *Arkanoid* or *Road Fighter*.

Run'n'Gun

Género donde el jugador se desplaza libremente por la pantalla y el desplazamiento por la fase es provocado por el jugador, también puedes fijar el movimiento verticalmente u horizontalmente por lo que el jugador se desplazaría sólo hacia arriba (*Commando*) o hacia la derecha (*Jail Break*), en estos casos el objetivo usualmente es llegar al jefe final (boss) que espera al final de la fase.

Selecciona este género si quieres hacer juegos similares a *Gauntlet*, *Commando*, *Jail Break*, *Ikary Warriors*, *Mercs* o *Shockway Troopers*.

También es perfecto para hacer juegos clásicos de pantalla por pantalla como *Berzerk* or *Robotron*.

Maze (laberintos)

Género donde el jugador tiene un movimiento de 4 direcciones (4-Way), la fase es por pantalla y generalmente es un juego de 'pillar al héroe'.

Selecciona este género si quieres hacer juegos similares a *Pac-Man*, *Eyes* o *Bomberman*.

Platform (plataformas)

Género donde el jugador se ve lateralmente, se mueve libremente por la pantalla y hay un efecto de gravedad que interactúa con el jugador, el desplazamiento de la fase es provocada por el jugador.

El jugador tiene otros movimientos a parte de moverse izquierda y derecha como agacharse, saltar o subir y bajar escaleras.

Selecciona este género si quieres hacer juegos similares a *Ghosts 'n' Goblins*, *Contra*, *Black Tiger*, *Toki*, *Sunset Riders* o por qué no *Metal Slug*.

En este género y con un buen conocimiento del programa se podrían incluso hacer juegos de lucha similares a *Kung Fu Master*, *Shinobi* o *Dragon Ninja*.

Otros géneros

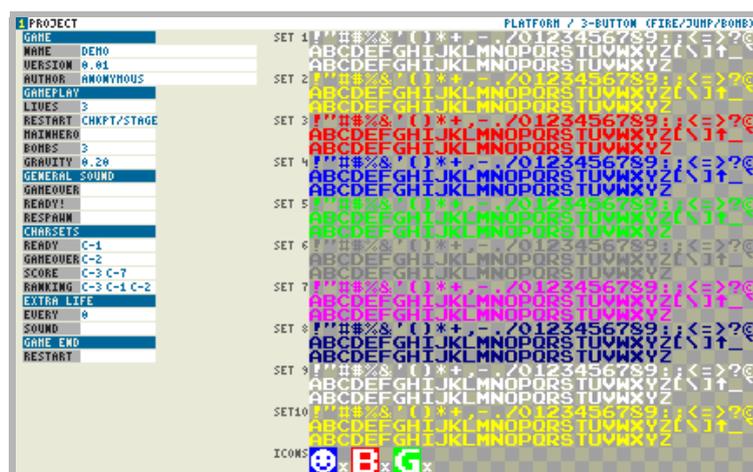
Aunque los géneros anteriores cubren un amplio espectro de los juegos arcade de los '80, hay otros generos que podrán ser añadidos con el tiempo, pero no quiero hacer ninguna promesa, estos incluyen juegos de 3ª persona como *Cabal*, *Beat'em-Ups* como *Final Fight* o *Fighting games* (juegos de lucha) como *Yie Ar Kung-Fu*.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.

Desarrollador Independiente
de Videojuegos
[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)
 [english version](#) 

Pantalla Project (proyecto) ayuda

Entra en la pantalla **PROJECT** pulsando [1], es la primera pantalla que veras cuando comienzas o cargas un proyecto, aquí tienes una imagen de como aparece.



Aquí es donde se define la configuración general del juego, aparece una lista de propiedades, veámoslas una por una:

NAME: nombre del juego, tiene un máximo de 25 caracteres.

VERSION: versión del juego, usualmente un código x.xx, pero puedes insertar el texto que quieras, tiene un máximo de 25 caracteres.

AUTHOR: tun nombre o el de tu compañía, tiene un máximo de 25 caracteres.

Gameplay settings (configuración general de jugabilidad)

LIVES: vidas, puede ser fijado a infinito.

RESTART: fija el modo de comienzo después de morir, puede ser **CHECKPOINT/STAGE**, por lo que se comienza desde el último checkpoint (punto de control) o la fase completa si no se coloca ninguno, o **RESPAWN** si quieres que el héroe continúe desde la posición donde murió.

MAIN HERO: es el héroe por defecto para el juego, se usará o no dependiendo de la propiedad que tiene asignada en cada fase (ver pantalla Stages para saber como).

BOMBS: bombas, son el arma secundaria, esto incluye cualquier acción secundaria como pueda ser también crear un campo de fuerza (shields), giros o cualquier cosa que queramos limitar, se puede deshabilitar para usar el arma secundaria (si la usamos en el héroe) como otra primaria.

GRAVITY: Gravedad, usado sólo para juegos **PLATFORM** (plataformas), fija la fuerza de la gravedad a cada actor que tiene la propiedad **GRAVITY** en **DEFAULT**.

General sound settings (configuración general de sonido)

GAME OVER: melodía (tune) que se reproduce cuando sale el mensaje "Game Over", mostrandose 5 segundos.

READY!: melodía (tune) que se reproduce cuando sale el mensaje "Ready!" mostrado 2 segundos, este mensaje no se muestra si la propiedad **RESTART** esta en **RESPAWN**.

RESPAWN: efecto de sonido que se reproduce al regenerar al héroe (respawn).

TIME UP!: melodía (tune) que se reproduce cuando se agota el tiempo de la fase (definido en la stage), sale el mensaje "TIME UP!", mostrandose 3 segundos.

Charsets settings (configuración de textos)

Ve [aquí](#) para leer más sobre charsets.

GAME OVER: texto usado para el mensaje "Game Over".

READY!: texto usado para el mensaje "Ready!".

SCORE: texto usado para la etiqueta de la puntuación y su valor, el primero define la etiqueta.

RANKING: texto usado para la tabla de puntuación, se introducen en este orden: posición, nombre y puntuación.

Extra Life (vida extra)

EVERY: puntos para conseguir una vida extra.

SOUND: efecto de sonido al conseguir una vida extra.

Game End (fin del juego)

RESTART: stage(fase) a la que salta una vez se complete la última stage, si no el juego salta a game over.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



Desarrollador Independiente
de Videojuegos

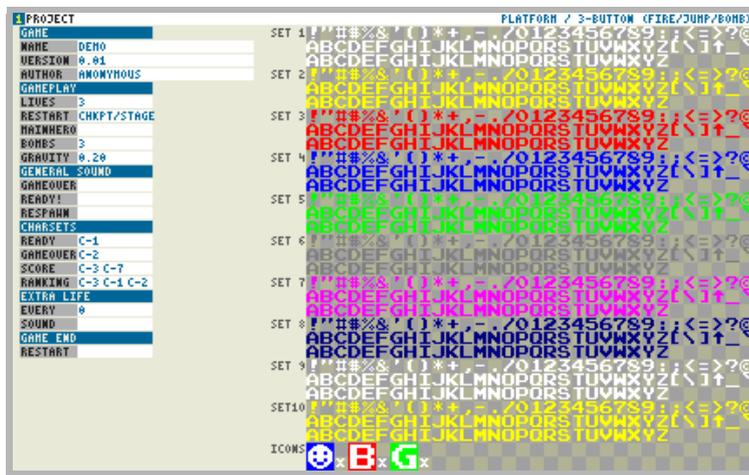
[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

 [english version](#) 

Charsets (textos) ayuda

Charsets son las fuentes de texto que utilizas para tu juego, son el Arial, Verdana o Tahoma para PC.

Aunque no utilices ningún texto en el juego, **ARGS** utilizará el charset que trae por defecto para mostrar la puntuacion, la tabla de resultados, o los textos para 'Ready!' y 'Game Over'.



Puedes cargar tu propio charset cuando quieras, ve a la pantalla PROJECT[1], selecciona SET CHARSET en el menú y carga la imagen, o puedes volver al charset por defecto seleccionando SET DEFAULT CHARSET.

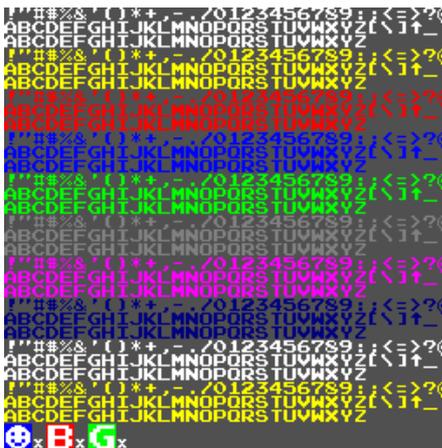
Un charset válido es una imagen PNG de 256*256 pixels, dividida en dos secciones:

- Una sección superior que contiene 10 párrafos con los caracteres ASCII, cada caracter es un icono de 8*8 pixels. El párrafo 9º y 10º se usan para texto parpadeante.
- Una sección inferior que tiene tres iconos en fila de 24*16 pixels, el primero representa las vidas, el segundo el arma secundaria (si esta habilitada) usada generalmente para bombas, escudos (*Phoenix*) o giros (*1942*). El tercero no se usa todavía.

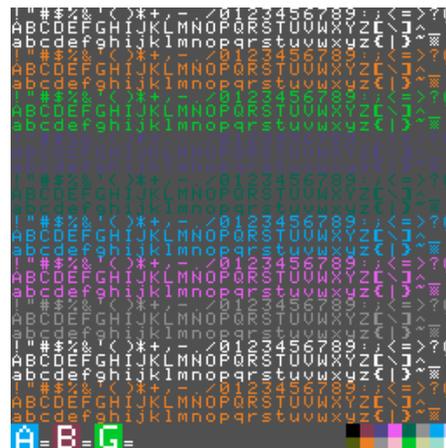
Mapa de caracteres del charset:

```
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~"¢
```

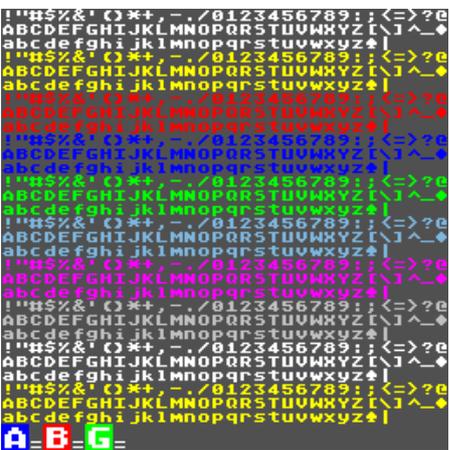
Aquí tienes algunos charsets predefinidos, pulsa el botón derecho del ratón y luego selecciona "guardar imagen como" para copiarlo a tu ordenador.



