

SpriteMaker Tool Documentation

M.U.G.E.N, (c) Elecbyte 1999-2009

Documentation for version 1.0 (2009)

Updated 19 September 2009

Contents

- [SpriteMaker Tool Documentation](#)
- [Overview](#)
 - [Terminology](#)
- [Using Sprmake2](#)
 - [Using lz5 compression](#)

Overview

Sprites in M.U.G.E.N are contained in sprite files with a .sff extension. These sprite files are also referred to as SFFs. Aside from sprites, SFFs may contain palettes as well.

In M.U.G.E.N 1.0, the version of SFF is 2.0, and it is referred to as SFF v2. Versions of M.U.G.E.N 2002.04.12 beta and older use SFF v1, which is now deprecated. M.U.G.E.N 1.0 is capable of loading both SFF v1 and SFF v2 files.

The tool Sprmake2 is used to generate SFF v2 files from source images in PNG format. Sprmake2 is used in conjunction with a definition file to generate each SFF file.

Terminology

Axis

A point on a sprite that corresponds to the point on the screen, at the drawing coordinates of the sprite. Conceptually, this the axis corresponds to "where the sprite is". The axis is numerically represented as the location of that point on the sprite's bitmap, with respect to the upper-left corner of the sprite.

Using Sprmake2

Sprmake2 is a command-line tool, and requires basic knowledge of navigating the Windows command prompt.

Typing sprmake2.exe at the command prompt in the M.U.G.E.N directory will display help.

```
SprMaker ver 2.00beta (2009-08-03)                               Elecbyte 2009

Usage: sprmake2 [options] [-o <out_file>] <in_file>
Options: c - automatically crop images and adjust axes
         f - link duplicate files
         o - override output filename
         p - link duplicate successive palettes
         q - quiet mode (except warnings & errors)
         Q - very quiet mode (no errors)
         v - verbose mode (reports linked/cropped/etc sprites and palettes)
         V - very verbose mode (reports everything)
         txt2def - convert in_file (sff1 .txt) to out_file (sff2 .def)
```

in. file refers to the definition file that will be used as input. The SFF definition file is a set of instructions that instructs Sprmake2 on how to assemble the source images into an SFF file.

For example:

```
sprmake2.exe -o chars\kfm\kfm.sff work\kfm\kfm-sff.def
```

The above example builds chars/kfm/kfm.sff. The output filename may instead be specified in the definition file.

