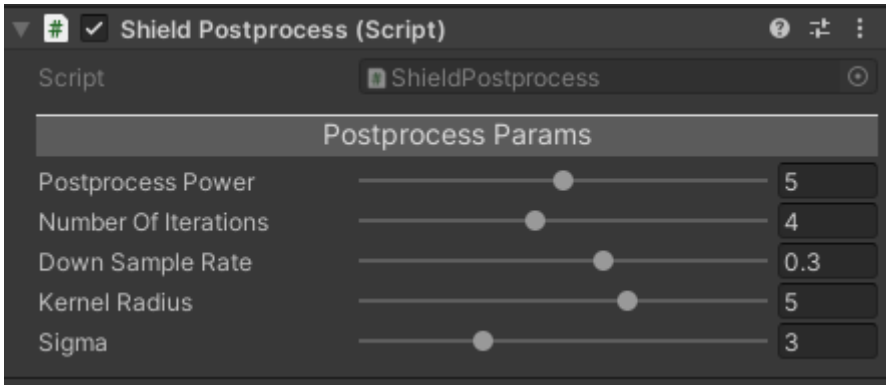
Found a bug, need a feature or help with implementation. Contact at:
fx.valley.contact@gmail.com

Installation & Setup
v2.2.2

1. **BuiltIn Render Pipeline**

- Asset is ready to use after importing.

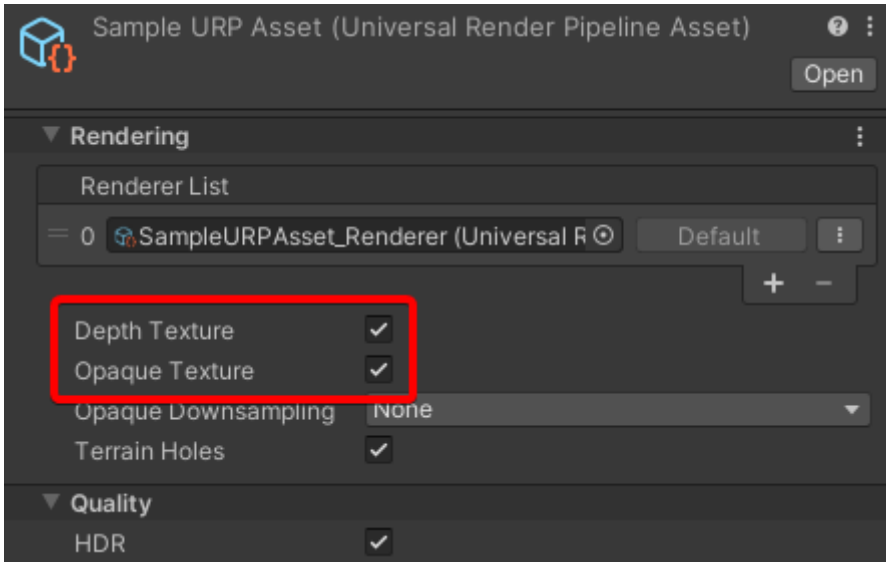
- To make everything work on your scene add *FXV/ShieldEffect/Scripts/ShieldPostprocess.cs* script to the camera:



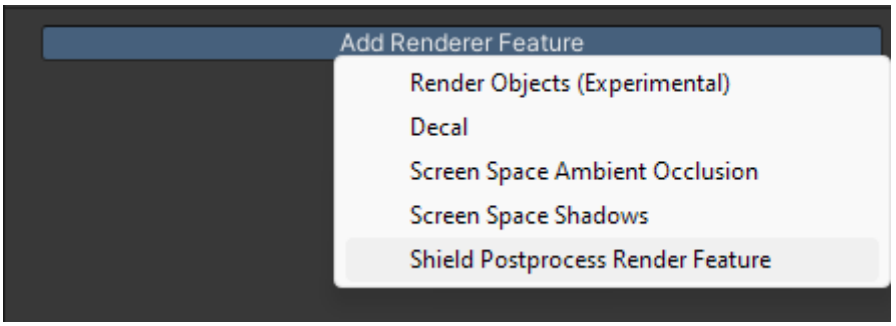
2. **Universal Render Pipeline**

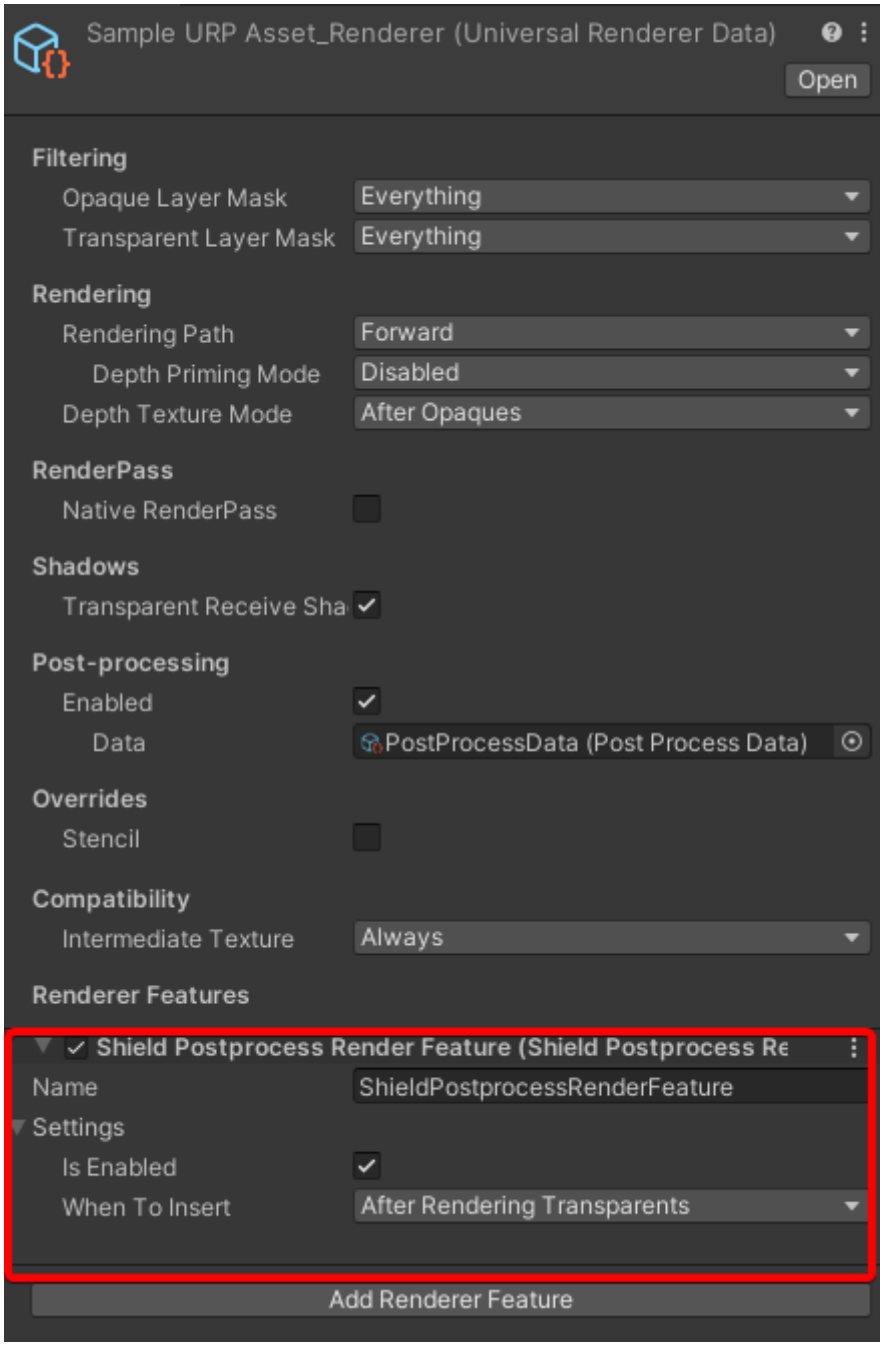
- After importing you will need to install the URP package *FXV/ShieldEffect/InstallURP.unitypackage* that contains all required assets.

