

Distort FX Manual

FAQ

Q: There's a missing script on the camera in the Example scene.

A: This is the unity post processing stack behavior that is referenced by the camera, if you don't have the Post Processing Stack imported it will show up as a missing script. Simply importing the Post Processing Stack into the project will fix this.

Q: The examples look weird (colors are too strong, no bloom, etc.)

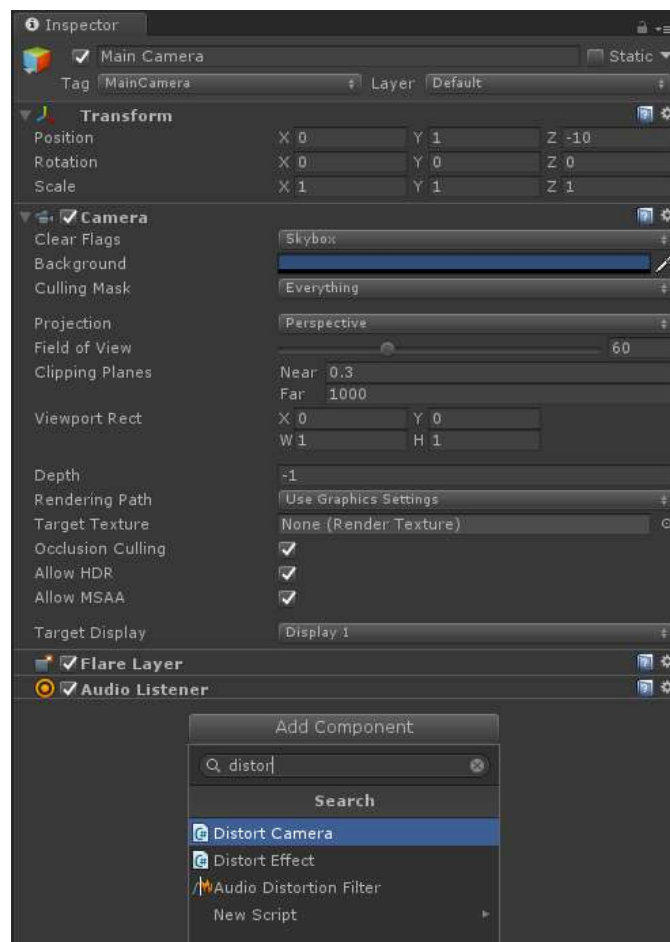
A: Make sure you have the Unity Post Processing Stack from the Asset Store installed. Also make sure you have set the color space to "Linear" in the Player Settings.

Q: Does the Distort Effect go before or after the Post Processing Behaviour component from Unitys Post Processing Stack?

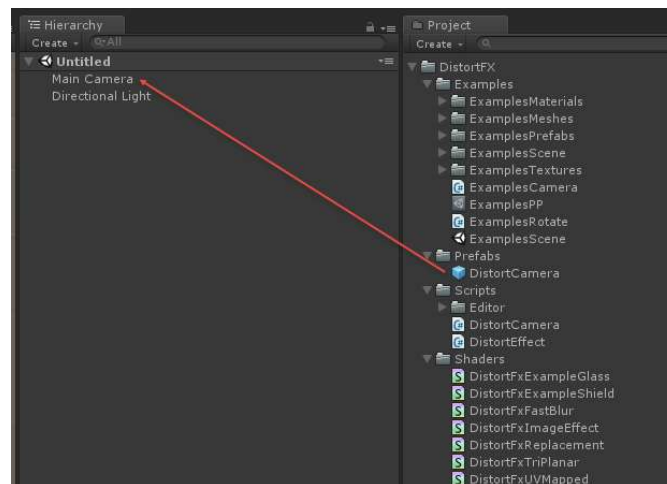
A: This is personal preference; the end result of the distortion is slightly different depending on if you put the distort effect before or after the Post Processing Behaviour

Camera Setup

Getting started with Distort FX is simple, start by adding the "Distort Effect" component to your main camera.



