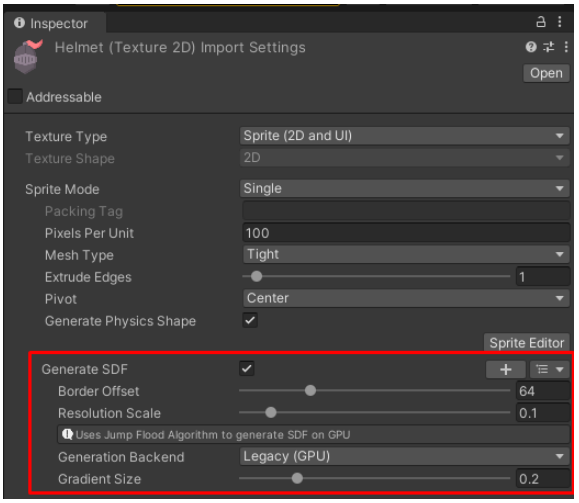
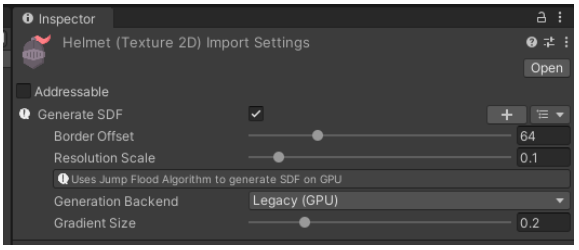
SDF Import Settings

Editor overview

SDFImporter window was deprecated, and now all import settings is displayed in texture inspector.



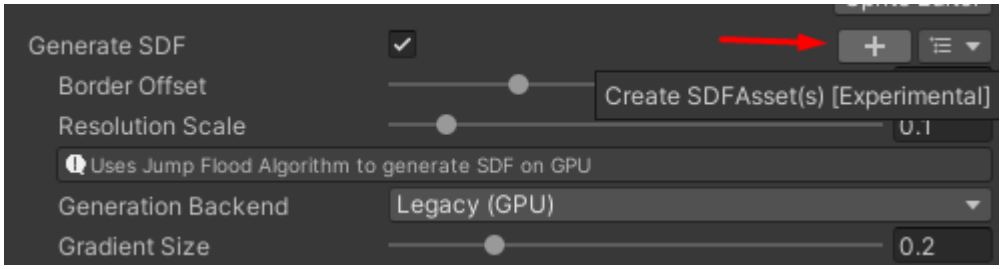
💡 This is done by TextureImporterEditor override, if you have other custom editor for TextureImporter it might take control, in this case import settings will be displayed in header)



Buttons

There is additional controls in SDF import settings section.