- For all features to work properly depth texture and opaque texture needs to be enabled in *Universal Render Pipeline Asset:*

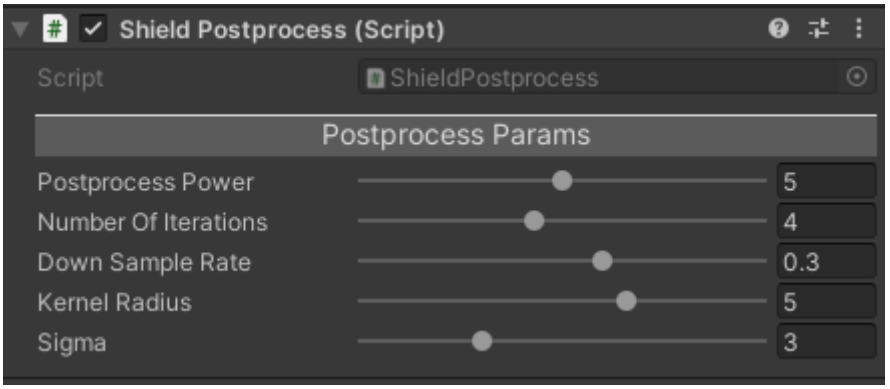


- To enable postprocess add *ShieldPostprocessRenderFeature* to Universal Render Pipeline Renderer. Set it to render After Transparents.





- To make everything work on your scene add *FXV/ShieldEffect/Scripts/ShieldPostprocess.cs* script to the camera:



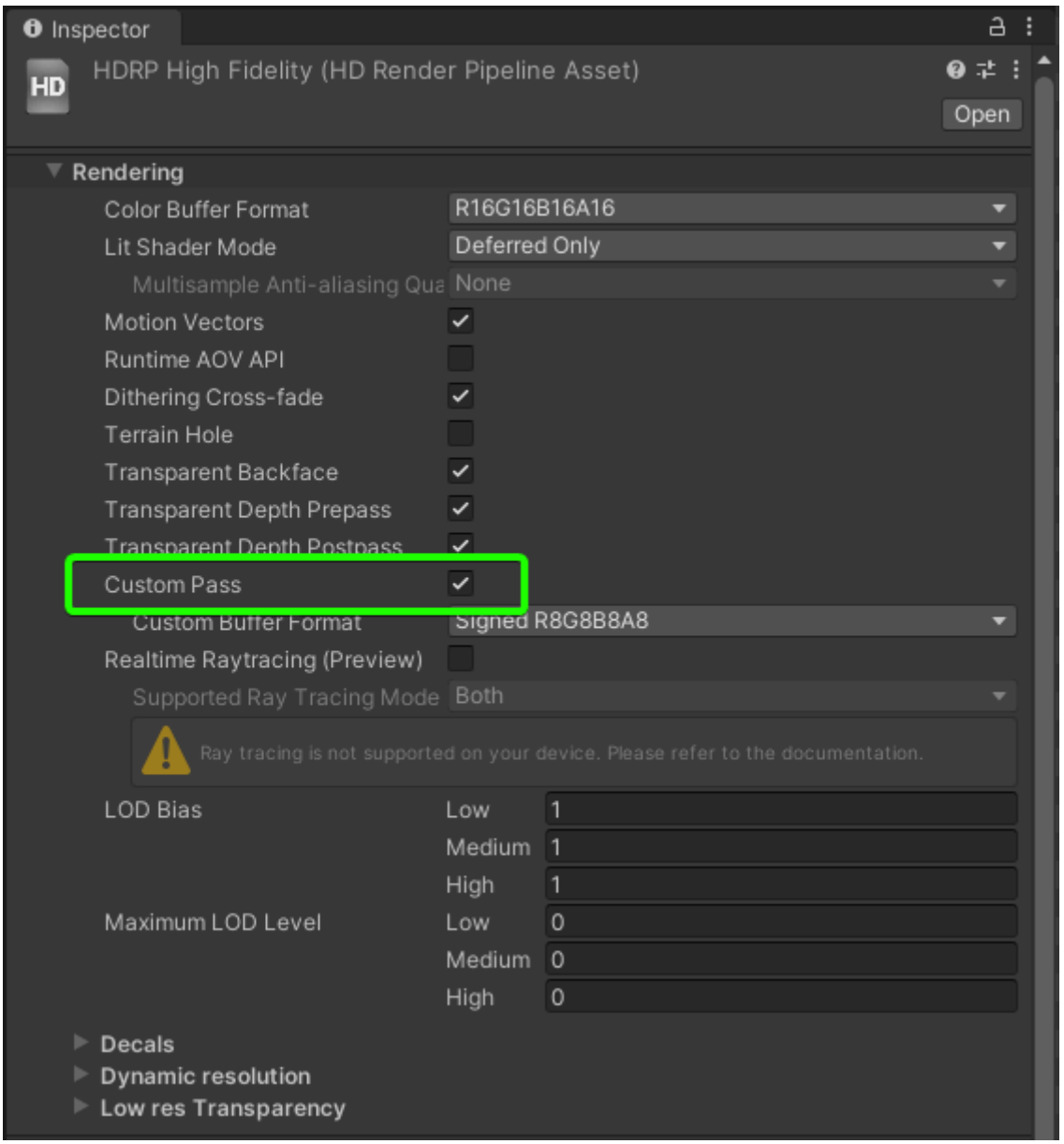
-

3. High Definition Render Pipeline

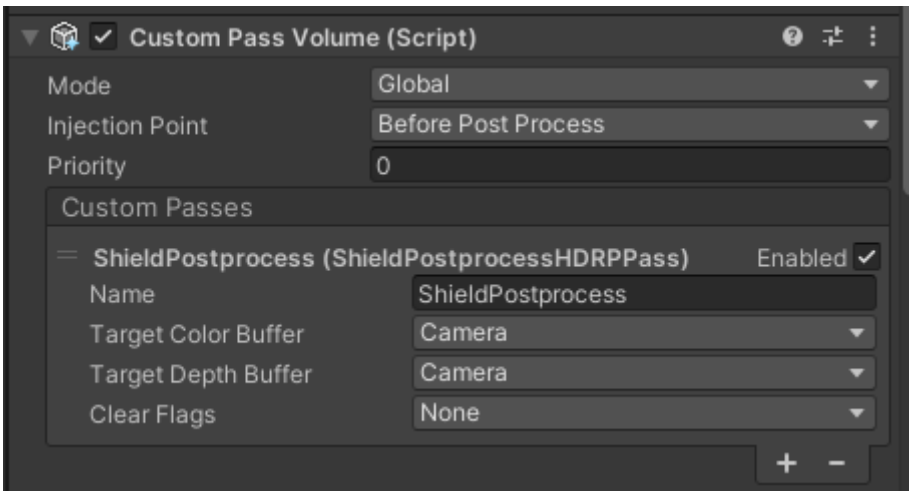
- After importing you will need to install the HDRP package *FXV/ShieldEffect/InstallHDRP.unitypackage* that contains all required assets.

- To make everything work on your scene add *FXV/ShieldEffect/Scripts/ShieldPostprocess.cs* script to the camera.

- Make sure your HDRP asset have Custom Pass Enabled (this is also required for hit effects on SkinnedMeshes to work)



- Also add *Custom Pass Volume* with *Shield Postprocess HDRP Pass* configured as on screenshot below*:*



-

4. ***Updates***

When updating assets to the new version remember to unpack specific Render Pipeline package once again.