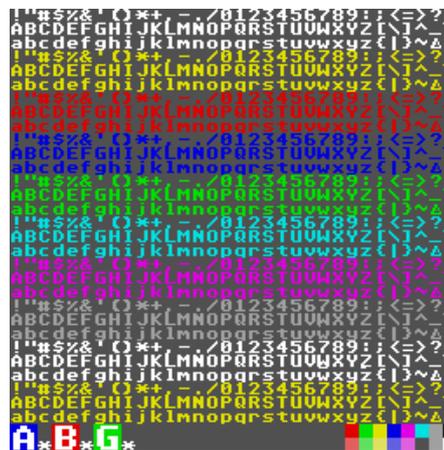
Arcade por defecto



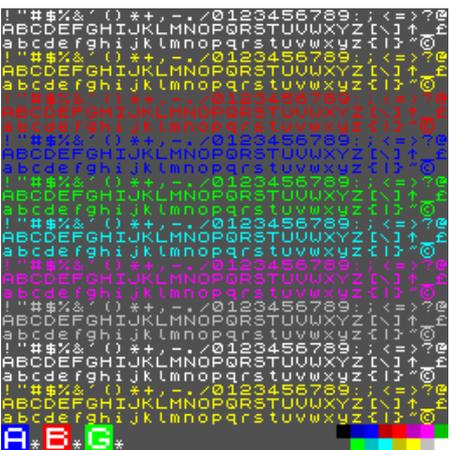
Apple II



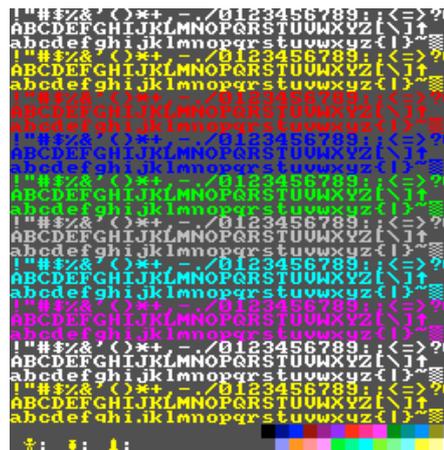
Atari 400/800



Atari ST



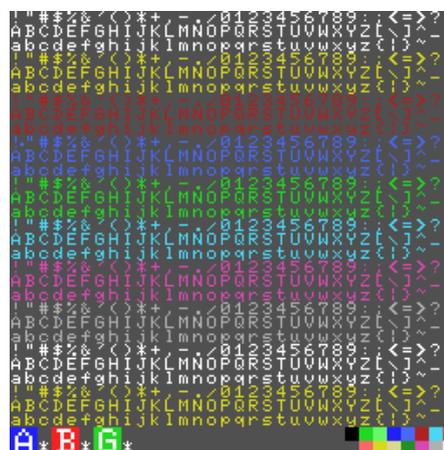
ZX Spectrum



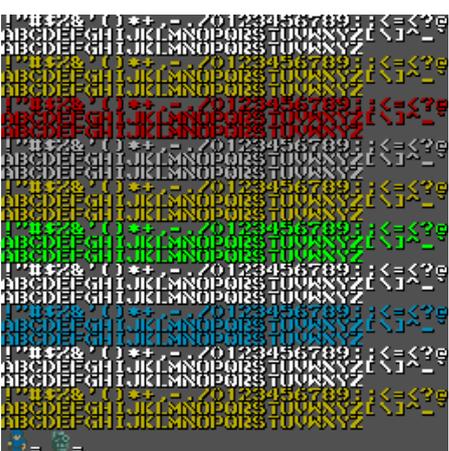
Amstrad CPC



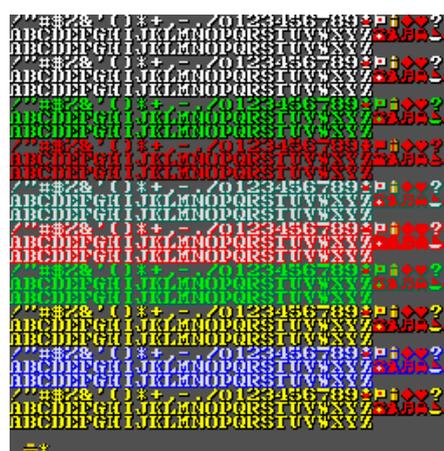
Commodore 64



MSX



Commando (Capcom 1985)



Gun.Smoke (Capcom 1985)

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.

Desarrollador Independiente
de Videojuegos
[Inicio](#)
[Ayuda y Tutoriales](#)
[Descargar](#)
[Juegos](#)
[Volver a principal](#)

[english version](#)


Pantalla Stages (fases) ayuda

Entra en la pantalla **STAGES** pulsando [2], aquí es donde creas las fases de tu juego.



Ante todo, fases no significan "jugar", las fases representan los diferentes actos de tu juego, como las fases donde no hay "disparos" y que se usan para contar una historia, como la introducción en *Ghosts'n'Goblins*.

Hay 5 tipos de fases: **NOTICE** (Aviso), **TITLE** (Título), **RANKING** (Marcador), **INTERMISSION** (Intermedio) and **GAME** (Juego).

Cuando creas un nuevo proyecto automáticamente se crean 3 fases: **NOTICE**, **TITLE** y **RANKING**, sólo tienes una de cada y no se pueden mover ni eliminar, veamos para que se utilizan.

NOTICE: esta fase es para tu presentación, logo o cualquier mensaje de aviso que quieras dar antes de empezar el juego, sólo aparece una vez al cargar el juego, se puede deshabilitar, luego veremos cómo.

TITLE: esta es la fase donde deberías poner el título del juego y es a donde los jugadores van antes de empezar una nueva partida (pulsando cualquier botón de disparo), se puede configurar para que se intercambie con la fase **RANKING** cada cierto tiempo.

RANKING: esta es la fase que muestra la clasificación, se puede deshabilitar pero entonces no se guardarán los nombres de los jugadores al acabar la partida. Una tabla de clasificación vacía se mostraría como en la imagen inferior, se puede inicializar y se genera un fichero a **.hi** que guarda los 8 mejores jugadores. Se puede configurar los charsets de la tabla (posición, nombre y puntos) en la pantalla **PROJECT**.



Puedes configurar estas fases de cualquier manera excepto que ocupan sólo una pantalla (no tienen scroll) y que no puedes colocar un actor heroe en ellas.

Hay otras dos fases, estas son las fases **INTERMISSION** y **GAME**.

INTERMISSION: esta fase es donde se debe crear las escenas de la historia, introducción al juego o cualquier cosa que se quiera contar entre fases del juego, puedes crear tantas como quieras.

GAME: esta fase es donde juegas, dependiendo del genero que hayas escogido tendrá diferentes propiedades y reglas, como veremos mas adelante.

Una cosa que debes saber de las fases es que se ejecutan de manera **sucesiva**, esto quiere decir que la posición en la lista indicará la posición en la que se ejecutará en el juego (excepto de la fase **RANKING** que salta hacia atrás a la fase **TITLE**), así que una fase **INTERMISSION** o **GAME** pasa a la siguiente de la lista cuando finaliza, y cuando llega al final el juego termina. Puedes mover las fases de posición usando [R] y [F] mientras mantienes pulsado [SHIFT], recuerda que **NOTICE**, **TITLE** y **RANKING** no pueden cambiar su posición.

Pulsa [F1] en el programa para conocer otras funcionalidades.

Fases Notice, Title, Ranking y Intermission

Sin depender de cómo configures la fase, todas ellas tienen en común sus propiedades.



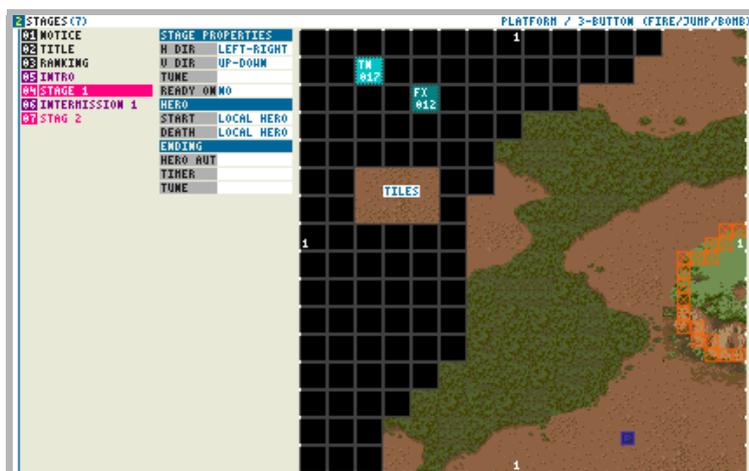
TIMER: establece el tiempo (en segundos) de la fase antes de pasar a la siguiente, si se establece a 0, hace falta que el jugador pulse **disparo** para pasar. Se pueden deshabilitar las fases **NOTICE** y **RANKING**.

TEXT: establece la velocidad de aparición del texto, también se le puede asociar un efecto de sonido cuando muestra un caracter.

TUNE: es la melodía que suena para la fase, se reproduce una sola vez, si se quieres que suene continuamente puedes colocar la melodía en el escenario directamente (luego veremos cómo).

Fases Game

Dependiendo del género escogido tendrán diferentes reglas y propiedades.



H DIR: sólo para juegos run'n'gun y plataformas, define el desplazamiento horizontal de la fase, hacia la derecha y/o izquierda.

V DIR: sólo para juegos run'n'gun y plataformas, define el desplazamiento vertical de la fase, hacia la derecha y/o izquierda.

SCROLL: para shooters, define la velocidad de desplazamiento (scroll) de la fase.

TUNE: es la melodía que suena para la fase y se reproduce de manera continua, por otra parte también puedes colocar un sonido en el escenario directamente (luego veremos cómo).

Propiedades **HERO** (héroe):

START: (movienzo) cuando colocamos un héroe en la escena esto no indica necesariamente que sea el héroe que usemos cuando comience la fase, eso dependerá de esta propiedad:

- **LOCAL HERO** usa el héroe colocado en la fase cuando esta comienza.
- **MAIN HERO** usa el héroe definido en **MAIN HERO** en la pantalla de proyecto.
- **PREVIOUS HERO** usa el último héroe de la fase anterior, para conservar el héroe entre fase y fase.

DEATH: (muerte) igual que el anterior pero se usa su propiedad para determinar el siguiente héroe después de morir.

COUNTDOWN: (tiempo) activa un contador de tiempo en segundos que si llega a 0 se pierde una vida.

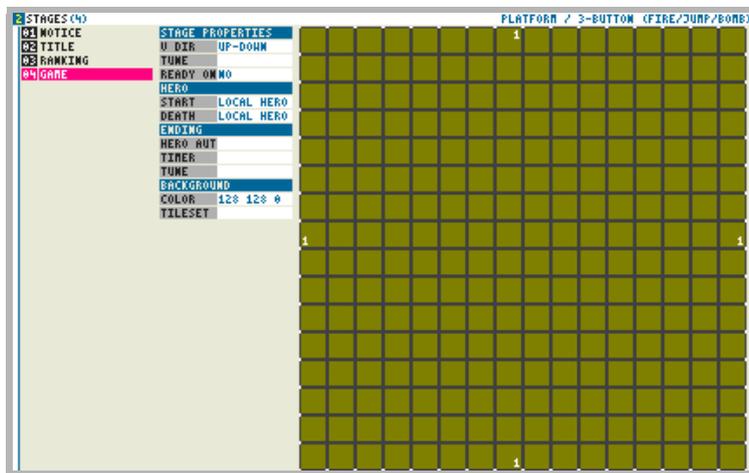
Propiedades **ENDING** (final):

HERO AUT: usualmente, cuando se termina una fase, quieres que el héroe haga una animación o movimiento predefinido, esta propiedad establece un héroe (automatizado) para esto.

TIMER: establece el tiempo (en segundos) de la fase antes de pasar a la siguiente, pero sólo cuando la fase ya ha terminado (cuando un actor **CHECK** lo indica).

TUNE: es la melodía que suena cuando la fase ha terminado.

Version 1.1 tiene dos nuevos parámetros **BACKGROUND**, estos son **COLOR** y **TILESET**.



COLOR: establece el color de fondo para la fase en formato RGB.

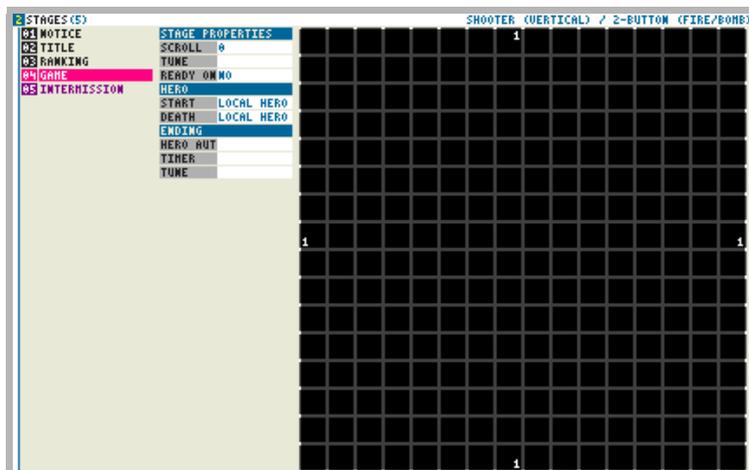
TILESET: establece un tileset estático como fondo de la fase.

Version 1.2 añade 3 parámetros a **TILESET**: **MODE** (modo), **SPEED** (velocidad), **DIRECTION** (dirección)

- **FIXED:** para un fondo estático
- **BOXED:** el fondo se mueve proporcionalmente al tamaño de la fase (solo visible en fases **GAME**), el fondo (**TILE**) debe ser mayor que la pantalla 256x256 para notar el efecto.
- **MOVE:** se puede establecer 2 parámetros al fondo (**SPEED** y **DIRECTION**), el fondo se desplazará hasta que el borde es alcanzado, útil para por ejemplo mostrar un mapa como en Ghosts'n'Goblins.
- **MOSAIC:** se puede establecer 2 parámetros al fondo (**SPEED** y **DIRECTION**), el fondo se desplazará cíclicamente, perfecto para fondos de estrellas.

Editor

El editor es la zona a la derecha de la pantalla, está dividida en 16*16 celdas, cada celda es 16*16 pixels, así que eso hace un área de 256*256 pixels, o lo que es lo mismo, la resolución de pantalla para cada juego.



Sólo fases **GAME** pueden ser mayores que el tamaño de pantalla, puedes mover la fase con las teclas [WASD] (pulsa [SHIFT] para desplazarte rápidamente), cuando el cursor está sobre el editor muestra una etiqueta con el modo de edición en el que está, este puede ser uno de los siguientes: **TILES**(fondos), **WALL/EVENT**(muro/evento), **ACTORS**(actores), **TEXT**(texto) y **SOUNDS**(sonidos). Usa [Q] y [E] para cambiar el modo de edición.

