

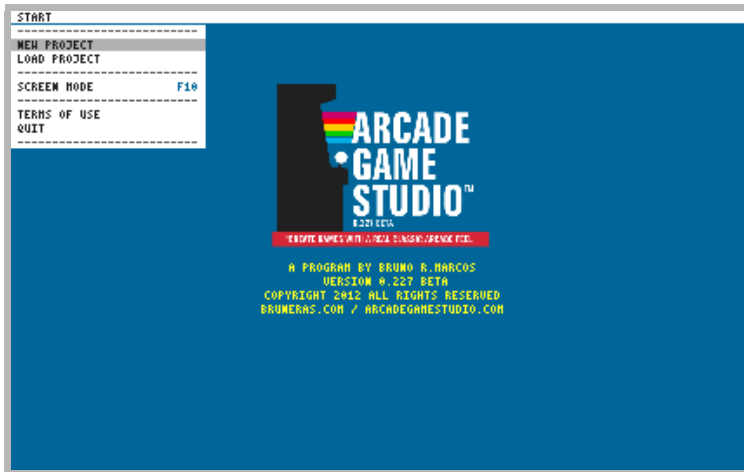
Independent  
Video Game Developer
[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)


version española



## Start screen and game genres help topic

The first screen you see when you run **Arcade Game Studio** is the **START** screen.

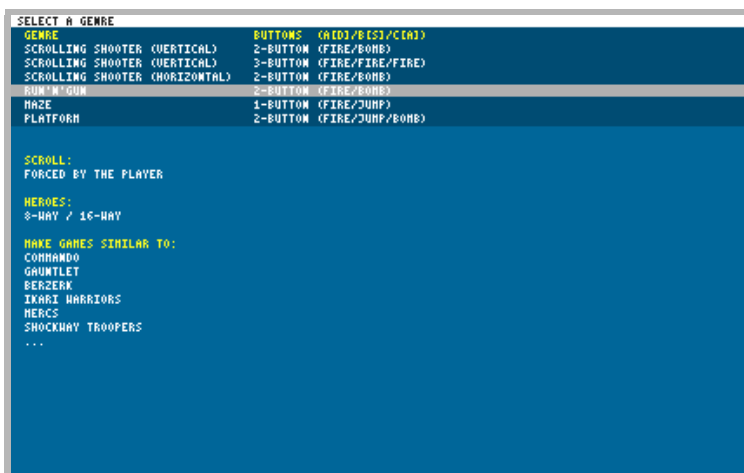


Press [F10] to change screen mode, as all games are 256\*256 pixels, **ARGS** workspace must be pixel perfect accordingly to this resolution so you can get a black zone on top and bottom of the screen. Press [F11] to switch between windowed or fullscreen. Press [F9] to save a screenshot.

Some screen modes will look the same, this is due to that in some cases on the **STAGES** screen you could experience some annoying lines on the backcells of the stage, if this happens try different screen modes to select a valid one.

Select **NEW PROJECT** to start your first game, a file wizard will show up to save your project file (can be different from your game name), I recommend to create a new folder for your project because a each project will create the following folders: actors, sounds, sprites, stages and tiles besides general files in the main directory.

Once you have saved your project you will be asked to select a game genre / button set.



**Arcade Game Studio** actually supports several genres, each designed with a specific mechanics, but with enough freedom to create completely different and original games.

Let's take a closer look to this genres, also take note that this is a my particular view of arcade genres:

## Scrolling Shooter

---

Genre where the player moves upward along the game stage (vertical Shooter) or rightward (horizontal Shooter), the screen movement (scroll) is automatic, and the objective usually is to get to the enemy boss at the end of the stage.

Select this genre to make games similar to *1942*, *Xevious*, *Flying Shark*, *Raiden*, *Gyrodine*, *Gunsmoke* (vertical), or *Scramble*, *R-Type*, *Side Arms*, *Gradius* (horizontal).

If scroll is disabled then we can get to make a classic Shoot'em-Up style game similar to *Space Invaders*, *Galaxian* or *Galaga*.

Also with some workarround tricks you could get to make games similar to *Arkanoid* or *Road Fighter*.

## Run'n'Gun

---

Genre where the player moves freely arround the stage, stage scroll is forced by the player, you can fix the stage movement vertically or horizontally so player can only move upward along the stage (*Commando*) or rightward (*Jail Break*), in this case the objective usually is to get to the enemy boss at the end of the stage.

Select this genre to make games similar to *Gauntlet*, *Commando*, *Jail Break*, *Ikary Warriors*, *Mercs* or *Shockway Troopers*.

Is also perfect to make classic screen by screen games similar to *Berzerk* or *Robotron*.

## Maze

---

Genre where the player has a 4-Way movement, stages are screen fitted and usually is a 'trap the hero' game.

Select this genre to make games similar to *Pac-Man*, *Eyes* or *Bomberman* for example.

## Platform

---

Genre where the player is right-left faced, moves freely arround the stage and a gravity effect interacts with the actors, stage scroll is forced by the player.

Player has other movements aside from moving left or right, like crouching, jumping or climbing ladders.

Select this genre to make games similar to *Ghosts 'n' Goblins*, *Contra*, *Black Tiger*, *Toki*, *Sunset Riders* or why not *Metal Slug*.

With this genre and good knowledge of the program you could even get to make fighting games similar to *Kung Fu Master*, *Shinobi* or *Dragon Ninja*.

## Other genres

---

Although the above genres cover a great range of the 80's arcade games, there are other genres that may be added with time, but I won't make any promises about it, this include 3rd Person Shooter games like *Cabal*, Beat'em-Ups games like *Final Fight* or Fighting games like *Yie Ar Kung-Fu*.

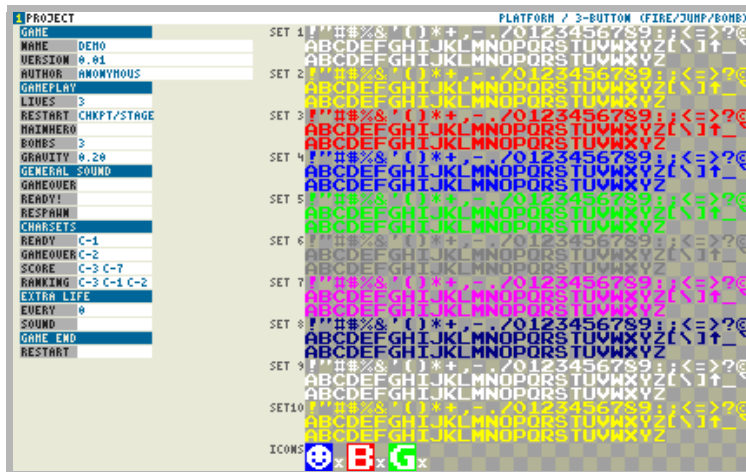
Independent  
Video Game Developer
[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)


version española



## Project screen help topic

Enter the **PROJECT** screen by pressing [1], is the first screen you will see when you start or load a project, here is a screenshot of what it looks like.



Here is where you define the game general settings, you have a list of properties, let's see them one by one:

**NAME:** is your game name, max. length is 25.

**VERSION:** is the game version, usually a x.xx code, but you can insert any text, max. length is 25.

**AUTHOR:** your name or company name, max. length is 25.

## Gameplay settings

**LIVES:** game lives, can be set to infinite.

**RESTART:** sets the restart mode after death, this can be set to **CHECKPOINT/STAGE**, so you restart at the last checkpoint reached or the entire stage if no checkpoint is set, or **RESPAWN** if you want your hero to continue from where you just died.

**MAIN HERO:** is your general hero actor for the game, this is used depending on a stage property value, see stage screen help topic to know how.

**BOMBS:** bombs are the secondary weapon, this includes any secondary event like shields, spins, or whatever, you can disable it if you want to use secondary gun as other primary weapon or to not use it.

**GRAVITY:** used only on **PLATFORM** games, defines a general gravity value for every actor that has **GRAVITY** property set to **DEFAULT**.

## General sound settings

**GAME OVER:** tune that plays on "Game Over" message, shown for 5 seconds.

**READY!:** tune that plays on "Ready!" message, shown for 2 seconds, this message is not shown if **RESTART**

property is set to **RESPAWN**.

**RESPAWN**: sound effect that plays on player respawn event.

**TIME UP!**: tune that plays on "Time Up!" message, shown for 3 seconds, this message is not shown if **COUNTDOWN** property is set on stage.

## Charsets settings

---

Go [here](#) to read more about charsets.

**GAME OVER**: charset used for "Game Over" text.

**READY!**: charset used for "Ready!" text.

**SCORE**: charset used for the player score and hi-score labels and values, first define the one for the labels.

**RANKING**: charset used for the ranking table, in this order: rank, name and score.

## Extra Life

---

**EVERY**: score interval to win an extra life.

**SOUND**: sound effect when you get an extra life.

## Game End

---

**RESTART**: game stage to jump once the last stage is played, if not set then game ends.

Copyright © 2003-2013 bruneras.com. All rights reserved.



Independent  
Video Game Developer

[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)



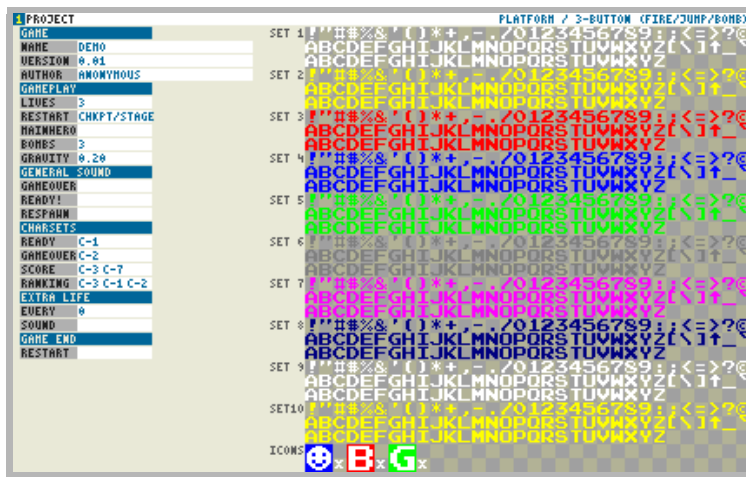
[version española](#)

## Charsets help topic

---

Charsets are the text fonts you use in your game, it's like the Arial, Verdana or Tahoma fonts for PC.

Even if you don't use any text messages in your game **ARGS** will use it's default Charset for the Score, the Best Ranking table, or the 'Ready!' and 'Game Over' messages.



You can load your custom charset any time, just go to the **PROJECT[1]** screen, select **SET CHARSET** menu option and load your custom Charset image file, or select **SET DEFAULT CHARSET** to return to the default charset.

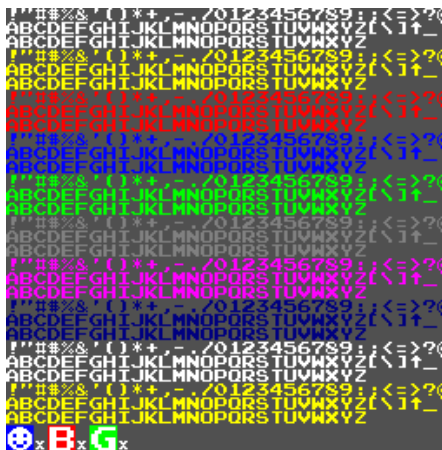
A valid charset is a 256\*256 **PNG** file, divided in two sections:

- Top section that represents 10 rows of ASCII charsets, where each character is a 8\*8 pixel icon. The 9th and 10th charsets are used for blinking text.
- Bottom section with three 24\*16 pixel icons in a row, the first icon represents the **LIVES**, the second icon represents the secondary weapon (if enabled), usually used for bombs, shields (*Phoenix*) or spins (*1942*). The third is not yet used.