Troubleshooting

If after proper initialization and setup you are experiencing any problems, please contact support by email or via discord channel providing as much information as you can:

- Unity version
- Platform (PC, Android, iOS, console etc.)
- is it VR project
- Graphics Api (DirectX (version), OpenGL (version) OpenGL ES (version), Vulkan, Metal etc.)
- Render pipeline (BuiltIn, URP, HDRP)
- Render path (Forward, Forward+, Deferred)
- Camera mode - orthographic, perspective
- Where does problem occur - edit mode, play mode, deployed build
- GPU or console/mobile device model
- does the problem also occur when asset is imported to empty Unity project
- errors/warnings from the console if there are any related to the asset
- editor/player log file if the problem is related to deploying and running a build - logs location are described here (`Editor.log, Player.log`): https://docs.unity3d.com/Manual/log-files.html

Known problems & solutions

1. Unity 6 Compatibility Mode

If you are getting warning in console:

Execute is not implemented, the pass FXV.ShieldHitsRenderPass won't be executed in the current render loop.
Execute is not implemented, the pass FXV.ShieldPostprocessRenderPass won't be executed in the current render loop.

It means that compatibility mode is ON in your project.

You can just turn OFF this mode in Preferences to solve this:

https://docs.unity3d.com/6000.0/Documentation/Manual/urp/compatibility-mode.html

But if you want to keep compatibility mode ON then you have to edit 3 scripts. Go to:

FXV\ShieldEffect\URP\Scripts\ShieldPostprocessRenderFeature.cs
FXV\ShieldEffect\URP\Scripts\ShieldPostprocessRenderPass.cs
FXV\ShieldEffect\URP\Scripts\ShieldHitsRenderPass.cs

Change all line occurrences of **UNITY_6000_0_OR_NEWER** to **UNITY_6000_0_OR_NEWER_off**

Package contents

- Each render pipeline has its own demo scene:

  BuiltIn:
  FXV/ShieldEffect/Demo/Demo.unity

  URP:
  FXV/ShieldEffect/URP/Demo_URP.unity

  HDRP:
  FXV/ShieldEffect/HDRP/Demo_HDRP.unity

- There are multiple prefabs and materials that you can use right away or use as a base for tweaking own variations:

  FXV/ShieldEffect/Prefabs

- All the prefabs and materials are presented on the demo scene. When in play mode you can change type using a leaver in the center and turn on/off each shield using a leaver near the shield object.

- For hit effect you can shoot using the left mouse button. Hold left ctrl for a bigger bullet to test different hit effect sizes.

- There is special scene with setup for custom rim texture baking when using custom meshes:

  FXV/ShieldEffect/Demo/CustomTextureBaking.unity

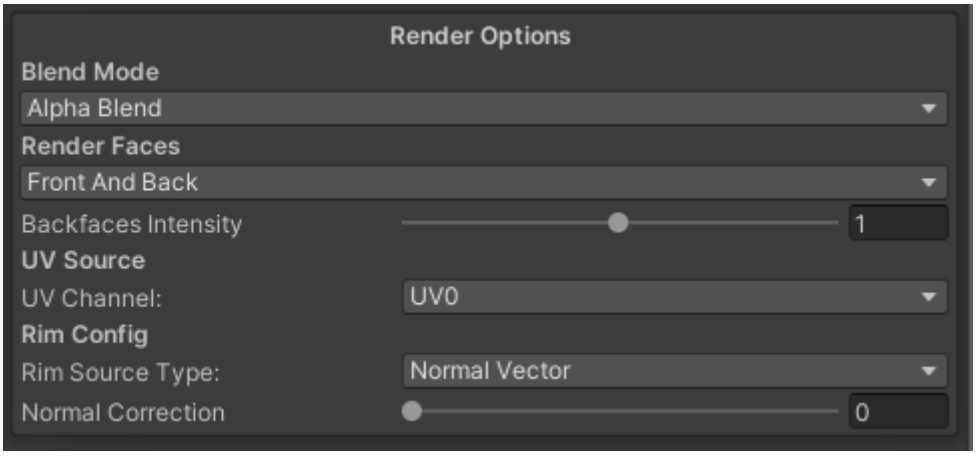
Using Shield Effect

1. **Setting up new object**

- Create material and assign to it FXV/FXVShieldEffect shader
- Create new object with mesh renderer and set the newly created material in its Renderer. Multiple renders in the hierarchy are also supported.
- Add Shield.cs script to the object. Shield component will detect all renderers in the hierarchy that are using the FXV/FXVShieldEffect shader and handle them.

2. **Tweaking material properties**

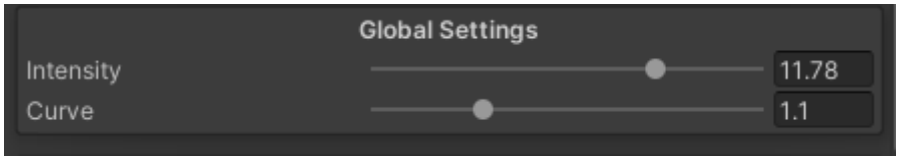
- **Rendering options**



Blend Mode	select blending mode alpha or additive
Render Faces	option to render back faces - in situations if rendering inside of the shield is required
Backface Intensity	Available only when Render Faces are set to **Front And Back.** Specify the color intensity of backfaces.
UV Source	define which UV channel to use for texture mapping of the effect. **UV0**, **UV1** are mesh uv maps. **Generated UV** is a procedurally generated UV - this is experimental and

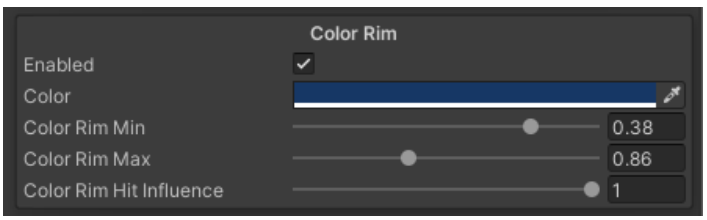
	might give unexpected results.
Rim Config	what is the source for rim calculation: - **Normal** - rim will be aligned to view space normal. - **Texture** - rim will be calculated from provided texture (R - rim color, G - texture rim, B- pattern rim)
Normal Correction	Available only for **Rim Config: Normal.** Use it to blend between mesh normals and a calculated normal based on local vertex position. This might improve visuals for complex meshes.

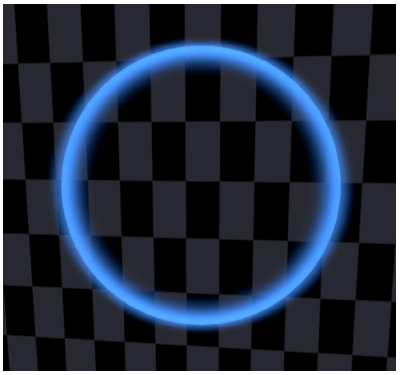
- **Global Settings - settings for final tweaking of effect intensity**



Intensity	final multiplier of output color
Curve	shape of the fade curve

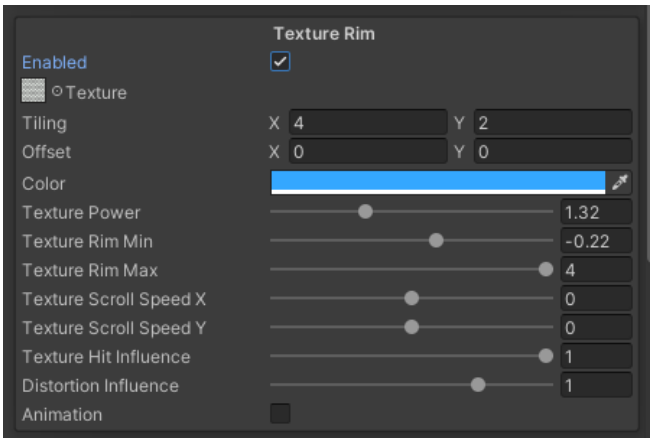
- **Color Rim - adds solid color rim effect**

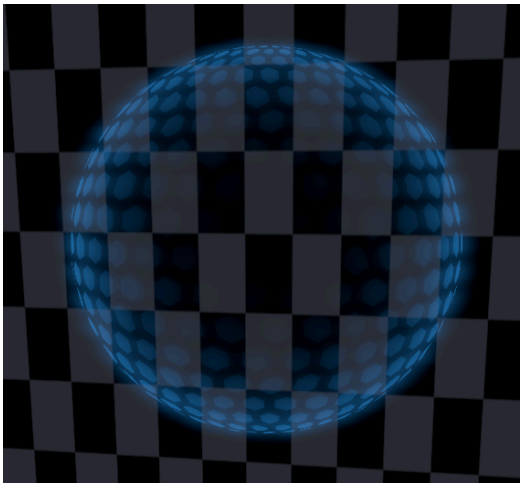


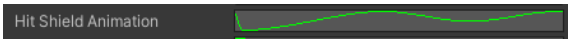


Color Rim Min/Max	size of the rim effect
Color Rim Hit Influence	how much the Hit Shield Animation curve from Shield script component affects it.

