# Advanced Dissolve

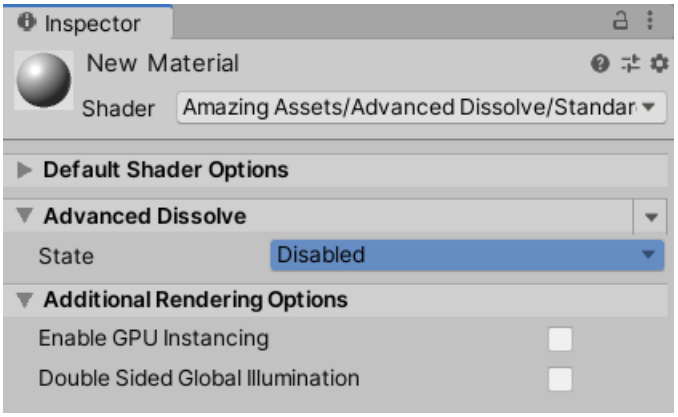


amazingassets.world

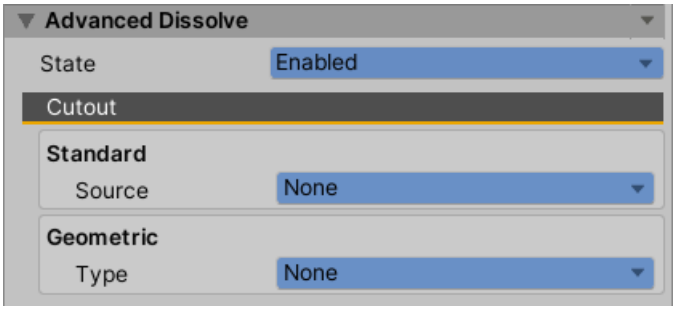
# TABLE OF CONTENTS

# QUICK START

Create simple mesh object and assign new material with **Amazing Assets/Advanced Dissolve** shader.

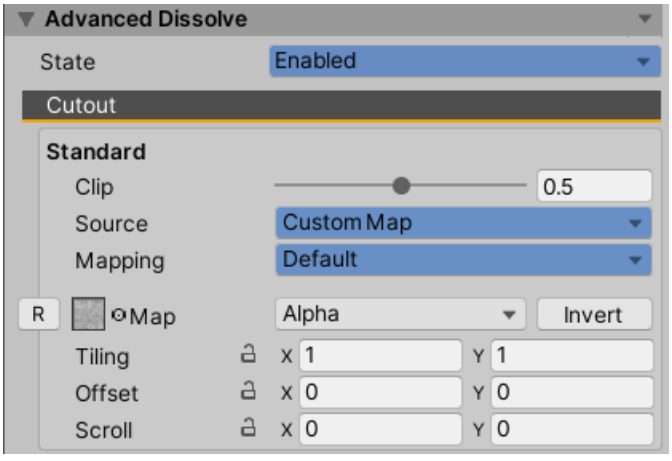


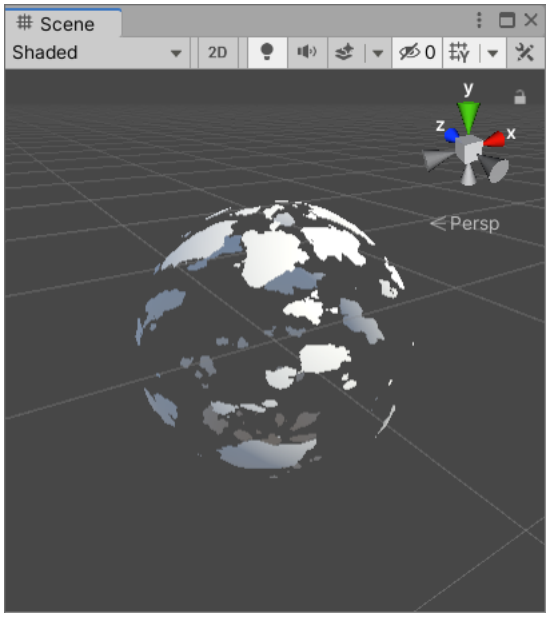

Inside material editor open **Advanced Dissolve** tab and enable **State** option. **Cutout** section with two groups: **Standard** and **Geometric** will become available.



