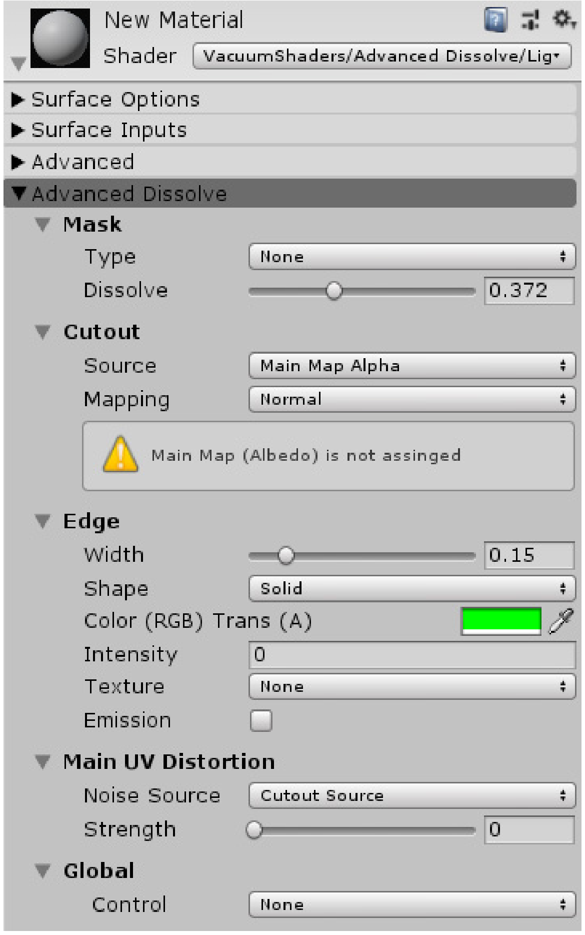
Quick Start

Create simple sphere and assign new material with Advanced Dissolve shader.
No dissolve effect at this stage.



New Material
Shader VacuumShaders/Advanced Dissolve/Lig

▶ Surface Options
▶ Surface Inputs
▶ Advanced
▼ Advanced Dissolve

▼ **Mask**

Type: None
Dissolve: 0.372

▼ **Cutout**

Source: Main Map Alpha
Mapping: Normal

⚠ Main Map (Albedo) is not assinged

▼ **Edge**

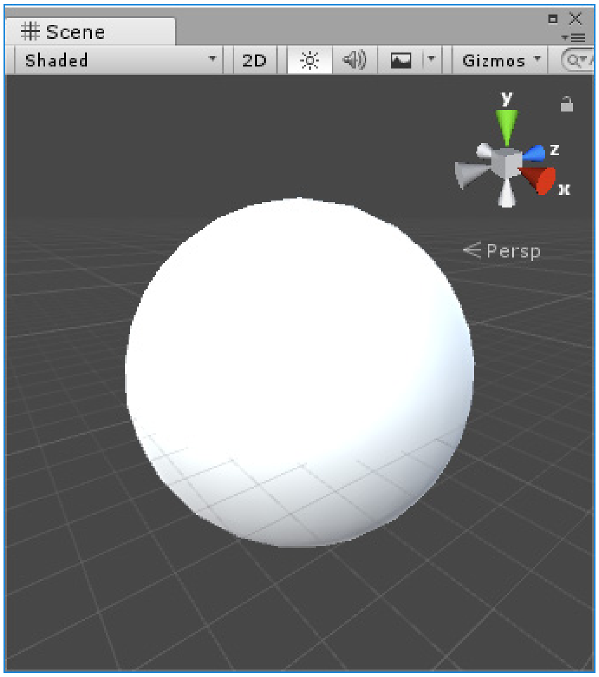
Width: 0.15
Shape: Solid
Color (RGB) Trans (A): 🟩
Intensity: 0
Texture: None
Emission: □

▼ **Main UV Distortion**

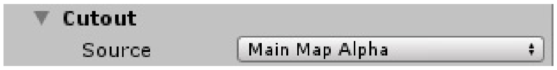
Noise Source: Cutout Source
Strength: 0

▼ **Global**

Control: None



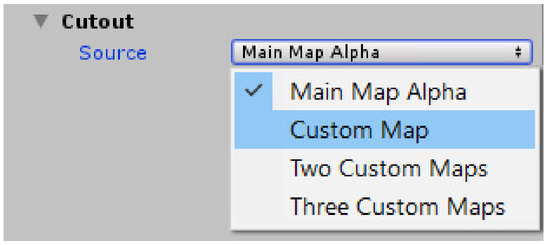
By default shader expects cutout value from the MainMap's alpha channel.



▼ **Cutout**

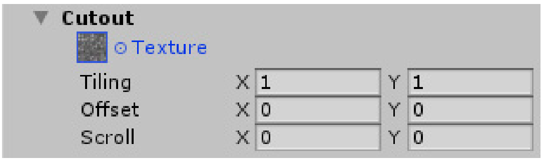
Source: Main Map Alpha

We can assign MainMap texture or tell shader to use **Custom Map**. Choose **Custom Map** option and select one of the included dissolve textures (e.g. D_1.jpg).



▼ **Cutout**

Source: Main Map Alpha

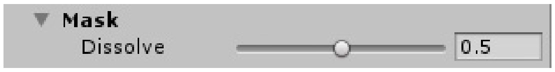


▼ **Cutout**

Texture

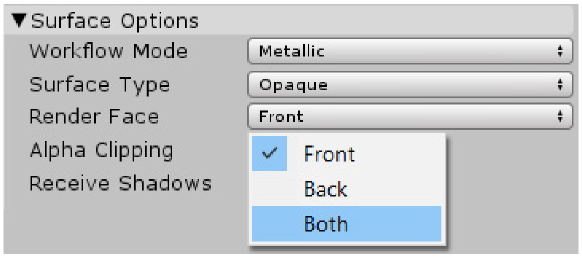
Tiling	X	1	Y	1
Offset	X	0	Y	0
Scroll	X	0	Y	0

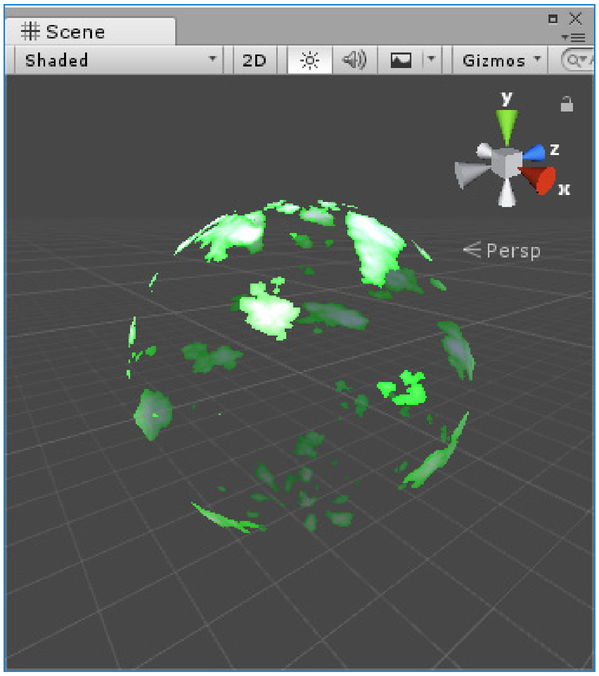
Now changing **Dissolve** parameter will animate cutout effect.


Mask
Dissolve 0.5

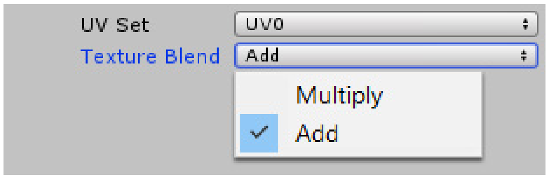
Note: Cutoff value is read from texture's Alpha channel.

Note: Setting **Render Face** option to **Both** will render both sides of a mesh.