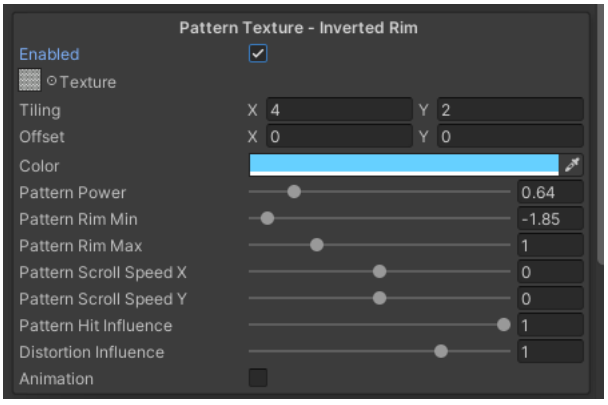
- **Texture Rim - add rim effect with texture**

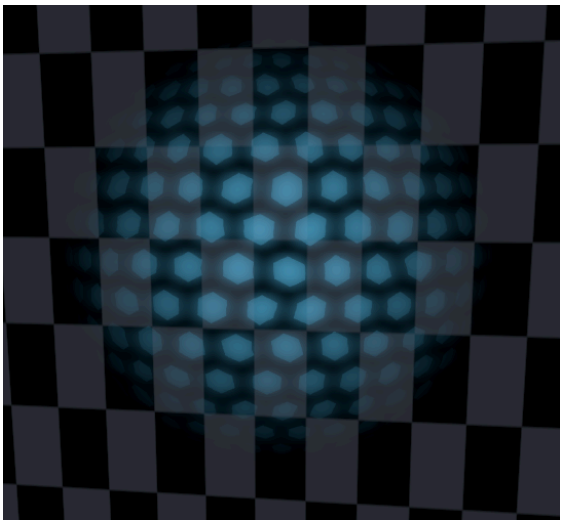


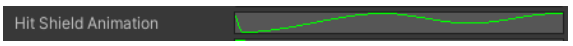


Texture Rim Min/Max	size of rim effect
Texture Power	how much to multiply final color
Texture Scroll Speed X/Y	speed of texture position scroll over time for X/Y axis
Texture Hit Influence	how much the Hit Shield Animation curve from Shield script component affects it 
Animation	Sin fade animation based on texture value
Distortion Influence	(available when Distortion is on) how much distortion affects this texture

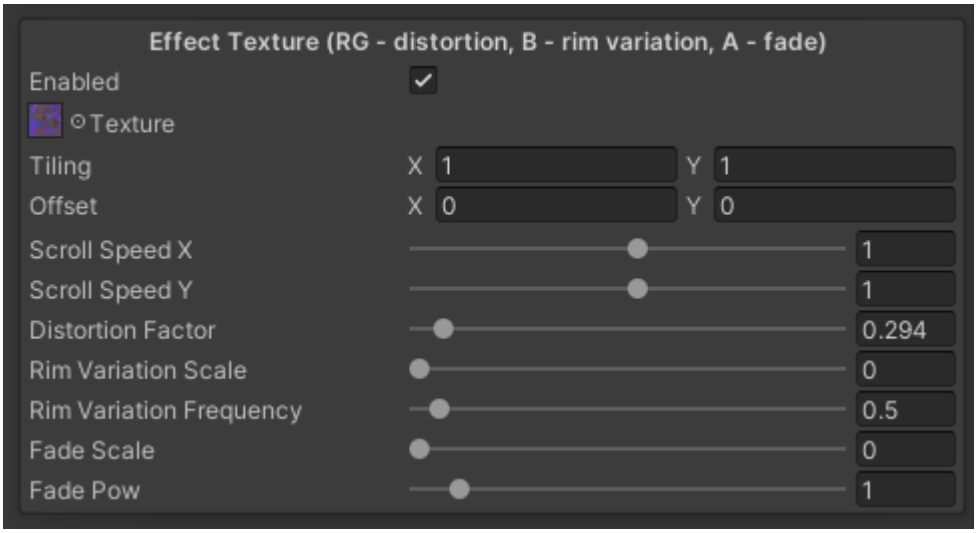
- **Pattern Texture - add inverted rim effect with texture so it will add nicely with Texture and Color Rim**



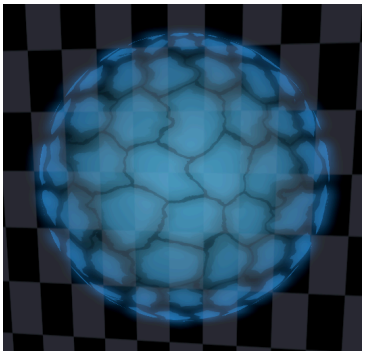
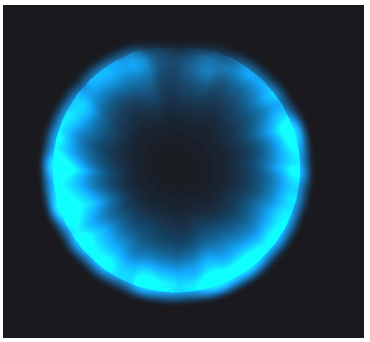
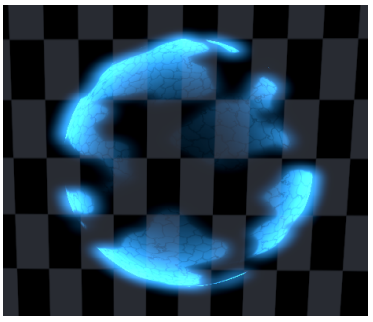


Pattern Rim Min/Max	size of rim effect (inverted)
Pattern Power	how much to multiply final color
Pattern Scroll Speed X/Y	speed of texture position scroll over time for X/Y axis
Pattern Hit Influence	how much the Hit Shield Animation curve from Shield script component affects it 
Animation	Sin fade animation based on texture value
Distortion Influence	(available when Distortion is on) how much distortion affects this texture

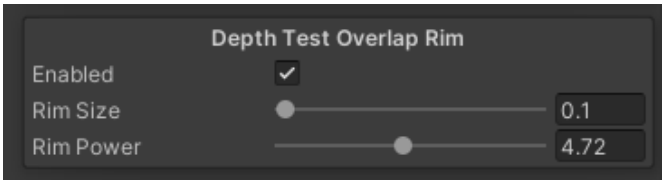
- **Effect texture - special texture with Red and Green chanell for distortion, Blue channel for rim effect variation, Alpha channel for fade out effect**



Speed X/Y	how fast texture will move over time
Distortion Factor	how much to distort by red and green channel of this texture (globally) 0 - off