TILES: (fondos) este modo es para colocar fondos. Ve en la pantalla **TILES** [3], selecciona un área, vuelve pulsando [2] y colócalo en cualquier sitio de la fase. Pulsa el botón derecho del ratón para de-seleccionar el fondo, mantén pulsado [SHIFT] para borrar la celda de la fase. Una fase es tan grande como el fondo que contiene, o hasta que se encuentra una fila o columna vacía, siempre comenzando desde la parte inferior izquierda.

WALL/EVENTS: este modo edición es para colocar muros, escaleras o celdas de evento, usa [Z] y [X] para

cambiar el evento seleccionado, este puede ser uno de los siguientes:

-  **WALL** (muro), actores con colisiones activo no puede atravesar estas celdas.
-  **PLATFORM** (plataforma), sólo para juegos de plataformas, actores pueden estar encima de ellos verticalmente y a través de ellos horizontalmente.
-  **LADDER** (escalera), sólo para juegos de plataformas, héroes puede subir y bajar por estas celdas.
-  **AREA** (área), sólo para juegos run'n'gun, los actores héroe pueden cambiar su estado al atravesarlos (como si tu héroe entrara en el agua), también se puede definir como muro, así que por ejemplo, un héroe *tanque* no puede atravesarlo, pero un héroe *hombre* si puede.
-  **DEATH** (muerte) el héroe muere automáticamente al colisionar con él.
-  **CHECKPOINT** (punto de control), marca un punto de control en la fase, si el héroe lo alcanza al morir la fase se reanuda desde este punto. La propiedad **RESTART** en la pantalla de proyecto debe estar en **CHECKPOINT/STAGE**.
-  **TRACK** (trayectoria), un actor que colisiona con estas celdas cambiará su movimiento a la dirección marcada si tiene activado su icono de comportamiento **TRACK**.
-  **END** (final), si tu héroe llega a esta celda la fase se da por terminada, realizará su evento de fin de fase y pasará a la siguiente.

ACTORS: (actores), este modo es para colocar los actores en la escena. Usa [Z] y [X] para seleccionar un actor o pulsa [] (Tecla entre Esc y Tab) para seleccionarlo de una lista. Los actores se activan cuando están en el borde de la escena (parte visible de la fase). Un punto parpadeará en blanco y negro en el origen de cada actor. con el cursor sin ningún actor seleccionado puedes arrastrar los actores en pantalla.



Haz click para colocar un actor, si desplazas el ratón mientras colocas al actor aparecerá una flecha indicando la posición de la que partir cuando el actor se activa, además, en juego con scroll, mantén pulsado [SHIFT] para que además sea el punto de aparición del actor cuando el scroll alcanza esa posición.

TEXT: (texto) este modo es para escribir texto en la fase. Usa [Z] y [X] para cambiar de charset y pulsa en el editor para entrar en modo edición, puedes utilizar los cursores para mover el cursor de texto. Pulsa [ESC] o el botón derecho para salir del modo edición. Usa el botón derecho para eliminar caracteres.

SOUNDS: (sonidos) este modo es para colocar sonidos en la fase. Usa [Z] y [X] para seleccionar un sonido o pulsa [] (Tecla entre Esc y Tab) para seleccionarlo de una lista. Los efectos de sonidos o melodías comenzarán cuando se activen para la en la escena, esto por ejemplo, sirve para cambiar la música cuando nos acercamos al final de la fase.

Pulsa [Tab] para ocultar/mostrar los elementos del modo seleccionado.

Pulsa [espacio] para entrar en modo de pruebas (test mode) de la fase, la fase siempre comenzará centrada en el héroe situado en la parte más inferior y a la izquierda de la fase. En modo test pulsa [Tab] para mostrar/ocultar los muros y eventos. EN la versión 1.1 también se mostrará el bounding box(superficie) de colisiones con muros (blanco) y colisiones de golpeo (rojo).

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



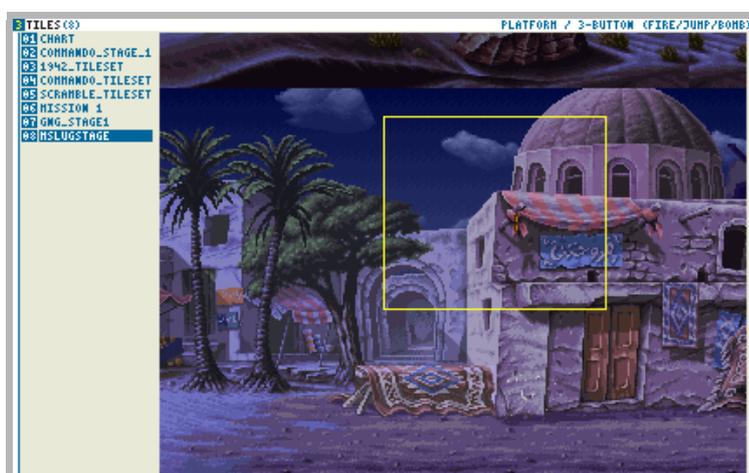
Desarrollador Independiente
de Videojuegos

[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

 [english version](#) 

Pantalla Tiles (fondos) [ayuda](#)

Entra en la pantalla **TILES** pulsando [3], aquí es donde se cargan los fondos para tu juego.



Los ficheros de imagen deben estar en formato **PNG** y un tamaño máximo de 2048*2048 pixels.

Si el fondo es mayor que la pantalla muévelo con [WASD]. Haz click en cualquier sitio de la imagen para comenzar a seleccionar un área, puedes mover el fondo mientras pulsas para seleccionar un área mayor, la primera celda seleccionada será el punto de anclaje, usa esto para colocarlo más fácilmente en la fase.

El área seleccionable mas pequeño es de 16*16 pixels.

También puedes sustituir (replace) o eliminar (delete) un fondo, y exportar o importar todos tus fondos en un único fichero.

Pulsa [F1] en el programa para conocer otras funcionalidades.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



Desarrollador Independiente
de Videojuegos

[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

 [english version](#) 

Pantalla Actors (actores) ayuda

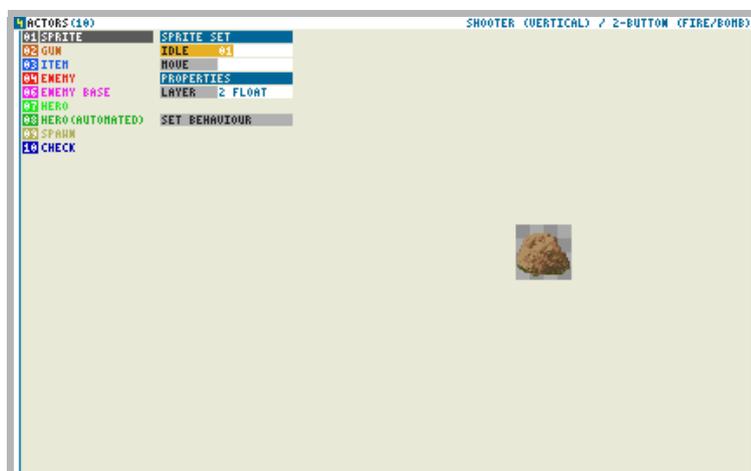
Entra en la pantalla **ACTORS** pulsando [4], aquí es donde se crean los actores de tu juego.



Todos los juegos, en mayor o menor medida, tienen los mismos actores, estos son básicamente los siguientes: **HEROES**(héroes), **ENEMIES**(enemigos), **GUNS**(balas), **ITEMS** y **SPRITES**, cada uno con una finalidad diferente.

Sprites

Un **SPRITE** es el actor más simple que se puede usar, esencialmente hace... nada, pues no causa ni daño ni se le puede matar.



Úsalo para crear los extras del juego, como rocas, árboles superpuestos, nubes, humo, fuego de disparo o cualquier otra cosa, esto no significa que no se puedan hacer comportamientos complejos con él, por ejemplo, puedes convertir un simple sprite en una maquina de matar usándolo como un generador de balas, gracias a la flexibilidad de los comportamientos.

SPRITE spritesets:

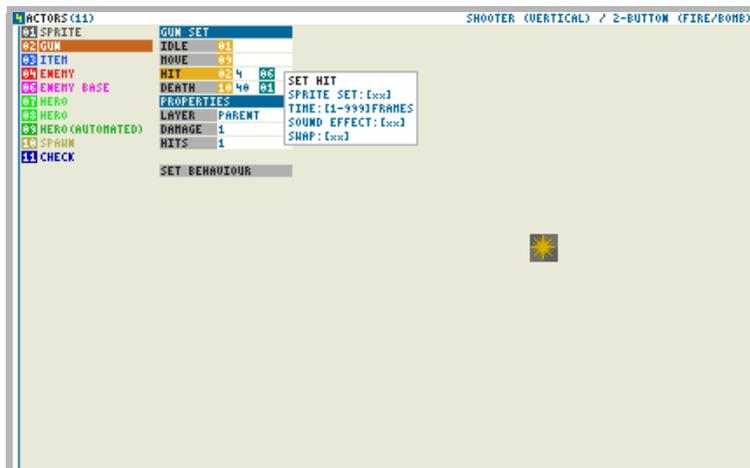
- **IDLE**: (parado) spriteset que se muestra cuando el actor no se mueve.
- **MOVE**: (moviendo) spriteset que se muestra cuando el actor se mueve.

SPRITE propiedades:

- **LAYER**: capa de dibujo (1-6, capas altas se dibujan sobre capas bajas), y el estado **float** del actor.
- **GRAVITY**: (gravedad) usado sólo en juegos de plataformas, establece la fuerza de gravedad al actor, un valor **DEFAULT** hace que se use el valor definido en la pantalla principal de proyecto.

Guns (balas)

Un actor **GUN** es un actor que se usa como arma, causa daños al héroe y a los enemigos dependiendo de quién lo creó. Úsalo para crear misiles, lasers o escudos...



GUN spritesets:

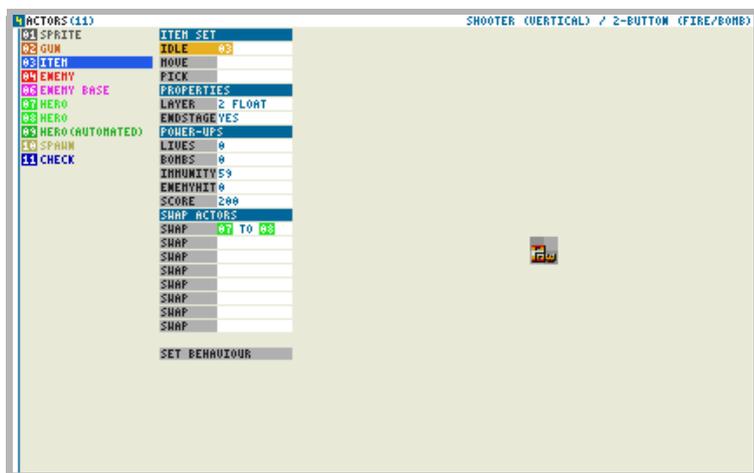
- **IDLE**: (parado) spriteset que se muestra cuando el actor no se mueve.
- **MOVE**: (moviendo) spriteset que se muestra cuando el actor se mueve.
- **HIT**: (golpear) spriteset que se muestra cuando la bala golpea a un enemigo (si es una bala del héroe) o al héroe, tiene otros parámetros como muestra la imagen superior: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de desaparecer, **SOUND EFFECT** efecto de sonido cuando golpea y/o **SWAP** (intercambio) del actor a otro cuando golpea.
- **DEATH**: (muerte) spriteset que se muestra cuando la bala muere, la muerte de la bala se produce con el icono de comportamiento **DEATH** o cuando choca con un muro y esta la colisión con muros habilitada, tiene los mismos atributos que **HIT**.

GUN propiedades:

- **LAYER**: capa de dibujo (1-6, capas altas se dibujan sobre capas bajas), y el estado **float**. El valor se puede establecer a **PARENT** si quieres que copie los valores del actor creador.
- **DAMAGE**: (daño) valor del daño causado al enemigo o héroe, el valor **VITALITY**(vitalidad) se decrementará en esa cantidad.
- **HITS**: (golpes) establece cuantos golpes consecutivos puede dar la bala antes de desaparecer, por ejemplo, un rayo que lo destruya todo deberá tener este valor en **INFINITE**.
- **HIT BOX**: Establece el área de colisión con actores **ENEMY/HERO**: puede ser **FULL SPRITE**(el sprite completo) o **BOUNDING BOX**(área definida en el spriteset).
- **GRAVITY**: (gravedad) usado sólo en juegos de plataformas, establece la fuerza de gravedad al actor, un valor **DEFAULT** hace que se use el valor definido en la pantalla principal de proyecto.