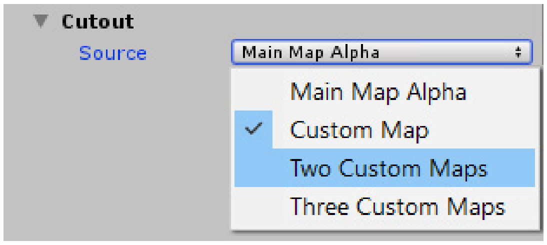

Surface Options
Workflow Mode: Metallic
Surface Type: Opaque
Render Face: Front
Alpha Clipping
Receive Shadows

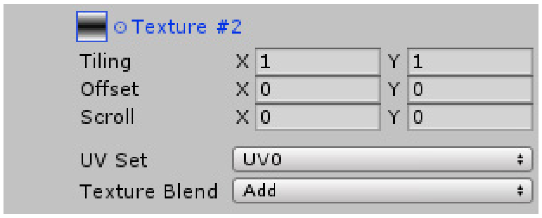


Advanced Dissolve shaders can use three textures simultaneously. They can be mixed by **Multiplying** or **Combining** (**Add**) alpha channel values.

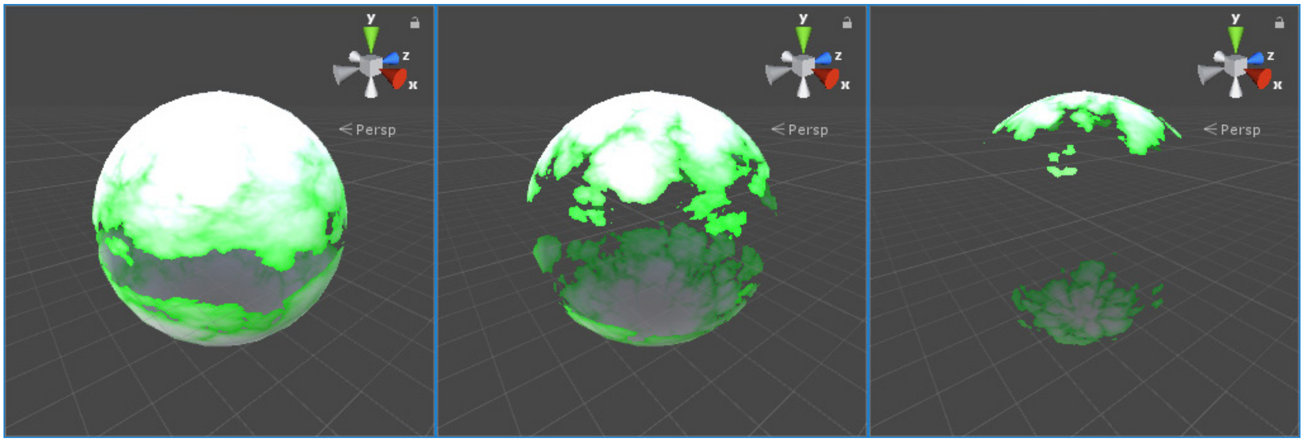

UV Set: UV0
Texture Blend: Add

Choose **Two Custom Maps** option for **Cutout Source** and select grayscale ramp_H1.jpg for the second texture. Leave **Texture Blend** option to be **Add**.

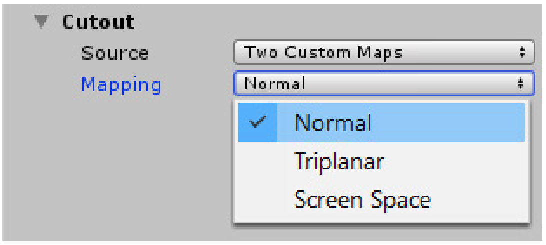

Cutout
Source: Main Map Alpha


Texture #2
Tiling X 1 Y 1
Offset X 0 Y 0
Scroll X 0 Y 0
UV Set: UV0
Texture Blend: Add

Changing **Dissolve** parameter will cut the middle part of a mesh moving toward top and bottom.



Experiment with various textures, blend and **Mapping** modes.

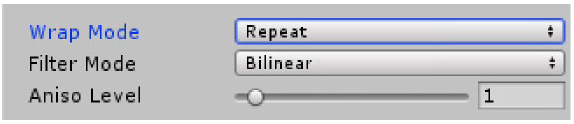


▼ **Cutout**

Source: Two Custom Maps

Mapping: Normal

Note: Cutout source textures can be animated by **Scroll** parameter, but make sure **Wrap Mode** is set to be **Repeat** inside Texture Import settings.

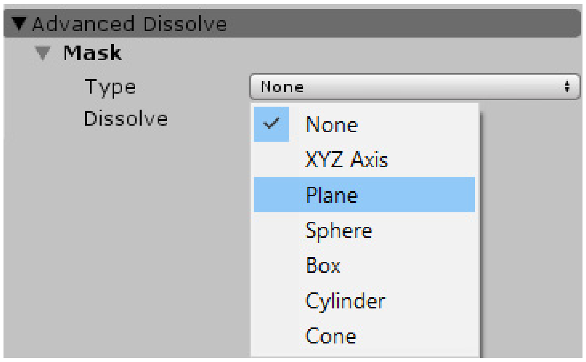


Wrap Mode: Repeat

Filter Mode: Bilinear

Aniso Level: 1

Let's try some dynamic Masks. For example **Plane**.



▼ Advanced Dissolve

▼ **Mask**

Type: None

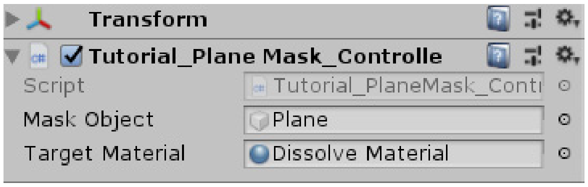
Dissolve

Note: After changing mask type some parts of the mesh may disappear unexpectedly. It is normal and will fix it.

Dynamic masks have no controller parameters inside material editor (except **XYZ Axis**) and they must be updated from custom scripts.

Plane mask needs two parameters to be updated from script **Position** and **Normal**.

Create empty gameobject in the scene and assign **Tutorial_PlaneMask_Controller** script to it. Script has very simple logic, it sends `transform.position` and `transform.up` vector data to the material.



