The format NecroDancer uses for modded weapon shape sprites is…

...a bit complicated.

I'm going to explain how to create weapon sprites in the format the game uses, with the help of a program I made. This will allow your weapons to work with all modded weapon materials. Note that I'll mainly be covering how to make sprites that work for weapon shapes. I will not be covering how to actually program weapon shapes.

You will need:

An image editor that works good for pixel art (GIMP, Aseprite, etc)

NecroWeaponTemplateHelper (get it here:
https://github.com/doublestar621/NecroWeaponTemplateHelper)
(its very likely you've already got this program on hand, since i'm distributing this readme with the program)

Foreword:

I, uh, didn't pay any attention at all to where the page breaks ended up, so there might be a few sentences or lists that start on one page and end on the next.

Step 0: Palette

There's a specific set of colors you'll need to use for some aprts. It should be right next to this file, named special_colors.png. You can import it as a palette into your image editor or color-pick from it.

I've also included the palette I typically use for the main sprite in the 'weapon main sprite palette' folder. Feel free to use it, or use your own colors.

Step 1: Primary sprite

First thing's first. Make an image with your image editor, and draw out your weapon. You'll have to use specific colors, though:

Material – White/Grey (weapon blade / main color)
Handle – Red (or brown? orange? unclear)
Grip – Blue

Tip – Cyan
Guard – Magenta
Detail – Green

For examples, check sample-images / 1 - template.

Oh, and if you want a weapon with multiple frames (i.e. crossbow, rifle), now is the time to draw those extra frames. And if you want a set of pixels that always gets added on top of your weapon without ever being changed, don't add that to the sprite yet. I'll explain more about that in step 7.

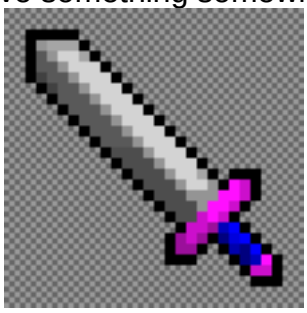
Make sure you get the shape of the weapon right at this stage. You can mess with the colors later, but messing with the shape will be trickier.

Note: You don't need to use all of these colors. Most base game weapons only use 3.

From what I can tell, the game is pretty lenient with what colors you use, so you can probably just eyeball the colors.

Note 2: Regarding handle color: I legitimately have no clue what's up with that. Modding API docs say to use brown, but trident uses orange and spear uses red, and they both seem to work. Also, bow uses a pale yellow for the string which seems to register as handle color (I think?).

Once you're done, you should have something somewhat like this:

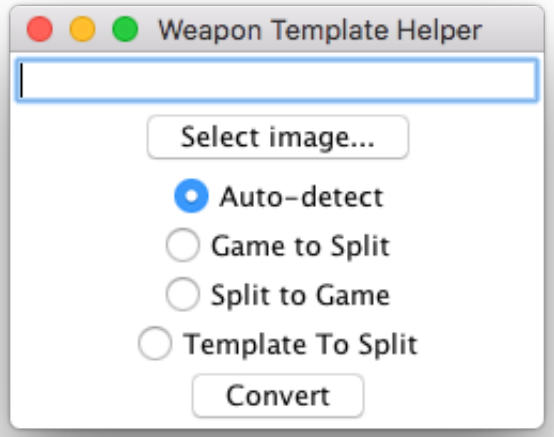


Save it as a png, and put it somewhere you'll be able to find it.

Step 2: Converting to Split format

Open NecroWeaponTemplateHelper. You should be able to just run the 'run' file corresponding to your operating system. If that doesn't work, try running the jar file from command line, terminal, or your operating system's equivalent.

You should be greeted with this GUI:

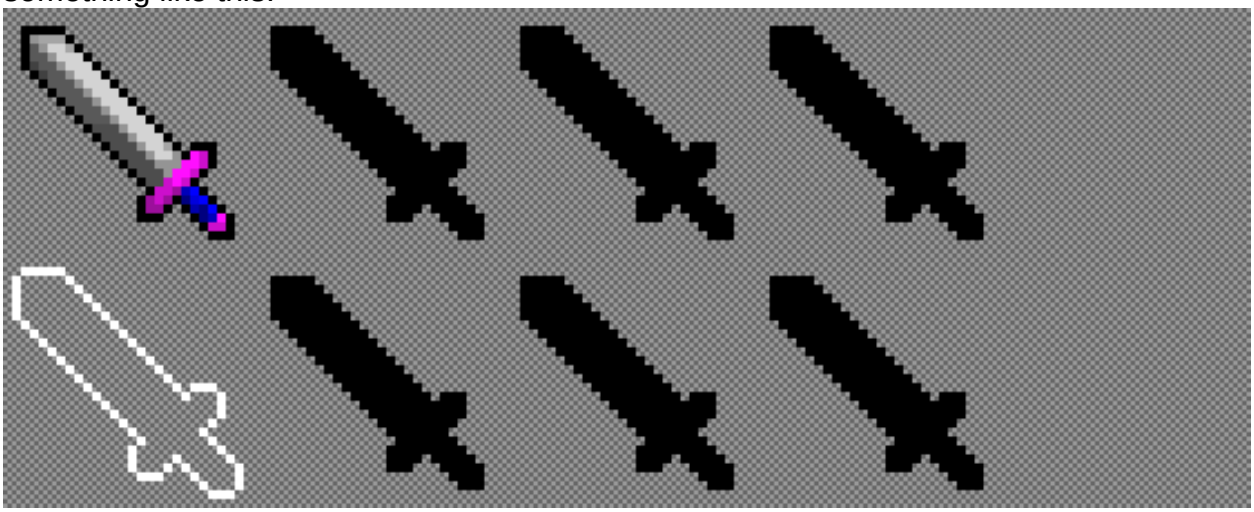


(will look different depending on your operating system probably)

Click 'Select image...' to open a file browser. Select image you saved from the previous step.

If you don't like the file browser and you're on MacOS, you can drag the image from a Finder window into the file browser to quickly select it. If you're on another operating system, you can probably paste the file path into the textbox above the 'Select image...' button.

Select 'Template to Split', then press 'Convert'. This will make a new image in the same directory as the selected image, with the same name, but prefixed with 'split-'. It'll look something like this:

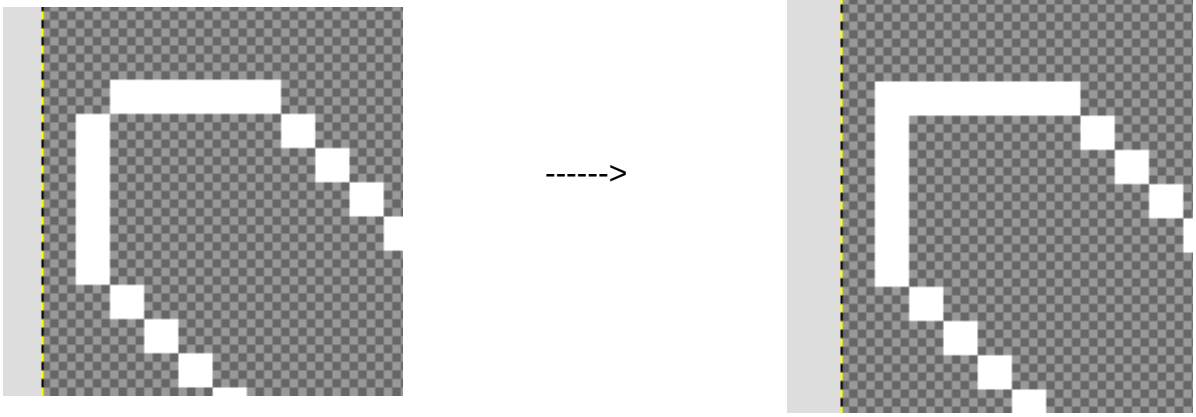


Open it in your image editor of choice.

For the rest of the sprite-making process, you can find examples in sample-images / 2 – split.

Step 3: Weapon glow

The bottom left frame is the weapon glow, which blood, gold, onyx and obsidian weapons use. It generally consists of an outline and some sparkles. You'll note that there's already an outline there, which was automatically generated for you. You may want to tweak it a bit, such as making the outline more pointy on pointy parts of the weapon, like the tip of the blade.



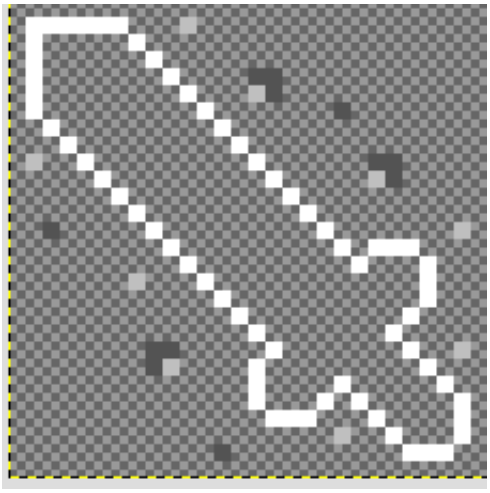
I recommend keeping the outline as full-bright white (#FFFFFF), since this is what all the base game weapons do.

Next, the sparkles. You'll need to use some specific colors for this. These are the second and third colors on the top row in special_colors.png. Here's the hex and RGB values, incase you need those:

#BFBFBF (rgb: 191, 191, 191)
#535353 (rgb: 83, 83, 83)

Sparkles from lighter colors are guaranteed to be visible when particles from darker colors are visible, but not the other way around. Design your sparkles with this in mind.