1. Plus button is a shortcut to create SDFAsset, main asset of Starting Decoupled Pipeline



2. Dropdown button shows what SDFAsset's in project using this texture

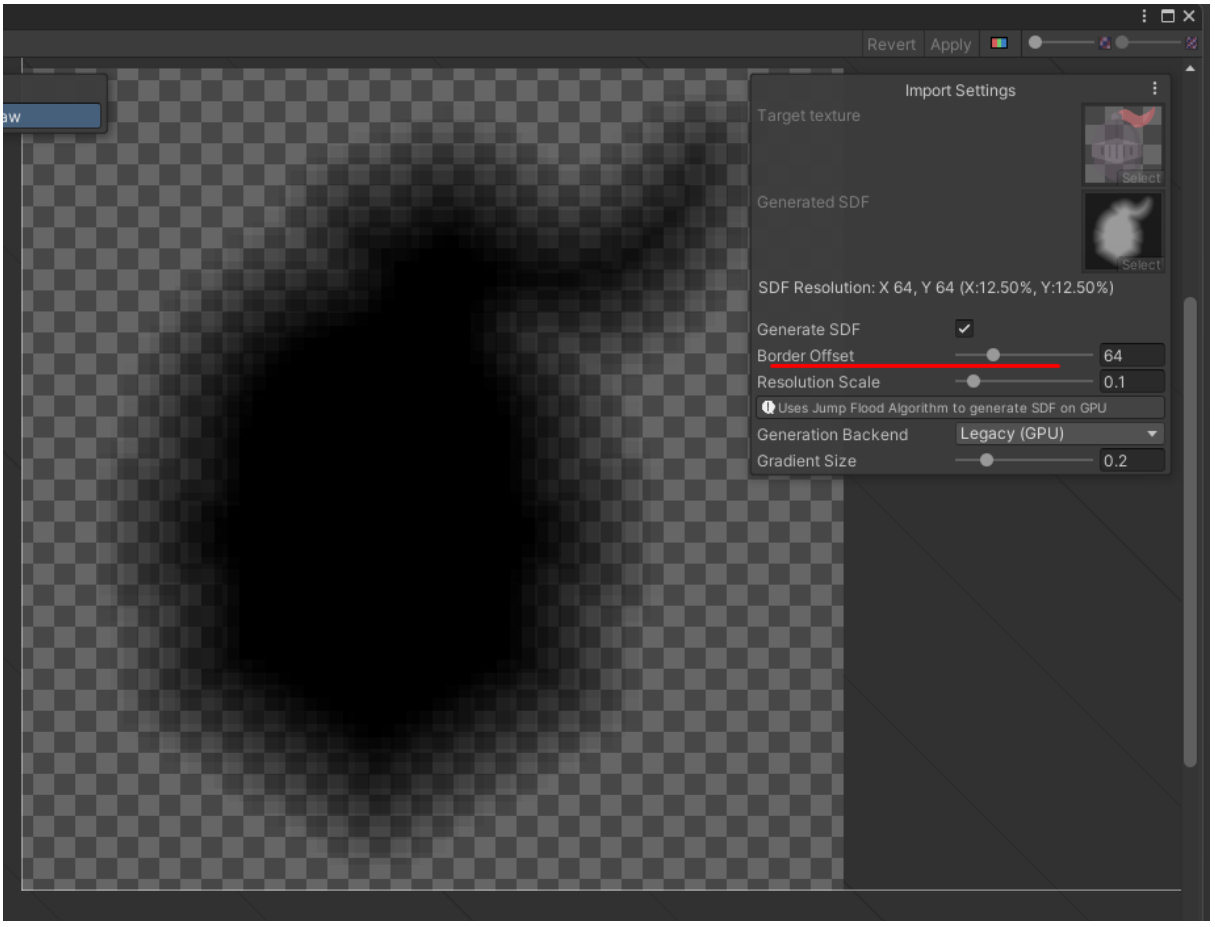
Import Settings

Second section is **Import Settings**, it has all the SDF Import settings.