Transform

☑ Tutorial_Plane Mask_Controlle

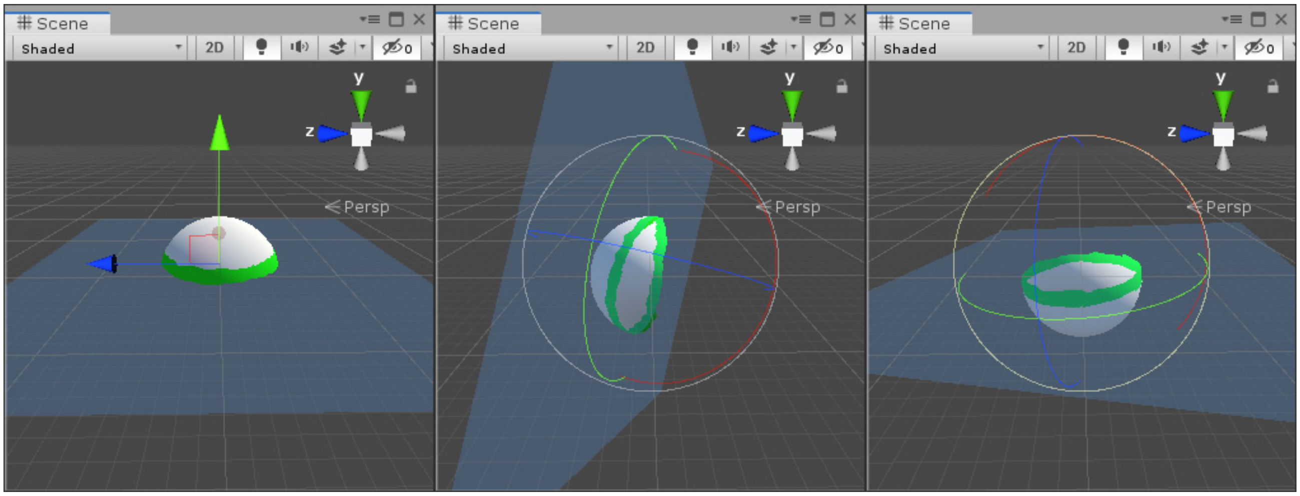
Script Tutorial_PlaneMask_Contr

Mask Object Plane

Target Material Dissolve Material

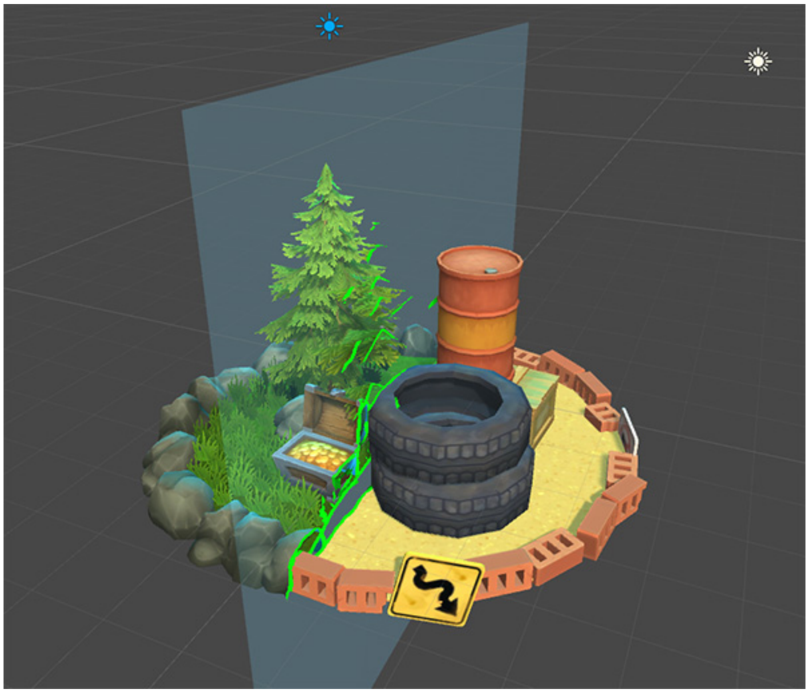
Create gameobject with Plane 3D mesh and select it for the **Mask Object** parameter.
For **Target Material** select our working dissolve material.

Now moving and rotating Plane 3D mesh effects dissolve material.



Note: When using dynamic masks **Cutout Source** textures serve as an edge noise controllers.

Another example: Open **9. Two Worlds** example scene.
Move and rotate **Plane** object. It hides one part of the scene meshes and reveals another one.



Scene mesh materials are already using Advanced Dissolve shaders with Plane Mask enabled. Those materials receive Plane mask info from **Mask Controller (Plane)** object that has attached to identical scripts updating two set of materials, one with **Invert** enabled.

This script is similar to the **Tutorial_PlaneMask_Controller** script, but can update Materials array with 4 mask objects simultaneously.

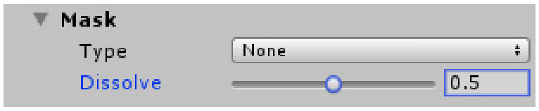
Note: Script sets materials mask count in the **Start** function to 1. Modify it if script updates more mask objects.

Now let's try Sphere mask.
Disable **Mask Controller (Plane)** and **Plane** objects.
Enable **Mask Controller (Sphere)** and **Sphere** objects.
For each material change **Mask Type** from Plane to Sphere inside material editor.
Now moving and scaling of the **Sphere** object will effect materials.

Materials with Sphere mask enabled need mask **Position** and **Radius** to be updated from script.
Mask controller scripts attached to the **Mask Controller (Sphere)** object are used for that.

For all dynamic mask types package includes ready-to-use **MaskController** scripts updating material parameters with mask objects data in required format. Those scripts are used in example and demo scenes.

Mask



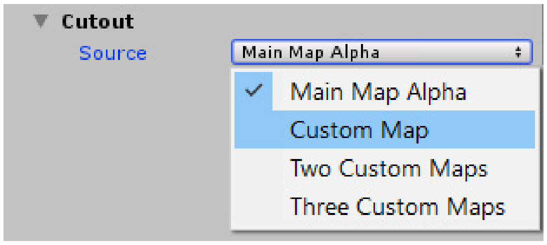
Mask

Type: None

Dissolve: 0.5

Controls dissolve effect amount and shape.

- **None** – Is the default option. **Dissolve** parameter controls cutout amount.
  Note: **Cutout Source** textures alpha channel defines cutout area.

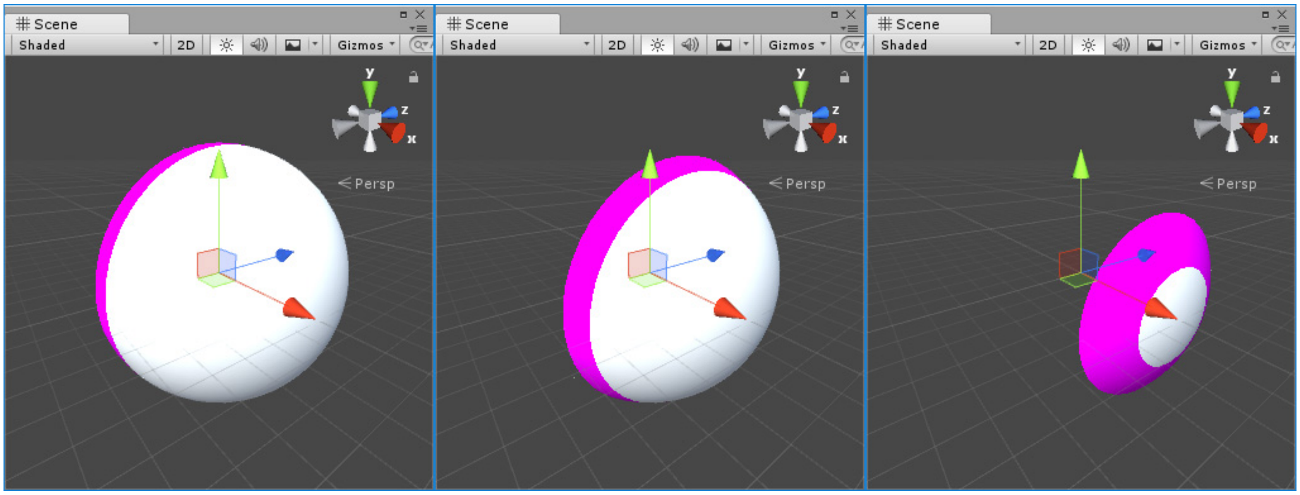


Cutout

Source: Main Map Alpha

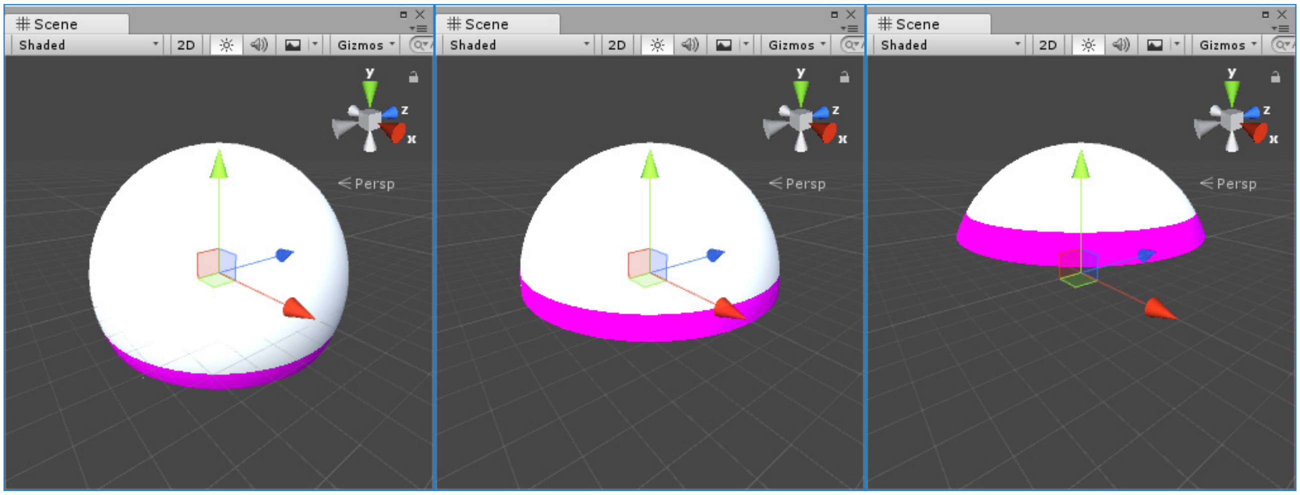
- **XYZ Axis** – Pixel is cutoff based on its World (or Local) X, Y and Z component values.

X Axis example: All pixels which world position X value is less than **Offset** parameter are dissolved.