Items

Un **ITEM** es un actor que el héroe puede recoger, usualmente representa un power-up, un arma o vehículo o simplemente monedas que dan puntos.



ITEM spritesets:

- **IDLE**: (parado) spriteset que se muestra cuando el actor no se mueve.
- **MOVE**: (moviendo) spriteset que se muestra cuando el actor se mueve.
- **PICK**: (recoger) spriteset que se muestra cuando recoges el item (colisión), tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de desaparecer, **SOUND EFFECT** efecto de sonido cuando recoge y/o **SWAP** (intercambio) del actor a otro.

ITEM propiedades:

- **LAYER**: capa de dibujo (1-6, capas altas se dibujan sobre capas bajas), y el estado **float** del actor.
- **ENDSTAGE**: lanza el evento de final de fase (propiedades **ENDING** de la fase).
- **GRAVITY**: (gravedad) usado sólo en juegos de plataformas, establece la fuerza de gravedad al actor, un valor **DEFAULT** hace que se use el valor definido en la pantalla principal de proyecto.

ITEM power-ups:

- **LIVES**: (vidas) incrementa/decrementa las vidas.
- **BOMBS**: (bombas) incrementa/decrementa el arma secundaria si está habilitada.
- **IMMUNITY**: (inmunidad) establece un periodo de inmunidad en fotogramas.
- **ENEMYHIT**: todos los enemigos son dañados an la cantidad indicada.
- **SCORE**: (puntos) puntos que se consiguen cuando se recoge.

ITEM swap actors (intercambios):

Intercambia actores a otros actores, si no se establece un actor destino el actor se elimina. Puedes usar esto para cambiar tu héroe a otro más poderoso, abrir puertas (eliminando la puerta), o convertir todos los enemigos en items, por ejemplo.

Enemies (enemigos)

Un **ENEMY** es un actor al que generalmente hay que esquivar o destruir.



ENEMY spritesets:

- **IDLE:** (parado) spriteset que se muestra cuando el actor no se mueve.
- **MOVE:** (moviendo) spriteset que se muestra cuando el actor se mueve.
- **HIT:** (golpe) spriteset que se muestra cuando el actor es alcanzado por una bala, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de volver al estado normal, si se ignora (tecla TAB) tomará el valor por defecto del spriteset, y **SOUND EFFECT** efecto de sonido cuando es golpeado. Un actor enemigo permanece inmóvil el tiempo del spriteset cuando es golpeado.
- **DEATH:** (muerte) spriteset que se muestra cuando el actor muere, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset, si se ignora (tecla TAB) tomará el valor por defecto del spriteset, **SOUND EFFECT** efecto de sonido cuando muere, y si queremos **SWAP** (intercambio) si se quiere cambiar el actor a otro cuando finaliza la animación de la muerte.
- **FIRE:** (disparo) spriteset que se muestra cuando el actor dispara, solo se produce si **GUN 1** está definido, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset, si se ignora (tecla TAB) tomará el valor por defecto del spriteset, **SOUND EFFECT** efecto de sonido cuando dispara, y **SWAP** (intercambio) si se quiere cambiar el actor a otro cuando finaliza la animación de disparo.

ENEMY propiedades:

- **LAYER:** capa de dibujo (1-6, capas altas se dibujan sobre capas bajas), y el estado **float** del actor.
- **HIT BOX:** Establece el área de colisión con actores **GUN/HERO:** puede ser **FULL SPRITE**(el sprite completo) o **BOUNDING BOX**(área definida en el spriteset), también establece un valor **SOLID** si quieres que el área de colisión del actor sea sólido, también se puede establecer un valor **PLATFORM** si queremos que actúe como tal (sólo para el género platform).
- **VITALITY:** (vitalidad) cuando alcanza 0 el actor muere, este valor hace al actor más o menos poderoso. Un valor de 0 da al actor vida infinita. Tiene también un efecto **SHADE**(tono).
- **DAMAGE:** (daño) causado al héroe si hay colisión.
- **SCORE:** (puntos) incrementa los puntos del jugador.
- **GRAVITY:** (gravedad) usado sólo en juegos de plataformas, establece la fuerza de gravedad al actor, un valor **DEFAULT** hace que se use el valor definido en la pantalla principal de proyecto.

ENEMY fire guns (armas):

- **DELAY:** (retardo) cuando un evento de disparo de activa, esta propiedad indica el retardo en fotogramas en crear la bala, por ejemplo, si tenemos una planta carnívora que su animación es abriendo la boca y disparando un ojo (*Ghosts 'n' Goblins*), la bala (ojo) es lanzada a la mitad de la animación (**FIRE** spriteset), este campo es para -encajar- la animación con la creación de la bala.
- **GUN 1-4:** (bala), tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de volver a llamar al evento disparo. el spriteset **FIRE** sólo se mostrará si **GUN 1** está definido.

Cada arma tiene sus propias posiciones en el actor. Para fijar una posición primero selecciona un **GUN** y luego selecciona el **SPRITESET** que quieres definir, sólomente se pueden definir spriteset con la etiqueta **G**. Desplázate sobre un sprite a la derecha, una cruz (**X**) indica que no hay posición fijada en el sprite, haz click en cualquier parte del sprite para fijar la posición de donde saldrá la bala, mantén la tecla [shift] para centrar la posición en el sprite, o pulsa el botón derecho para desactivar la posición. El actor sólo disparará si el sprite tiene la posición activa.

Enemies (base) (enemigos base)

Un actor **ENEMY BASE** es un actor al que generalmente hay que esquivar o destruir, la diferencia con **ENEMY** es que puede tener enemigos adjuntos, por ejemplo un tanque, el enemigo base es el vehículo y el cañon y la ametralladora son enemigos adjuntos.



ENEMY BASE spritesets:

- **IDLE**: (parado) spriteset que se muestra cuando el actor no se mueve.
- **MOVE**: (moviendo) spriteset que se muestra cuando el actor se mueve.
- **HIT**: (solo para bases enemy en estado **ACTIVE**, ver **BASEMODE**) (golpe) spriteset que se muestra cuando el actor es alcanzado por una bala, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de volver al estado normal, si se ignora (tecla TAB) tomará el valor por defecto del spriteset, y **SOUND EFFECT** efecto de sonido cuando es golpeado. Un actor enemigo permanece inmóvil el tiempo del spriteset cuando es golpeado.
- **DEATH**: (muerte) spriteset que se muestra cuando el actor muere, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset, si se ignora (tecla TAB) tomará el valor por defecto del spriteset, **SOUND EFFECT** efecto de sonido cuando muere, y si queremos **SWAP** (intercambio) si se quiere cambiar el actor a otro cuando finaliza la animación de la muerte.

ENEMY BASE properties:

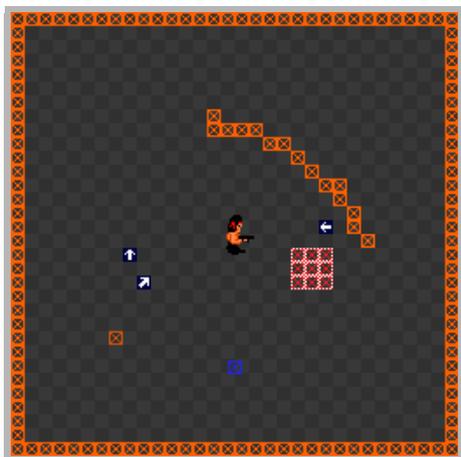
- **BASEMODE**: (modo base), tiene dos estados **ACTIVE**(activo): la base y los enemigos adjuntos actúan como un todo, se necesita destruir la base, pues destruir sus enemigos adjuntos no es suficiente. **PASIVE**(pasivo): la base sólo actúa como soporte para un grupo de enemigos adjuntos, se necesita destruir todos ellos para destruir la base, por lo que en modo pasivo el actor base debe contener al menos un enemigo adjunto para existir.
- **LAYER**: capa de dibujo (1-6, capas altas se dibujan sobre capas bajas), y el estado **float** del actor.
- **HIT BOX**: Establece el área de colisión con actores **GUN**: puede ser **FULL SPRITE**(el sprite completo) o **BOUNDING BOX**(área definida en el spriteset), también establece un valor **SOLID** si quieres que el área de colisión del actor sea sólido.
- **VITALITY**: (vitalidad) sólo para bases **ACTIVE**, cuando vitality alcanza un valor 0 el actor muere y todos sus enemigos adjuntos, esta valor hace al actor más o menos poderoso. Un valor de 0 da al actor vida infinita. Tiene también un efecto **SHADE**(resplandor).
- **DAMAGE**: (daño) causado al héroe si hay colisión.
- **SCORE**: (puntos) incrementa los puntos del jugador.
- **GRAVITY**: (gravedad) usado sólo en juegos de plataformas, establece la fuerza de gravedad al actor, un valor **DEFAULT** hace que se use el valor definido en la pantalla principal de proyecto.

ENEMY BASE attachments (adjuntos):

- **ENEMY 1-4**: enemigos adjuntos, cada uno tiene su propia posición el actor base. Para fijar una posición primero selecciona un **ENEMY** y luego selecciona el spriteset **IDLE** (marcado con A). Desplázate sobre un sprite a la derecha y haz click en cualquier parte del sprite para fijar la posición, mantén la tecla [shift] para centrar la posición en el sprite.

Test mode (modo de pruebas)

Pulsa [espacio] para entrar en el modo de pruebas del actor seleccionado, si el actor no es un héroe un cuadrado azul aparecerá haciendo del mismo, puedes moverlo con los cursores.



Coloca muros y otros elementos en tiempo real con el ratón, [Z,X] para cambiar elemento, pulsa [F1] en la pantalla actores para ver como colocar otros elementos. además con ['](tecla entre ESC TAB) se cambia el color de fondo, y [TAB] muestra/oculta los boundingbox de colisión con muros (blanco) y colision de vida (rojo).

Estado *floating* (flotante) del actor

En la propiedad **LAYER** de cada actor se define el estado **flotante** (float), para juegos donde las fases son mayores que la pantalla esto establece si el actor se desplaza con el movimiento (scroll) o mantiene la posición en pantalla, por ejemplo, en shooters verticales como *Raiden* sólo los actores de suelo (casas, tanques,..) se desplazan con el scroll, mira este [video de 1942](#) de Capcom para ver el efecto, si te fijas en los aviones enemigos puedes percibir cuando se desplazan horizontalmente como no son arrastrados por el scroll vertical, el efecto pasa desapercibido, pero es fundamental para que el juego gane en jugabilidad.

Si activamos el estado flotante del actor da lo mismo que velocidad de scroll tengamos, el actor no se vera arrastrado por él, en *1942* lo que esencialmente tenemos es un inmenso juego *Galaga* de la 2ª GM con un fondo que se desplaza.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



Desarrollador Independiente
de Videojuegos

[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

 [english version](#) 

Actores: Héroes ayuda

Los héroes son los actores que puedes controlar en el juego, es decir, el jugador, dependiendo de género escogido puedes crear un héroe particular, veámoslos por género.

Importante: Los bounding box de los Spritesets deberían tener la misma posición y tamaño para no experimentar comportamientos erróneos al mover el Actor.

Shooter vertical (2-botones)

Puedes crear un héroe 1-WAY y 3H-WAY, a parte del específico uso de los spriteset de x-ways(x-direcciones) son basicamente iguales, excepto que el héroe 3H-WAY tiene la propiedad **ROLL**.



HERO spritesets:

- **IDLEMOVE**: como juego de scroll automático, el spriteset se usa tanto para cuando no se mueve como en movimiento.
- **DEATH**: (muerte) spriteset que se muestra cuando el héroe muere, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de que la fase reinicie, **SOUND EFFECT** efecto de sonido cuando muere y/o **SWAP** (intercambio) del héroe a otro, puede ser un héroe **1-WAY**.
- **BUTTON-A**: spriteset que se muestra al disparar [D], sólo se produce si **GUN 1** está definido, tiene los mismos atributos que **DEATH** sin la opción **SWAP**.
- **BUTTON-B**: spriteset que se muestra al disparar bombas [S], sólo se produce si **GUN 3** está definido, tiene los mismos atributos que **DEATH**.

HERO propiedades:

- **LAYER**: capa de dibujo (1-6, capas altas se dibujan sobre capas bajas).
- **WALLCOLS**: establece la colisión con muros.
- **HIT BOX**: Establece el área de colisión con actores **GUN**: puede ser **FULL SPRITE**(el sprite completo) o **BOUNDING BOX**(área definida en el spriteset).
- **SPEED**: (velocidad) de movimiento en pixels, puede tener decimales.
- **VITALITY**: (vitalidad) cuando alcanza 0 el héroe muere. Un valor de 0 da al héroe vida infinita. Se puede definir un efecto **SHADE** (tono) al recibir un impacto.
- **FIXED**: establece el movimiento del héroe.
- **ROLL**: (girar) sólo para héroes **3H-WAY**, establece si la animación del spriteset **IDLEMOVE** se revierte cuando la dirección cambia, esto genera un efecto de giro.

