Regular Pipeline Scripting

Status	Up-to date

SDF Import

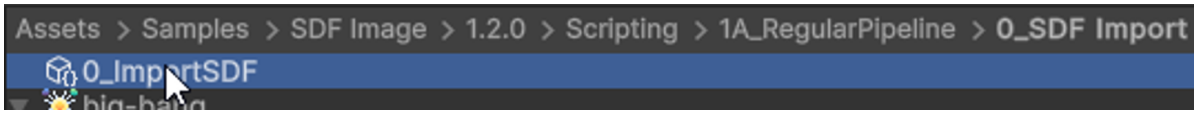
Sample that goes over how to import SDF for texture trough code. And how to change its `SDFImportSettings` .

Fix texture (Import SDF)

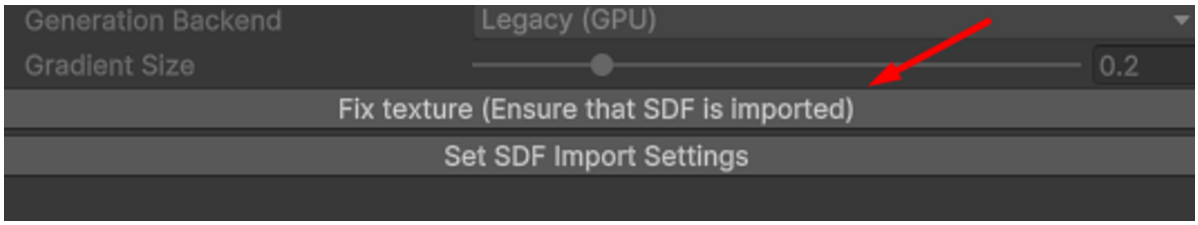
This function ensures that sdf is generated for texture.

> 💡 Texture need to be configured to sprite mode, and have sprites defined, function changes only sdf import settings.

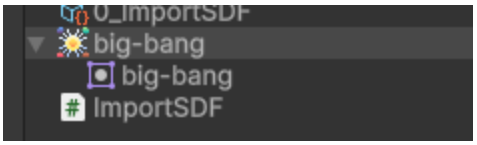
- Select 0_Import SDF : `ScriptableObject`



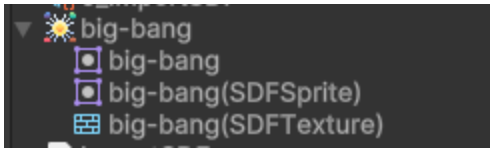
- Press *Fix texture (Ensure that SDF is imported)*