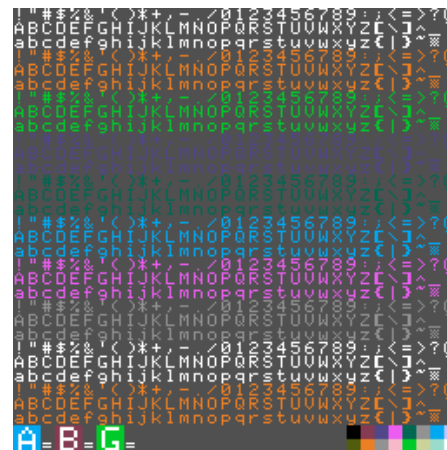
Charset map:

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

Here you have some predefined charsets, just right-click on any of them and then select "save image as" to save it.

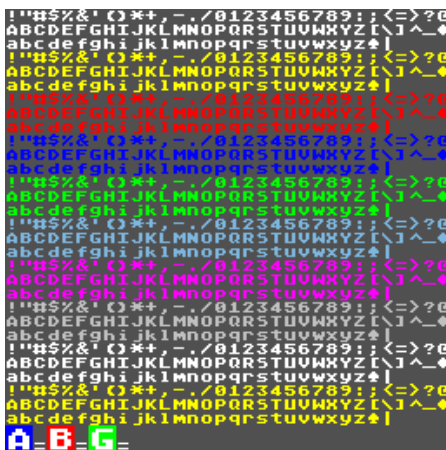


Default Arcade

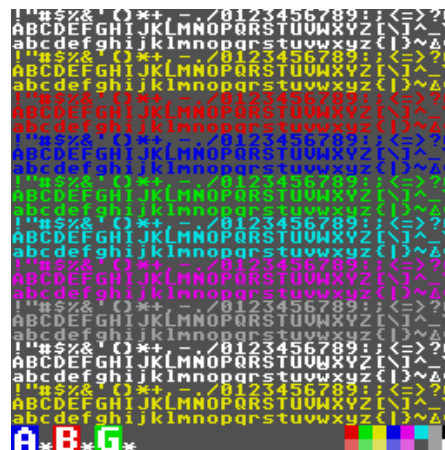


Apple II

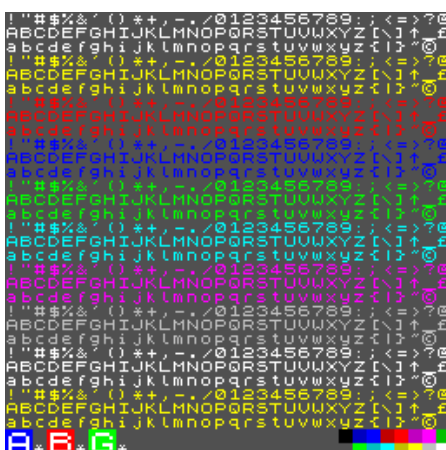




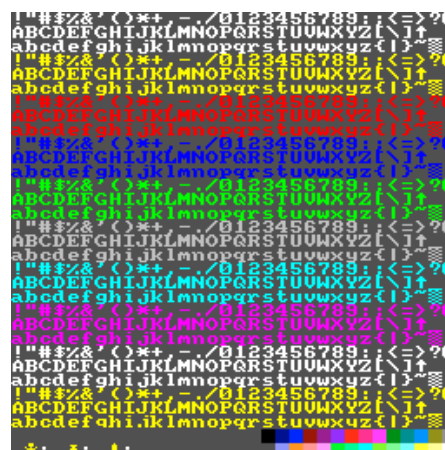
Atari 400/800



Atari ST



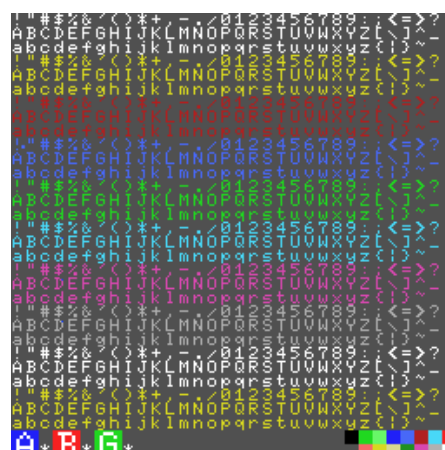
ZX Spectrum



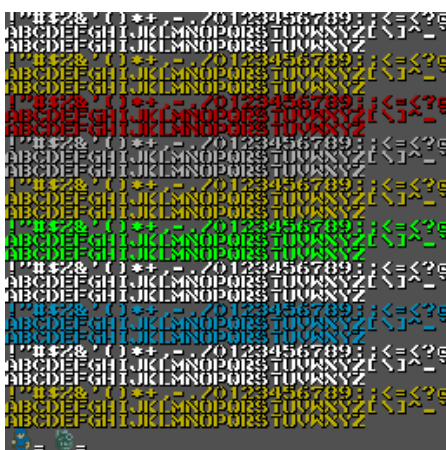
Amstrad CPC



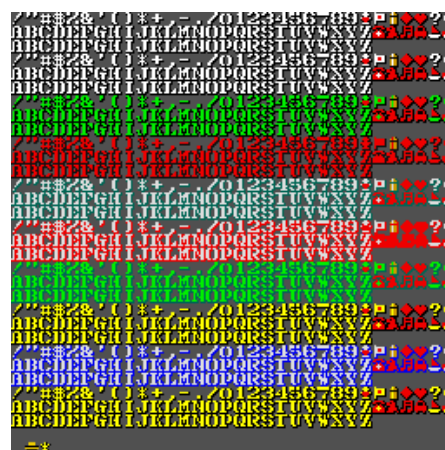
Commodore 64



MSX



Commando (Capcom 1985)



Gun.Smoke (Capcom 1985)

Copyright © 2003-2013 bruneras.com. All rights reserved.

Independent  
Video Game Developer
[Home](#)   [Help & Tutorials](#)   [Download](#)   [Games](#)   [Back to Main](#)


version española

## Stages screen help topic

Enter the **STAGES** screen by pressing [2], here is where you create the stages for your game.



First of all, stages do not mean "play", stages represent the diferent acts in your game, like stages with no "shooting" but usually used to tell a story, like the introduccion in *Ghosts'n'Goblins*.

There are 5 kind of stages: **NOTICE**, **TITLE**, **RANKING**, **INTERMISSION** and **GAME**.

When you start a new project you automatically get 3 stages: **NOTICE**, **TITLE** and **RANKING**, you only have one of each and cannot be moved or deleted, let's see what are they stand for.

**NOTICE:** this stage is for your presentation, logo or any typical warning message for the game, it only appears once at game start-up, you can disable it if you like, will see its properties latter.

**TITLE:** this is where you should place the game title and where players go before they start a new game (by pressing any fire button), it can swap with **RANKING** stage if you want.

**RANKING:** this is where the best ranking table appears, it can be disabled but then you won't get name score rank input. An empty ranking stage will show up as the image below, it can be initialized and it will generate a **.hi** file with the game top 8 players. You can customize the ranking table charsets (rank, names and scores) in the **PROJECT** screen.



You can customize this stages in any way except that they are one screen (no background scrolling) and that you cannot set a hero actor in it.

There are other two stages, this are the **INTERMISSION** stages and the **GAME** stages.

**INTERMISSION:** this is where you should set your game story intro or anything you want to tell between game stages, you can have as many as you want.

**GAME:** is where you play the game, depending on the genre you have chosen it will have diferent properties and rules, we will se this later on this topic.

One thing you must know about stages is that they play in a **successive** manner, this means that the position in the list will indicate the playing position in the game (except for the **RANKING** stage that moves back to the **TITLE** stage), so an **INTERMISSION** or **GAME** stage will move to the next in the list once ended, y there are no stages left the game ends. You can move a stage position in the list by using [R] and [F] keys while pressing [SHIFT], remember that **NOTICE**, **TITLE** and **RANKING** cannot be moved from their position.

Press [F1] in the program to see all keyboard shortcuts any time.

## Notice, Title, Ranking and Intermissions stages

Without relying on how you customized this stages, they all have in common their properties.



**TIMER:** sets the time (in seconds) to play this stage before moving to the next in the list, if is set to 0, player will need to press **fire** button to move. A **disabled** value is available for **NOTICE** and **RANKING** stages.

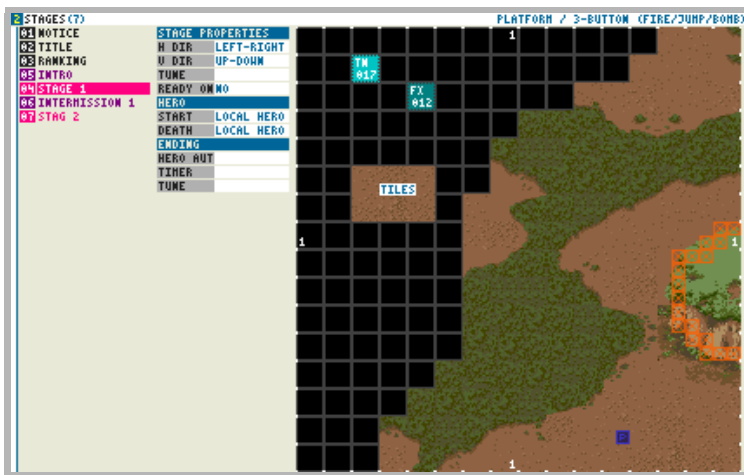
**TEXT:** sets the delay speed for the text, and you can set a sound effect for every time a letter pops out.

**TUNE:** is the playing music for the stage, it will play once, if you want it to loop set a **TUNE** sound element directly onto the stage (we will see later how).



## Game stages

Depending on the genre you have chosen, they will have different properties and rules.



**H DIR:** for **RUN'N'GUN** and **PLATFORM** games, sets the stage horizontal movement, **right** and/or **left** faced.

**V DIR:** for **RUN'N'GUN** and **PLATFORM** games, sets the stage vertical movement, **up** and/or **down** faced.

**SCROLL:** for **SHOOTER** games, you set the scrolling speed of the stage.

**TUNE:** is the playing music for the stage, it will restart when ended, you can set a sound element directly onto the stage (we will see later how).

**HERO** properties:

**START:** when we set a custom hero on a stage this doesn't mean that it will be the hero when playing the game, it depends on this property:

- **LOCAL HERO** uses the current hero set on the stage.
- **MAIN HERO** uses the **MAIN HERO** defined in the **PROJECT** screen
- **PREVIOUS HERO** uses the last hero played in previous stage, it's to keep track of your hero between stages.

**DEATH:** same as above but used after player gets killed on a stage.

**COUNTDOWN:** (time) activates a countdown in seconds, when 0 is reached hero losses a life.

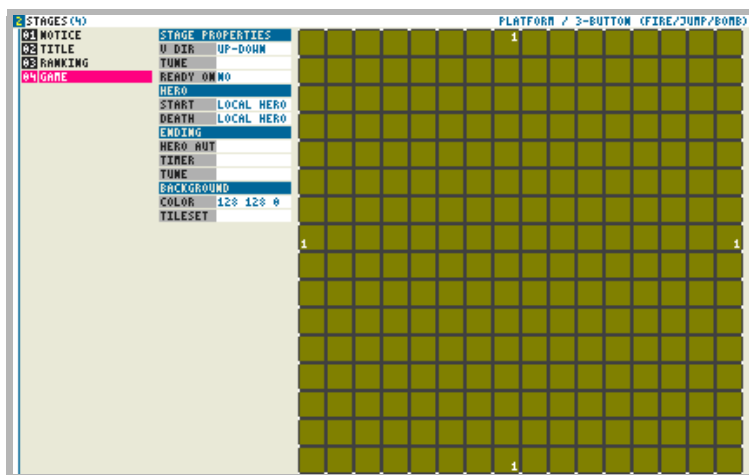
**ENDING** properties:

**HERO AUT:** usually, when you finish a stage you want your hero to perform an automatic movement or animation, sets a **HERO (AUTOMATED)** for this.