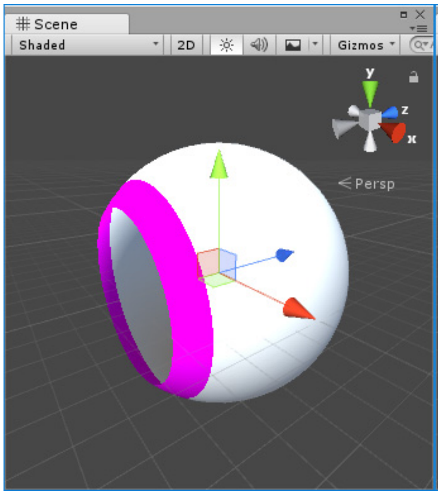
(Offset: 0) (Offset: 0.2) (Offset: 0.4)

Y Axis example:

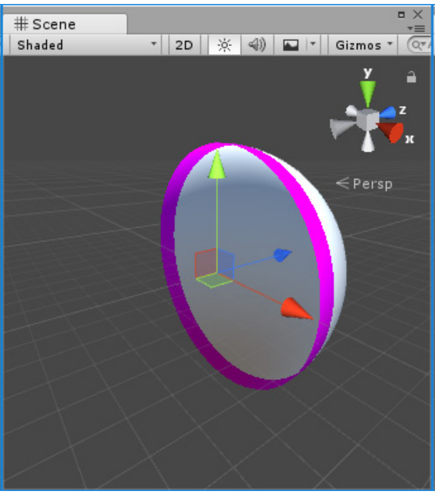


(Offset: -0.2) (Offset: 0) (Offset: 0.2)

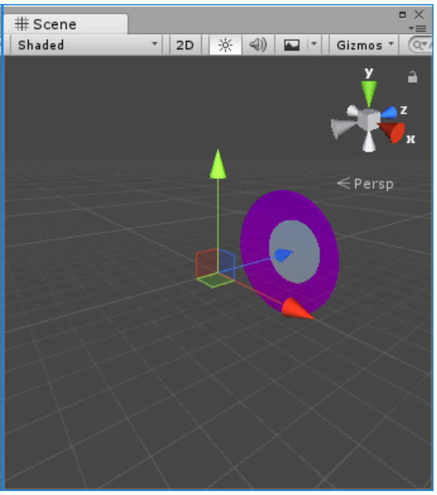
Z Axis example:



(Offset: -0.4)



(Offset: 0)



(Offset: 0.4)

Check **2. Mask (XYZ Axis)** example scene.

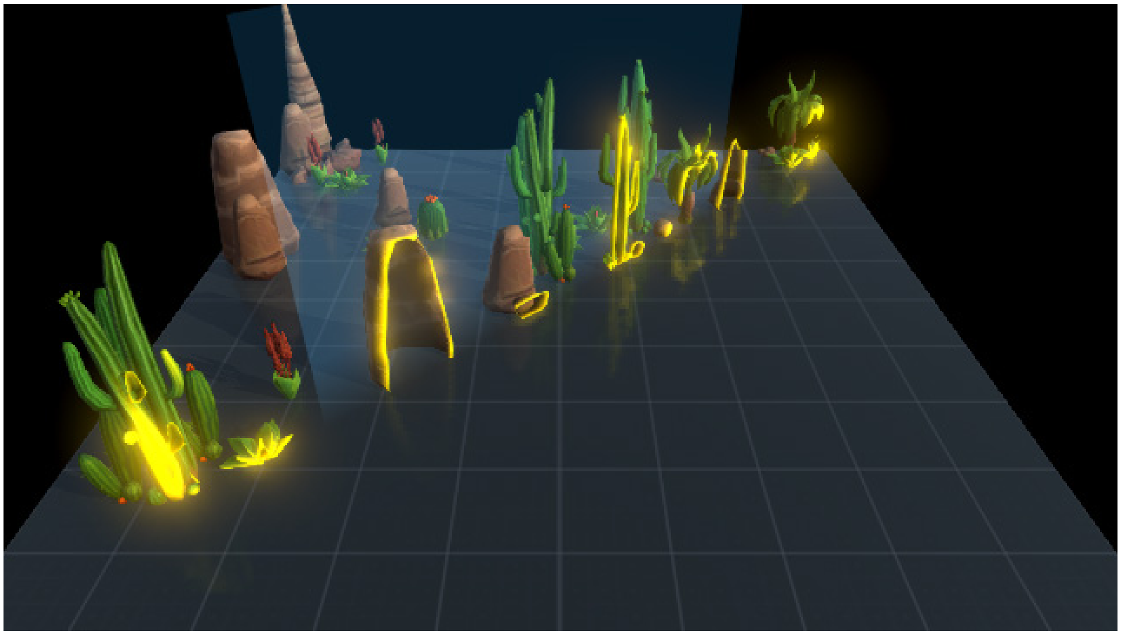
All mask types bellow have no parameter controllers inside material editor and expects them to be updated from custom script.

Simultaneously can be used only 4 mask objects.

For each mask type package includes separate example scenes and controller scripts.

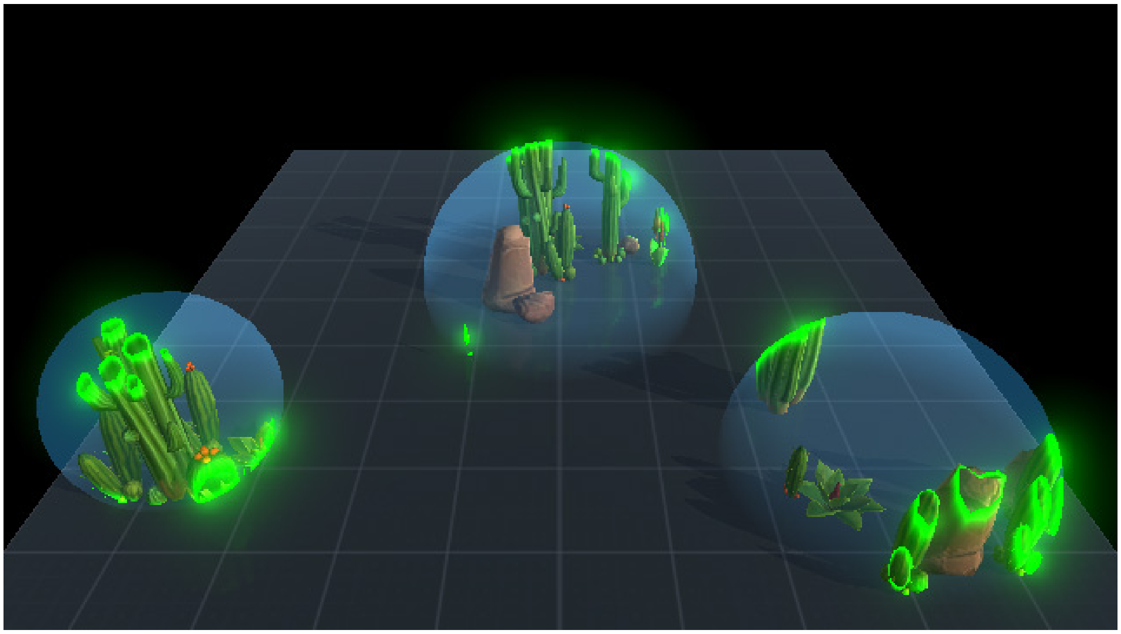
- **Plane** – Pixel visibility is defined by ‘*plane*’ position in the scene and its normal direction. Shader requires ‘*plane*’ **Position** and **Normal** to be updated from script.

  Check **3. Mask (Plane)** example scene and **Controller_Mask_Plane** script used there for updating shader parameters.