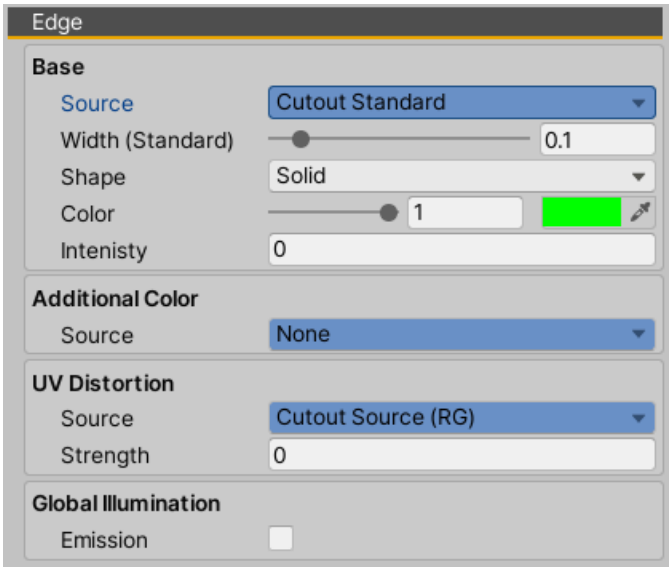

Change **Standard Source** option from **None** to the **Custom Map** and for the map parameter choose one of the package included textures, for example *CutoutMap1*.

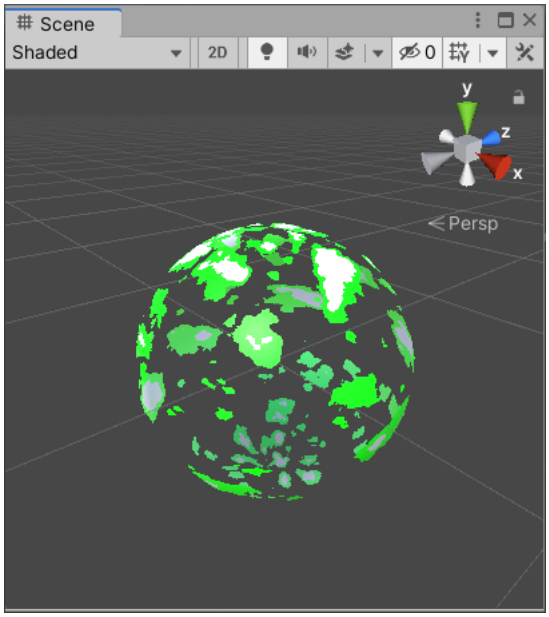
Now, changing **Clip** slider will affect mesh dissolve effect. This is simple basic dissolve effect.



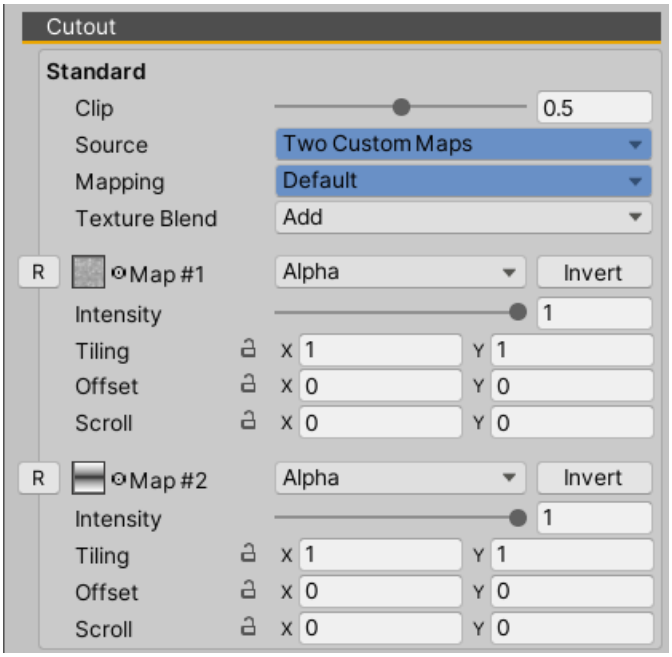



Inside **Edge** section change **Base Source** option from **None** to **Cutout Standard**.
Dissolve effect now will have visible colored edge.