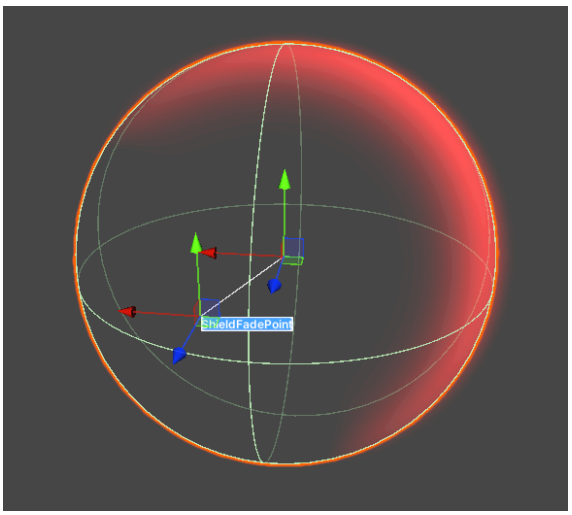
	
Rim Variation Scale	how much rim effect will be scaled by blue channel of this texture, 0 - off 
Rim Variation Frequency	specify how quickly rim variation changes
Fade Scale	how much to fade the shield effect with alpha channel of this texture, 0 - off 
Fade Pow	change the edge/curve of the fade effect

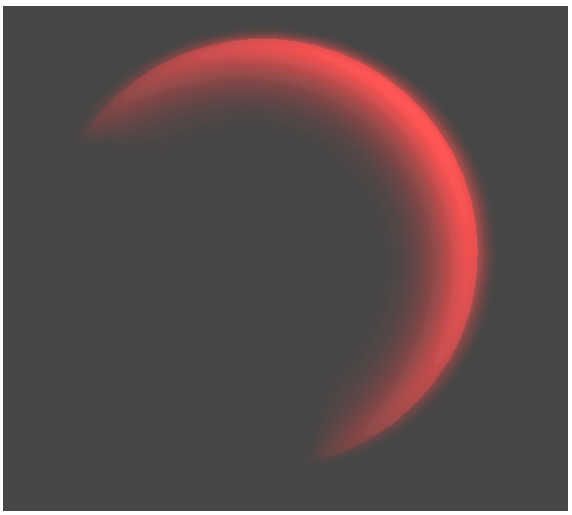
- **Depth Test Overlap Rim - enable color rim for intersections with geometry - this one needs to have Color Rim enabled**

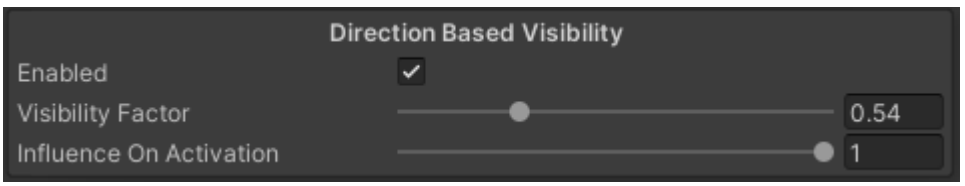


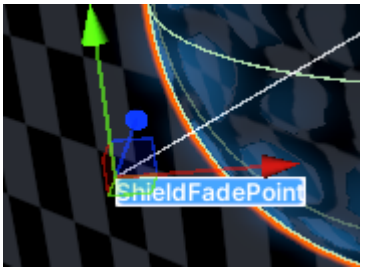
Rim Size	set size of the effect based on depth
Rim Power	how much to multiply color from Color Rim

- **Direction Based Visibility - adds nice fade effect based on ShieldFadePoint position**

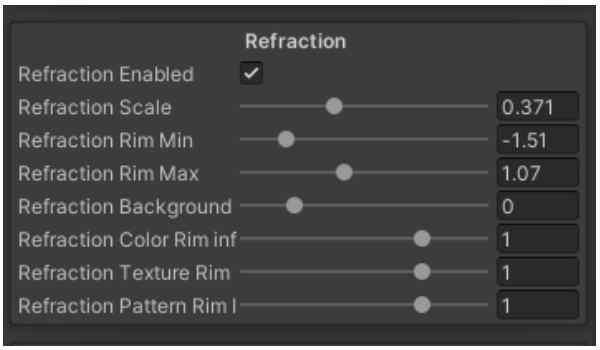


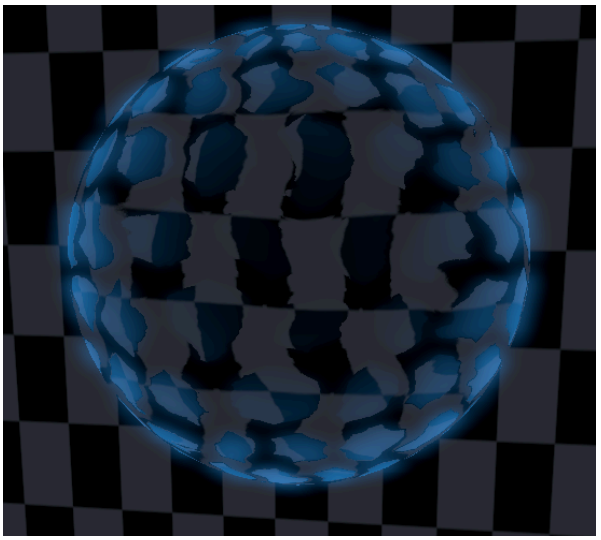




ShieldFadePoint (scene view)	position on scene - for fade direction - this point is visible in scene view when shield object is selected 
Visibility Factor	parameter for fade size
Influence On Activation	how much directional visibility will affect activation effect

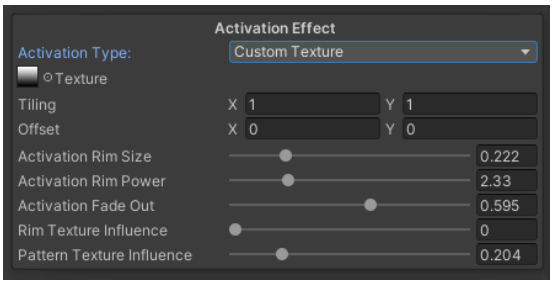
- **Refraction - distorts everything behind the shield**

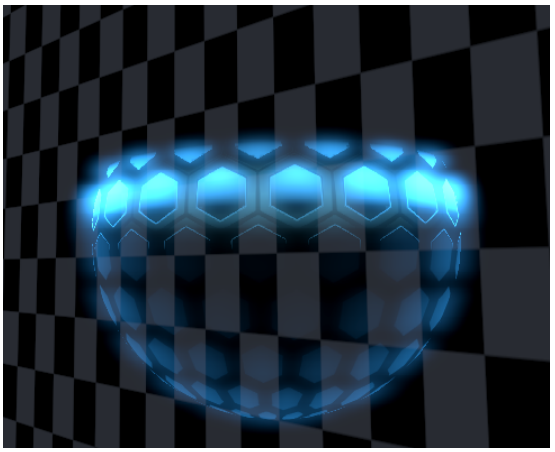




Refraction scale	how much to distort
Refraction Rim Min/Max	scales refraction strength based on rim effect parameters
Refraction Background Exposure	lightens or darkens the background
Refraction Color/Texture/Pattern Influence	specify how much each layer affects the distortion effect

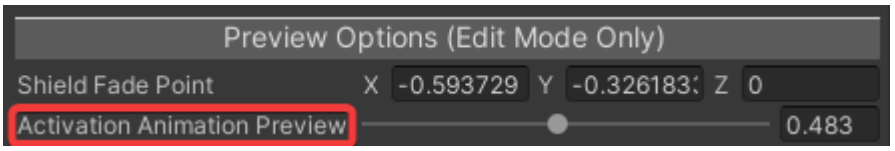
- **Activation Effect - animation effect when shield is turned on/off**