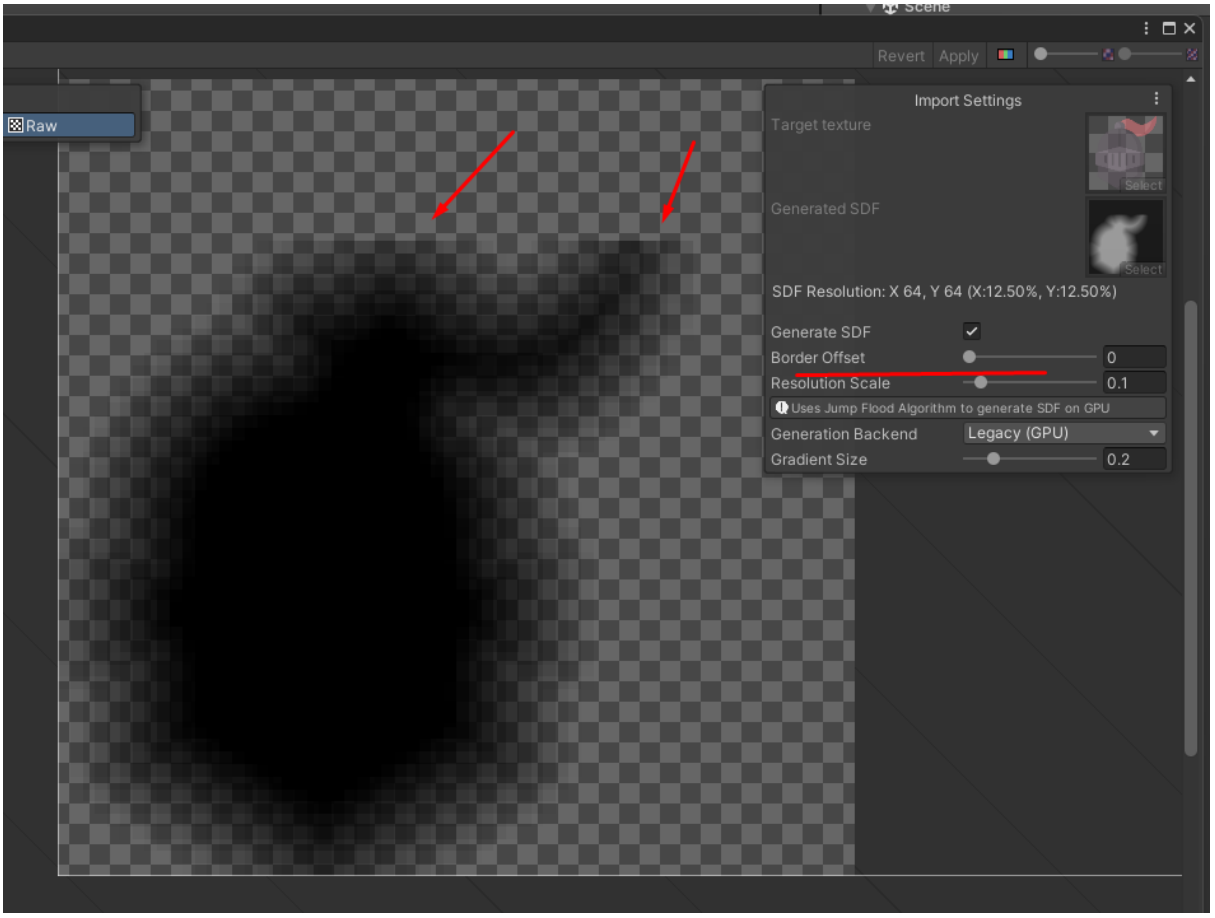
`Generate SDF`

What it does is pretty self-explanatory, if enabled SDF Texture will be generated and `SDFImage` component can display proper outline, without it will act almost as built-in `Image`.

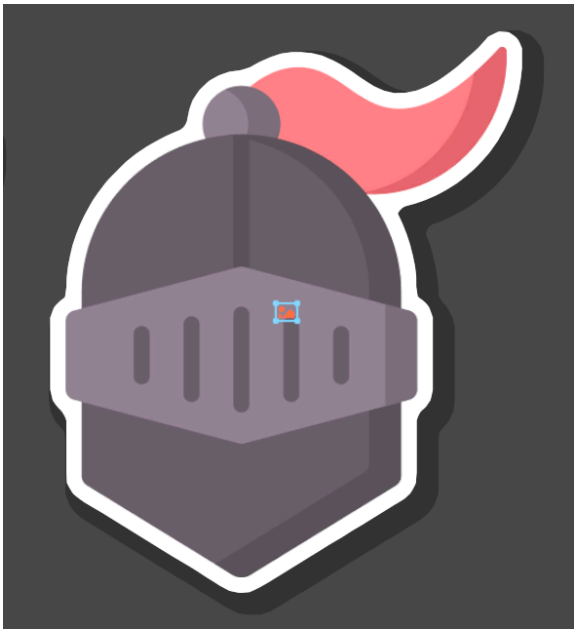
`Border Offset`



Default value of 64 determines how much pixels is added to SDF texture at each edge. Since distance field might expand over original texture edges we need to add some additional pixels to it. Value of 64 okay resolution for 512×512 sprite resolution. This value is adapted to account for texture compression and resolution scale so the final number of pixels added is different.



If `Border Offset` set to 0 we can see distinct edge on SDF texture, meaning some of SDF pixels is lost. On the final image artifacts will look like that.