**TIMER:** sets the time (in seconds) to wait before the game moves to the next stage, but only after the stage as ended (and that is when a **CHECK** stage actor says so).

**TUNE:** sets the tune that plays after the stage as ended.

Version 1.1 has two new **BACKGROUND** parameters for each stage: **COLOR** and **TILESET**.



**COLOR:** sets the background color of the stage in a RGB format.

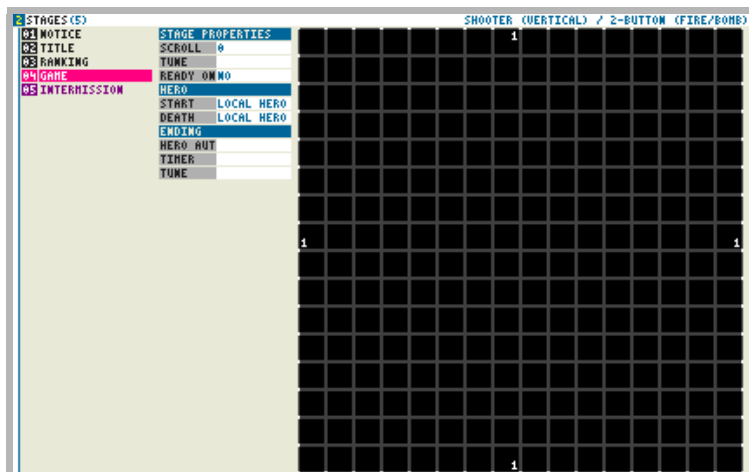
**TILESET:** sets a static tileset as background.

Version 1.2 adds 3 parameters to **TILESET**: **MODE**, **SPEED** and **DIRECTION**

- **FIXED**: a static tile on background
- **BOXED**: the background moves proportionally to the stage size (only visible on **GAME** stages), tile must be larger than 256x256 to see the effect.**GAME**).
- **MOVE**: sets a movement(**SPEED** and **DIRECTION**) to the background tile, stops when edge is reached, usefull for example to show a map on intermissions like Ghosts'n'Goblins does.
- **MOSAIC**: sets a movement(**SPEED** and **DIRECTION**), background will loop, perfect for star fields..

## Stage editor

The editor is the area in the right side of the stages screen, it's divided in 16\*16 cells, each cell is 16\*16 pixels, so this makes a 256\*256 pixel area, or what is the same, the exact screen resolution for every game.



Only **GAME** stages can be larger than a screen size, you can move around the stage with [WASD] keys (keep [SHIFT] down to move faster), when the cursor is over the area it will show a tag indicating the edit mode you are in, it can be one of the following: **TILES**, **WALL/EVENT**, **ACTORS**, **TEXT** and **SOUNDS**. Use [Q] and [E] keys to change the edit mode.

**TILES**: this mode is for placing background tiles. To place a tile go to the **TILES** screen [3], and select an area from a tile, go back [2] and place it anywhere on the stage. Press right-click to deselect the tile, and keep [SHIFT] down to delete a background cell from the stage. A stage is as large as the background that contains, or until an empty background row or column is reached, starting from the bottom-left corner.

**WALL/EVENTS**: this mode is for placing wall cells, ladders or specific collision events, use [Z] and [X] to change the selected event, that can be one of the following:



**WALL** cell, actors with collisions enabled cannot go through this cells.



**PLATFORM** cell, used only for **PLATFORM** games, actors can jump onto them vertically and through them horizontally.



**LADDER** cell, used only for **PLATFORM** games, hero actors can climb a ladder cell.



**AREA** cell, used only for **RUN'N'GUN** games, hero actors can change their state when going through them (like if your hero gets into water), also can be defined as a wall, so for example a *tank* hero can't go through it, but a *man* hero can.



**DEATH** cell, hero automatically dies on collision.



**CHECKPOINT** cell, marks a checkpoint on the stage, if a hero reaches it, game will restart at this point on hero death. **RESTART** property on **PROJECT** screen must be set to **CHECKPOINT/STAGE**.

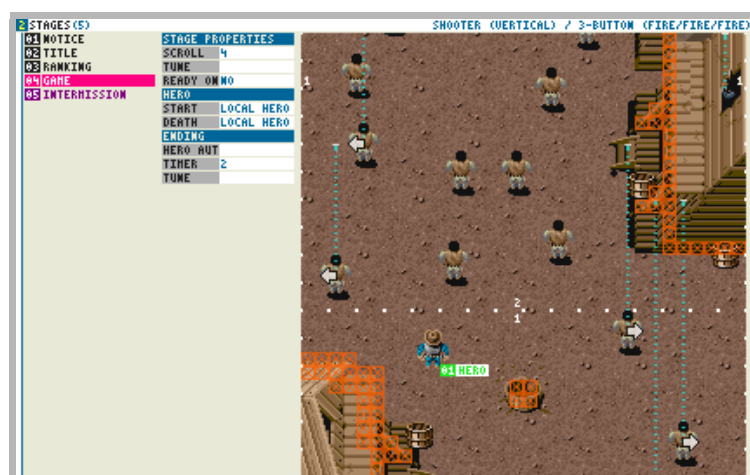


**TRACK** cell, an actor that collides with this cell will change its direction if it has its **TRACK** behaviour icon set to on.



**END** cell, if your hero reaches this cell the stage will perform the ending event and will move to the next stage.

**ACTORS:** this mode is for placing actors on the stage. Use [Z] and [X] to select an actor or press [ ] (Key between ESC and TAB) to select it from a list. Actors will activate when they get at the edge of the sight. A blinking black & white spot will indicate the origin of each actor.



Left-click to place an actor, if you move in any direction while placing an arrow will show up pointing the edge where the actor will start on the screen, also on **SCROLLING** games keep [SHIFT] pressed to set the actor start spot when the scroll reaches that point.

**TEXT:** this mode is for writing texts on the stage. Use [Z] and [X] to select a charset and click on the stage to change to **EDITOR** mode, this is like any text editor, you can also use the cursor keys to move the cursor. Press [ESC] or right-click to exit **EDITOR** mode. You can remove letters with a right-click over them.

**SOUNDS:** this mode is for placing sounds on the stage. Use [Z] and [X] to select a sound or press [ ] (Key between ESC and TAB) to select it from a list. Sound effects or tunes will play and loop when they get into sight, this is useful, for example, if you want to change the music when reaching the end of a stage.

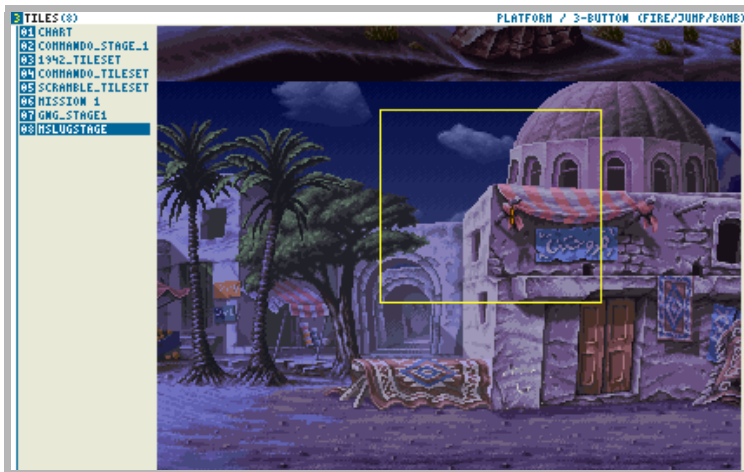
Press [TAB] to show/hide current mode elements.

Press [SPACE] to test the stage, note that the stage will always start centered on the **nearest bottom-left hero** set on the stage. On test mode press [TAB] to hide/show wall and events. In version 1.1 you also will see the bounding box for wall collisions (white) and hit collisions (red).

Independent  
Video Game Developer[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)[version española](#)

## Tiles screen help topic

Enter the **TILES** screen by pressing [3], here is where you load the tiles used for your game backgrounds.



Tile image files must be in **PNG** format and with a maximum size of 2048\*2048 pixels.

Move large tiles with [WASD]. Click any place to start a selected area, you can move the tile while dragging to select a greater area, the first selected cell will represent the gripping point, use this to place it easily on a stage.

The smallest area selectable is a 16\*16 pixels cell.

You can replace or delete a tile, and also import or export all your tiles into one unique file.

Press [F1] to see other features.

Copyright © 2003-2013 bruneras.com. All rights reserved.

Independent  
Video Game Developer[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)[version española](#)

## Actors screen help topic

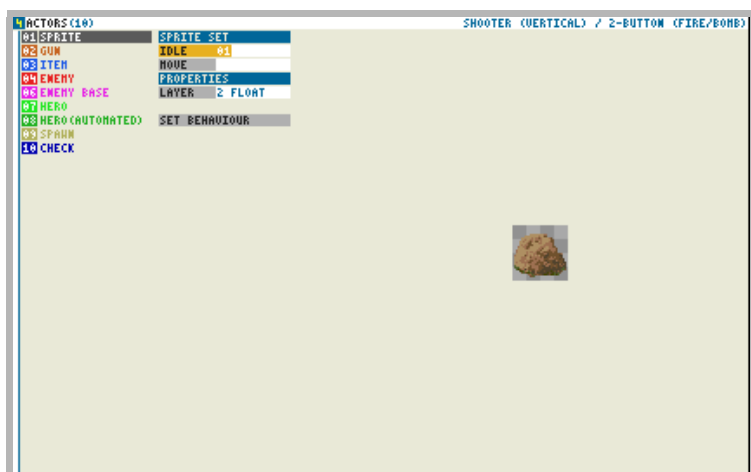
Enter the **ACTORS** screen by pressing [4], here is where you create the actors for your game.



All games have the same actors in a greater or lesser extent, this actors are basically: **HEROES**, **ENEMIES**, **GUNS**, **ITEMS** and **SPRITES**, each with a different finality.

## Sprites

A **SPRITE** is the simplest actor you can use in your game, in essence it does...nothing, as it don't cause damage to your hero or can't be killed.



Use it to create extras to your game like rocks, foreground trees, clouds, smoke, gunfire or anything else, this doesn't mean that you cannot make complex behaviours with it, for example, you can convert a simple sprite in a killing machine using it as a gun spawn point, thanks to the flexibility of the behaviour panel.

**SPRITE** spritesets:

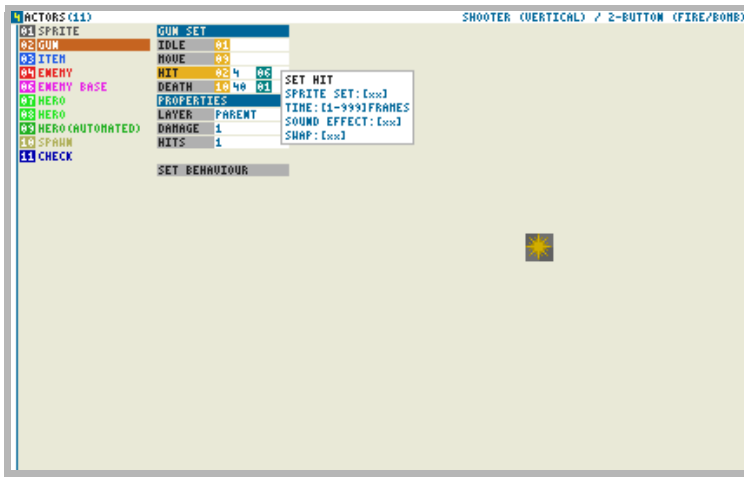
- **IDLE**: spriteset displayed when not moving.
- **MOVE**: spriteset displayed when moving.

**SPRITE** properties:

- **LAYER**: actor drawing layer (1-6 higher layers are drawn over lower layers), and the actor **float** state.
- **GRAVITY**: used only on **PLATFORM** games, sets the actor gravity, a **DEFAULT** value will use the project general gravity value.