HERO Button-A:

- **RANGED**: (oscilar) establece un movimiento oscilatorio de disparo al desplazarse similar a *Gyrodine*.
- **GUN 1-2**: Arma primaria, tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de poder volver a disparar y un parámetro **AUTOFIRE**, que marca cuantos disparos seguidos puede realizar por cada pulsación. el spriteset **BUTTON-A** sólo se mostrará si **GUN 1** está definido.

HERO Button-B:

- **GUN 3-4**: Arma secundaria, tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de poder volver a disparar y un parámetro **AUTOFIRE**, que marca cuantos disparos seguidos puede realizar por cada pulsación. el spriteset **BUTTON-B** sólo se mostrará si **GUN 3** está definido. Decrementa el valor bombas si la propiedad **BOMB** de la ventana proyecto esta habilitada.

Cada arma tiene sus propias posiciones en el actor. Para fijar una posición primero selecciona un **GUN** y luego selecciona el **SPRITESET** que quieres definir, sólomente se pueden definir spriteset con la etiqueta **G**. Desplázate sobre un sprite a la derecha, una cruz (**X**) indica que no hay posición fijada en el sprite, haz click en cualquier parte del sprite para fijar la posición de donde saldrá la bala, mantén la tecla [shift] para centrar la posición en el sprite, o pulsa el botón derecho para desactivar la posición. El héroe sólo disparará si el sprite tiene la posición activa.

Shooter vertical (3-botones)

Puedes crear un héroe **3H-WAY** con un botón de disparo para cada dirección como *Gunsmoke* de Capcom.



HERO spritesets:

- **IDLEMOVE**: como juego de scroll automático, el spriteset se usa tanto para cuando no se mueve como en movimiento.
- **DEATH**: (muerte) spriteset que se muestra cuando el héroe muere, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de que la fase reinicie, **SOUND EFFECT** efecto de sonido cuando muere y/o **SWAP** (intercambio) del héroe a otro, puede ser un héroe **1-WAY**.
- **3-W FIRE**: spriteset que se muestra al disparar, la configuración es: [A] hacia la **izquierda**, [S] hacia **arriba** and [D] hacia la **derecha**, sólo se produce si **GUN 1** está definido, tiene los mismos atributos que **DEATH** sin la opción **SWAP**.

HERO propiedades:

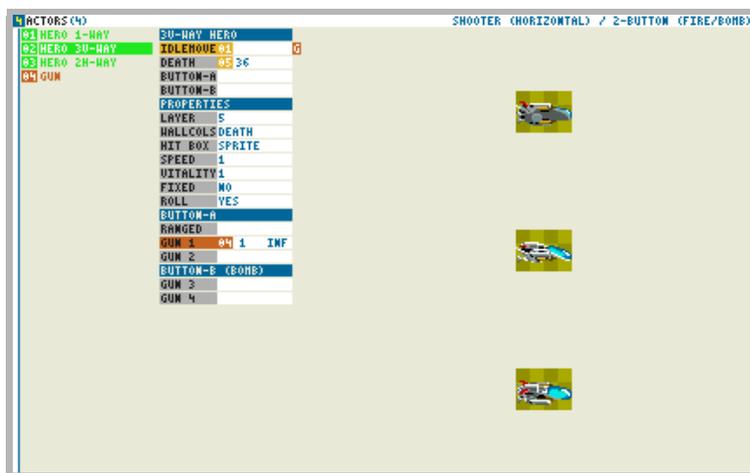
- las mismas que shooter vertical (2-botones) sin la propiedad **ROLL**.

HERO 3-w fire (disparo 3-direcciones):

- **GUN 1-2**: tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de poder volver a disparar, y un parámetro **AUTOFIRE** que marca cuantos disparos seguidos puede realizar por cada pulsación. el spriteset **3-W FIRE** sólo se mostrará si **GUN 1** está definido.

Shooter horizontal (2-botones)

Puedes crear un héroe **1-WAY**, **3V-WAY** y **2H-WAY**, a parte del específico uso de los spriteset de x-ways(x-direcciones) son basicamente iguales, excepto que le héroe **3H-WAY** tiene una propiedad **ROLL**.



Los spritesets, propiedades y configuración de los botones son los equivalentes horizontales a los de género shooter vertical definido arriba.

Maze(laberinto) (2-botones)

Puedes crear un héroe 4-WAY.



HERO spritesets:

- **IDLE:** (parado) spriteset que se muestra cuando el actor no se mueve.
- **MOVE:** (moviendo) spriteset que se muestra cuando el actor se mueve.
- **DEATH:** (muerte) spriteset que se muestra cuando el héroe muere, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de que la fase reinicie, **SOUND EFFECT** efecto de sonido cuando muere y/o **SWAP** (intercambio) del héroe a otro, puede ser un héroe **1-WAY**.
- **JUMP:** (salto) spriteset que se muestra cuando salta [S], la propiedad **SPEED jump** (velocidad de salto) debe estar definida.
- **BUTTON-A:** spriteset que se muestra al disparar [D], sólo se produce si **GUN 1** está definido, tiene los mismos atributos que **DEATH** sin la opción **SWAP**. Si se usa, el héroe permanece inmóvil el tiempo del spriteset.

HERO propiedades:

- **LAYER:** capa de dibujo (1-6, capas altas se dibujan sobre capas bajas).
- **HIT BOX:** Establece el área de colisión con actores **GUN:** puede ser **FULL SPRITE**(el sprite completo) o **BOUNDING BOX**(área definida en el spriteset).
- **SPEED:** (velocidad) de movimiento en pixels, puede tener decimales. Se puede activar el atributo **LOCK** para bloquear la velocidad por lo que el héroe permanecerá moviéndose, y establecer la velocidad de salto
- **VITALITY:** (vitalidad) cuando alcanza 0 el héroe muere. Un valor de 0 da al héroe vida infinita. Se puede definir un efecto **SHADE** (tono) al recibir un impacto.
- **EDGECONT:** Establece margen continuo, si el héroe sale por un lado de la pantalla aparece por el lado

contrario.

- **GRAVITY:** (gravedad) establece la fuerza de gravedad, válido si el valor de velocidad de salto esta definido.
- **JUMP MOV:** (movimiento en salto) activa/desactiva control del héroe mientras se esta saltando.

HERO Button-A:

- **GUN 1-2:** tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de poder volver a disparar, y un parámetro **AUTOFIRE** que marca cuantos disparos seguidos puede realizar por cada pulsación. el spriteset **BUTTON-A** sólo se mostrará si **GUN 1** está definido.

Cada arma tiene sus propias posiciones en el actor. Para fijar una posición primero selecciona un **GUN** y luego selecciona el **SPRITESET** que quieres definir, sólomente se pueden definir spriteset con la etiqueta **G**. Desplázate sobre un sprite a la derecha, una cruz (**X**) indica que no hay posición fijada en el sprite, haz click en cualquier parte del sprite para fijar la posición de donde saldrá la bala, mantén la tecla [shift] para centrar la posición en el sprite, o pulsa el botón derecho para desactivar la posición. El héroe sólo disparará si el sprite tiene la posición activa.

Run 'N' Gun (2-botones)

Puedes crear un héroe **8-WAY** y **16-WAY**, son basicamente iguales, excepto que el héroe **16-WAY** tiene un movimiento rotatorio mientras realiza un giro por lo que tiene un mayor rango de tiro.



HERO spritesets:

Los héroes **RUN'N'GUN** tienen 2 grupos de spritesets: el normal y el **AREA-X** (opcional). en juegos run'n'gun puedes utilizar una celda **área** para las fases, este segundo grupo de spritesets es por si quieres que tu héroe interactue con estas celdas, por ejemplo, si las áreas representan agua puedes hacer que el héroe sea diferente cuando esté dentro o fuera.

- **IDLE:** (parado) spriteset que se muestra cuando el actor no se mueve. Si no se define se toma la última imagen de **MOVE**
- **MOVE:** (moviendo) spriteset que se muestra cuando el actor se mueve. Si no se define se toma la imagen de **IDLE**
- **DEATH:** (muerte) spriteset que se muestra cuando el héroe muere, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de que la fase reinicie, **SOUND EFFECT** efecto de sonido cuando muere y/o **SWAP** (intercambio) del héroe a otro, puede ser un héroe **1-WAY**.
- **BUTTON-A:** spriteset que se muestra al disparar [D], sólo se produce si **GUN 1** está definido, tiene los mismos atributos que **DEATH** sin la opción **SWAP**. Si se usa, el héroe permanece inmóvil el tiempo del spriteset.
- **BUTTON-B:** spriteset que se muestra al disparar bombas [S], sólo se produce si **GUN 3** está definido, tiene los mismos atributos que **DEATH**. Si se usa, el héroe permanece inmóvil el tiempo del spriteset.

HERO propiedades:

- **LAYER:** capa de dibujo (1-6, capas altas se dibujan sobre capas bajas).

- **WALLCOLS:** establece la colisión con muros.
- **HIT BOX:** Establece el área de colisión con actores **GUN:** puede ser **FULL SPRITE**(el sprite completo) o **BOUNDING BOX**(área definida en el spriteset).
- **SPEED:** (velocidad) de movimiento en pixels, puede tener decimales.
- **VITALITY:** (vitalidad) cuando alcanza 0 el héroe muere. Un valor de 0 da al héroe vida infinita. Se puede definir un efecto **SHADE** (tono) al recibir un impacto.
- **AREA-X:** establece la interacción con las áreas, por ejemplo, se puede fijar no se pueda entrar o salir de ellas.

HERO Button-A:

- **DIR LOCK:** (bloqueo de dirección) mantiene la dirección de disparo mientras se mueve.
- **GUN 1-2:** Arma primaria, tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de poder volver a disparar y un parámetro **AUTOFIRE**, que marca cuantos disparos seguidos puede realizar por cada pulsación. el spriteset **BUTTON-A** sólo se mostrará si **GUN 1** está definido.

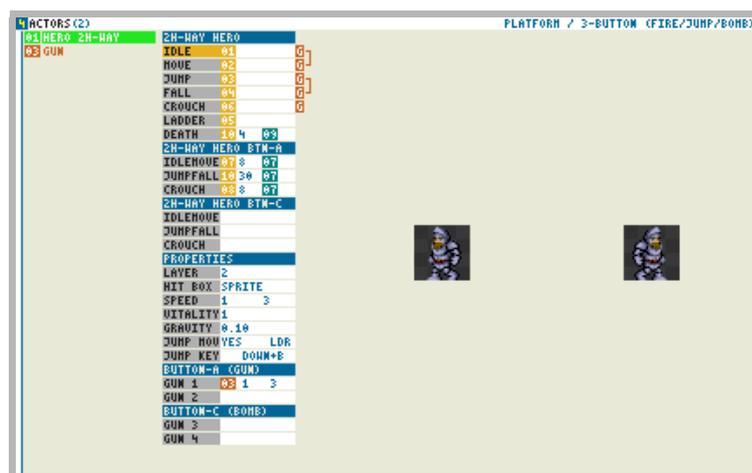
HERO Button-B:

- **GUN 3-4:** Arma secundaria, tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de poder volver a disparar y un parámetro **AUTOFIRE**, que marca cuantos disparos seguidos puede realizar por cada pulsación. el spriteset **BUTTON-B** sólo se mostrará si **GUN 3** está definido. Decrementa el valor bombas si la propiedad **BOMB** de la ventana proyecto esta habilitada.

Cada arma tiene sus propias posiciones en el actor. Para fijar una posición primero selecciona un **GUN** y luego selecciona el **SPRITESET** que quieres definir, sólomente se pueden definir spriteset con la etiqueta **G**. Desplázate sobre un sprite a la derecha, una cruz (**X**) indica que no hay posición fijada en el sprite, haz click en cualquier parte del sprite para fijar la posición de donde saldrá la bala, mantén la tecla [shift] para centrar la posición en el sprite, o pulsa el botón derecho para desactivar la posición. El héroe sólo disparará si el sprite tiene la posición activa.

Platform(plataformas) (3-botones)

Puedes crear un héroe **2H-WAY** y **10-WAY**, son basicamente iguales, la mayor diferencia es que mientras un héroe **2H-WAY** sólo dispara a derecha e izquierda, un héroe **10-WAY** lo puede hacer en 8 direcciones.



adelante o hacia atrás en función.