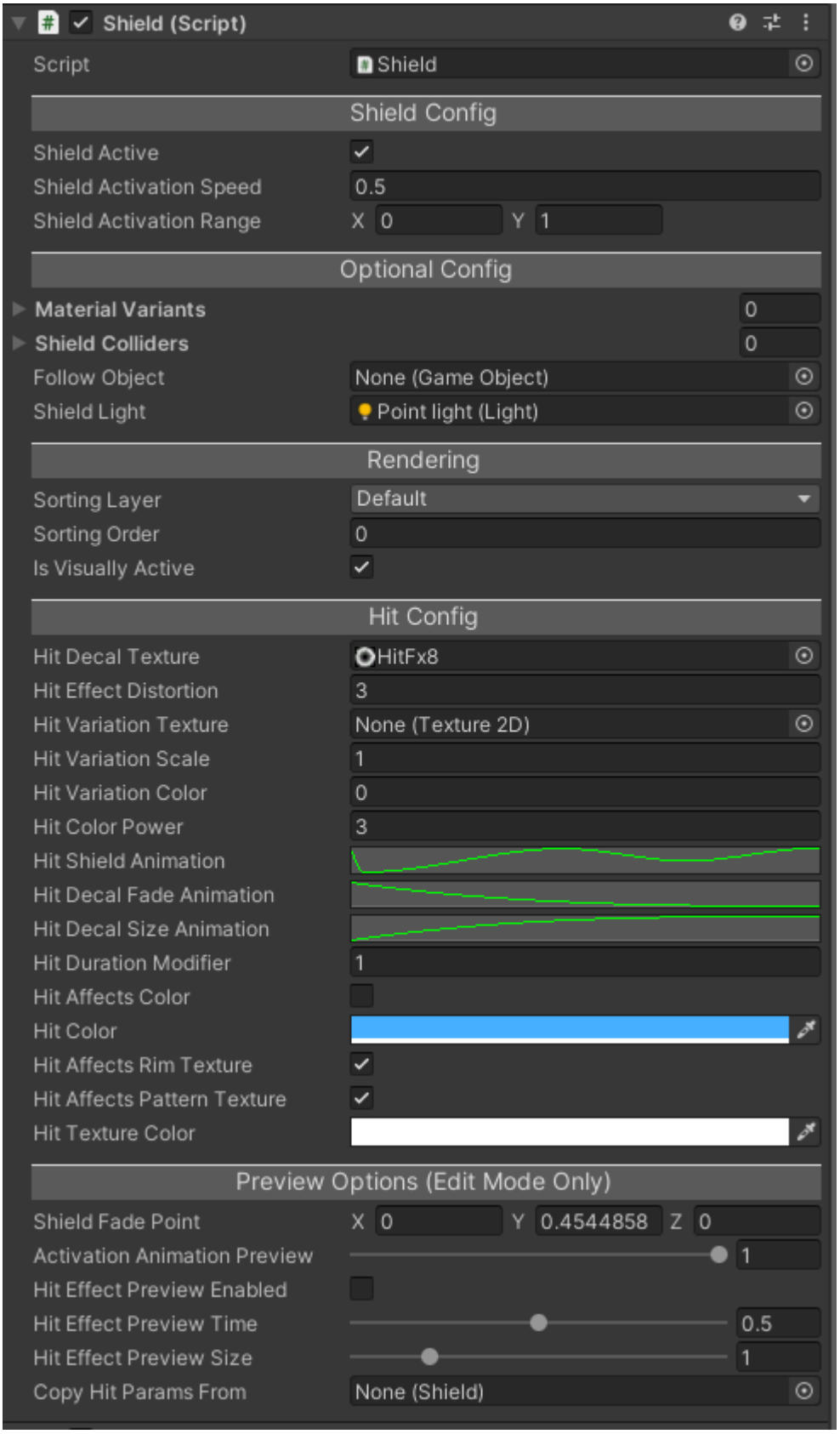


Activation Type	Select the source for activation animation - this will be compared to activation time for rim effect calculation: - **FinalColor** - final material color - **FinalColor_and_UVx** - final material color mixed with UV.x value - **FinalColor_and_UVy** - final material color mixed with UV.y value - **CustomTexture** - custom texture mapped onto object - **UVx** - uv.x value - **UVy** - uv.y value - **FinalColor_and_PosY -** final material color mixed with local position Y
Activation Rim Size	size of the rim effect
Activation Rim Power	how much rim color will be multiplied
Activation Fade Out	when (in time) to start fade out of rim effect 0 - no fadeout
Rim Texture Influence	how much rim texture will have effect on activation time
Pattern Texture Influence	how much pattern texture will have effect on activation time

For preview and tweaking purposes there is slider for activation animation preview in Shield component:



3. Configuring Shield Component



- Shield Config

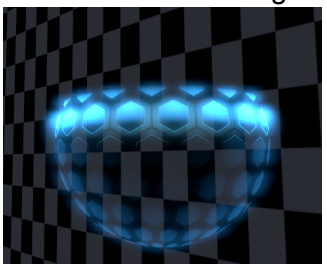
Shield Active	is shield active when the scene is started in play mode
Shield Activation Speed	how fast the activation animation will be
Shield Activation Range	specify activation min max values so that the shield is fully invisible at time 0, and fully visible at time 1. Use the Activation Animation Preview slider to tweak this.
Material Variants	Add materials to this list if you want to switch shield materials at runtime (for example to have different shield colors). Use SetMaterialVariant(int index) to change material variant at runtime (can be empty)
Shield Collider	By default Shield component will take collider from it's gameObject. Use this property to specify a collider that is outside of Shield gameObject.
Follow Object	in situation when shield cannot be a child of an object, specify transform here, that the shield should follow (can be empty)
Shield Light	light that the shield should affect when turning on/off (can be empty)
Sorting Layer	Sorting layer for rendering order - useful for 2d
Sorting Order	Order in layer for rendering – useful for 2d
Is Visually Active	Use this option to hide shield effect rendering while having hit effects active

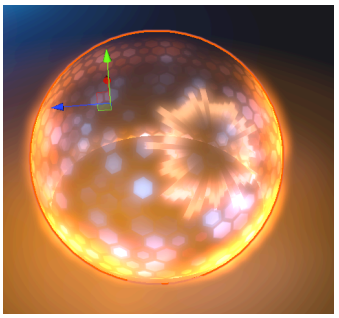
- Hit Config

Hit Decal Texture	texture for decal pattern, if empty hit will have attenuation calculated by it's radius
Hit Effect Distortion	how much hit will distort the shield
Hit Variation Texture	optionally specify texture that will affect hit size and color dynamically over time. See **/FXV/ShieldEffect/Textures/Hits/HitVariance***.png** - for example variance textures
Hit Variation Scale	how much hit size will be affected by variation texture
Hit Variation Color	how much hit color intensity will be affected by variation texture

Hit Color Power	intensity of hit decal final color
Hit Shield Animation	curve for global shield color influence when shield is hit. this might be scaled for each layer individually by using **Color Rim/ Texture/Pattern Hit Influence** sliders in material
Hit Decal Fade Animation	the curve for fade animation over hit effect life time
Hit Decal Size Animation	the curve for size animation over hit effect life time. Affects **hitScale** from OnHit(Vector3 hitPos, float hitScale, float hitDuration)
Hit Duration modifier	modifier for hit lifetime, use when for example. Affects **hitDuration** from OnHit(Vector3 hitPos, float hitScale, float hitDuration)
Hit Affects Color/Texture/Pattern	set if hit effect should be mixed together with each layer of shield effect material
Hit Color	color of hit decal that is used when Hit Affects Color is enabled
Hit Texture Color	color of hit decal that is used when Hit Affects Texture/Pattern is enabled