You'll end up with something like this:



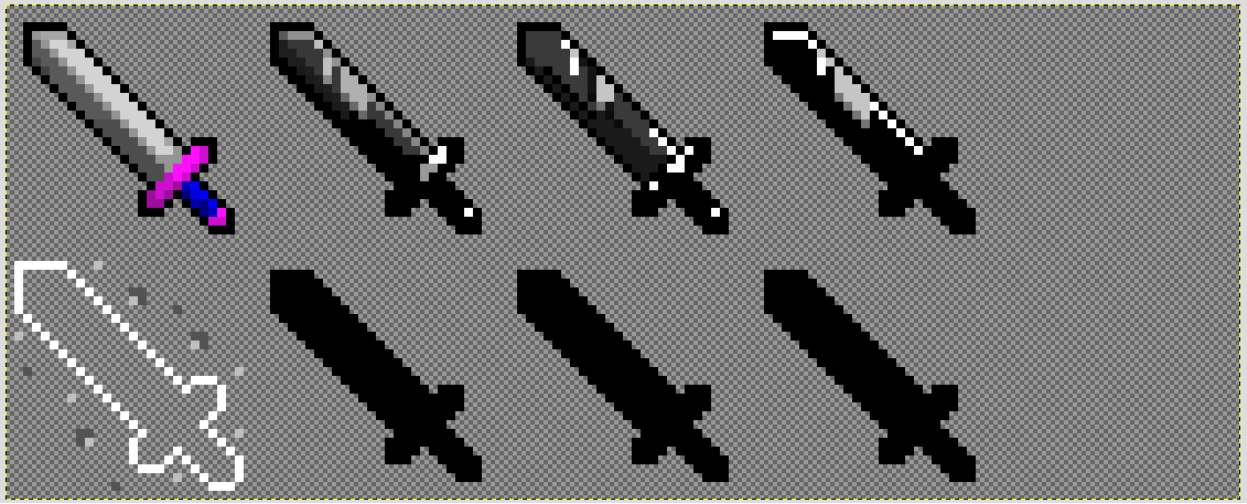
Step 4: Shine

So, you know those black silhouettes of your weapon shape on the spritesheet? It's time to work with those!

The upper three are for shine layers, which materials can use to give weapons a bit of extra shine. You'll want to draw three unique shine patterns, which will be overlaid onto your weapon's main sprite. Note that these layers are meant to be monochrome, so only use white, black and greys for them. Also note that pure black pixels will never be visible in-game.

I recommend referencing the examples available in the sample-images folder.

In case you're wondering which shine layer is used by which material: Gold uses the left one, glass uses the middle one, and obsidian uses the right one. Note that modded materials can pull from any of these, or even a combination of them with differing intensities on each.



Step 5: First aux layer – jewels

The bottom left silhouette is a bit more complicated.

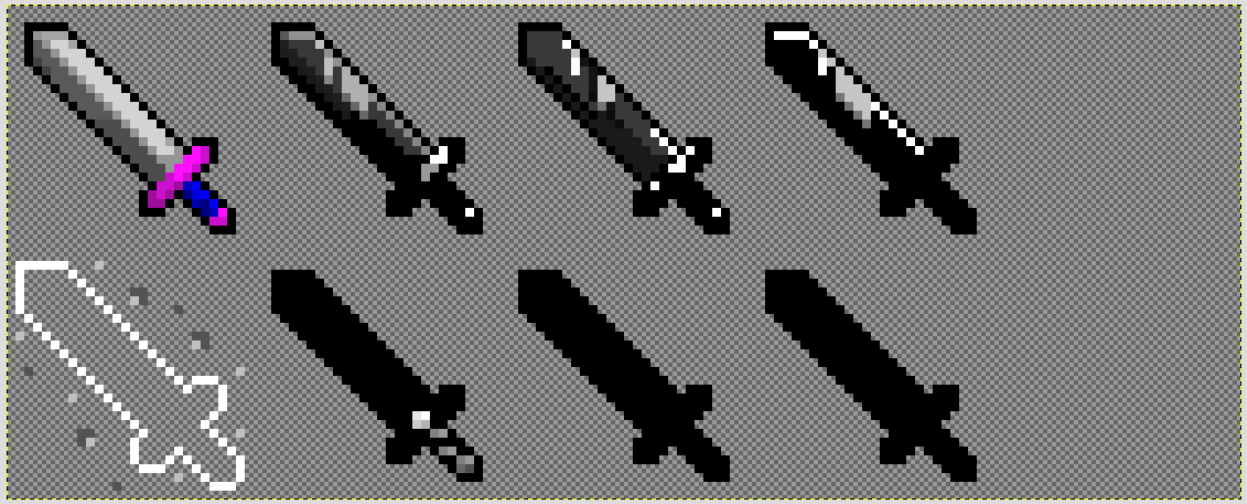
...I'm going to be honest, I don't entirely know how this one works. I don't know why the colors are what they are, nor do I have any way of finding out.