- **DEATH:** (muerte) spriteset que se muestra cuando el héroe muere, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset antes de que la fase reinicie, **SOUND EFFECT** efecto de sonido cuando muere y/o **SWAP** (intercambio) del héroe a otro, puede ser un héroe **1-WAY**.

Nota importante: Los bounding box de los Spritesets de **IDLE/MOVE/JUMP/FALL/LADDER** deben tener la misma posición y tamaño para no experimentar comportamientos erróneos al mover el Actor.

2H-WAY HERO BUTTON-A spritesets:

- **IDLEMOVE:** (parado/moviendo) spriteset que se muestra al disparar [D], sólo se produce si **GUN 1** está definido, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset y **SOUND EFFECT** efecto de sonido. Si se usa, el héroe permanece inmóvil el tiempo del spriteset.
- **JUMPFALL:** (salto/caer) spriteset que se muestra al disparar [D] mientras salta o cae, sólo se produce si **GUN 1** está definido, tiene otros parámetros: **TIME**(tiempo) en fotogramas que se muestra el spriteset y **SOUND EFFECT** efecto de sonido.
- **CROUCH:** (agachado) spriteset que se muestra al disparar [D] mientras está agachado, las mismas propiedades que **JUMPFALL**.

2H-WAY HERO BUTTON-C spritesets:

Igual que **BUTTON-A** pero para bombas.

HERO propiedades:

- **LAYER:** capa de dibujo (1-6, capas altas se dibujan sobre capas bajas).
- **HIT BOX:** Establece el área de colisión con actores **GUN:** puede ser **FULL SPRITE**(el sprite completo) o **BOUNDING BOX**(área definida en el spriteset).
- **SPEED:** (velocidad) de movimiento en pixels, puede tener decimales. También establece la velocidad de salto.
- **VITALITY:** (vitalidad) cuando alcanza 0 el héroe muere. Un valor de 0 da al héroe vida infinita. Se puede definir un efecto **SHADE** (tono) al recibir un impacto.
- **GRAVITY:** (gravedad) establece la fuerza de gravedad, válido si el valor de velocidad de salto está definido.
- **JUMP MOV:** establece algunos tributos de salto: si es controlable en el salto, saltos consecutivos o si el héroe puede saltar desde escaleras.
- **JUMP KEY:** establece si el salto se realiza pulsando la tecla **arriba** en vez del botón-B [S], o si pulsando **abajo** hace caer al héroe de una celda plataforma.

HERO Button-A:

- **GUN 1-2:** Arma primaria, tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de poder volver a disparar y un parámetro **AUTOFIRE**, que marca cuantos disparos seguidos puede realizar por cada pulsación. el spriteset **BUTTON-A** sólo se mostrará si **GUN 1** está definido.

HERO Button-C:

- **GUN 3-4:** Arma secundaria, tiene un actor **GUN**, un parámetro **FIRE RATE**(ratio de disparo) que indica en fotogramas (*2) el tiempo antes de poder volver a disparar y un parámetro **AUTOFIRE**, que marca cuantos disparos seguidos puede realizar por cada pulsación. el spriteset **BUTTON-B** sólo se mostrará si **GUN 3** está definido. Decrementa el valor bombas si la propiedad **BOMB** de la ventana proyecto está habilitada.

Cada arma tiene sus propias posiciones en el actor. Para fijar una posición primero selecciona un **GUN** y luego selecciona el **SPRITESET** que quieres definir, sólo se pueden definir spriteset con la etiqueta **G**. Desplázate sobre un sprite a la derecha, una cruz (**X**) indica que no hay posición fijada en el sprite, haz click en cualquier parte del sprite para fijar la posición de donde saldrá la bala, mantén la tecla [shift] para centrar la posición en el sprite, o pulsa el botón derecho para desactivar la posición. El héroe sólo disparará si el sprite tiene la posición activa.

Veamos ahora las diferencias frente a un actor **10-WAY**.



10-WAY HERO spritesets:

Los spriteset son iguales que los de 2H-WAY HERO pero todos deben ser 10-WAY spritesets, excepto los de CROUCH (agacharse), que son 2H-WAY, también se añade un set de disparo para escaleras.

BUTTON-A spritesets usa las mismas posiciones de disparo para IDLE, MOVE, JUMP y FALL.

BUTTON-C spritesets usa las mismas posiciones de disparo para IDLE, MOVE, JUMP y FALL.

HERO (AUTOMATED) (héroe automático)

Un HERO (AUTOMATED) es un 'sprited' hero, esto quiere decir que actúa como un actor SPRITE por lo que le puedes dar un comportamiento al héroe, pero a la vez lo puedes controlar, este tipo de héroe no puede 'matarse', en esencia es un hero con total inmunidad.



¿Pero cuándo lo usamos?, se usa para crear acciones automáticas, por ejemplo, el giro en 1942: el héroe 3H-WAY se intercambia a un héroe automático en el spriteset **BUTTON-B** por lo que cabía al pulsar el botón secundario, el héroe automático realiza el giro y vuelve al héroe con el icono **SWAP** del comportamiento.

También podemos usarlo para hacer una transición de un héroe a otro, por ejemplo, si queremos que nuestro héroe se suba a un vehículo (ITEM) podemos intercambiar el héroe actual a uno automático (en las propiedades del actor ITEM) y que realice una animación montándose en él, luego lo cambiamos por un héroe con forma de vehículo con el icono **SWAP** del comportamiento.

Otro ejemplo es cuando la fase termina y queremos que nuestro héroe realice un movimiento o animación, para esto ponemos el héroe automático en la propiedad **HERO AUT** de la fase.



Desarrollador Independiente
de Videojuegos

[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)



english version 

Actores especiales: Spawn y Check ayuda

Spawn

A veces se quiere crear un grupo de actores que estén relacionados para un propósito común, por ejemplo, si queremos crear un item de mejora (power-up) sólo cuando hayamos destruido un grupo de enemigos. Los actores **SPAWN** son exactamente para eso.



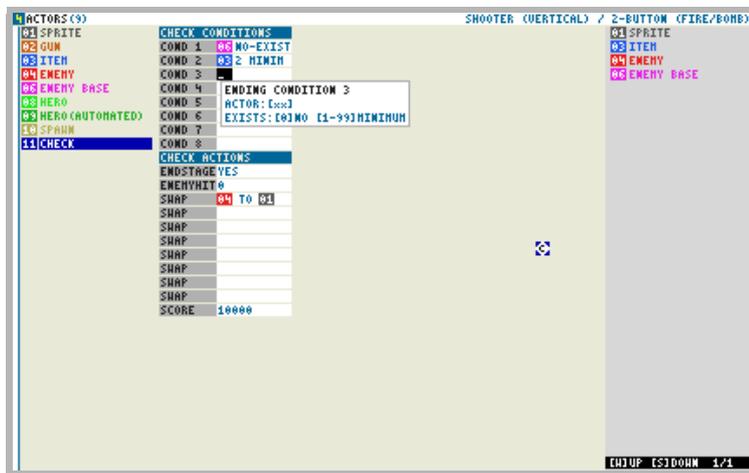
Los actores **SPAWN** no tiene spritesets porque son invisibles al jugador, tienen las siguientes propiedades:

- **LAYER**: (capa) del actor, y un estado **float** (flotante).
- **SCORE**: (puntos) puntos que se consiguen cuando se completa(destruye) el spawn.
- **SWAP**: (intercambio) a otro actor cuando se completa(destruye) el spawn, for ejemplo, un actor item de mejora.

Los actores que el actor **SPAWN** (enjambre) crea para formar parte del grupo se crean en la pantalla de comportamientos, usa los iconos create actor para crearlos, sólo los actores **ENEMY** se contabilizan para el grupo, cada paso debe tener como mínimo un icono **TIMER** y icono **CREATE ACTOR**, una vez alcanzado el último paso el spawn point pasará de verde a rojo así que estará completado y esperará a que los enemigos creados sean destruidos.

Check

A veces necesariamente necesitas comprobar el estado del juego, por ejemplo, necesitas comprobar que una fase ha terminado porque el jefe final (boss) ha sido derrotado, o abrir una puerta porque se han recogido todas las llaves, el actor **CHECK** es para esto.



CHECK conditions (condiciones):

- **COND 1-8:** condicion a comprobar, comprueba si un actor no existe o si existe un mínimo número de ellos en pantalla, por ejemplo, luchando contra 3 jefes finales, intercámbialos (swap) a otro actor cuando mueran (a un actor estático **SPRITE** por ejemplo), ahora sólo hay que comprobar si existen 3 de esos actores **SPRITE** para confirmar que la fase ha terminado.

CHECK actions (acciones):

- **ENDSTAGE:** lanza el evento de final de fase (propiedades **ENDING** de la fase).
- **ENEMYHIT:** todos los enemigos son dañados an la cantidad indicada.
- **SWAP:** (intercambio) actores a otros actores, si no se establece un actor destino el actor se elimina. Puedes usar esto para cambiar tu héroe a otro más poderoso, abrir puertas (eliminando la puerta).
- **SCORE:** (puntos) puntos que se consiguen cuando se cumple.

Una vez cumplido el actor **CHECK** se auto-elimina.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



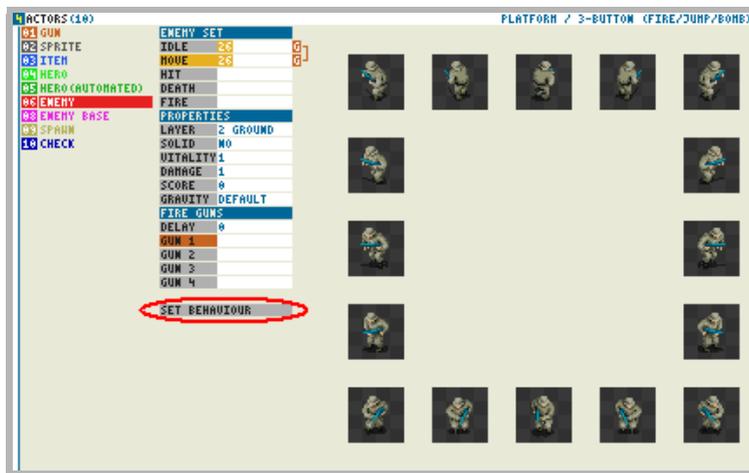
Desarrollador Independiente
de Videojuegos

[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

[english version](#)

Pantalla behaviour (comportamiento) ayuda

A todos los actores se les puede definir un comportamiento excepto a los héroes, para entrar en la pantalla behaviour (comportamiento) pulsa en la opción **SET BEHAVIOUR** al final de la lista de propiedades como indica la siguiente imagen.



Aparecerá una pantalla con un panel de iconos y 16 columnas, cada columna representa un paso(step) del comportamiento, así que un actor puede tener hasta 16 pasos.



Cuando un actor se activa en el juego pasa a leer la primera columna del comportamiento, si no hay ningún icono, se detiene el paso. Para usar un icono de comportamiento sólo tienes que **arrastrar** el icono que quieras usar a una columna, hay un espacio designado para cada icono en la columna, e iconos de la misma clase ocupan el mismo lugar por lo que se sustituirán unos a otros aunque haya sido arrastrado a otra posición en la columna.

Algunos iconos tienen parámetros, para verlos selecciona el icono en la columna y aparecerán debajo del panel.

Puedes copiar un icono de una columna a otra, manten pulsado la tecla [SHIFT] y arrastra el icono de la columna que quieras copiar a otra.

Esta es la lista de iconos y sus comportamientos:



Establece un contador para el paso, una vez cumplido pasa al siguiente paso, tiene 2 parámetros:

- **FRAMES:** (fotogramas) es el tiempo en fotogramas, los juegos van a 59 fotogramas por segundo.
- **REPEAT:** (repetir) define las veces que quieres repetir el paso.



Intercambia(swaps) el actor a otro actor.



Salta automáticamente a otro paso, si se usa no ejecuta ningún otro comportamiento en la columna.

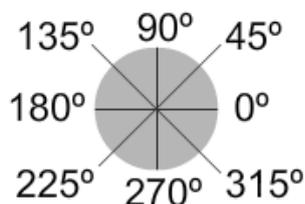


Ejecuta el evento -kill-(muerte) del actor, sólo para los actores **ENEMY** y **GUN**.



Elimina el actor.

Comportamientos de dirección, casi todo utilizan ángulos para definir la dirección, así que aquí tienes un recordatorio:



Establece el ángulo de dirección para el actor, tiene 2 parámetros:

- **ANGLE:** (ángulo) define el ángulo deseado, puede tener decimales.
- **SPEED:** (velocidad) define la velocidad angular del actor para alcanzar el ángulo definido, el efecto es que el actor gira lentamente hacia su destino, también puede ser instantáneamente.



Suma un ángulo a la dirección del actor, tiene 2 parámetros:

- **ANGLE:** ángulo a incrementar/decrementar.
- **SPEED:** igual que el del anterior icono.