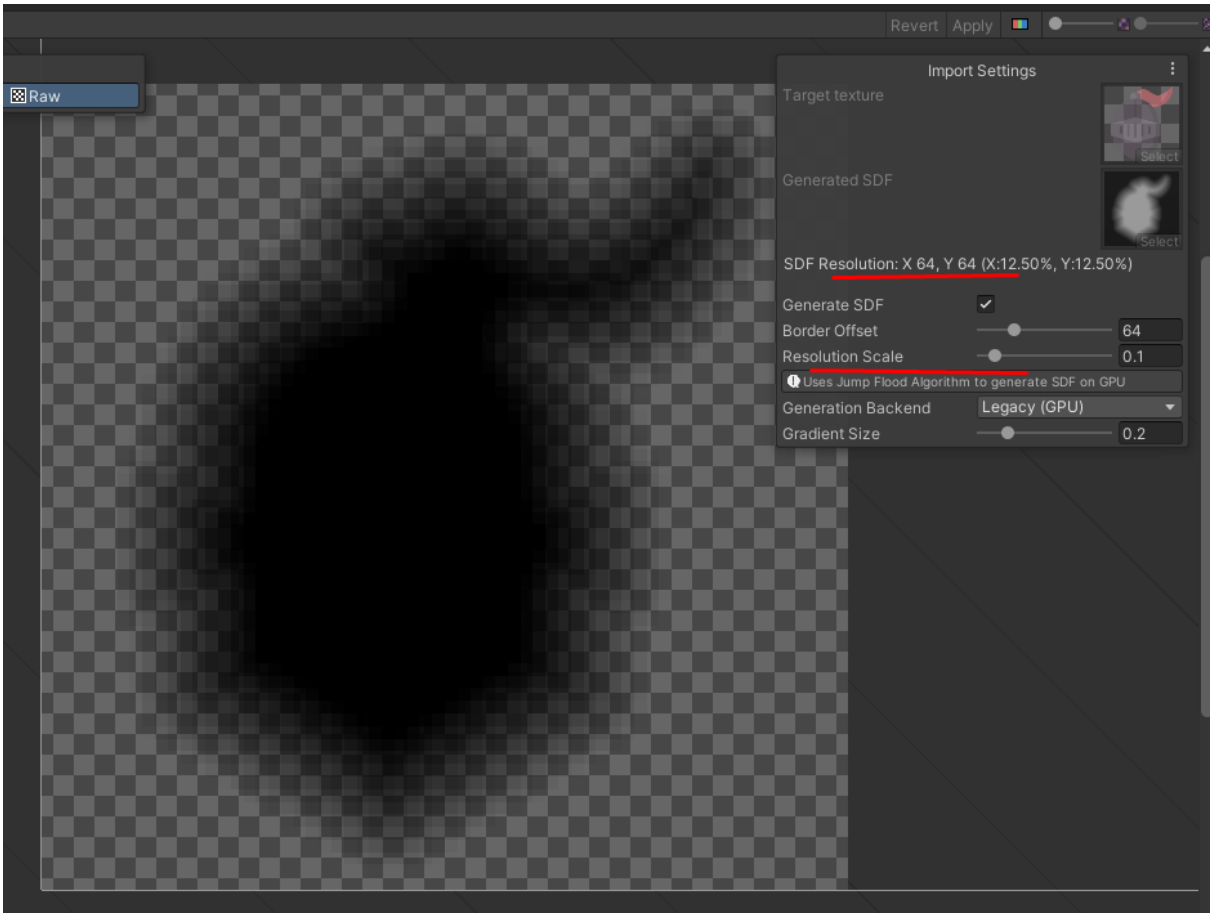


And that is how proper sdf texture with border offset looks like

If you want border offset can be added to the source texture, but usually it's easier to just use `Border Offset`

We will talk more about it when covering multiple-sprites workflow.