## Guns

A **GUN** is an actor used as weapons, it cause hits to hero and enemies, depending who created it, use it to create bullets, lasers, shields or homming missiles...



### GUN spritesets:

- **IDLE**: spriteset displayed when not moving.
- **MOVE**: spriteset displayed when moving.
- **HIT**: spriteset displayed when gun hits an enemy (if its a hero gun) or hero, this set has some extra attributes (see image): **TIME** frames the spriteset is shown before it disappears, **SOUND EFFECT** on hit or **SWAP** gun to other actor when hit.
- **DEATH**: spriteset displayed when gun dies, death is set by the **DEATH** behaviour icon or by the collision icon and gun collides with a wall. Has the same attributes as **HIT**.

### GUN properties:

- **LAYER**: actor drawing layer, and **float** state. Value can be set to **PARENT** if you want it to copy its creator layer and floating values.
- **DAMAGE**: damage value to an enemy or hero, enemy or hero **VITALITY** will be decreased.
- **HITS**: gun consecutive hits, this is how many collision hits a gun can perform before it disappears, for example, a ray gun will have this value set to **INFINITE**.
- **HIT BOX**: sets the collision box with **ENEMY/HERO** actors: **FULL SPRITE** or **BOUNDING BOX**.
- **GRAVITY**: used only on **PLATFORM** games, sets the actor gravity, a **DEFAULT** value will use the project general gravity value.

## Items

An **ITEM** is an actor that heroes can pick up, it usually represent a power-up, weapon, coin or vehicle.



### ITEM spritesets:

- **IDLE**: spriteset displayed when not moving.
- **MOVE**: spriteset displayed when moving.
- **PICK**: spriteset displayed when you pick up the item (collision), this set has some extra attributes: **TIME** frames the spriteset is shown before it disappears, **SOUND EFFECT** or **SWAP** item to other actor when



picked up.

#### ITEM properties:

- **LAYER**: actor drawing layer, and **float** state.
- **ENDSTAGE**: triggers the end of the stage event.
- **GRAVITY**: used only on **PLATFORM** games, sets the actor gravity, a **DEFAULT** value will use the project general gravity value.

#### ITEM power-ups:

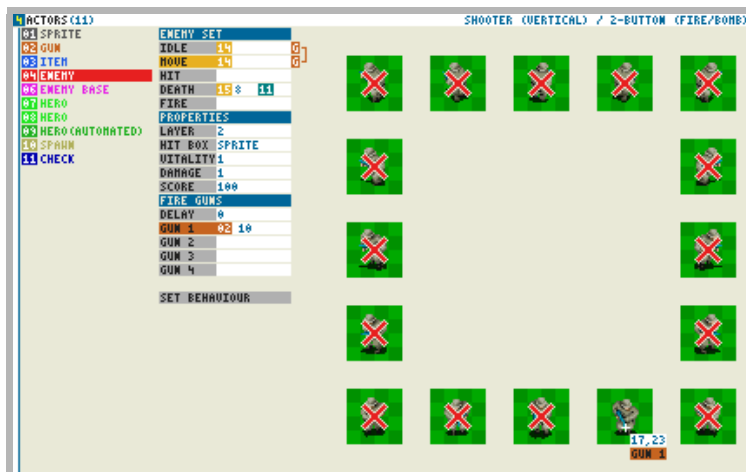
- **LIVES**: increase/decrease lives.
- **BOMBS**: increase/decrease secondary weapon, if enabled.
- **IMMUNITY**: sets immunity to hero for a certain time in frames.
- **ENEMYHIT**: all enemy actors are hit by the indicated amount.
- **SCORE**: increments player score.

#### ITEM swap actors:

Swaps defined actors to another ones, if no destination is set, actors are deleted. You can use this to change your hero to another more powerful, open doors, or to get on a vehicle.

## Enemies

An **ENEMY** is an actor you usually have to elude or destroy.



#### ENEMY spritesets:

- **IDLE**: spriteset displayed when not moving.
- **MOVE**: spriteset displayed when moving.
- **HIT**: spriteset displayed when actor gets hit by a gun, this set has some extra attributes: **TIME** frames the spriteset is shown before returns to its normal state, if skipped (TAB key) it sets spriteset default time, and **SOUND EFFECT** on hit. An enemy actor does not move when its on this state.
- **DEATH**: spriteset displayed when actor gets hit by a gun, this set has some extra attributes: **TIME** frames the spriteset is shown, if skipped (TAB key) it sets spriteset default time, **SOUND EFFECT** on death and **SWAP** if you want to change enemy to other actor when animation is done.
- **FIRE**: spriteset displayed when actor fires a gun, only performed if **GUN 1** is defined, this set has some extra attributes: **TIME** frames the spriteset is shown, if skipped (TAB key) it sets spriteset default time, **SOUND EFFECT** on fire and **SWAP** if you want to change enemy to other actor after when animation is done.

#### ENEMY properties:

- **LAYER**: actor drawing layer, and **float** state.
- **HIT BOX**: sets the collision box with **GUN/HERO** actors: **FULL SPRITE** or **BOUNDING BOX**, also sets a **SOLID** value if you want the bounding box act like a wall, or **PLATFORM** value if you want the actor to act like such (only for platform genre).

- **VITALITY**: when vitality reaches 0 the actor dies, this makes the actor more or less powerful. An initial value of 0 gives the actor an infinite vitality. Also has a hit shade color value.
- **DAMAGE**: damage value inflicted to hero on collision.
- **SCORE**: increments player score.
- **GRAVITY**: used only on **PLATFORM** games, sets the actor gravity, a **DEFAULT** value will use the project general gravity value.

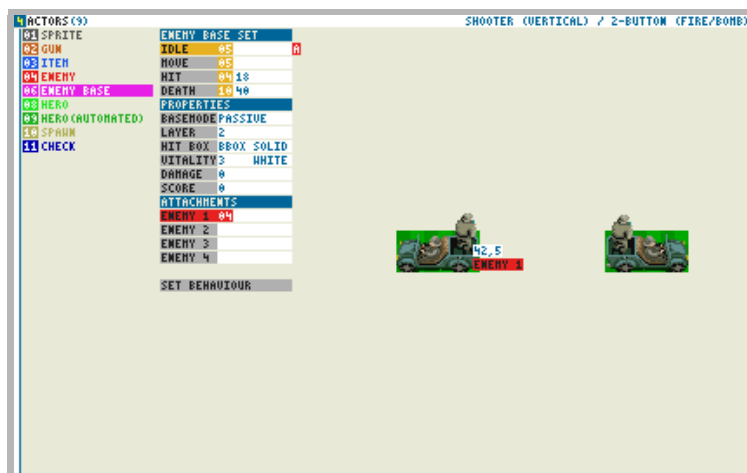
#### ENEMY fire guns:

- **DELAY**: when a fire event triggers this indicates the delay before actor creates a **GUN**, for example, imagine an eating plant that opens its mouth and throws an eye-ball to the hero (*Ghosts 'n' Goblins*), the **GUN** (eye-ball) is thrown at the middle of the open-mouth animation (**FIRE** spriteset), this delay is for fitting the animation with the gun creation.
- **GUN 1-4**: Custom fire guns, has a **GUN** actor and a fire rate, frames(\*2) before it calls the fire event again. **FIRE** spriteset will only trigger if **GUN 1** is set.

Each fire gun has its own firing positions on the actor. To set a firing position first select a **GUN** and then select a **SPRITESET** you want to use, only spritesets labeled with **G** can be used, move over a sprite. Move on to any sprite on the right, a cross (X) indicates that there is no firing position set for the current sprite, click on any part of the sprite to set its fire position, keep [SHIFT] down to center it on the sprite, or right-click to deactivate. Actor will only shoot if a fire position is set on its facing direction.

## Enemies (base)

An **ENEMY BASE** is an enemy actor that you usually have to elude or destroy, the difference is that can have attached other **ENEMY** actors, for example, a tank, the enemy base would be the vehicle and the cannon and/or the machine gun are attached enemies.



#### ENEMY BASE spritesets:

- **IDLE**: spriteset displayed when not moving.
- **MOVE**: spriteset displayed when moving.
- **HIT**: (only for **ACTIVE** enemy base, see **BASEMODE**) spriteset displayed when actor gets hit by a gun, this set has some extra attributes: **TIME** frames the spriteset is shown before returns to its normal state, if skipped (TAB key) it sets spriteset default time, and **SOUND EFFECT** on hit. An enemy actor does not move when its on this state.
- **DEATH**: spriteset displayed when actor gets hit by a gun, this set has some extra attributes: **TIME** frames the spriteset is shown, if skipped (TAB key) it sets spriteset default time, **SOUND EFFECT** on death and **SWAP** if you want to change enemy to other actor when animation is done.

#### ENEMY BASE properties:

- **BASEMODE**: base mode, has two states **ACTIVE**: base and attachments work as a whole, you need to destroy the enemy base, destroying its attachments won't destroy it. **PASSIVE**: base just acts like a stand for its attachments, you need to destroy all attachments to destroy it, so in passive mode base must have at least one attachment to keep existing.
- **LAYER**: actor drawing layer, and **float** state.

- **HIT BOX**: sets the collision box with **GUN** actors: **FULL SPRITE** or **BOUNDING BOX**, also sets a **SOLID** value if you want the bounding box act like a wall.
- **VITALITY**: (only for **ACTIVE** bases) when vitality reaches 0 the enemy dies and all its attachments, this makes an enemy more or less powerful. An initial value of 0 gives the actor an infinite vitality. Also has a hit shade color value.
- **DAMAGE**: damage value inflicted to hero on collision.
- **SCORE**: increments player score.
- **GRAVITY**: used only on **PLATFORM** games, sets the actor gravity, a **DEFAULT** value will use the project general gravity value.

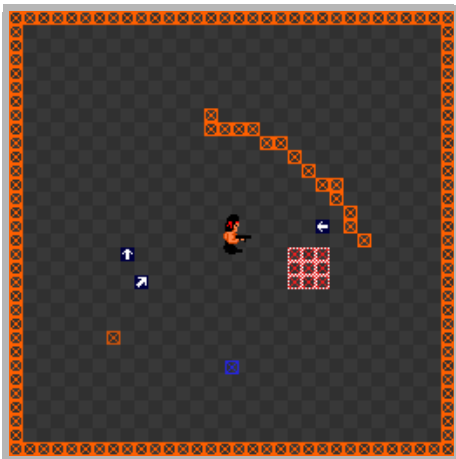
#### ENEMY BASE attachments:

- **ENEMY 1-4**: attached enemy actor, each attachment has its own positions on the base. To set a base position first select a **ENEMY** and then select **IDLE** spriteset (labeled with **A**), move over a sprite, click on any part of the sprite to set the attachment for this base position, keep [SHIFT] down to center it on the sprite.

## Test mode

---

Press [SPACE] to enter the test mode for the current actor, if the actor isn't a hero a blue cell will appear representing it, you can move it with the CURSOR keys.



Place walls and cell events in real-time with the mouse for testing, to switch use [Z,X] keys , press [F1] on **ACTORS** screen to view test mode keys. Also pressing [`] (Key between ESC TAB) you can change background cell colors, and [TAB] to hide/show wall actor collisions bounding box(white) and hit boundingbox(red).

## Actors *float* state

---

Actor **LAYER** property defines the **float** state of the actor, for games where the stage is larger than the screen this sets if the actor moves along with the scroll or not, for example, in scrolling shooter games like *Raiden* only grounded actors (houses, tanks,..) move along with the scroll, look at this Capcom's [1942 gameplay video](#) to see the effect, if you look at the planes you can perceive when they move horizontally that they don't get dragged by the scroll, the effect goes unnoticed but it is essential to make the game gain in gameplay.