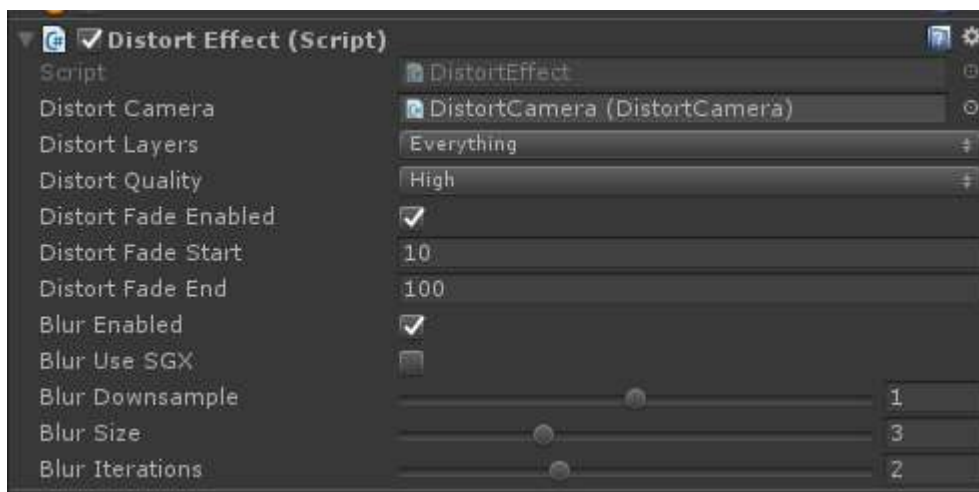
After that drag the “DistortCamera” prefab from Assets/DistortFX/Prefabs as a child to the camera



This is all that’s required to get Distort FX working.

Effect Settings

Let’s go through the settings on the Distort Effect.



- Distort Camera – The camera that is used to render the distortion volumes and textures, this should be found automatically by Distort FX and doesn’t need to be set manually.
- Distort Layers – This is the layer mask which defines which layers the objects which are to apply distortion to the world are in, for general purpose use “Everything” should be fine – but you might want to set this to specific layers to increase performance in complex scenes.
- Distort Quality – Controls the resolution of the distortion texture and thus the final quality of the distort, in general for everything except mobile platforms leaving this as “High” is fine.
- Distort Fade Enabled – Enables/Disables the global fade value of the distort, the “Start” and “End” values control in which range the fade takes place. The fading is used to fade out distortion over distance from the camera so that objects which are far away don’t apply as much distortion.
- Blur Enabled – Enables/Disables the ability for distortion volumes to blur what they distort, this is not recommended for mobile as it requires an extra blur pass on the main render target of the camera.

Object Setup

To add a distortion volume to the world, simply create a new material in Unity and select one of the three pre-made shaders that comes with Distort FX and then assign it to the object you want the distortion to be applied with.

- Distort Fx/Tri-Planar Mesh – intended for distortion volumes which benefits from tri-planar mapping, for example spheres or other meshes which don't neatly unwrap for use with repeating textures.
- Distort Fx/UV Mesh – intended for volumes which has a well-defined UV unwrap which allows the texture to scroll over it without seams.
- Distort Fx/UV Plane – intended for planar meshes like particles, trails or quads.

These three shaders are simply “pass through” and don't render anything into the world itself, they are only used for adding distortion volumes.

For a lot of use cases like shockwaves and heat distortion this works well, but for other cases it's easier to hook a custom shader into the distortion process, see the next section for information on how to set that up.

Adding Support To Custom Shaders

A lot of uses cases are easier handled by making the shader that's rendering the mesh in the world also perform the distortion itself. The “Shield” and “Glass” shaders which come with Distort FX are good examples of this.

Distort FX allows you to easily hook into the distortion process with any custom shader by adding properties and a tag to the shader source. To hook into either the Tri-Planar Mesh or UV Mesh distortion add the following shader properties:

```
_DistortTexture("Distort Texture", 2D) = "bump" {}  
[DistortSettingsProperty(Mesh)] _DistortSettings("Settings", Vector) = (10, 0, 1, 1)
```

And then add a new tag named “Distort” with either “TriPlanar” or “UVMesh” as its value:

```
Tags {  
    "Queue" = "Transparent"  
    "IgnoreProjector" = "True"  
    "RenderType" = "Transparent"  
    "PreviewType" = "Sphere"  
    "Distort" = "UVMesh"  
}
```

For hooking into the UV Plane distortion process add the following properties to the shader

```
_DistortTexture ("Distort Texture", 2D) = "bump" {}  
_DistortTextureMask ("Distort Texture Mask", 2D) = "white" {}  
[DistortSettingsProperty(Plane)] _DistortSettings ("Settings", Vector) = (10, 0, 0, 0)
```

And set the “Distort” tag to “UVPlane”:

```
Tags {  
    "Queue" = "Transparent"  
    "IgnoreProjector" = "True"  
    "RenderType" = "Transparent"  
    "Distort" = "UVPlane"  
}
```

Adding Custom Distort Modes

If you're an advanced user you might want to be add your own distortion mode instead of the three main ones: TriPlanar, UVMesh and UVPlane.

This can be done by editing the Assets/DistortFX/Shaders/DistortFxReplacement.shader file, a full tutorial on how to create a custom shader pass is out of scope of this documentation, but the result of the fragment shader color output should be the following:

R+G: Distortion Amount

B: Reserved for future use

A: Blur Amount