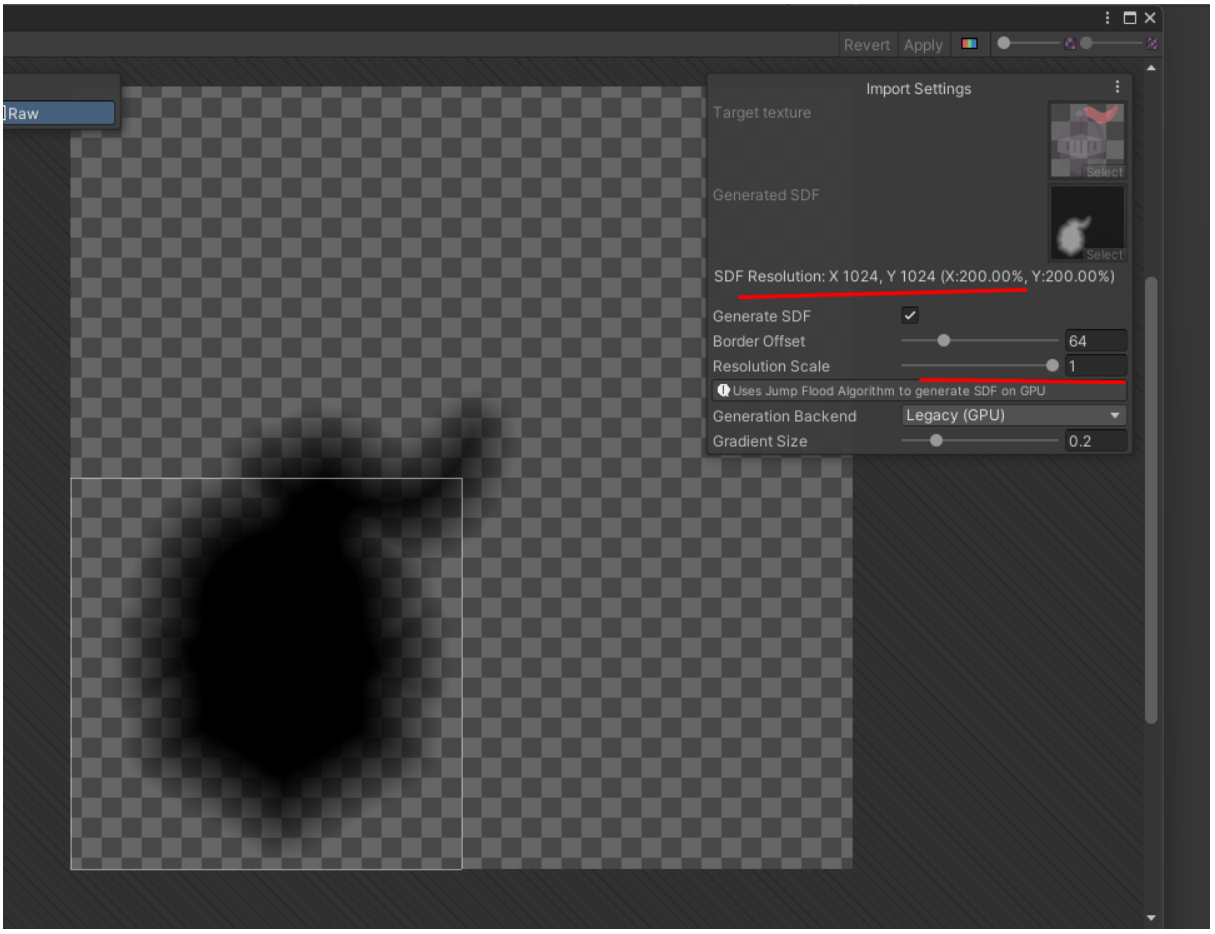
`Resolution Scale`



`Resolution Scale` 0.1

Range 0 - 1, where `0` SDF texture is has 0% `width` x `height` of original texture, and `1` where sdf texture is 100% of original texture size. (minimum size is clamped to be 4 pixels)

The actual size will not be precisely the `10%` of original texture, since border offset is added, and final sdf texture is packed in atlas texture (for multiple sprites texture) and packing might not be efficient. More about that in multiple sprites section.