If we set the floating state of the actor you can set whatever scrolling speed to the stage, the actor won't get pushed out of the screen, in *1942* we essentially got a huge WWII *Galaga* game with a scrolling background.

Independent  
Video Game Developer
[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)


version española



## Actors: Heroes help topic

Heroes are the actors you can control in your game, ie the player, depending on the chosen genre you can create a particular hero or another, let's see them by genre.

Important: Hero Spritesets bounding boxes should have same position and size or you could experience weird behaviours when moving the actor.

### Vertical Shooter (2-Buttons)

You can create a **1-WAY** and **3H-WAY** heroes, aside of the specific use of x-ways spriteset they are essentially the same, except that **3H-WAY** hero has a **ROLL** property.



#### HERO spritesets:

- **IDLEMOVE**: as a scrolling game, hero is always moving so there should not be a difference between idle and moving spritesets.
- **DEATH**: spriteset displayed when hero dies, this set has extra attributes: **TIME** frames the spriteset is shown before stage resumes with next life, **SOUND EFFECT** on death or **SWAP** to other hero actor, can be a **1-WAY** hero.
- **BUTTON-A**: spriteset displayed when hero shoots [D], only performed if **GUN 1** is set, has the same attributes as **DEATH** without **SWAP** option.
- **BUTTON-B**: spriteset displayed when hero shoots [S] bombs, only performed if **GUN 3** is set, has the same attributes as **DEATH**.

#### HERO properties:

- **LAYER**: actor drawing layer (1-6 higher layers are drawn over lower layers).
- **WALLCOLS**: sets wall collisions.
- **HIT BOX**: sets the collision box with **GUN** actors: full sprite or bounding box.
- **SPEED**: in pixels, can have decimal values.
- **VITALITY**: when vitality reaches 0 the hero dies. An initial value of 0 gives the hero an infinite vitality. Also has a hit shade color value.
- **FIXED**: sets hero movement.
- **ROLL**: only for **3H-WAY** hero, sets if the **IDLEMOVE** spriteset animation is reversed when direction is changed, this creates a roll effect.

**HERO Button-A:**

- **RANGED**: makes a upward ranged fire effect, similar to *Gyrodine*.
- **GUN 1-2**: Primary guns, has a **GUN** actor, a **FIRE RATE** parameter that sets the time in frames(\*2) before you can shoot again, and an **AUTOFIRE** parameter, this sets how many guns are fired on one press. **BUTTON-A** spriteset will only trigger if **GUN 1** is set.

**HERO Button-B:**

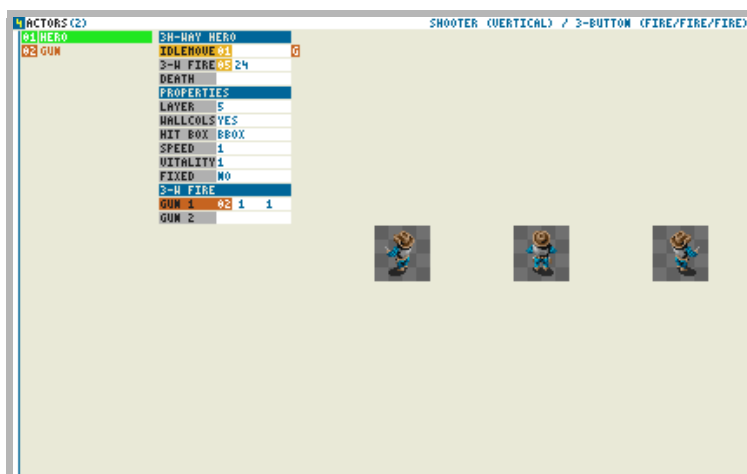
- **GUN 3-4**: Secondary guns, has a **GUN** actor, a **FIRE RATE** parameter that sets the time in frames(\*2) before you can shoot again, and an **AUTOFIRE** parameter, this sets how many guns are fired on one press. **BUTTON-B** spriteset will only trigger if **GUN 3** is set. Decreases bombs if **BOMB** project property is enabled.

Each fire gun has its own firing positions on the actor. To set a firing position first select a **GUN** and then select a **SPRITESSET** you want to use, only spritesets labeled with **G** can be used, move over a sprite, a **X** indicates that there is no firing position set for the current sprite, click on any part of the sprite to set its fire position, keep [SHIFT] down to center it on the sprite, or right-click to deactivate. Hero will only shoot if a fire position is set on its facing direction.

## Vertical Shooter (3-Buttons)

---

You can create a **3H-WAY** heroe with a one button per fire direction as Capcom's *Gunsmoke*.

**HERO spritesets:**

- **IDLEMOVE**: as a scrolling game, hero is always moving so there should not be a difference between idle and moving spritesets.
- **DEATH**: spriteset displayed when hero dies, this set has extra attributes: **TIME** frames the spriteset is shown before stage resumes with next life, **SOUND EFFECT** on death or **SWAP** to other hero actor, can be a **1-WAY** hero.
- **3-W FIRE**: spriteset displayed when hero shoots, fire set buttons are: [A] for **left**, [S] for **up** and [D] for **right**, only performed if **GUN 1** is set, has the same attributes as **DEATH** without **SWAP** option.

**HERO properties:**

- the same as Vertical Shooter (2-Button) with no **ROLL** property.

**HERO 3-w fire:**

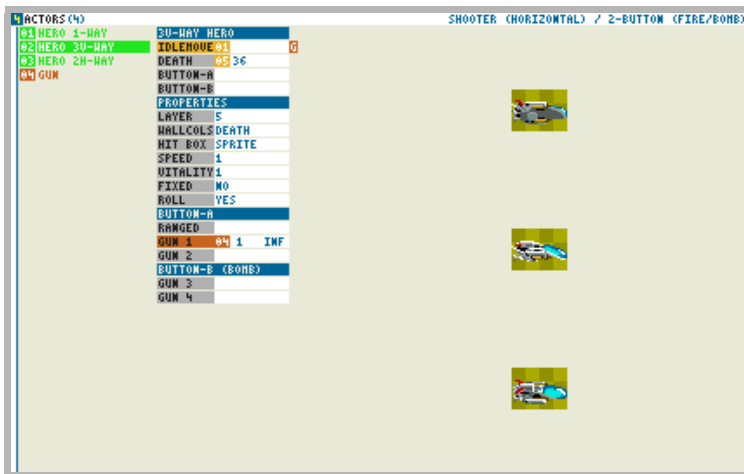
- **GUN 1-2**: has a **GUN** actor, **FIRE RATE** parameter that sets the time in frames(\*2) before you can shoot again, and an **AUTOFIRE** parameter, this sets how many guns are fired on one press.

## Horizontal Shooter (2-Buttons)

---

You can create a **1-WAY**, **3V-WAY** and **2H-WAY** heroes, aside of the specific use of x-ways spriteset they are

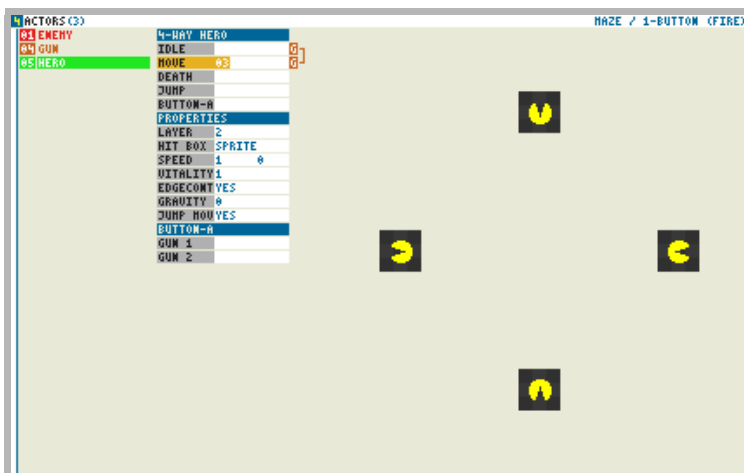
essentially the same, except that **3H-WAY** hero has a **ROLL** property.



**HERO** spritesets, properties and buttons attributes are the horizontal equivalent of Vertical Shooter.

## Maze (2-Buttons)

You can create a **4-WAY** hero.



**HERO** spritesets:

- **IDLE**: spriteset displayed when not moving.
- **MOVE**: spriteset displayed when moving.
- **DEATH**: spriteset displayed when hero dies, this set has extra attributes: **TIME** frames the spriteset is shown before stage resumes with next life, **SOUND EFFECT** on death or **SWAP** to other hero actor, can be a **1-WAY** hero.
- **JUMP**: spriteset displayed when hero jumps [S] (**SPEED** jump must be set).
- **BUTTON-A**: spriteset displayed when hero shoots [D], only performed if **GUN 1** is set, has the same attributes as **DEATH**. If is used, hero stands still while shooting.

**HERO** properties:

- **LAYER**: actor drawing layer (1-6 higher layers are drawn over lower layers).
- **HIT BOX**: sets the collision box with **GUN** actors: full sprite or bounding box.
- **SPEED**: in pixels, can have decimal values, you can **LOCK** the speed so hero always move, and also sets the jump speed.
- **VITALITY**: when vitality reaches 0 the enemy dies. An initial value of 0 gives the hero an infinite vitality. Also has a hit shade color value.
- **EDGECONT**: enables/disables continuous edge, if the hero exits the screen it will appear on the opposite side.
- **GRAVITY**: sets a gravity value, only useful if speed jump value is set.



- **JUMP MOV**: enables/disables hero moveable while jumping.

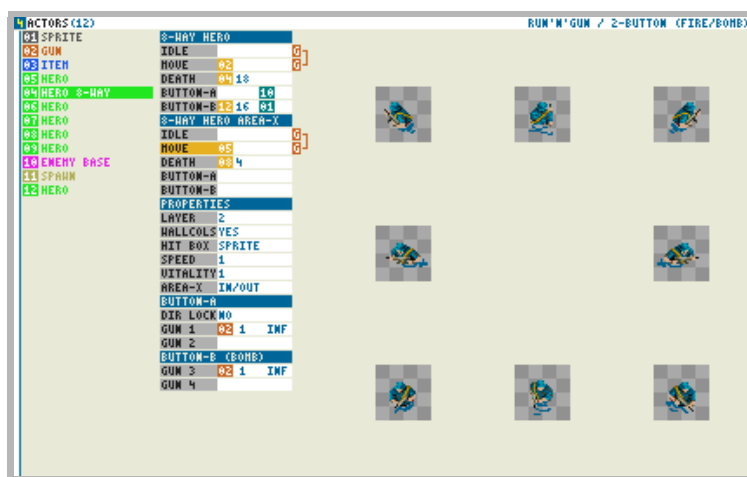
#### HERO Button-A:

- **GUN 1-2**: has a **GUN** actor, **FIRE RATE** parameter that sets the time in frames(\*2) before you can shoot again, and an **AUTOFIRE** parameter, this sets how many guns are fired on one press.

Each fire gun has its own firing positions on the actor. To set a firing position first select a **GUN** and then select a **SPRITESET** you want to use, only spritesets labeled with **G** can be used, move over a sprite, a **X** indicates that there is no firing position set for the current sprite, click on any part of the sprite to set its fire position, keep [SHIFT] down to center it on the sprite, or right-click to deactivate. Hero will only shoot if a fire position is set on its facing direction.

## Run 'N' Gun (2-Buttons)

You can create a **8-WAY** and **16-WAY** heroes, they are essentially the same, except that a **16-WAY** hero has a rotating movement when making a turn of 90 degrees or less, this makes a smoother movement and also adds more fire intervals to the hero.



#### HERO spritesets:

**RUN'N'GUN** heroes have 2 sets of spritesets, a normal one and an **AREA-X** one (optional). in run'n'gun games you can use a special area cell for stages, this second set is for this, if you want your hero to have interaction with area cells, for example if the area represents water, you can make your hero look different if is onto an area.