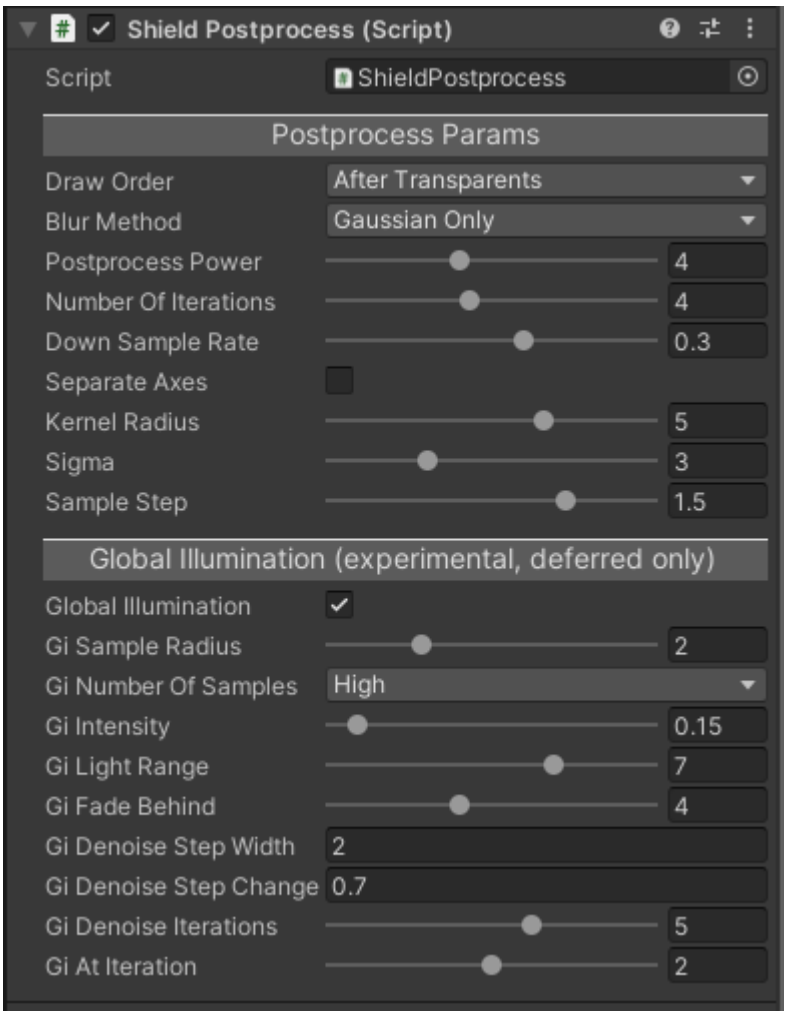
- Edit mode options

Shield Fade Point	manually set ShieldFadePoint position 
Activation Animation Preview	use this to visualize and tweak activation effect on scene during edit mode 

Hit Effect Preview Enabled	check this to enable hit effect preview 
Hit Effect Preview Time	when hit preview is enabled use this slider to preview hit over time
Hit Effect Preview Size	when hit preview is enabled use this slider to preview hit with different sizes
Copy Hit Params From	Drag and drop different Shield object to quickly copy Hit properties.

4. Configuring Postprocess

Postprocess effects add the most for the final quality of the effect. Add ShieldPostprocess script to the camera on your scene. See **Installation & Setup** for detailed installation info for each render pipeline.



Postprocess power	Final intensity of postprocess effect.
Draw Order	Built In only - specify when to draw psotprocess
Blur Method	change blurring method used - affects blur shape
Number of Iterations	Number of iterations for posprocess effect. This will have an effect on blur quality and size. Use it with connection with other parameters. **Higher = slower**.
Down Sample Rate	How much to scale down the texture in each iteration. The higher the value the faster the overall effect will be. As there will be less pixels to compute. **Higher = faster**
Separate Axes	Enable this to control parameters for horizontal and vertical axes separately. When enabled additional parameters for Kernel Radius, Sigma and Sample Step will

	appear for the vertical axis.
Kernel Radius	Radius in pixels samples for post process blur effect. The higher the value the more blur will be applied in each iteration. **Higher = slowe**r.
Sigma	Offset coefficient for Kernel sampling offset. Does not affect performance, but will have an effect on blur shape.
Sample Step	Sampling offset scale - increase for larger blur, decrease for smaller blur.

Global Illumination is an experimental feature that is still work in progress. It adds nice illumination from
the shield to surrounding geometry. It uses some expensive computations so use it with caution to specific hardware. There are number of parameters to tweak for quality/performance. **Global Illumination** is only available with **Deferred** rendering path.

Gi Sample Radius	Sampling radius for the geometry that can be affected by lights. The higher the radius the more objects will be affected but artifacts might appear.
Gi Number of Samples	Number of Sample that will be taken inside **Gi Sample Radius**. Increase for better quality. **Higher = slower**.
Gi Intensity	Intensity of the final effect.
Gi Light Range	Max range for light attenuation function. If it's increased also **Gi Sample Radius** should be increased.
Gi Fade Behind	Fade out gi directly behind shield effect. 0.0 no fade
Gi Denoise Step Width	Offset factor for denoise function sampling
Gi Denoise Step Change	Chow much Gi Denoise Step Width will change with each denoise iteration
Gi Denoise Iterations	Number of iterations for denoise shader to run. This parameter should be tweaked for best effect together with Step Width and Step Change
Gi At Iteration	This parameter will specify at which downsample iteration from postprocess effect to calculate GI. The higher the value

	the lower will be the resolution of source texture - and performance will be improved. This parameter will affect performance the most. **Higher = faster**

5. Scripting

class: Shield.cs

Core class of shield effect instance, handles enabling disabling and hit effects.

public void SetMaterial(Material newMat)

- Set new material at runtime, material should use FXShield shader

public void SetRimColor(Color c)

- *Set rim color at runtime (will create material instance use SetMaterialVariant for better performance)*

public void SetTextureRimColor(Color c)

- *Set texture rim color at runtime (will create material instance use SetMaterialVariant for better performance)*

public void SetPatternColor(Color c)

- *Set pattern texture color at runtime (will create material instance use SetMaterialVariant for better performance)*

public void SetHitColor(Color c)

- *Set hit color at runtime*

public bool GetIsShieldActive()

- *Return true if shield is in active state*

public bool GetIsDuringActivationAnim()

- *Return true if activation animation is in progress*

public void SetShieldVisuallyActive(bool active)

- *Set shield effect rendering on/off but leaves collider intact*

public void SetShieldActive(bool active, bool animated = true)

- *Turns shield on/off both visually and physically. Use animated = true fr activation animation, or animated = false for instant state change without animation*

public void SetShieldEffectFadePointPosition(Vector3 localPos)

- *Sets ShieldFadePoint position at runtime*

public void OnHit(Vector3 hitPos, Vector3 hitNormal, float hitScale, float hitDuration)

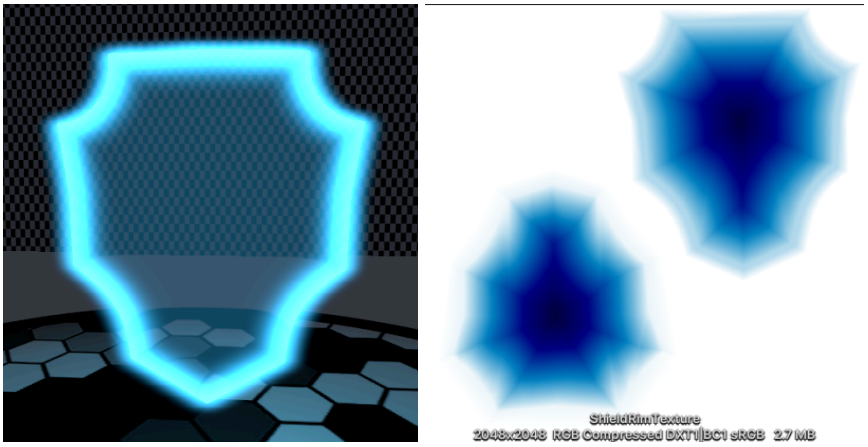
- *Spawn hit effect at position and normal at specific hit point, with specified scale and duration. Parameters will be multiplied by curves defined in Shield component hit config.*

public void SetMaterialVariant(int index)