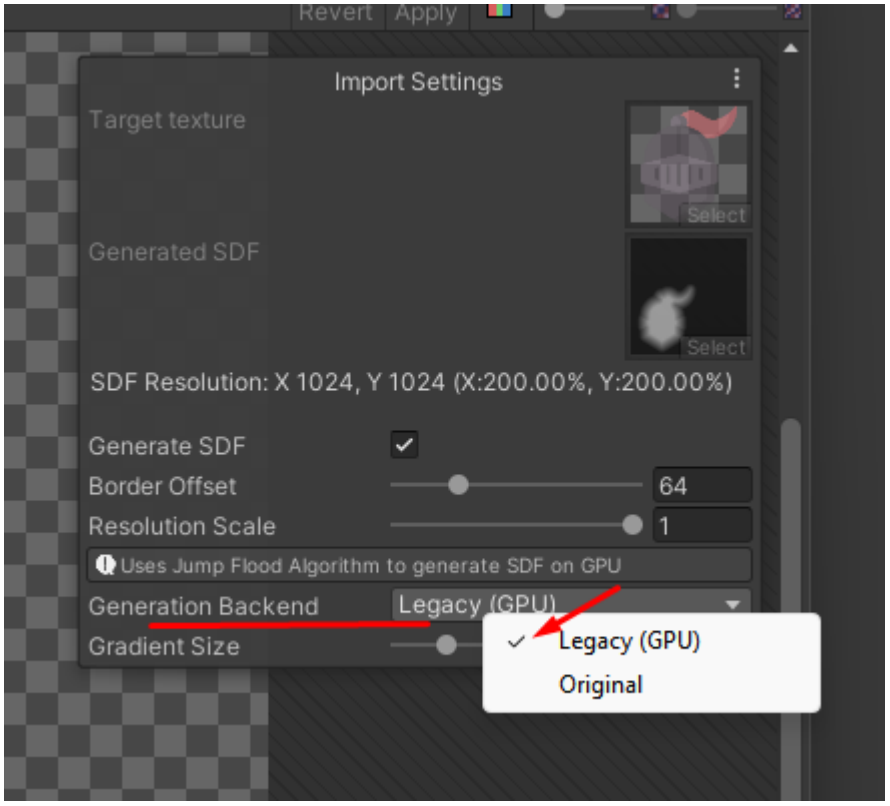


`Resolution Scale` 1

Here is texture with scale of `1`, it is twice as big as big as original since `Border Offset` is added and final texture packing fits generated signed distance field to smallest texture rounded to the power of 2 (32, 64, 128, 256, 512, 1024)

But usually value of `0.1 - 0.2` is enough.

`Generation Backend`

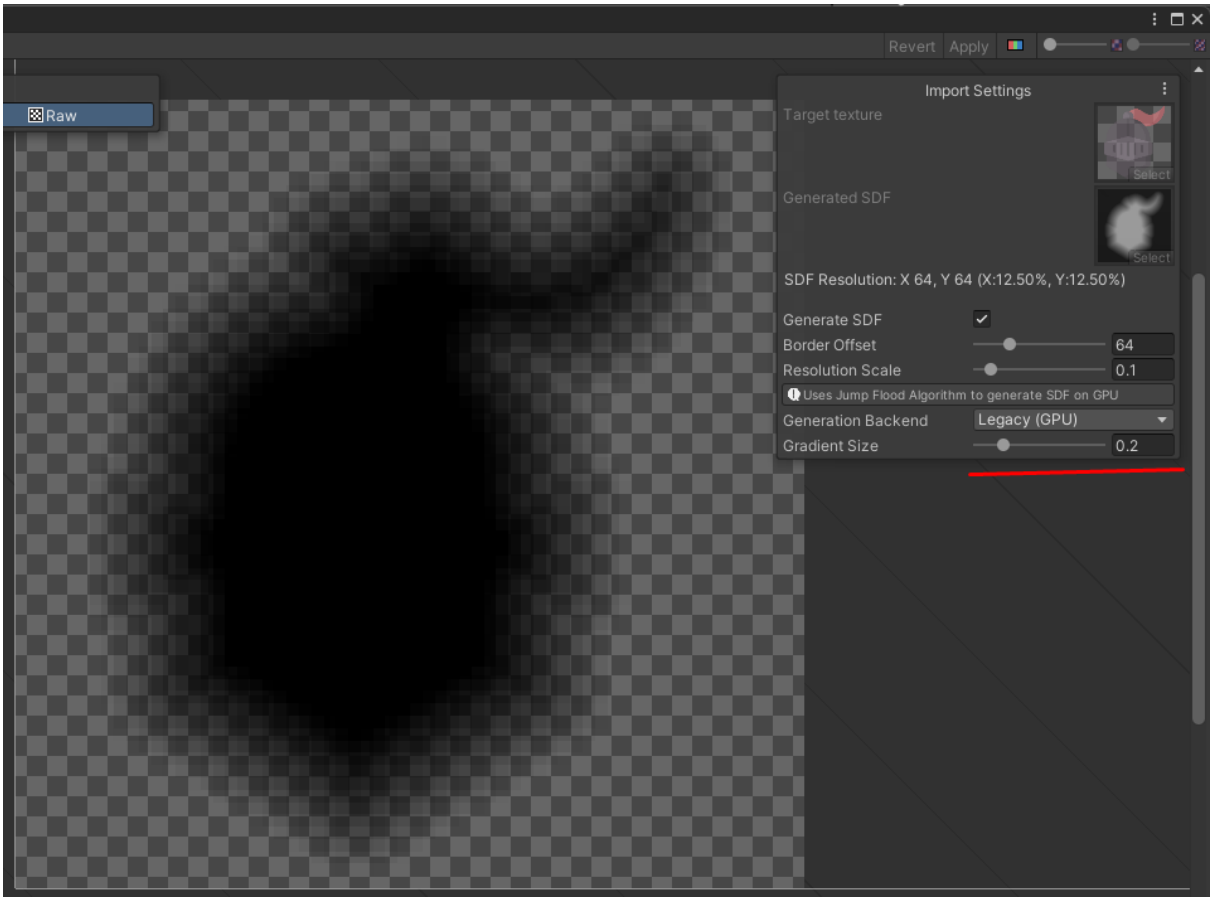


This settings now has only two options:

- **Legacy (GPU)** - generates SDF on GPU
- **Original** - does not generates SDF, just applies resolution scale and border offset to produced texture.