Establece una dirección al azar, tiene 2 parámetros:

- **ANGLE:** ángulo (multiplo de), si por ejemplo lo fijamos en 5, el resultado será sólo ángulos aleatorios multiples de 5, esto es: 0 - 5 - 10 - 15...
- **SPEED:** igual que el del anterior icono.



Establece movimiento **HOMING** (buscador de blancos) al actor, todo actor que tenga este icono activo se desplazará en busca del héroe, a menos que sea creado por el propio héroe, en este caso buscará al enemigo más cercano. Tiene sólo el parámetro **SPEED**, igual que el del anterior icono.



Establece la dirección por posición, tiene 3 parámetros:

- **X HOR:** posición horizontal, puede ser **DEFAULT** posición actual, **POINT** (apuntar) or **HERO** (apuntar al héroe horizontalmente). Posición relativa o global en el caso de **POINT** y si el actor se desplaza a la posición o sólo apunta a ella.
- **Y VER:** igual que **X HOR** pero para valores verticales.
- **SPEED:** (velocidad) define la velocidad angular del actor para alcanzar el ángulo definido, el efecto es que el actor gira lentamente hacia su destino, también puede ser instantáneamente.



Establece un movimiento giratorio al actor, tiene 2 parámetros:

- **TURNS:** (giros) establece los giros a realizar, valores positivos lo harán girar hacia la izquierda, valores negativos hacia la derecha, los decimales se pueden usar para porciones de giro, un valor por ejemplo de 2.5 realizará dos giros completos y medio.
- **SPEED:** velocidad angular para el giro, mientras mayor sea más rápido girará.



Mantiene la posición del actor en la del héroe, perfecto para balas usadas como escudos, armas de cuerpo a cuerpo... Los actores con este icono actuarán como elementos del héroe.



Establece un movimiento 4-Way (4-direcciones), sólo para juegos **MAZE**, tiene 2 parámetros:

- **TO HERO:** (al héroe) establece si quieres que el actor "busque" al héroe mientras mueve.
- **EVERY:** fotogramas(frames) antes de cambiar la dirección del actor.

Todos los anteriores pertenecen a la misma clase: comportamientos de dirección.



Establece un movimiento parabólico al actor, usado en todos los géneros excepto en el de plataformas, tiene 3 parámetros:

- **HEIGHT:** (altura) en pixels, es la altura que alcanza en el punto medio.
- **INITIALH:** (altura inicial) en pixels, si quieres que el actor empiece a cierta altura.
- **BOUNCES:** (botes) número de botes a realizar, cada bote será menor que el anterior.



Úsalo con:  para fijar el tiempo en el que quieres que realice el movimiento, el valor por

defecto es 100 fotogramas.

Úsalo también con:  para fijar el destino, por ejemplo si quieres hacer un disparo parabólico a la posición del héroe.



Realiza un salto, sólo para juegos de plataformas, tiene 2 parámetros:

- **SPEED:** (velocidad) velocidad vertical en pixels, se decrementará cada fotograma por el valor del parámetro **GRAVITY** del actor.
- **BOUNCES:** (botes) número de botes a realizar, cada bote será menor que el anterior, y si no esta establecido a **INFINITE**(infinito), pasará al siguiente paso una vez finalizado el salto.



Establece la velocidad del actor, tiene 2 parámetros:

- **SPD IN:** velocidad inicial.



- **SPD OUT:** velocidad final, se esta habilitado usar con  para establecer el tiempo en el que realizar la aceleración/desaceleración.



Establece la velocidad del actor a 0.



Establece un movimiento oscilatorio (swing) al actor, tiene 2 parámetros.

- **HOR MOVE:** establece el movimiento horizontal **RADIO** en pixels y una velocidad angular **SPEED**.
- **VER MOVE:** establece el movimiento vertical **RADIO** en pixels y una velocidad angular **SPEED**.



Igual que el anterior pero para un movimiento circular.



Establece interacción con el héroe.

- **DISTANCE:** (distancia) comprueba la distancia al héroe, cambia de paso si el actor esta mas cerca que el valor introducido, también puede hacer una comprobación horizontal o vertical.
- **LOOK AT:** (mirar) el actor mira al héroe, si quieres que el actor esté mirando al héroe mientras se desplaza.



Establece colisiones con muros.



Establece colisiones con muros, si se produce la colisión el actor muere, es para actores **ENEMY** y **GUN**, si no lo elimina.



Establece bote con muros.



Establece bote con los márgenes(edges) de la pantalla.



Establece margen continuo, si el actor sale por un lado de la pantalla aparece por el lado contrario.

Un actor se puede usar como creador(spawn) de otros actores, por ejemplo un actor bala se convierte en multiples balas direccionales, o crear sprites aleatorios para un efecto de explosión. Hay los siguientes iconos:



Crea un actor en la posición del actor actual, tiene 1 parámetro:

- **ACTOR:** actor a crear.



Crea actores alineados al actor actual, tiene 4 parámetros:

- **ACTOR:** actor a crear.
- **ANGLE+:** los actores son creados perpendicularmente al actual, puedes sumar o restar un ángulo con este parámetro.
- **RADIUS:** radio desde el centro de la posición del actor actual.
- **INSTANCE:** (instancias) número de instancias del actor a crear.
- **RANDOM:** si activado creara las instancias en posiciones aleatorias a lo largo de la línea.



Crea actores en forma de cuña al actor actual, tiene 4 parámetros:

- **ACTOR:** actor a crear.
- **ANGLE:** la cuña en grados.
- **RADIUS:** radio desde el centro de la posición del actor actual.
- **INSTANCE:** (instancias) número de instancias del actor a crear.
- **RANDOM:** si activado creara las instancias en posiciones aleatorias a lo largo del arco.



Crea actores en una posición aleatoria alrededor del actor actual, tiene 3 parámetros:

- **ACTOR:** actor a crear.
- **RADIUS:** radio del círculo virtual donde se crearán los actores.
- **INSTANCE:** (instancias) número de instancias del actor a crear.

Todos los anteriores pertenecen a la misma clase: iconos de creación, así que sólo puede haber uno por paso, ten cuidado con posibles bucles recursivos como: A crea a B y luego B crea a A.



Establece un efecto de parpadeo al actor, tiene 1 parámetro:

- **VISIBILITY:** (visibilidad) velocidad de parpadeo, se puede establecer a invisible.



Reproduce un sonido, tiene 1 parámetro:

- **SOUND:** efecto de sonido.



Establece una probabilidad aleatoria de salto, tiene 2 parámetros:

- **CHANCE:** Establece el valor de probabilidad, valores bajos incrementan la probabilidad se salto de paso.
- **STEP:** paso al que saltar.



Sólo para actores **ENEMY**, Establece la propiedad track(trayectoria) para el paso, si el actor colisiona con una celda track cambiará su dirección a la dirección marcada por la celda.



Sólo para actores **ENEMY**, Establece la propiedad "no disparar" para el paso.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



Desarrollador Independiente
de Videojuegos

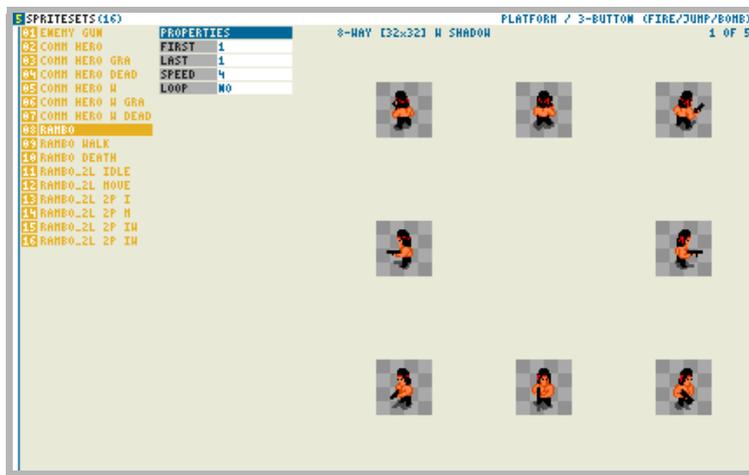
[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)



[english version](#) 

Pantalla Spritesets ayuda

Entra en la pantalla **SPRITESETS** pulsando [5], aquí es donde cargas tus **SPRITE SHEETS** (plantillas de imagenes) para los actores de tu juego.



Los ficheros sprite sheets deben cumplir unos requisitos para poder cargar correctamente, lee [aquí](#) para saber más.

Una vez hayas cargado un sprite sheet podrás ver el número de subimágenes que contiene para cada dirección en la parte de arriba a la derecha de la pantalla. Un sprite sheet puede contener varias animaciones diferentes: andando, saltando, muriendo, etc..., por lo que tendrás que definir un **SPRITESET** para cada animación que quieras que tu actor represente, esto es fácil, sólo tienes que editar las siguientes propiedades:

FIRST: primera imagen de la animación.

LAST: última imagen de la animación.

SPEED: número de fotogramas (pasos) para pasar a la siguiente imagen de la animación. Un segundo son 59 fotogramas.

LOOP: (bucle) si la animación se repite, si la animación tiene multiples imágenes también puedes definir cuantas quieres que se repitan, las 2 últimas, las 3 últimas, etc.

Marca un área sobre cualquier sprite para fijar el **bounding box**, esto es la zona que se utiliza para la colisión con las zonas sólidas, en los juegos de plataforma la parte de abajo del área se utiliza para comprobar la colisión con el suelo.

Para probar el spriteset pulsa **espacio** o haz doble click sobre el registro. Pulsa **Esc** para parar la animación.

También puedes sustituir (replace), duplicar (duplicate) o eliminar (delete) un spriteset, y exportar o importar todos tus spritesets en un único fichero.

Es recomendable dividir las plantillas que sean muy grandes en ficheros, cada uno para una animación para optimizar la velocidad de carga del juego.

Pulsa [F1] en el programa para conocer otras funcionalidades.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



Desarrollador Independiente
de Videojuegos

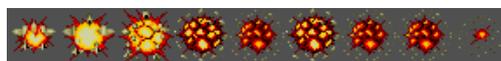
[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

 [english version](#) 

Sprite sheets (plantillas de imagenes) [ayuda](#)

Sprite sheets son los ficheros de imagen que utilizas para los actores (actors) de tu juego, estos son los héroes, enemigos, balas, objetos o simplemente sprites que aparecen en el juego. Puedes cargar diferentes tipos de sprite sheets, estos tienen que estar en formato **PNG** format el nombre del fichero debe tener una nomenclatura específica para cargar correctamente, los tipos están definidos a continuación:

1-WAY sprite sheet (1-posición)



Utiliza un único sprite (animado o sin animar) para todos los movimientos, usualmente se usa para imágenes estáticas, muertes, explosiones, balas, etc.

Formato del fichero: *nombrefichero_1w*numero_de_subimagenes.png

Tamaño máximo para cada subimagen: 192*192 pixels.

2-WAY horizontal sprite sheet (2-posiciones horizontales)



El sprite sheet tiene 2 filas: la primera es para las acciones moviendo hacia la **derecha** y la segunda hacia la **izquierda**, se usa principalmente para juegos de plataformas.

Formato del fichero: *nombrefichero_2wh*numero_de_subimagenes.png

Tamaño máximo para cada subimagen: 112*112 pixels.

2-WAY vertical sprite sheet (2-posiciones verticales)



El sprite sheet tiene 2 filas: la primera es para las acciones moviendo hacia **arriba** y la segunda hacia **abajo**, a lo mejor para un enemigo específico.

Formato del fichero: *nombrefichero_2wv*numero_de_subimagenes.png

Tamaño máximo para cada subimagen: 112*112 pixels.

3-WAY horizontal sprite sheet (3-posiciones horizontales)



El sprite sheet tiene 3 filas: la primera es para las acciones moviendo hacia **arriba**, **abajo** y **parado**, la

segunda hacia la **derecha** y la tercera hacia la **izquierda**, se usa generalmente para los shooters verticales.

Formato del fichero: [nombrefichero_3whnumero_de_subimagenes.png](#)

Tamaño máximo para cada subimagen: 72*72 pixels.

3-WAY vertical sprite sheet (3-posiciones verticales)



El sprite sheet tiene 3 filas: la primera es para las acciones moviendo hacia la **derecha**, **izquierda** y **parado**, la segunda hacia **arriba** y la tercera hacia **abajo**, se usa generalmente para los shooters horizontales.

Formato del fichero: [nombrefichero_3wvnumero_de_subimagenes.png](#)

Tamaño máximo para cada subimagen: 72*72 pixels.

