

# Adding an Intro to FF3us

This short tutorial will cover how to add an intro, or commonly called *splash screen*, to FF3us. The code and process is based on a [more general tutorial](#) that is somewhat outdated due to the utilities used not being able to run on x64 systems.

## Requirements

You'll need the following to complete the tutorial:

1. [the files used in this tutorial](#)
2. [bass](#)
3. [superfamiconv](#) (or [snesgfx262](#))
4. [snesgfx262](#) (or [superfamiconv](#))
5. an image editor (as an example [Gimp](#))
6. a hex editor (optional, as an example [HxD](#))
7. an IPS patching utility (optional, as an example [Floating IPS](#))

## Preparing the Image

You'll need first to have an image in PNG format. The image must be 256×224 and indexed to 256 colors. The image can have less than 256 colors though. This preparation can be done with Gimp and showing in detail how to do these steps is a bit out of scope for this tutorial. However to index to 256 color, you need to select Image → mode → Indexed and then select an optimal palette of 256 colors.

The image used in this tutorial is the following one and was based on [this](#).



## Using superfamiconv

This is the most straightforward conversion method. If you really want a GUI application, go to next section right away.

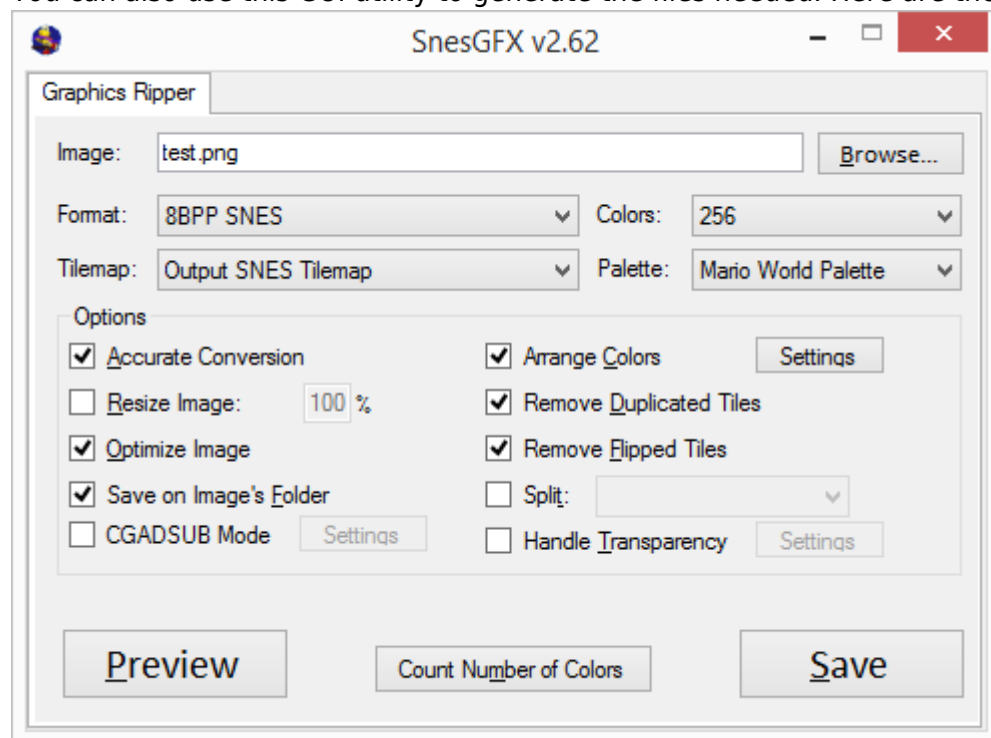
You simply put your image (here test.png) in same folder as the superfamiconv utility and type the following in Windows command prompt (after moving in said directory):

```
superfamiconv -i test.png -p palette.bin -t tiles.bin -m map.bin --no-flip -  
-bpp 8 --map-width 32 --map-height 28
```

This will generate 3 files: tiles.bin, palette.bin and map.bin. You can now skip next section and continue after.

## Using snesgfx262

You can also use this GUI utility to generate the files needed. Here are the suggested settings:



Now after the conversion you'll need to rename the 3 files:

test.bin → tiles.bin pal.mw3 → palette.bin test\_map\_16.bin → map.bin

You can optionally trim the [last two bytes of palette file](#) and the [extra \\$100 byte on map file](#) but anyway the DMA transfers coded are transferring 0x200 and 0x700 bytes so there is no worry. These extra step will make you save a bit of space that's all.

## Assembling the intro

When you have your 3 files ready, move them in same directory as bass and intro.asm. You can edit the ASM file to specify a bank for your code and data; since the hack take almost a full bank, it is suggested to input an offset of a bank start, here \$F30000. You can also specify if you need to press a key to proceed to the beginning of FF6 intro or not. In this case input a 1 instead of the 0 on the varKeyPress variable:

```
// splash screen data and code offset
constant slashScreen($F30000)           // change this offset, it is recommended
                                         // to use a whole bank.

// variable to toggle keypress option
constant varKeyPress(0)                 // 0 = wait for keypress
                                         // 1 = automatic fadeout after a few seconds
```

After this all is needed is assembling the hack with bass:

```
bass -o romname.smc intro.asm
```

From:

<https://www.ff6hacking.com/wiki/> - ff6hacking.com wiki

Permanent link:

<https://www.ff6hacking.com/wiki/doku.php?id=ff3:ff3us:tutorial:splashscreen&rev=1522976600>

Last update: 2019/02/12 12:29