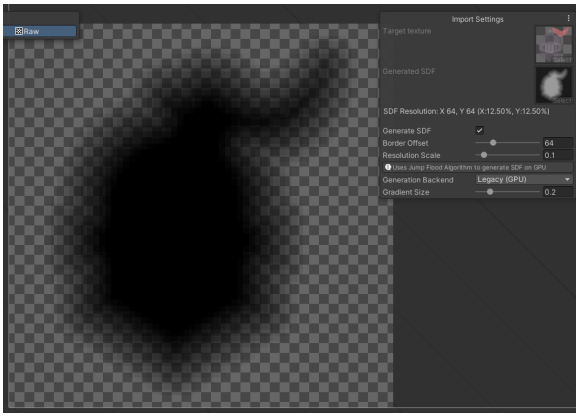
Currently only viable option is **Legacy (GPU)**, but in future **Jobs (CPU)** backend might be added, since CPU is more reliable for such algorithms, and all unity 2D packages uses only CPU. Right now all known bugs for **Legacy (GPU)** is fixed, but GPU's might behave different on different platforms.

`Gradient Size`



Property specific to **Legacy (GPU)** backend, it determines how big signed distance field is, how much it goes from original image edge, lets see the examples.

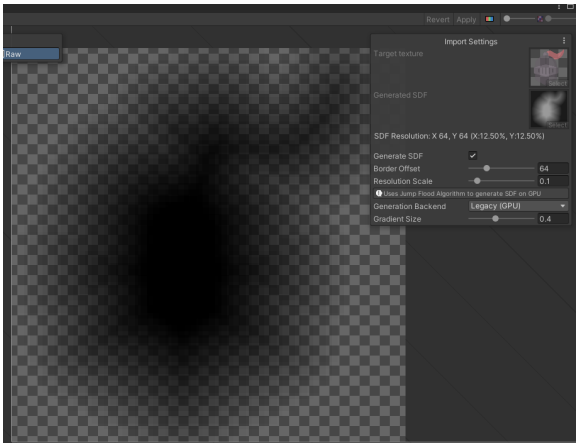
`Gradient Size = 0.2`



Raw 0.2 view

Rendered 0.2 view

`Gradient Size = 0.4`

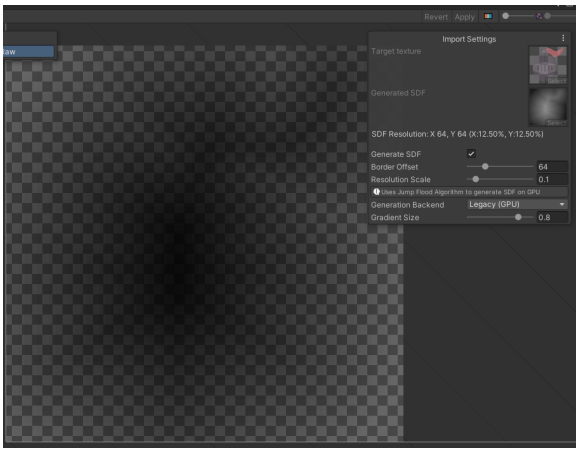


Raw 0.4 view

Rendered 0.4 view

As you can see 0.4 results in much bigger width of final outline/shadow. Lets try something bigger now.

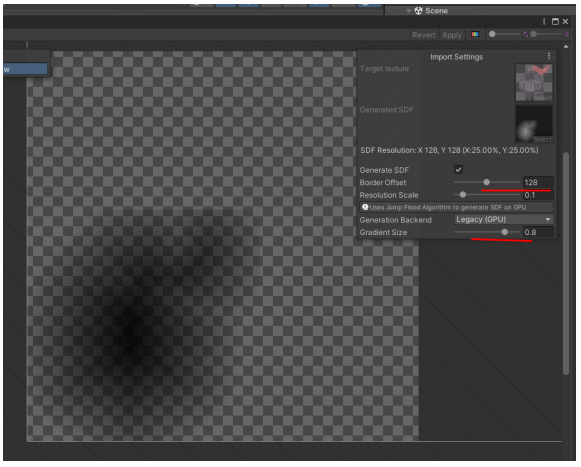
`Gradient Size = 0.8`



Raw 0.8 view

Rendered 0.8 view

0.8 is causing artifacts, to lets try to fix them



To fix artifacts `Border Offset` needs to be changed. `128` in this case.