A snippet of kfm-sff.def is shown below, with inline comments:

```
[Output]
;Filename of the SFF file to create (required).
filename = chars/kfm/kfm.sff

[Option]
;Input directory where files are to be read from. Defaults to the same
;directory as this def file if omitted.
;input.dir =
;Sprite compression method for 5-bit sprites (32 color). Choose from:
;none - No compression. This generates very big files (default).
;lz5 - Lossless encoding for 5-bit (32 color) sprites. Compression is slow;
;      decompression is fast. Compression ratio is good.
;rle5 - Lossless encoding for 5-bit (32 color) sprites. Compression is fast;
;      decompression is fast. Compression ratio is average.
sprite.compress.5 = lz5
;Sprite compression method for 8-bit sprites (256 color). Choose from:
;none - No compression. This generates very big files (default).
;rle8 - Lossless encoding for sprites with color depth up to 8-bits (256
;      colors). Compression and decompression are fast. Compression ratio is
;      low. Suitable for images with 33 to 256 colors.
;      Advanced: best compression ratio is achieved on images where most of
;      the color indices are in the range 0-127 and 192-255.
sprite.compress.8 = rle8
;Sprite compression method for 24-bit sprites (true color). Choose from:
;none - No compression. This generates very big files (default).
;rle16- Lossless encoding for 16-bit (high color) sprites. Input files in 24-
;      bit format will be downsampled to 16-bits before being encoded.
;      (NOT YET AVAILABLE)
;rle24- Lossless encoding for 24-bit (true color) sprites.
;      (NOT YET AVAILABLE)
sprite.compress.24 = none
;This option affects the program that uses the sprites.
;0 - sprites are to be decompressed when needed (default).
;1 - sprites are to be decompressed when loaded from disk. This uses more
;      memory when the program runs, but the sprite can be drawn faster. This
;      setting is suitable for images that are drawn very frequently, especially
;      large ones (eg. in backgrounds).
sprite.decompressonload = 0
;0 - Detect identical files only through filename. Duplicate files references
;      are linked to the original in order to reduce SFF file size as well as
;      memory usage (default).
;1 - More thorough comparison of sprite files; actual bytes are compared. Use
;      this setting to reduce file size.
;      Note: Comparisons are done before cropping.
sprite.detectduplicates = 0
;0 - Do not auto-crop sprites (default).
;1 - Automatically crop sprites before adding to them the SFF file. Use this
;      setting to reduce file size.
sprite.autocrop = 1
;0 - Detect identical files only through filename. Duplicate files references
;      are linked to the original in order to reduce SFF file size as well as
;      memory usage (default).
;1 - More thorough comparison of palette files; actual bytes are compared. Use
;      this setting to reduce file size.
pal.detectduplicates = 1
;0 - Do not discard duplicate palettes (default).
;1 - Discard any duplicate palettes that are detected, instead of linking
;      them. Any sprite using a palette that is removed will be set to use the
;      non-duplicate palette instead, eg. if palette (1,2) is a duplicate of
```

```

; (1,1), palette (1,2) will be discarded, and any sprites using palette
; (1,2) will be set to use palette (1,1) instead.
; Use this setting to reduce the number of palettes in use, especially
; when adding many palettes through the auto-add palette option
; (sprite.usepal = -1).
pal.discardduplicates = 1
;0 - Do not reverse ACT file color order.
;1 - Reverse the order of colors in the palette when loading from an ACT file.
; Can be used to correct the color order if your image editor saves ACT
; files with the color order reversed. Photoshop is known to do this.
pal.reverseact = 0
;0 - Do not reverse PNG file color order.
;1 - Reverse the order of colors when loading from a PNG file.
; Can be used to correct the color order if your image editor saves it in
; reverse. This affects both sprites and palettes read from PNG files.
pal.reversepng = 0

;Palettes
;Accepts .act as well as .pcx/.png files (the palette is extracted from the
;pcx or png).
;Character palettes should be numbered from 1,1 to 1,6. Extra palettes for
;projectiles, helpers, etc. may use any other number.
;Format:
; group,itemno, fname, startcol,endcol
[Pal]
1,1, kfm.act, 0,255
1,2, kfm2.act, 0,255
1,3, kfm3.act, 0,255
1,4, kfm4.act, 0,255
1,5, kfm5.act, 0,255
1,6, kfm6.act, 0,255

[Option] ;You can set option states at any time
sprite.usepal = -1 ;Use -1 to add and use self's pal (autopal)

;Sprites
;Accepts 8-bit .png files
;group,itemno, fname, axisx, axisy
[Sprite]
9000, 1, f-faceb.png, 0, 0 ? usepal = -1 ;Use own palette (adds as 9000,1)

[Option] ;You can set option states at any time
sprite.usepal = 1,1 ;Use kfm.act for all these sprites
;Sprite's own pal data will be discarded
;Use -1 to add and use self's pal (autopal)
;Removes colors from sprites in order to reduce color depth.
;Specify the range of colors to remove, or use -1 to disable this feature
;(default).
sprite.removecolors = -1

[Sprite]
9000, 0, f-face.png, 0, 0
0, 0, stand00.png, 18,105
0, 1, stand01.png, 18,104
0, 2, stand02.png, 18,104
0, 3, stand03.png, 18,104
0, 4, stand04.png, 18,104
0, 5, stand05.png, 18,104
; and so on

```

The definition file is parsed sequentially, where the [Option] groups set or change the behavior of Sprmake2, and the [Pal] and [Sprite] groups specify palettes and sprites to add.

Using lz5 compression

The default behavior of Sprmake2 assumes all input files are 8-bit (256 color) images. This achieves a reasonable compression ratio with the rle8 algorithm. In order to reduce runtime memory consumption, sprites may be reduced to 32 colors and compressed using lz5.

From kfm-sff-optimized.def:

```
[Pal]
1,1, kfm.act, 224,255
1,2, kfm2.act, 224,255
1,3, kfm3.act, 224,255
1,4, kfm4.act, 224,255
1,5, kfm5.act, 224,255
1,6, kfm6.act, 224,255

...

sprite.removecolors = 1,224 ;Convert 8-bit to 5-bit (32 color) sprites
```

The palettes added are first reduced to 32 colors by the last two parameters of each line (224,255). Those specify the start and ending index of the colors to keep. The remaining colors will be discarded.

The line "sprite.removecolors = 1,224" specifies that color indices 1 through 224 will be discarded from any sprites added after. Colors 0 and 225-255 will be kept, for a total of 32 colors.