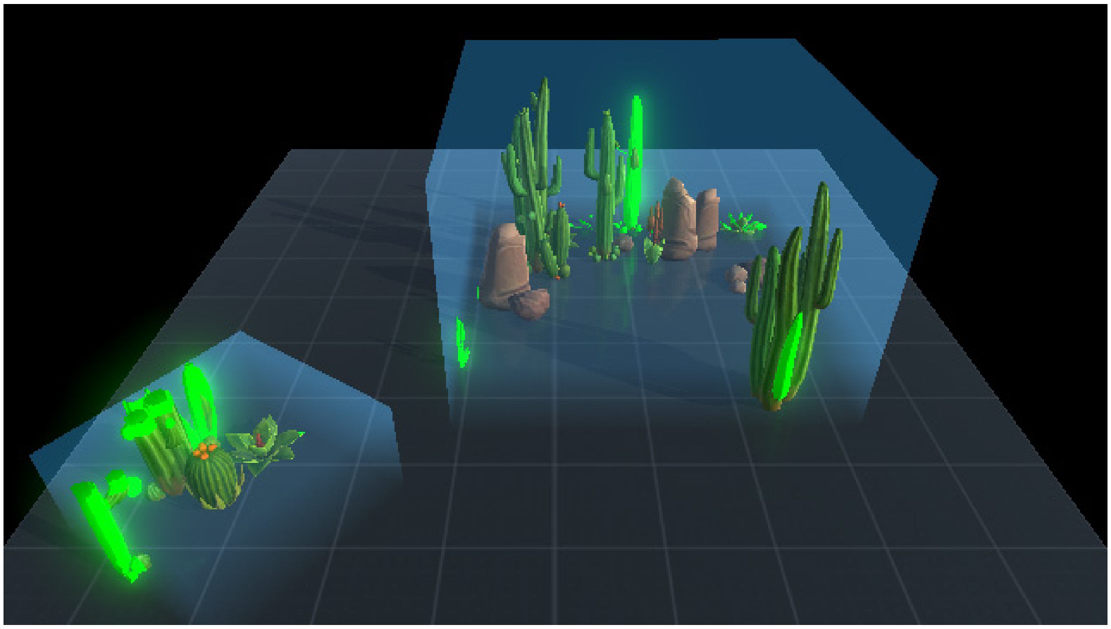
- **Sphere** – Pixels inside *'sphere'* are visible only. Shader needs *'sphere'* **Position** and **Radius** to be updated from script.

  Check **4. Mask (Sphere)** example scene and **Controller_Mask_Sphere** script used there for updating shader parameters.

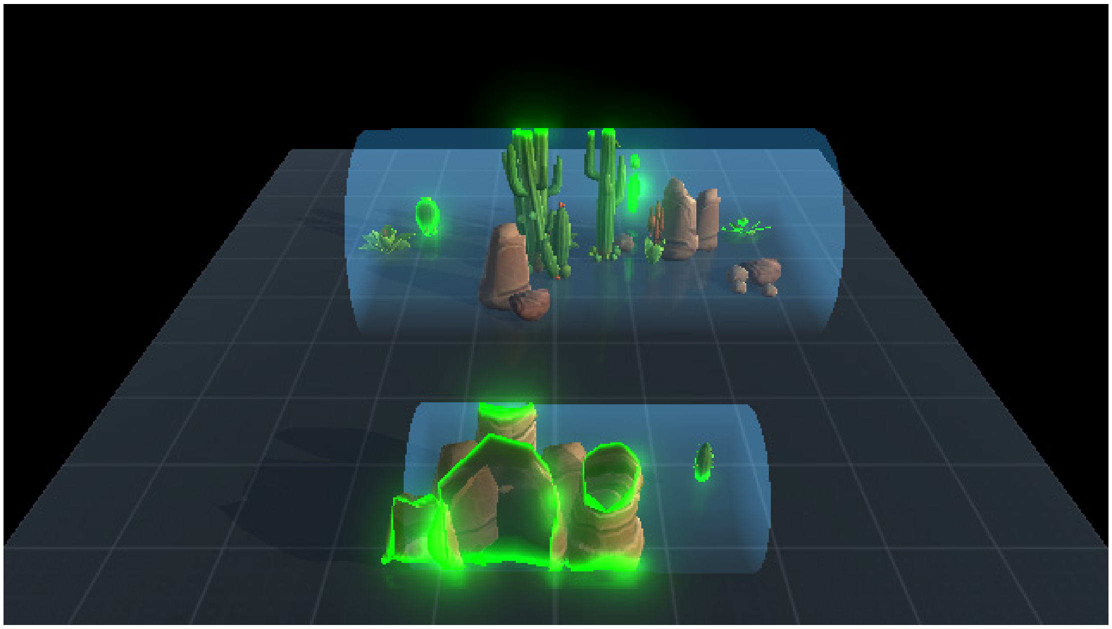
- **Box** – *'Box'* mask object requires calculation of its **Bound Box** and **Transformation Matrix** inside script.

  Check **5. Mask (Box)** example scene and **Controller_Mask_Box** script used there for updating shader parameters.

- **Cylinder** – Requires: **Position**, **Normal Direction**, **Height** and **Radius**, to be updated from script.

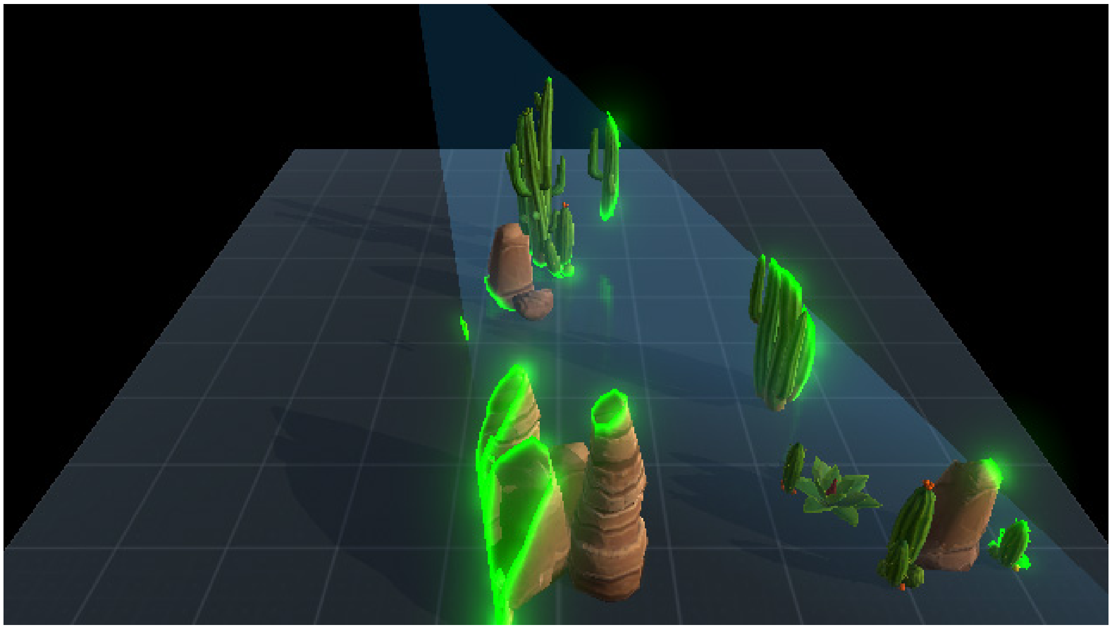
  Check **6. Mask (Cylinder)** example scene and **Controller_Mask_Cylinder** script used there for updating shader parameters.



  Note: example script uses Unity built-in cylinder mesh and converts its transform properties into shader 'friendly' form.

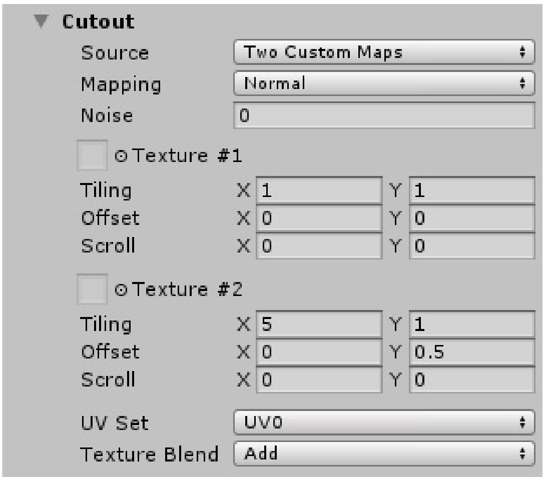
- **Cone** – Requires: **Position**, **Normal Direction**, **Height** and **Radius**, to be updated from script.

  Check **7. Mask (Cone)** example scene and **Controller_Mask_Cone** script used there for updating shader parameters.



  Note: example script uses Unity built-in spot light and converts its transform properties into shader 'friendly' form.

Cutout



▼ Cutout

Source: Two Custom Maps

Mapping: Normal

Noise: 0

Texture #1

Tiling X 1 Y 1

Offset X 0 Y 0

Scroll X 0 Y 0

Texture #2

Tiling X 5 Y 1

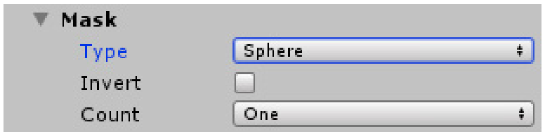
Offset X 0 Y 0.5

Scroll X 0 Y 0

UV Set: UV0

Texture Blend: Add

Properties in Cutout group define shape of dissolve effect.
If **Mask** is enabled, it becomes edge noise controlled by **Noise** parameter.