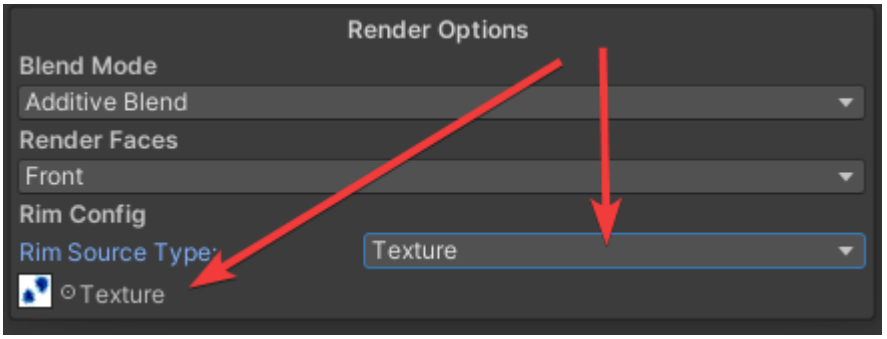
- *Use this method to switch between material variants at runtime - Material Variants list in Shield component have to be filled for this to work.*

6. Baking Custom Rim Texture

When using custom meshes, calculating rim effect with normal might not give good results. Here comes the **ShieldRimTextureBake** component. It bakes distance from the edge to texture from visible given BakeOrigin point:



This texture should be used together with Texture - Rim Source Type:



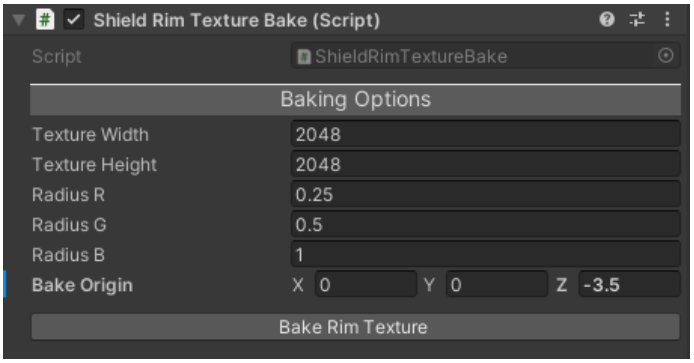
Each texture channel will be used for different rim value calculation:

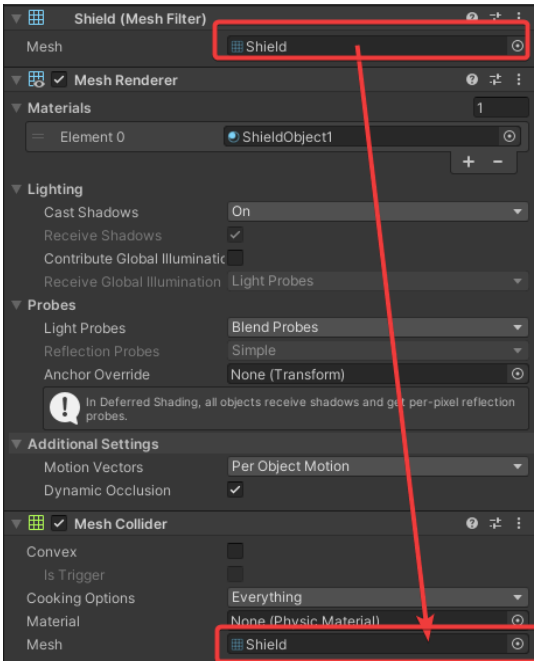
Red - color rim

Green - texture rim

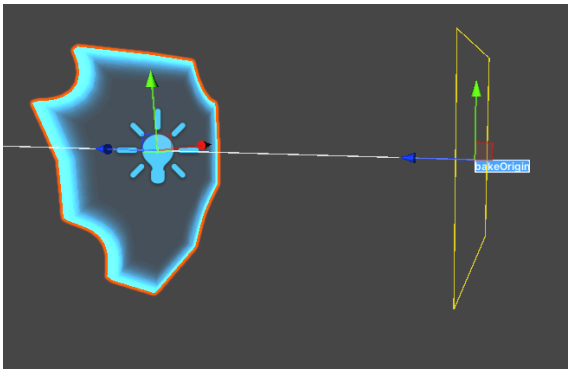
Blue - pattern rim

To bake rim texture add ShieldRimTextureBake component to the object , and set the **mesh collider** with the same mesh as the rendered object:





Texture Width/Height	size of the texture (big sizes might take time to bake)
Radius R/G/B	rim size for each layer
Bake Origin	point from which the rim edge is viewed for baking (probably somewhere in front of mesh)

Look at FXV/ShieldEffect/Demo/CustomTextureBaking.unity for sample setup.

It's good to do baking on an empty scene so that other colliders will not interfere with the process.

Changelog

2.2.2

- improved compatibility with different mobile devices
- bug fixes to BuiltIn forward rendering

2.2.1

- fixed SetShieldVisuallyActive method, added “Is Visually Active” checkbox in Shield component inspector
- Added setup check in HDRP if Custom Pass is enabled
- fixed Material Variants option with postprocess

2.2.0

- SkinnedMeshRenderer support.
- New materials and prefabs for SkinnedMeshRenderer demo, Demo scenes update
- Possibility to set Shield effect material as second material - useful for simplification when using as force field effect rendered on existing opaque mesh.
- Unity 6 and RenderGraph for URP support.
- Added “Hit Texture Color” parameter for more flexibility in hit color customization.
- Fixed “Hit Color Power” parameter in Shield component. **NOTE that this might have an impact on how your hits look after updating if the value of this parameter was different than 1.0. Change this parameter back to 1.0 to make everything render as before the update.**
- Improvements to "Generated UV" option in material
- Performance optimizations when rendering hit effects.

2.1.5

- Shield components now support multiple renderers in object hierarchy.
- Hit effects will no longer create GameObjects for each hit to improve performance and reduce resources used.

- Added giFadeBehind to postprocess, to control gi intensity directly behind shield effect
- Added sorting layer and sorting order option to shield component - useful for 2d rendering
- Added texture tiling and offset for rim source texture

2.1.4

- fixed bug with alpha write to rendered buffer for hit effects
- code compatibility improvements to avoid conflicts when importing asset

2.1.3

- fixed issue with shaders not rendering correctly in builds
- fixed bug with postprocess downsample rate not calculated correctly for multiple steps in Universal Render Pipeline
- VR support for Universal Render Pipeline
- fixed bug where shield effect appeared pink in build when there was no ShieldPostprocess script added to camera

2.1.2

- Added blur method option to postprocess for different blur shapes
- Fixes and improvements to hit effect rendering. OnHit method now will require hit normal - as mesh normal at specific hit point - this can be retrieved from collision info.
- Fixed Rim Variation parameter - this will now scale the rim based on its direction. Materials that were using this parameter will have to be readjusted.
- Added hit effects preview in Shield component - for easier hit parameters tweaking
- Added Hit Variation Texture that will scale size and color dynamically over hit life time.
- tweaking material parameters of demo materials
- improved demo scenes

2.1.1

- Added Material Variants option to easily switch materials at runtime
- Added possibility to define postprocess parameters on vertical and horizontal axis separately
- Added postprocess render order option for Built In Render Pipeline
- Fixing bugs with shield component enabling and disabling
- Added Shield Collider property as an option to specify collider that is outside shield Game Object
- performance optimizations

2.1.0

- New materials, prefabs and demo objects for shield around spaceship.
- Added experimental Global Illumination effect to postprocess
- Support for Renderers with multiple submeshes
- Hit effect are now included in posptrocess for improved quality
- Shader bugfixes

- Scripting bufixes