- **IDLE**: spriteset displayed when not moving, if not set will use last moving image.
- **MOVE**: spriteset displayed when moving, if not set will use the idle set.
- **DEATH**: spriteset displayed when hero dies, this set has extra attributes: **TIME** frames the spriteset is shown before stage resumes with next life, **SOUND EFFECT** on death or **SWAP** to other hero actor, can be a **1-WAY** hero.
- **BUTTON-A**: spriteset displayed when hero shoots [D], only performed if **GUN 1** is set, has the same attributes as **DEATH** without **SWAP** option. If is used, hero stands still while shooting.
- **BUTTON-B**: spriteset displayed when hero shoots [S] bombs, only performed if **GUN 3** is set, has the same attributes as **DEATH**. Decreases bombs if **BOMB** project property is enabled. If is used, hero stands still while shooting.

#### HERO properties:

- **LAYER**: actor drawing layer (1-6 higher layers are drawn over lower layers).
- **WALLCOLS**: sets wall collisions.
- **HIT BOX**: sets the collision box with **GUN** actors: full sprite or bounding box.
- **SPEED**: in pixels, can have decimal values.
- **VITALITY**: when vitality reaches 0 the enemy dies. An initial value of 0 gives the hero an infinite vitality. Also has a hit shade color value.
- **AREA-X**: sets hero interaction with areas, can be set to make hero move only inside or outside of them.

**HERO Button-A:**

- **DIR LOCK:** keeps firing direction while moving.
- **GUN 1-2:** Primary guns, has a **GUN** actor, **FIRE RATE** parameter that sets the time in frames(\*2) before you can shoot again, and an **AUTOFIRE** parameter, this sets how many guns are fired on one press. **BUTTON-A** spriteset will only trigger if **GUN 1** is set.

**HERO Button-B:**

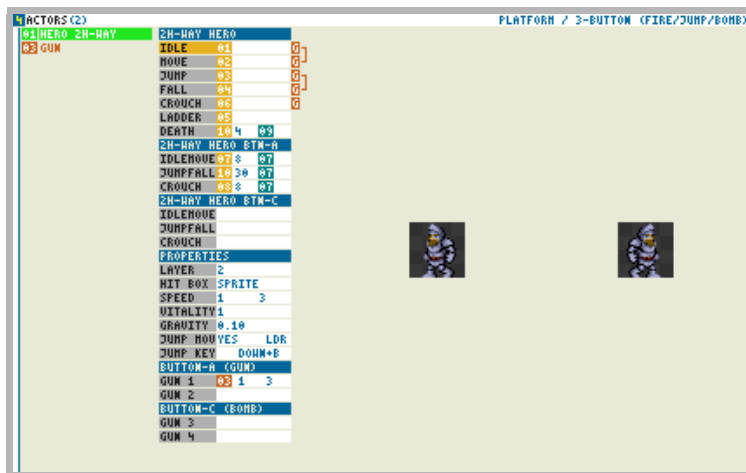
- **GUN 3-4:** Secondary guns, has a **GUN** actor, **FIRE RATE** parameter that sets the time in frames(\*2) before you can shoot again, and an **AUTOFIRE** parameter, this sets how many guns are fired on one press. **BUTTON-B** spriteset will only trigger if **GUN 3** is set.

Each fire gun has its own firing positions on the actor. To set a firing position first select a **GUN** and then select a **SPRITESSET** you want to use, only spritesets labeled with **G** can be used, move over a sprite, a **X** indicates that there is no firing position set for the current sprite, click on any part of the sprite to set its fire position, keep [SHIFT] down to center it on the sprite, or right-click to deactivate. Hero will only shoot if a fire position is set on its facing direction.

## Platform (3-Buttons)

---

You can create a **2H-WAY** and **10-WAY** heroes, they are essentially the same, the main difference is that while **2H-WAY** can only shoot left and right, a **10-WAY** hero can do it in 8 directions.

**2H-WAY HERO spritesets:**

- **IDLE:** spriteset displayed when not moving.
- **MOVE:** spriteset displayed when moving.
- **JUMP:** spriteset displayed when jumping. Can play a **SOUND EFFECT**.
- **FALL:** spriteset displayed when falling. Can play a **SOUND EFFECT**.
- **CROUCH:** spriteset displayed when crouching, if not used hero cannot crouch. Can play a **SOUND EFFECT**.
- **LADDER:** spriteset displayed onto a ladder, spriteset animation will go forward (up) or backward (down).
- **DEATH:** spriteset displayed when hero dies, this set has extra attributes: **TIME** frames the spriteset is shown before stage resumes with next life, **SOUND EFFECT** on death or **SWAP** to other hero actor, can be a **1-WAY** hero.

Important Note: **IDLE/MOVE/JUMP/FALL/LADDER** Spritesets bounding boxes should have same position and size or you could experience weird behaviours when moving the actor.

**2H-WAY HERO BUTTON-A spritesets:**

- **IDLEMOVE:** spriteset displayed when hero shoots [D], only performed if **GUN 1** is set, has **TIME** and **SOUND EFFECT** attributes. If is used, hero stands still while shooting.
- **JUMPFALL:** spriteset displayed when hero shoots while jumping or falling, only performed if **GUN 1** is set, has **TIME** and **SOUND EFFECT** attributes.
- **CROUCH:** spriteset displayed when crouching, same as above.

Same as **BUTTON-A** but for bombs.

- **LAYER**: actor drawing layer (1-6 higher layers are drawn over lower layers).
- **HIT BOX**: sets the collision box with **GUN** actors: full sprite or bounding box.
- **SPEED**: in pixels, can have decimal values, also sets the jump speed.
- **VITALITY**: when vitality reaches 0 the enemy dies. An initial value of 0 gives the hero an infinite vitality. Also has a hit shade color value.
- **GRAVITY**: sets a gravity value, only useful if speed jump value is set.
- **JUMP MOV**: sets some jumping attributes: moving while jumping, consecutive jumps and if hero can jump from ladders.
- **JUMP KEY**: sets if jump is performed with **up** key instead of Button-B [S], or if **down** key makes hero fall from a platform cell.

- **GUN 1-2:** Primary guns, has a **GUN** actor, **FIRE RATE** parameter that sets the time in frames(\*2) before you can shoot again, and an **AUTOFIRE** parameter, this sets how many guns are fired on one press.

- **GUN 3-4:** Secondary guns, has a **GUN** actor, **FIRE RATE** parameter that sets the time in frames(\*2) before you can shoot again, and an **AUTOFIRE** parameter, this sets how many guns are fired on one press.

Each fire gun has its own firing positions on the actor. To set a firing position first select a **GUN** and then select a **SPRITESET** you want to use, only spritesets labeled with **G** can be used, move over a sprite, a **X** indicates that there is no firing position set for the current sprite, click on any part of the sprite to set its fire position, keep [SHIFT] down to center it on the sprite, or right-click to deactivate. Hero will only shoot if a fire position is set on its facing direction.

Let's see now the differences with a **10-WAY** hero actor.



Spritesets are the same as the **2H-WAY HERO** above but where all sets must be a **10-WAY** spriteset, except the **CROUCH** ones, that are **2H-WAY**, it also adds a gun set for laddering positions.

**BUTTON-A** spritesets use the same firing positions for **IDLE**, **MOVE**, **JUMP** and **FALL** states.

**BUTTON-C** spritesets use the same firing positions for **IDLE**, **MOVE**, **JUMP** and **FALL** states.

## HERO (AUTOMATED)

A **HERO (AUTOMATED)** is a 'sprited' hero, this means that hero acts like a **SPRITE** actor, so you can set a behaviour to the hero, but it also can be controlled, this hero actor cannot be killed, essentially is a hero with total immunity.



But when do we use it?, we use it for actions that are automatically made, for example, the spinning in *1942*: the **3H-WAY** hero swaps to a automated hero set on **BUTTON-B** spriteset, the automated hero performs the loop and then we call it back with the **SWAP** behaviour icon).

We also can use it to make a transition from one hero to other, for example, if our hero get's a vehicle (**ITEM**) you can swap the current hero with and automated one that performs the animation and then swaps (swap behaviour icon) to another hero.

Other example is when a stage ends and we want our hero to perform a movement or an animation, set this on the **HERO AUT** property on the **STAGE** screen.

Copyright © 2003-2013 bruneras.com. All rights reserved.



Independent  
Video Game Developer

[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)

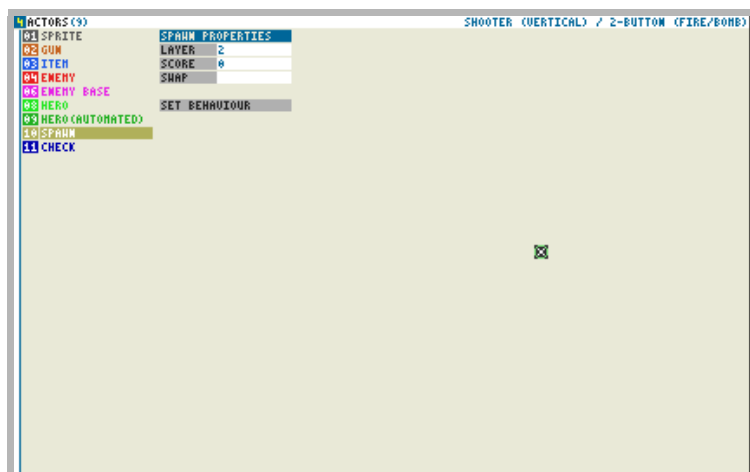


[version española](#) 

## Special Actors: Spawn and Check help topic

### Spawn

Sometimes you want to create a group of actors that are associated for a common purpose, for example, you want to create a power-up item but only by destroying a group of enemies, **SPAWN** actors are for this.



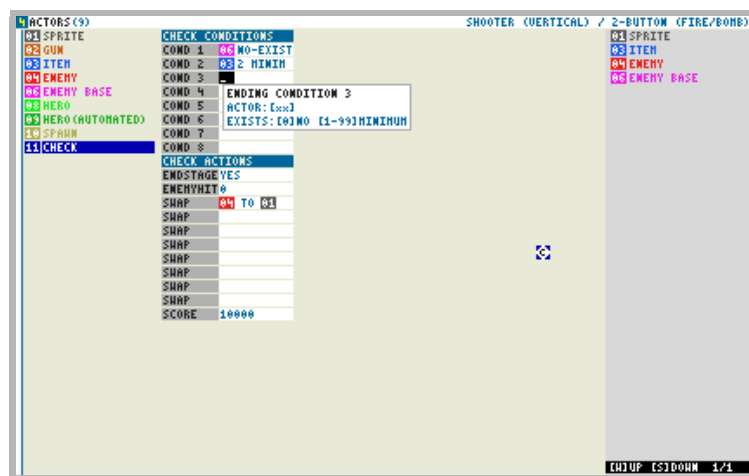
**SPAWN** actors do not have spritesets because they are invisible to the player, they have the following properties:

- **LAYER**: spawn layer, and **float** state.
- **SCORE**: score for a completed (destroyed) spawn.
- **SWAP**: swap to other actor when completed (destroyed), for example, a power-up item.

**SPAWN** are created on the behaviour panel, only **ENEMY** actors are taken into account for the group, use the spawn icons to create them, each step must have at least a **TIMER** and a **CREATE ACTOR** icon, once the last step is reached the spawn point will turn from green to red so its completed and will wait until all created enemies are killed.

## Check

Sometimes you necessarily need to check the game state, for example, you need to know when the stage has ended because you defeated a boss, or open a door if all keys have been picked up, **CHECK** actors are for this.



**CHECK** conditions:

- **COND 1-8**: checking condition, checks if an actor does not exists or if exists a minimum amount of them, for example, fighting 3 final bosses, swap them to other actor when killed (a plain **SPRITE** actor), you just need to check if there are 3 of those **SPRITE** actors to confirm that the stage is over.