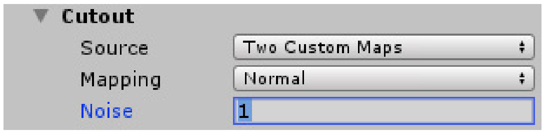


▼ Mask

Type: Sphere

Invert: □

Count: One



▼ Cutout

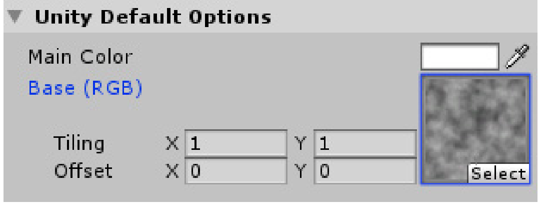
Source: Two Custom Maps

Mapping: Normal

Noise: 1

Source:

- **Main Map Alpha** – Dissolve cutout values are read from the MainMap alpha channel.



▼ Unity Default Options

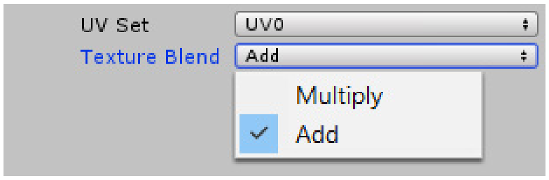
Main Color

Base (RGB)

Tiling X 1 Y 1

Offset X 0 Y 0

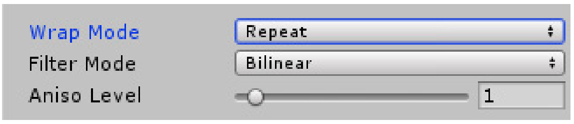
- **Custom Maps** (One, Two Three) – Custom dissolve textures. Can be animated using **Scroll** parameter and mixed by **Multiplying** or **Combining** (**Add**).



UV Set: UV0

Texture Blend: Add

Note: Cutout textures can be animated by **Scroll** parameter, but make sure **Wrap Mode** parameter is set to be **Repeat** inside Texture Import settings.



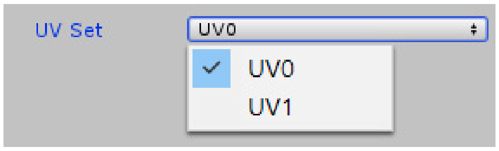
Wrap Mode: Repeat

Filter Mode: Bilinear

Aniso Level: 1

Mapping:

- **Normal** – Cutout textures are sampled by mesh UV coordinates. Optionally can be used UV0 or UV1 layout.



UV Set: UV0

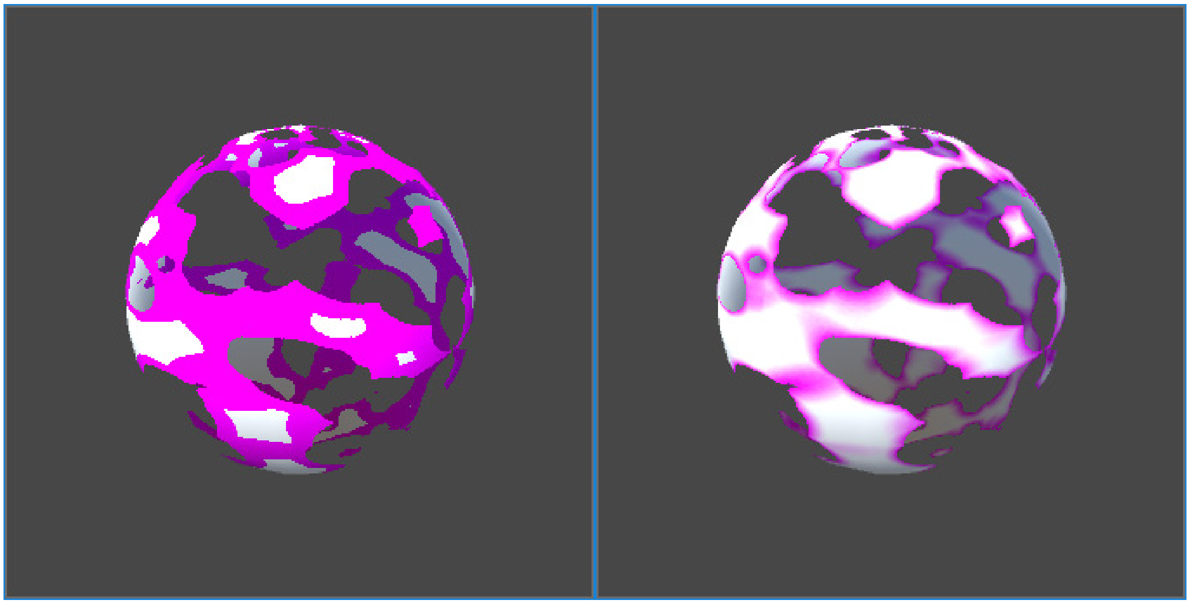
- **Triplanar** – Triplanar map projection. No mesh UV coordinates are required. Note: 1 texture projection requires 3 texture sampling.

- **Screen Space** – Textures are sampled in screen space coordinates. No mesh UV coordinates are required.

Edge

Controls visual appearance of the edge.

- **Width** – Controls dissolve edge size.
- **Shape** – Edge color can be solid or smooth (blending with background).



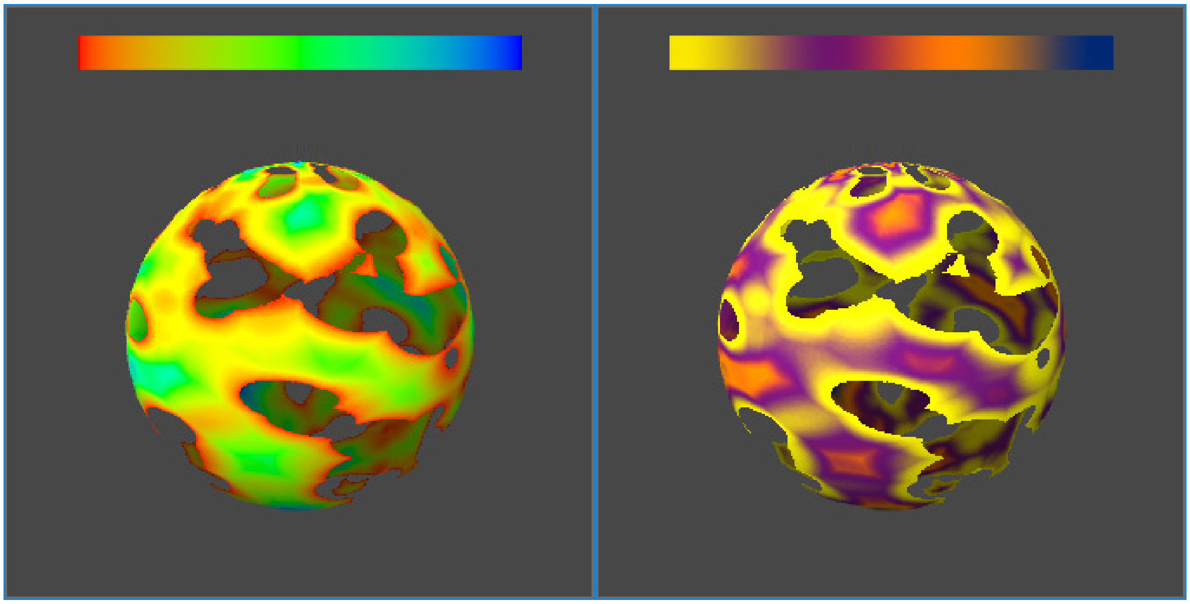
(Solid)

(Smooth)

- **Color** – Dissolve edge color. Alpha channel effects transparency.
- **Intensity** – Makes dissolve edge color emissive.
- **Texture** – Additional control over dissolve edge color.