The colors you need to use are the ones in the middle and bottom rows in special_colors.png. I recommend you view the example sprites and copy what they do.

Of note: The four brightest colors used for these sprites are visible on both frost and jeweled weapons, but the rest are not visible on frost weapons, but are visible on jeweled weapons.

...yeah, this one will probably take some trial and error to get right.

Oh, and this layer is meant to be monochrome. Same as the shine layers.



Step 6: Second and third aux layer – Gradients

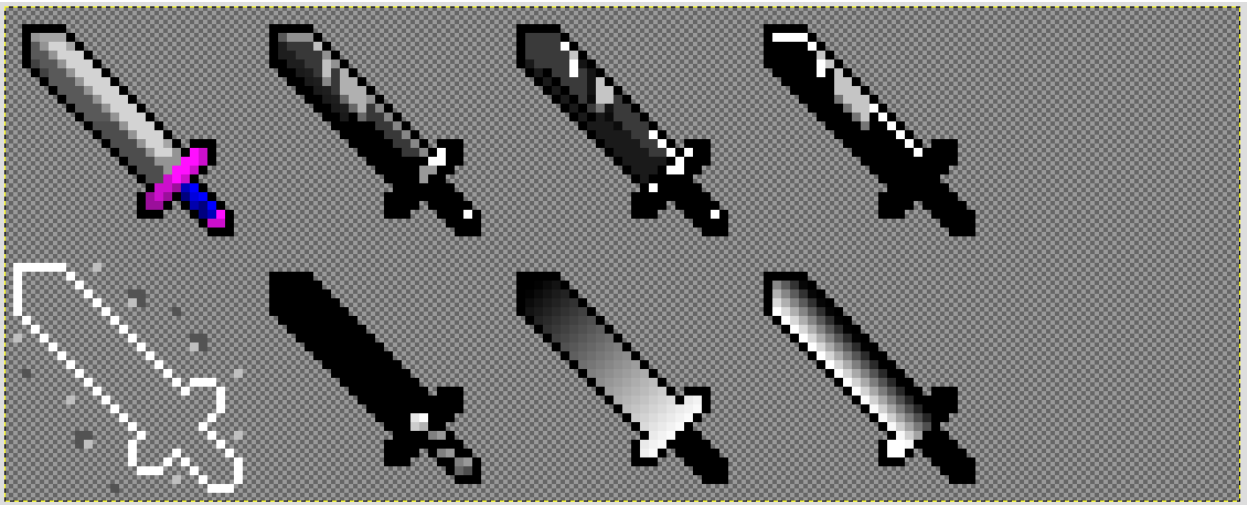
The last two frames are used for gradients. The middle one is for a gradient that goes parallel to the blade (usually), and the right one is for a gradient that goes perpendicular to the blade (usually). There are a ton of exceptions to this, though:

Bow, harp and trident have the gradient apply to the whole weapon.
Crossbow has it apply to the whole weapon minus the arrows.
Dagger has some gradient pixels on the bottom of the weapon.
Longsword has the gradient extend over the guard above the grip.

Your image editor probably has a gradient tool somewhere. Use that to create the gradient. If you're using GIMP, you can select the area you want to apply the gradient to, and then create the gradient. It'll only apply to the selected area.

Feel free to look through the examples and experiment a bit.

Also, these layers are also meant to be monochrome, just like the shine layers and first aux layer.



Step 7: Overlay

Although not many weapons use this feature, it's still worth mentioning.

...and by 'not many weapons use this feature' I mean 'literally only staff and rifle use it'.

The overlay frame is in the top-right corner in the split format, and is put on top of the weapon sprite with zero modifications, regardless of literally anything. The staff uses this for it's gem, and rifle uses it for smoke when not loaded.

If you want something like this on your weapon, I recommend using the staff sprite as an example.

Step 8: Converting back, and getting it in-game.

Save the image, then boot up NecroWeaponTemplateHelper again. Point it to the image you've saved, select 'Split to Game', then hit 'Convert'. A new image will appear in the same directory and with the same name, but prefixed with 'merged-'. This is the image you'll use in your mod. Congratulations! You did it!

All you need to do now is make your custom weapon shape use it in-game. Within your customWeapons.registerShape call, set texture to a string containing the file path of the weapon's sprite, like this:

```
customWeapons.registerShape {
    name = "Shortsword",
    friendlyName = "Shortsword",
    hint = "Knock back enemies, spin attack",
    texture = "mods/DSMisc/sprites/weapon/shortsword.png",
```

Your weapon should now be visible in-game.