**CHECK** actions:

- **ENDSTAGE**: triggers the end of the stage event. (**ENDING** properties of the stage)
- **ENEMYHIT**: all enemy actors are hitted by the indicated amount.
- **SWAP**: Swaps defined actors to another ones, if no destination is set, actors are deleted. You can use this to change your hero to another more powerful, open doors (deleting the door).
- **SCORE**: increments player score.

Once done **CHECK** actor it removes itself.

Copyright © 2003-2013 bruneras.com. All rights reserved.



Independent  
Video Game Developer

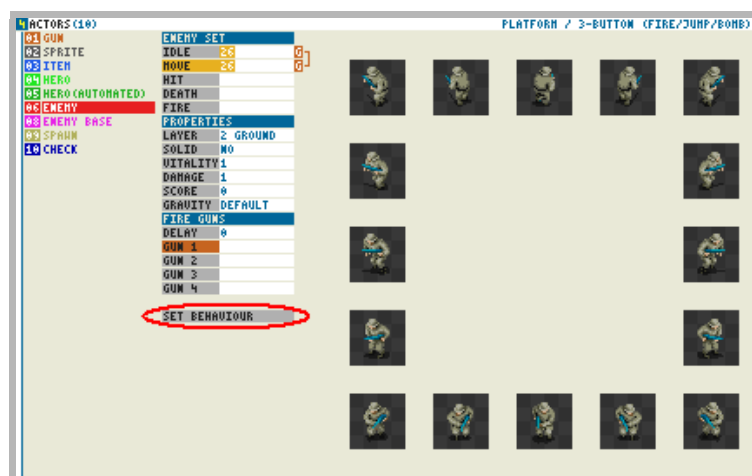
[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)



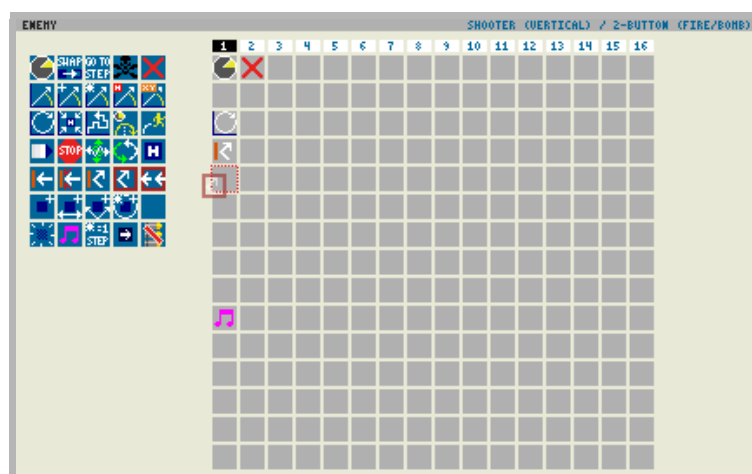
version española

# Actors behaviour screen help topic

Actors can have a behaviour (except heroes), to enter the **BEHAVIOUR** screen click the **SET BEHAVIOUR** option at the bottom of the property list.



You will see a icon panel at the left and 16 columns. Each column represents a behaviour step, so an actor can have up to 16 steps.



Once an actor gets active in the game it will play the first column step, if no icons are defined it stops. To set a behaviour icon you only need to **drag and drop** it to a column, there is a space designed for each icon, and icons of the same class occupy the same space, so they get replaced even though you dragged it to a different space in the column.

Most icons have parameters, to modify them select the icon from the column and parameters will appear below the icon panel.

You can copy an icon from one step to another, drag the icon and keep [SHIFT] down to the desired step.

Here is the list of icons and their behaviour:



Sets the timer for the step, once ended moves to next step, it has 2 parameters:

- **FRAMES:** sets the time in frames, games play at 59 frames per second.
- **REPEAT:** sets how many times should repeat the current step.



Swaps current actor to another actor.



Jumps automatically to other step, does not perform any other behaviour in the step.



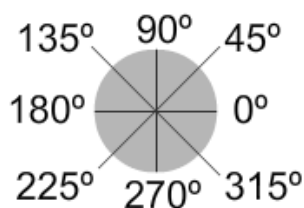


Performs the -kill- actor event, used only by **ENEMY** and **GUN** actors.



Deletes the actor.

Direction behaviours, most of them use angles to set the actor direction, so here you have a reminder:



Sets the direction angle for the actor, it has 2 parameters:

- **ANGLE**: desired angle, decimals included.
- **SPEED**: sets how should the actor turn to the desired angle, instantly or with an angle speed, so the effect will be that the actor turns slowly.



Adds an angle to the actor's direction angle, it has 2 parameters:

- **ANGLE**: increment/decrement angle.
- **SPEED**: same as above.



Sets a random direction angle, it has 2 parameters:

- **ANGLE**: random angle (multiple of), so for example if you set its value to 5 it will only generate random angles multiples of 5 this is 0 - 5 - 10 - 15...
- **SPEED**: same as above.



Sets a **HOMING** behaviour to the actor, any actor with this set will move towards the hero except if the actor is created by the hero, in this case homing will search for the nearest enemy on the screen and move towards it. has only **SPEED** property.



Sets direction angle by a positional location, has 3 parameters:

- **X HOR**: horizontal position, can be **DEFAULT** actual position, **POINT** (faces screen position) or **HERO** (faces hero horizontal position). Relative position or global if **POINT** is selected, and if the actor looks or moves to the position.
- **Y VER**: same as **X HOR** but with vertical values.
- **SPEED**: same as above.



Sets a turn movement to the actor, has 2 parameters:

- **TURNS**: turns to perform, positive values will make a left turn, negative values will make a right turn, decimals are taken into account so a value of 2.5 will perform two turns and a half.
- **SPEED**: angle speed for the turn.



Sticks the actor to the hero, perfect for shield guns, body wheapons, swords... Actors with this icon will act like a part of the hero.



Sets 4-Way movement, only for **MAZE** games, it has 2 parameters:

- **TO HERO**: sets if you want the actor to seek for the hero while it moves.
- **EVERY**: frames to perform before it changes its direction to a 4-Way direction.


All of the above belong to the same class: direction behaviours.




Sets a parabolic movement to the actor, used in all game genres except for **PLATFORM** games, it has 3 parameters:

- **HEIGHT**: in pixels, is the height at the midpoint.
- **INITIALH**: initial height in pixels, if you want the actor to start from a certain height.
- **BOUNCES**: number of bounces to perform, each bounce will be lower than the previous.



Use it in conjunction with:  to set the time to perform the movement, default is 100 frames.

Also use it in conjunction with:  to set the destination, for example if you want to make a parabolic gunshot to the hero.



Performs a jump, only for **PLATFORM** games, it has 2 parameters:


- **SPEED:** vertical speed in pixels, it will be decremented each frame by actor **GRAVITY** value.
- **BOUNCES:** number of bounces to perform, each bounce will be lower than the previous, and unless this parameter is set to **INFINITE**, behaviour will move to next step once the jump has ended.



Sets actor speed, it has 2 parameters:

- **SPD IN:** initial speed.



- **SPD OUT:** speed out, if enabled use it in conjunction with  to set the time to perform the increment/decrement of speed.



Sets actor speed to 0.



Sets a swing movement to the actor, it has 2 parameters.

- **HOR MOVE:** set the horizontal **RADIO** in pixels and an angle **SPEED** value.
- **VER MOVE:** set the vertical **RADIO** in pixels and an angle **SPEED** value.



Same as above but sets a circular movement to the actor.



Sets hero events.

- **DISTANCE:** checks the distance to the hero, moves to a step if the actor is closer than a given value, can make a horizontal or vertical position check.
- **LOOK AT:** actor looks at hero, if you want an actor to face the hero but have a different movement.



Sets wall collisions.



Sets wall collisions, if it collides with a wall death event is raised for **ENEMY** and **GUN** actors, otherwise deletes it.



Sets bounce on wall collisions.



Sets bounce on edge.



Sets continuous edge, if the actors exits the screen it will appear on the opposite side.

An actor can be used as a spawn of other actors, for example a gun can spread in multiple directional guns, or create random sprites for an explosion effect. You have this icons for this:



Creates an actor at the current actor position, has only 1 parameter:

- **ACTOR:** actor to create.



Creates aligned actors at the current actor position, has 4 parameters:

- **ACTOR:** actor to create.
- **ANGLE+:** actors are created perpendicular to the current actor direction angle, you can add or subtract an angle value with this property.
- **RADIUS:** radius distance to the center of the current actor.
- **INSTANCE:** number of instances to create along the line.
- **RANDOM:** if set instances are created randomly along the line.



Creates wedge actors at the current actor position, has 4 parameters:

- **ACTOR:** actor to create.
- **ANGLE+:** actors are created in a wedge form to the current actor direction angle.

- **RADIUS:** radius distance to the center of the current actor.
- **INSTANCE:** number of instances to create along the range.
- **RANDOM:** if set instances are created randomly along the range.



Creates random actors around current actor, has 3 parameters:

- **ACTOR:** actor to create.
- **RADIUS:** radius size of the virtual circle where the actor will be created.
- **INSTANCE:** number of instances to create.

All of the above belong to the same class: spawn icons, so only one per step, also beware of possible recursive loops like: A creates B and then B creates A.



Sets a blinking effect to the current actor, has 1 parameter:

- **VISIBILITY:** blink speed, it also can be set to invisible.



Plays a sound, has 1 parameter:

- **SOUND:** sound effect.



Sets a jump chance, has 2 parameters:

- **CHANCE:** sets the chance value, lower values will increment the chance to trigger the step jump.
- **STEP:** step to jump to.



Only for **ENEMY** actors, sets the track property for the step, so if the actor collides with a track cell will change its direction to the arrow pointing direction.



Only for **ENEMY** actors, sets the no shooting property for the step.

Copyright © 2003-2013 bruneras.com. All rights reserved.



Independent  
Video Game Developer

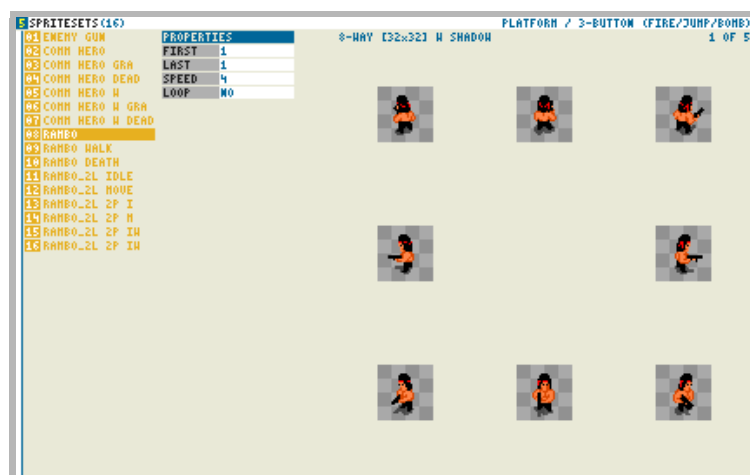
[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)



version española

## Spritesets screen help topic

Enter the **SPRITESETS** screen by pressing [5], here is where you load your **SPRITE SHEETS** for the actors of your game.



Sprite sheets image files must meet some requirements to load correctly, read [here](#) to know.

Once you have loaded a sprite sheet you can see the number of subimages for each Way in the top-right corner of the screen. As a sprite sheet can contain multiple animations, like walking, jumping, or dying, you will need to define a spriteset for every animation you want to perform, this is easy, you only have to set the following properties:

**FIRST:** first image for the animation.

**LAST:** last image for the animation.

**SPEED:** number of frames (steps) to advance to next image in the animation. One second has 59 frames.

**LOOP:** if the animation loops or not, you can also define a loop for only the last n images.

Select an area over any sprite to set the **bounding box** for the spriteset, bounding box is used for wall collisions, in Platform games the bottom of the bounding box is used as floor collision check.