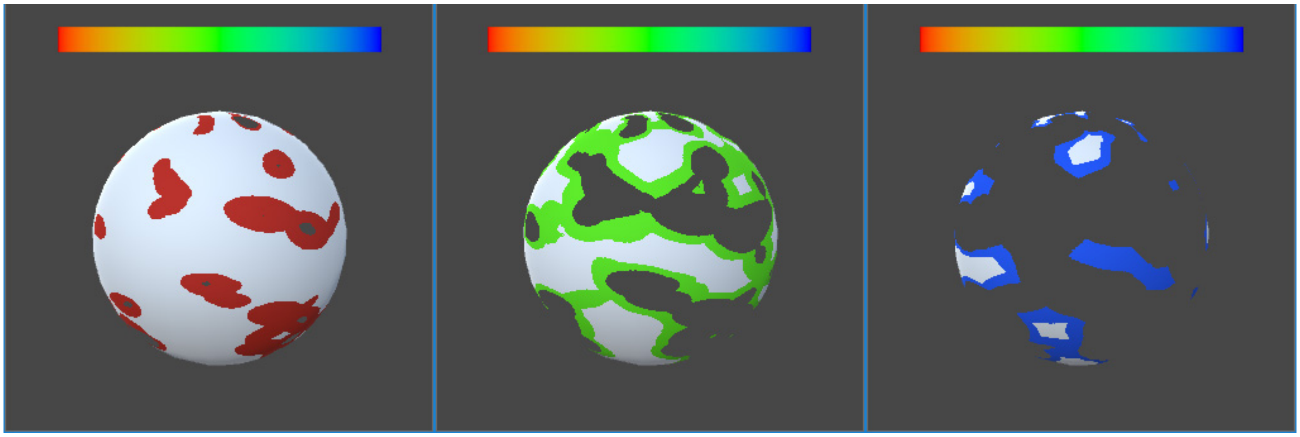
  Note: Textures alpha channel effects edge transparency. It can be adjusted by **Alpha Offset** parameter.

  1. **Gradient** – Ramp texture is used for calculating edge color.



Texture can be animated with **Phase Offset** parameter, but it needs texture with **Repeat** wrap mode inside Texture Import settings.

If **Is Dynamic** checkbox is enabled and **Mask** is not used, then edge color depends on **Dissolve** parameter value read as a **V** (from uv) coordinate of a Ramp texture.



(Dissolve: 0.1) (Dissolve: 0.5) (Dissolve: 0.9)

If **Mask** is enable dynamic gradient effect can be achieved by animating **Dissolve** property `_DissolveCutoff` from custom script.

2. **Main Map** – Dissolve edge color is multiplied by MainMap color.
3. **Custom** – Custom color texture. UV coordinates are the same as MainMap.

- **Emission** – Shader participates in Global Illumination system. Mesh must be **Static**.
  If option is enabled shader will need `_Dissolve_ObjectWorldPos` variable update from a custom script (dissolve color illumination is calculated inside shader Meta pass that has no info about mesh position).

  Note: LWRP does not support dynamic GI, only baked.

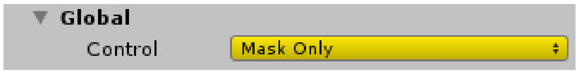
Main UV Distortion

Distorts main UV coordinates used by MainMap, BumpMap and other default textures.
Noise UV direction is read from MainMap or Cutout Source textures Red and Green channels.

Global Control

All material properties in Advanced Dissolve shaders can be controlled from custom scripts. If multiple materials need similar shader values it is better to use **Global** properties.

Check **8. Global Controller** example scene.
All objects here use the Plane mask. By default we would need to create "for" loop inside script and update mask parameters for all scene materials.
But mask parameters (position and normal direction) are same for all materials.
By setting global controller to **Mask Only** we have to update Plane mask properties only once and all materials will use that values.



▼ **Global**
Control Mask Only

Shader global parameters are updated by [Shader.SetGlobalVector]() and [Shader.SetGlobalFloat]() methods from script.

Package includes scripts for updating global properties for all groups of Advanced Dissolve shaders (path: `Assets\VacuumShaders\Advanced Dissolve\Example Scenes\Files\Scripts\Controllers`).

Global variable has the same names as non-global but with "**_Global**" in the end.
E.g. `_DissolveMaskPosition` and `_DissolveMaskPosition_Global`,
`_DissolveMaskNormal` and `_DissolveMaskNormal_Global`

Note: Global controller updates only variable parameters, they will be disabled (grayed out) inside material editor.
Shader keywords cannot be controlled globally and need to be modified manually using [Material.EnableKeyword]() and [Material.DisableKeyword]() methods.

Mask type keywords:

- **None -** `_DISSOLVEMASK_NONE`
- **XYZ Axis** - `_DISSOLVEMASK_XYZ_AXIS`
- **Plane** - `_DISSOLVEMASK_PLANE`
- **Sphere** - `_DISSOLVEMASK_SPHERE`
- **Box** - `_DISSOLVEMASK_BOX`
- **Cylinder** - `_DISSOLVEMASK_CYLINDER`
- **Cone** - `_DISSOLVEMASK_CONE`

Mask count keywords:

- **One** – `_DISSOLVEMASKCOUNT_ONE`
- **Two** – `_DISSOLVEMASKCOUNT_TWO`
- **Three** – `_DISSOLVEMASKCOUNT_THREE`
- **Four** - `_DISSOLVEMASKCOUNT_FOUR`

Cutout source:

- **Main Map Alpha** – _DISSOLVEALPHASOURCE_MAIN_MAP_ALPHA
- **Custom Texture** – _DISSOLVEALPHASOURCE_CUSTOM_MAP
- **Two Custom Textures** - _DISSOLVEALPHASOURCE_TWO_CUSTOM_MAPS
- **Three Custom Textures** - _DISSOLVEALPHASOURCE_THREE_CUSTOM_MAPS

Cutout source textures mapping keywords:

- **Normal** – _DISSOLVEMAPPINGTYPE_NORMAL
- **Triplanar** – _DISSOLVEMAPPINGTYPE_TRIPLANAR
- **Screen Space** - _DISSOLVEMAPPINGTYPE_SCREEN_SPACE

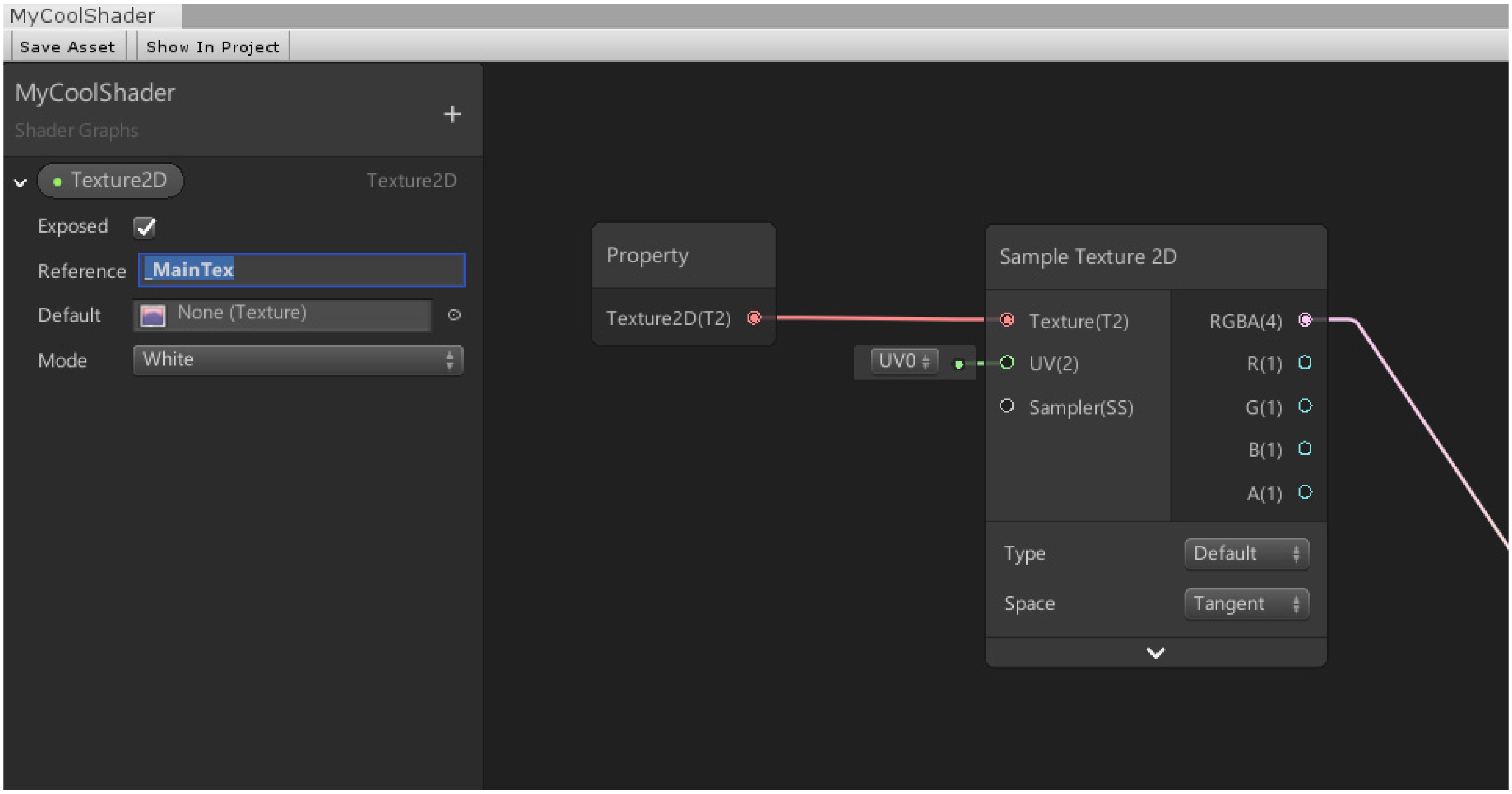
Edge color textures:

- **None** – _DISSOLVEEDGETEXTURESOURCE_NONE
- **Gradient** – _DISSOLVEEDGETEXTURESOURCE_GRADIENT
- **Main Map** - _DISSOLVEEDGETEXTURESOURCE_MAIN_MAP
- **Custom** - _DISSOLVEEDGETEXTURESOURCE_CUSTOM

Integration with Shader Graph

Advanced Dissolve effect cannot be directly constructed in Shader Graph using nodes, however it can be integrated into HLSL shader created by Shader Graph.

The only requirement: Shader Graph must use Sample Texture 2D node for **_MainTex**.



Exposed ☑

Reference _MainTex

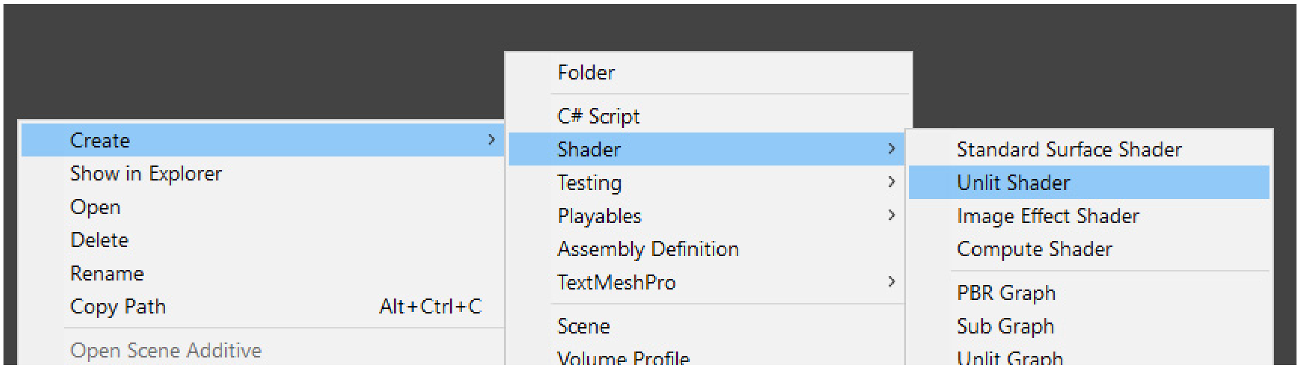
Default None (Texture)

Mode White

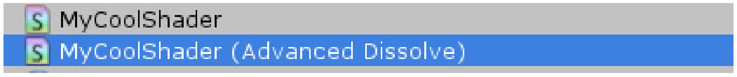
Type Default

Space Tangent

1. Create new Unlit shader file.

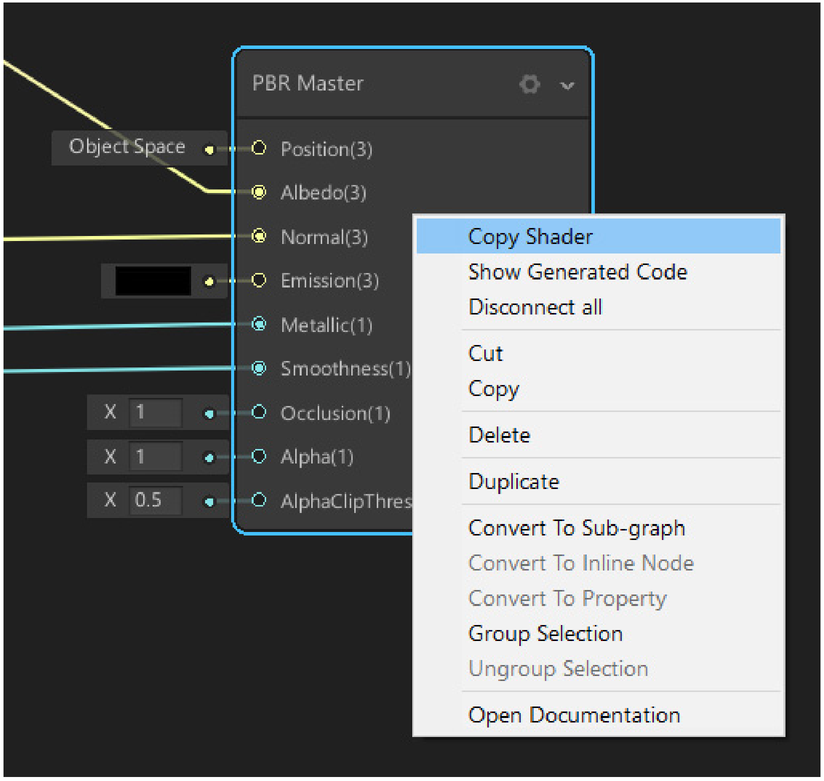


2. Rename it to be the same as ShaderGraph file but with ***(Advanced Dissolve)*** in the end.



(Not necessary step, but will make life easier later).

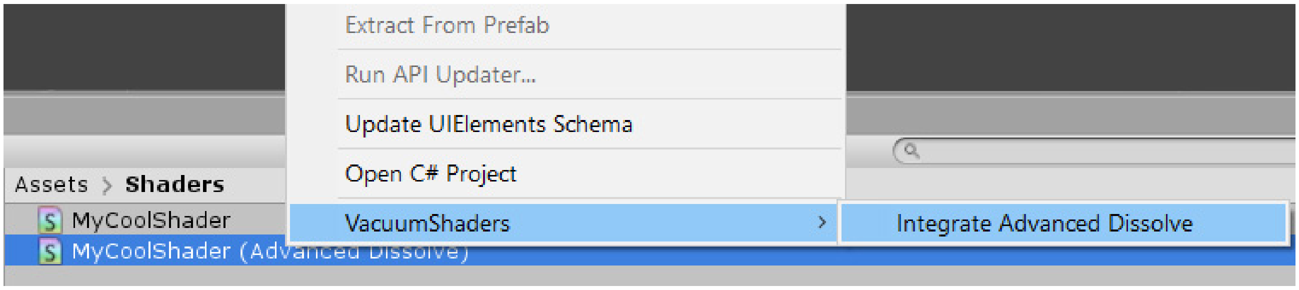
3. Inside Shader Graph right click on Master node and select Copy Shader.
   HLSL shader is now copied into system memory as text.



4. Open just created Unlit shader file in any text editor and delete all content.
5. Past shader copied into memory by CTRL+V.
6. Save file and close it.

Steps above creates HLSL shader from Shader Graph. We will add Advanced Dissolve effect inside this shader file.

7. Right click shader file and from context menu select **VacuumShader > Integrate Advanced Dissolve**



8. Shader is ready to use.
   It is a copy of Shader Graph effect but with Advanced Dissolve features.

Integration with Curved World (http://u3d.as/awm)

Follow steps described in **Advanced Dissolve/ Shaders/ cginc/ Integration_CurvedWorld.cginc** file.

Note: Curved World currently supports only Standard render pipeline.