- See how big-bang.png now has sdf generated inside of it



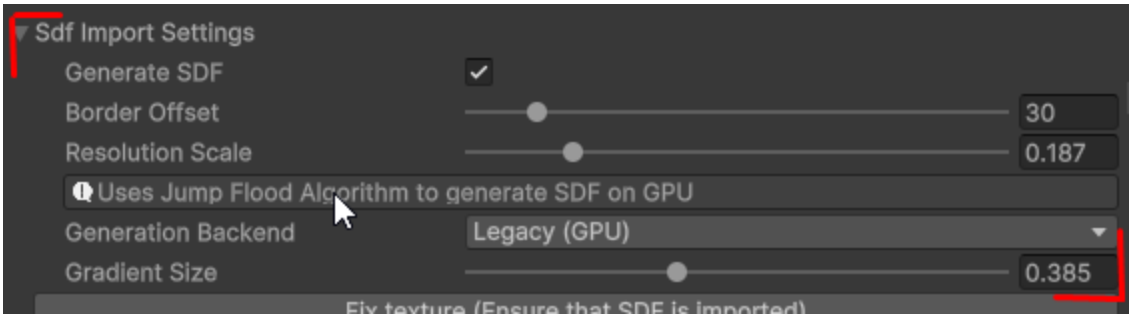
Before



After

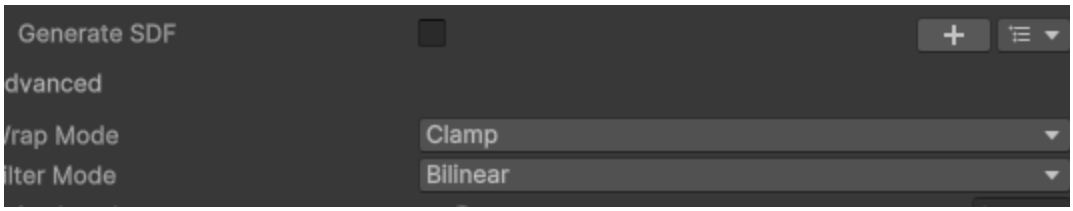
Set import settings

- Modify import settings

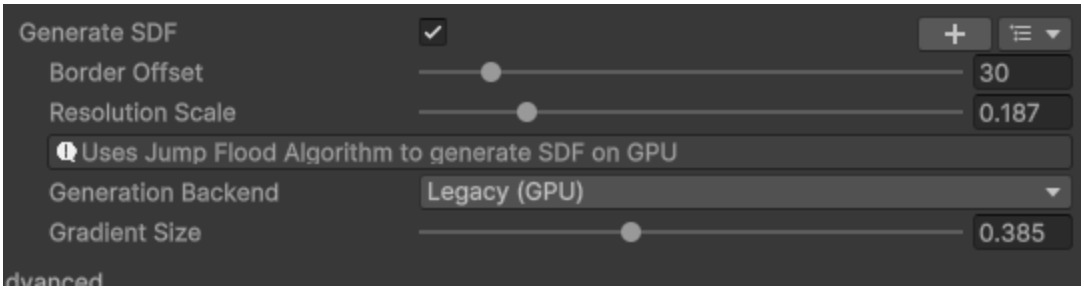


- Press *Set SDF Import Settings*

- See how these import settings gets applied to big-bang.png, and it gets re-imported.



Before



After

Code

```
ic class ImportSDF : SampleBaseAsset

[Header("Import SDF & Change settings")]
[SerializeField] private Texture _texture;  Serializable
[SerializeField] private SDFImportSettings _sdfImportSettings = new();  Serializable

[SampleButton(Name = "Fix texture (Ensure that SDF is imported)")]
private void FixTexture()
{
    Assert.IsNotNull(_texture, message:"_texture != null");
    SDFEditorUtil.FixTextures(new [] { _texture });
}

[SampleButton]
private void SetSDFImportSettings()
{
    Assert.IsNotNull(_texture, message:"_texture != null");
    SDFEditorUtil.SetImportSettings(textures:new [] { _texture }, _sdfImportSettings);
}
```

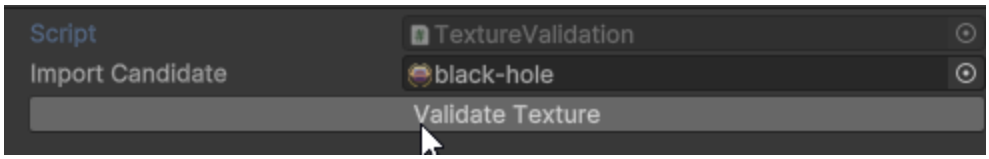
Both functions used if pretty simple, just pass persistent texture. And in case of `SetImportSettings` also provide sdf import settings.

`SetImportSettings` is much more versatile then `FixTextures`, while second one is basically a wrapper around first.

Texture Validation

Texture validation is basically core

- Press Validate button



- The validation result will be printed in console

Code

Validating texture is pretty simple

```
var shouldImportSDF = SDFEditorUtil.ShouldImportSDF(texturePath,
                out var importer,
                out var settings);
```

- in
  - texture path
- return
  - is sdf should be generated for this texture?

> 💡 Its not necessarily is generated, since SDFImporter might not yet processed it, or there is import errors. This function only determines is the texture settings is fit to generate sdf.

- out
  - loaded texture importer
  - sdf import settings loaded from importer

Why texture validation is useful?

When writing editor scripts for SDFAsset its might be useful to known is sdf generated for particular texture without loading it. `SDFEditorUtil.ShouldImportSDF()` will not load the texture, just validate its import settings, texture type, etc, this can be used as pre-processing step, to filter only sdf-valid textures.