4-WAY sprite sheet (4-posiciones)

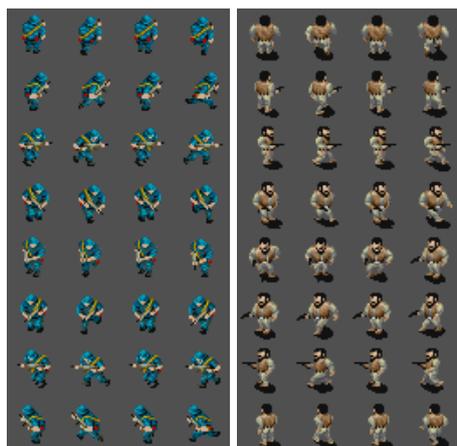


El sprite sheet tiene 4 filas: la primera es para las acciones moviendo hacia **arriba**, la segunda hacia la **derecha**, la tercera hacia **abajo** y la cuarta hacia la **izquierda**, se usa generalmente para los juegos de laberinto (maze games).

Formato del fichero: [nombrefichero_4wnumero_de_subimagenes.png](#)

Tamaño máximo para cada subimagen: 72*72 pixels.

8-WAY sprite sheet (8-posiciones)

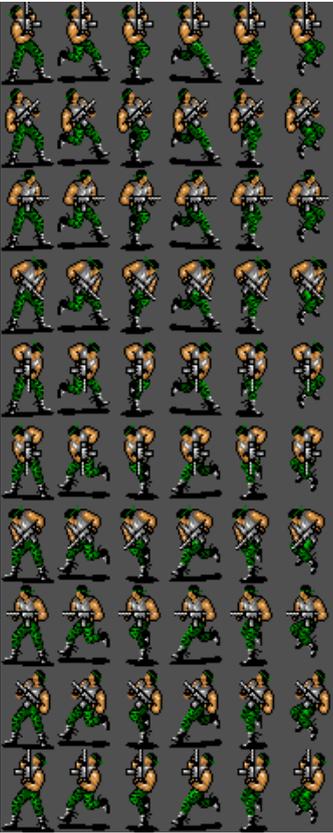


El sprite sheet tiene 8 filas que deben ser dibujadas siguiendo las agujas del reloj, esto es 1ª fila arriba, 2ª fila diagonal arriba-derecha, 3ª fila derecha, etc..., usualmente usado para juegos run'n'gun y shooters.

Formato del fichero: [nombrefichero_8wnumero_de_subimagenes.png](#)

Tamaño máximo para cada subimagen: 72*72 pixels.

10-WAY sprite sheet (10-posiciones)

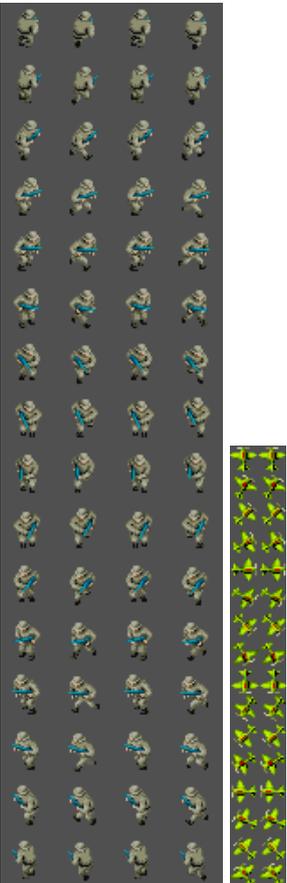


El sprite sheet tiene 10 filas, las 5 primeras son cuando mira hacia la derecha de arriba a abajo, y las 5 siguientes cuando mira hacia la izquierda de abajo a arriba, principalmente usado para plataformas.

Formato del fichero: [nombrefichero_10wnumero_de_subimagenes.png](#)

Tamaño máximo para cada subimagen: 56*56 pixels.

16-WAY sprite sheet (16-posiciones)



El sprite sheet tiene 16 filas que deben ser dibujadas siguiendo las agujas del reloj, como se muestra en las imágenes de ejemplo, usualmente usado para juegos run'n'gun y shooters.

Formato del fichero: *nombrefichero_16wnumero_de_subimagenes.png*

Tamaño máximo para cada subimagen: 40*40 pixels.

Shadow sprite sheet (sombras)



Si quieres que tu actor use sombras puedes crear un fichero de sombras, es fácil, haz una copia del sprite sheet y renómbalo colocando una 's' al final del fichero. **ARGS** lo cargará junto al principal como un todo.

Formato del fichero: *nombrefichero_formato numero_de_subimagenesS.png*

Notas importantes:

- Cada sprite puede tener hasta 99 imágenes por fila.
- El color transparente RGB es [Red=80, Green=80, Blue=80] o el hexadecimal **#505050**, como los usados en los ejemplos.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



Desarrollador Independiente
de Videojuegos

[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

 [english version](#) 

Pixel art ayuda

Dibujar sprites originales y animaciones es un trabajo duro, generalmente tienen que ser dibujados uno por uno, pixel por pixel, pero te sientes recompensado cuando ves a tu personaje moviéndose, saltando y disparando.

Hay buenos programas para realizar esta tarea, algunos sólo para dibujar sprites, otros también para animarlos, aquí tienes una lista de los más populares.

Cosmigo Pro Motion, uno de los mejores para diseñar y animar sprites.

Gimp, otro bueno, pero necesitarás configurarlo para usarlo para pixel art.

Adobe Photoshop, nada que decir que no sepas de él, un gran programa de diseño gráfico.

Graphics Gale, otro bueno.

Jasc Animation Shop, sin web oficial, lo encontrarás en *Google*.

Iconomaker, muy facil de usar.

Aseprite (Allegro Sprite Editor)

Pyxel Edit

MS Paint, simple, pero muy util para copiar+pegar sprites en una plantilla.

Otros programas de dibujo:

RealWorld Paint, **Spriter**, **Grafx2**, **mtPaint** y **D-Pixel**. (enlaces cortesía de Carlos)

Recursos

Porque dibujar sprites no es para todo el mundo hay muchos sitios de recursos con los que puedes crearte tu plantilla, sólo recuerda que este material suele tener copyright asi que tenlo en cuenta a la hora de publicar tu juego.

Aquí tienes una lista de webs con recursos:

OpenGameArt

The Sriters Resource

Sprite Database

NES SNES Sprites

The ShyGuy Kingdom

ScrollBoss

Si conoces algún programa para pixel art o alguna web de recursos que merezca estar en esta lista házmelo saber.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



Desarrollador Independiente
de Videojuegos

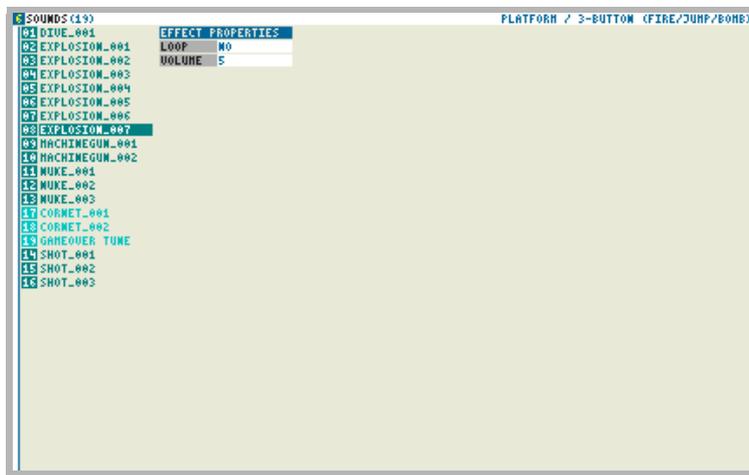
[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)



[english version](#)

Pantalla Sounds (sonidos) ayuda

Entra en la pantalla **SOUNDS** pulsando [6], aquí es donde se cargan los efectos de sonido y las melodías para tu juego.



SOUND EFFECTS (efectos de sonido) son los disparos, lasers, explosiones y toda clase de sonidos del juego, pueden sonar varios a la vez y tienen 2 propiedades **LOOP** (bucle) y **VOLUME** (volumen).

TUNES (melodías) la música de fondo del juego, también las melodías de 'Ready!' y 'Game Over', suena una cada vez y tienen la propiedad **VOLUME** (volumen). Los registros se muestran con un color celeste.

Para probar un sonido pulsa **espacio** o haz doble click en el registro. Pulsa **Esc** para detenerlo.

También puedes sustituir (replace) o eliminar (delete) un sonido, y exportar o importar todos tus sonidos en un único fichero.

Pulsa [F1] en el programa para conocer otras funcionalidades.

Herramientas y recursos

Crear tus propios sonidos puede ser incluso más difícil que crear tus sprite sheets, afortunadamente hay herramientas online que pueden facilitarte esta tarea enormemente, como www.bfxr.net, esta es una increíble herramienta online que te permite crear sonidos en segundos, y exporta directamente a **WAV**!

Los sonidos deben estar en formato **WAV** para funcionar correctamente en **ARGS**, así que si tienes efectos o melodías en mp3 por ejemplo, deberás convertirlas antes a **WAV**, hay docenas de convertidores gratuitos en internet.

La leyenda dice que los ficheros en formato **WAV** tienen siempre un tamaño gigante,... pero eso no es cierto, y lo bueno es que además puedes reducir su tamaño sin usar software de terceros, [aquí](#) tienes cómo.

Un editor de sonidos gratuito y muy bueno es [Audacity](#)

¿Y si hubiera un buscador como *Google* sólo para sonidos?, bueno, lo tienes: www.soungle.com

De todas formas, aquí tienes una recopilación de efectos para que no empieces con las manos vacías:

Aquí tienes algunos sitios de recursos donde puedes encontrar sonidos gratuitos y libres de licencia:

[OpenGameArt](#)

[freesound](#)

[SoundBible.com](#)

[Partners In Rhyme](#)

[Sound FX Center.com](#)

[Flash Kit](#)

[GRSites](#)[YoYo Games sounds](#)[YoYo Games music](#)[IndieGame Music](#)[Free Music Archive](#)[The Recordist](#)[mobilefish](#)

Si conoces alguna web de recursos que merezca estar en esta lista házmelo saber.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.



Desarrollador Independiente
de Videojuegos

[Inicio](#) [Ayuda y Tutoriales](#) [Descargar](#) [Juegos](#) [Volver a principal](#)

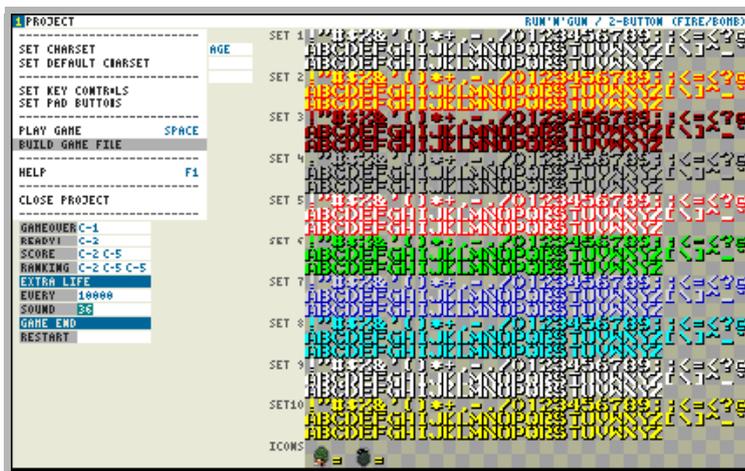


[english version](#)

Compilando y distribuyendo tus juegos ayuda

Una vez hayas completado y probado tu juego lo normal es que quieras distribuirlo y/o venderlo.

Para esto necesitarás haber adquirido la versión completa (full) de Arcade Game Studio para poder tener la opción build-game-file activa.



Cuando compilas tu juego se creará un único fichero con todos los datos del juego, este fichero tendrá la extensión **.args** y se guardará en la carpeta "build" de tu proyecto.

```
ARCADE GAME STUDIO 1.0
MAXIMUS ACTION CARNAGE 1.00 12/5/2012
BY BRUNO B MARCOS
RUM'N'GUM / 2-BUTTON (FIRE/BOMB)

MAC_ARGS
BUILD COMPLETED

>> PRESS SPACE TO CONTINUE <<
```

Como el fichero **.args** no es un ejecutable necesitarás el lanzador [cabinet.exe](#), así que copia tu juego en la carpeta que contiene cabinet.exe y luego lanza cabinet.exe para jugarlo.

Ten en cuenta que cabinet.exe buscará el primer fichero .args en la carpeta así que asegurate de que no hay otros.

Por lo tanto esto quiere decir que cuando distribuyas tu juego necesitarás distribuir 2 ficheros: tu juego .args con cabinet.exe para lanzarlo.

Puedes renombrar cabinet.exe al nombre de tu juego si quieres, lee Cabinet README.txt para saber las características avanzadas de cabinet.exe.

Copyright © 2003-2013 bruneras.com. Todos los derechos reservados.