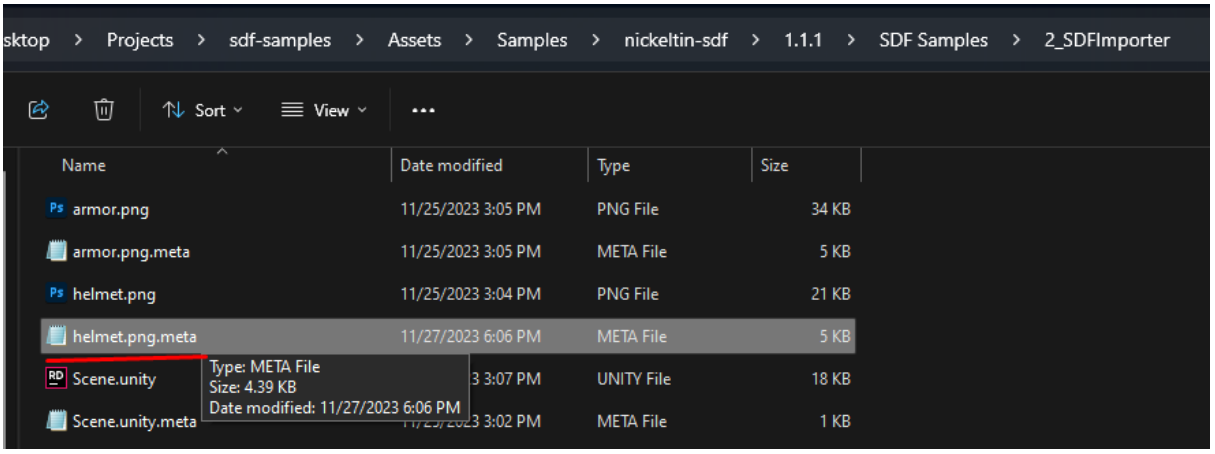
Now artifacts is fixed

What happens is that SDF pixels got cut of since `Border Offset` was to small, increasing it gives enough space for SDF generation, and artifacts is gone.

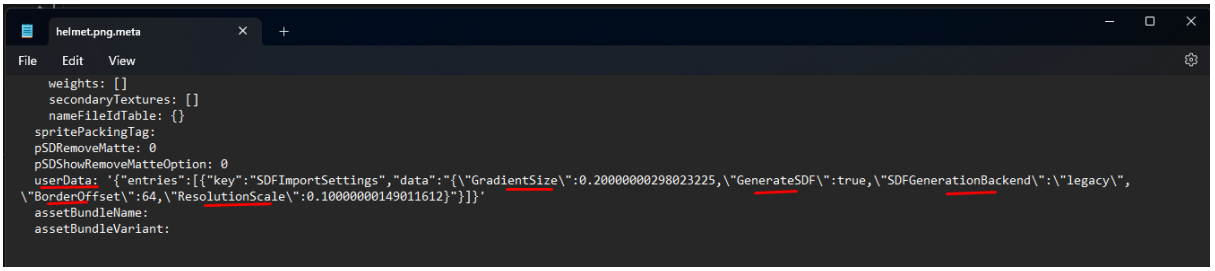
Usually such big outline is rarely used, but is you need it, it is possible.

> 💡 This property final value is different from what you specify in import settings, it is adjusted to be consistent over all texture sizes and compressions, so that single sprite 512×512 texture and big 4096×4096 sprite sheet with bunch of sprites will yield into same final outline width.

Where import settings is stored?



It is stored in texture meta file.



In field called `userData`, it is field that any `AssetImporter` has to store some custom data. Scripting API

Disadvantages of `userData` is that other assets might use this field and then data will be lost. So be careful and aware of assets in your project.

As soon as unity releases its Modular Importers API all SDF import settings will be moved to it.

SDF texture is Alpha8 UNorm texture without compression, it is usually smaller than original texture.

First is **Preview** which is pretty simple right now (old preview modes is removed for now, but maybe will be returned). **Regular** displays source texture, and **Raw** displays generated SDF texture.