To test a spriteset, press **space** or double-click on a register. Press **Esc** to stop the animation.

You can also replace, duplicate or delete a spriteset, and also import or export all your spritesets into one unique file.

For large animated files it is recommended to have multiple sprite sheet images instead of one unique image file, to prevent loading slow downs.

Press [F1] to see other features.

Copyright © 2003-2013 bruneras.com. All rights reserved.



Independent  
Video Game Developer

[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)

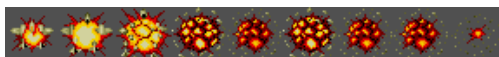


[version española](#)

## Sprite sheet formats help topic

Sprite sheets are the image files you use for your **ACTORS** in your game, this are the heroes, enemies, guns, items or just simple sprites that appear in the game. You can load a sort of different kind of sprite sheets, this image files have to be in **PNG** format and named with a specific nomenclature to load correctly, this are defined below:

### 1-WAY sprite sheet



Unique sprite set for all moving or facing directions, usually used for static images, deaths, explosions, bullets, etc.

Image file format: `filename_1wnumber_of_subimages.png`

Maximum single sprite size: 192\*192 pixels.

## 2-WAY horizontal sprite sheet

---



Sprite sheet has 2 rows: 1st row for **right** and 2nd row for **left** actions, mainly used for Platform games.

Image file format: [filename\\_2wh](#)[number\\_of\\_subimages](#).png

Maximum single sprite size: 112\*112 pixels.

## 2-WAY vertical sprite sheet

---



Sprite sheet has 2 rows: 1st row for **up** and 2nd row for **down** actions, maybe for some specific enemy.

Image file format: [filename\\_2wv](#)[number\\_of\\_subimages](#).png

Maximum single sprite size: 112\*112 pixels.

## 3-WAY horizontal sprite sheet

---



Sprite sheet has 3 rows: 1st row for **up**, **down** and **idle** actions, 2nd row for **right** and 3rd row for **left** actions, usually used for Vertical Shooters.

Image file format: [filename\\_3wh](#)[number\\_of\\_subimages](#).png

Maximum single sprite size: 72\*72 pixels.

## 3-WAY vertical sprite sheet

---



Sprite sheet has 3 rows: 1st row for **left**, **right** and **idle** actions, 2nd row for **up** and 3rd row for **down** actions, usually used for Horizontal Shooters.

Image file format: [filename\\_3wv](#)[number\\_of\\_subimages](#).png

Maximum single sprite size: 72\*72 pixels.

## 4-WAY sprite sheet

---



Sprite sheet has 4 rows: 1st row for **up**, 2nd row for **right**, 3rd row for **down** and 4th row for **left** actions, usually used for Maze games.

Image file format: [filename\\_4w](#)[number\\_of\\_subimages.png](#)

Maximum single sprite size: 72\*72 pixels.

## 8-WAY sprite sheet

---



Sprite sheet has 8 rows that must be drawn in a clockwise manner: 1st row for **up**, 2nd row for **up-right**, 3rd row for **right**, etc., usually used for shooters and run'n'gun games.

Image file format: [filename\\_8w](#)[number\\_of\\_subimages.png](#)

Maximum single sprite size: 72\*72 pixels.

## 10-WAY sprite sheet

---





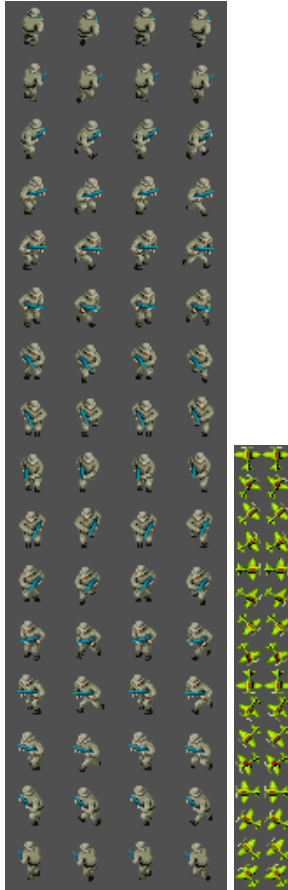
Sprite sheet has 10 rows, first five are for right-faced directions from **up** to **down**, next five are for left-faced directions, from **down** to **up**, mainly used for Platform heroes.

Image file format: [filename\\_10w](#)[number\\_of\\_subimages](#).png

Maximum single sprite size: 56\*56 pixels.

## 16-WAY sprite sheet

---



Sprite sheet has 16 rows that must be drawn in a clockwise manner (see image), usually used for Shooters and Run N'Gun games.

Image file format: [filename\\_16w](#)[number\\_of\\_subimages](#).png

Maximum single sprite size: 40\*40 pixels.

## Shadow sprite sheet

---



If you want your game Actor to use a shadow you can define a shadow image file associated to your sprite sheet, it's easy, you only have to make a copy of your sprite sheet file and rename it by placing a 's' at the end of it. **ARGS** will load your sprite sheet file and shadow file as a whole.

Image file format: [filename\\_format](#)[number\\_of\\_subimages](#)**S**.png

## Important notes

- Each sprite can have up to 99 images in a row.
- Transparent color is RGB [Red=80, Green=80, Blue=80] or HEX value **#505050**, as the one used in the examples.

Copyright © 2003-2013 bruneras.com. All rights reserved.



Independent  
Video Game Developer

[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)



[version española](#)

## Pixel art help topic

---

Drawing original sprites and animations is a tough task, they have to be drawn one by one and pixel per pixel, but you feel rewarded when you see your own character moving, jumping or shooting.

There are great programs for this task, ones just for drawing sprites, others also for animating them, here is a brief list.

**Cosmigo Pro Motion**, one of the best drawing/animating sprite tool you can find around.

**Gimp**, another good one, but you'll need to configured it for pixel art use.

**Adobe Photoshop**, nothing to say about it that you don't know, a great drawing program.

**Graphics Gale**, another good one.

**Jasc Animation Shop**, no official website, search for it in *Google*.

**Iconomaker**, very easy to use.

**Aseprite (Allegro Sprite Editor)**

**Pyxel Edit**

**MS Paint**, simple, but useful for copy+paste single sprites to a sprite sheet.

Other painting programs:

**RealWorld Paint**, **Spriter**, **Graf2**, **mtPaint** and **D-Pixel**. (links courtesy of Carlos)

## Sprite resources

---

Because drawing sprites is not for everyone there are many sprite resources sites for your sprite sheets, just remember that this is usually copyrighted stuff so take this into account if you publish your game.

Here are some links to great resources.

**OpenGameArt**

**The Spriters Resource**

[Sprite Database](#)[NES SNES Sprites](#)[The ShyGuy Kingdom](#)[ScrollBoss](#)

If you know about a good program for pixel art, animation or any sprite resource site let me know and I will post it here.

Copyright © 2003-2013 bruneras.com. All rights reserved.



Independent  
Video Game Developer

[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)



[version española](#)

## Sounds screen help topic

Enter the **SOUNDS** screen by pressing [6], here is where you load the **SOUND EFFECTS** and **TUNES** you will use in your game.



**SOUND EFFECTS** are the beeps, explosions, lasers, and all kind of sounds in your game, there can be playing many at a time, and have two properties **LOOP**, and **VOLUME**.

**TUNES** are used for the background music of your game, or the 'Ready!' and 'Game Over' melodies, and it only plays one at a time. this registers appear as light blue, and have only a **VOLUME** property.

To test a sound, press **space** or double-click on a register. Press **Esc** to stop all sounds.

You can also replace or delete a sound, and also import or export all sound files into one unique file.

Press [F1] to see other features.

## Sound tools and resources

Creating your own sounds is even tougher than creating Sprite Sheets, fortunately there are online tools that can make this task much easier for you, like [www.bfxr.net](http://www.bfxr.net), this is an incredible online tool where you

can create a sound effect in seconds, and exports directly to **WAV**!

Sounds must be in **WAV** format to work with **ARGS**, so if you have your sound files on mp3 or any other audio format you will need to convert them to **WAV** first, you can find dozen of tools for this on the internet.

The legend says that **WAV** sound files are always huge in size... but that is not true, and the great thing about this is that you can get a reduced file without using any third party software, [here](#) is how.

A free sound editor and a good one too is [Audacity](#)

And what if you could have a *Google like* search engine for sounds?, well, you got it! [www.soungle.com](http://www.soungle.com)

Anyway, here you have a compilation of some sound effects so you don't start empty handed.

There are also many resources sites where you can find free royalty sounds:

[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](#)

If you know about a free sound resource site let me know and I will post it here.

Copyright © 2003-2013 bruneras.com. All rights reserved.



Independent  
Video Game Developer

[Home](#) [Help & Tutorials](#) [Download](#) [Games](#) [Back to Main](#)



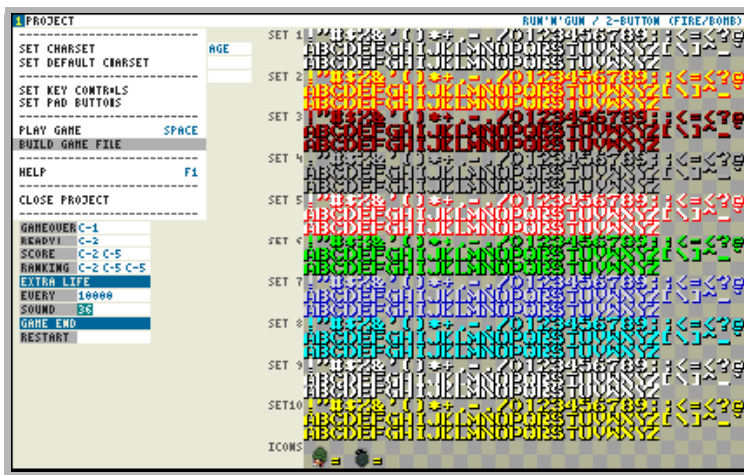
[version española](#)

## Building and distributing your games help topic

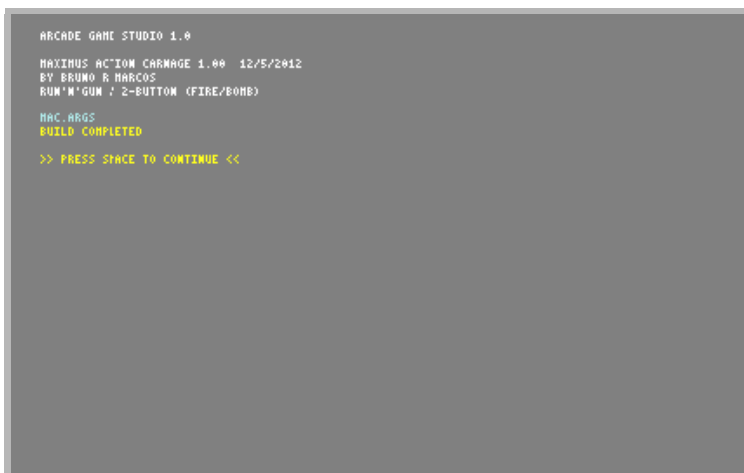
---

Once you have completed and tested your game as normal is that you'll want to distribute and/or sell it.

For this you will need to have acquired the full version of Arcade Game Studio to have the build-game-file feature activated.



When building your game, an unique file will be generated with all the game data, this file will have a **.args** extension and will be stored in a folder named "build" in the project folder.



As the **.args** file is not an executable you will need the game launcher [cabinet.exe](#), so copy your game file into the folder containing cabinet.exe, then launch cabinet.exe to play it.

Take note that cabinet.exe will search for the first .args file in the folder so make shure that there aren't other game files on the folder.

So this means that when you distribute your game you will need to distribute 2 files: your .args game file with cabinet.exe to launch it.

Cabinet.exe can be renamed to your game name if you like, read Cabinet README.txt for more advanced features of cabinet.exe.