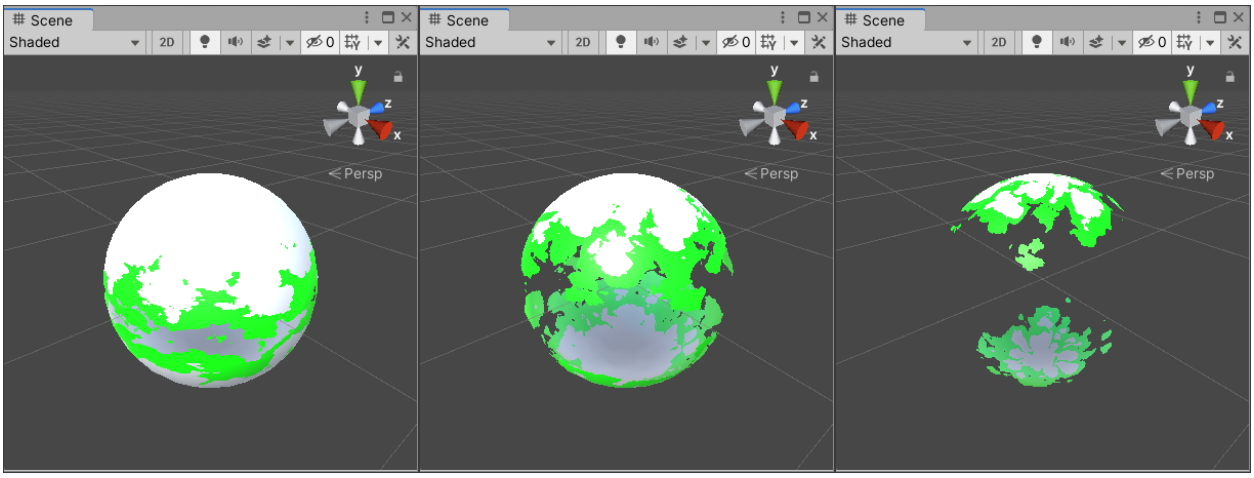



Currently material uses one texture map for calculating dissolve effect. Let's try two textures combination.
Inside **Cutout Standard** group choose **Two Custom Maps** and select *CutoutMap24* for the second map.
Set **Mapping** option to **Default** and **Texture Blend** to **Add**.



Now changing **Clip** slider will cut the middle part of the mesh first and then moving toward the top and bottom.

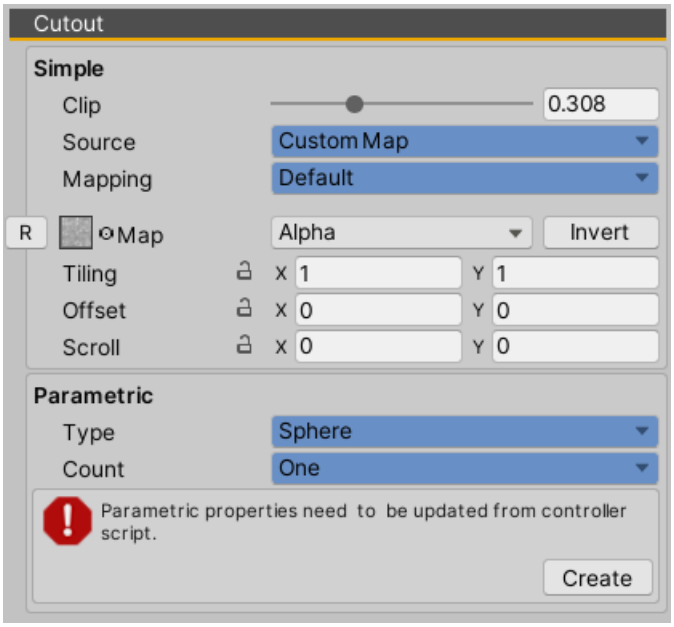



This is the end of the first part of the Quick Start chapter.
To summarize it: We have created simple dissolve effect using custom maps with enabled edge color control.
Before continue reading, manually try various texture combinations with different **Mapping** types and **Tiling/Scroll** parameters.

Note, pay attention to the textures wrap mode. Using **Clamp** or **Mirror One** modes may create unwanted results.

## PART 2: GEOMETRIC CUTOUT

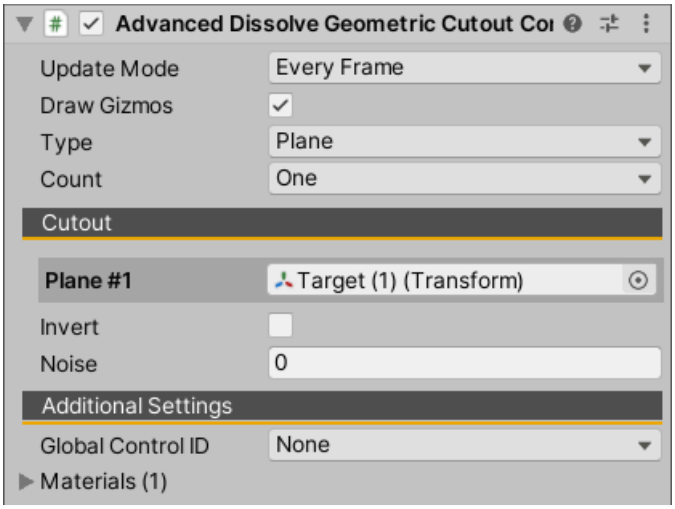
Inside **Cutout** section change **Standard Source** option back to the **Custom Map** and enable **Geometric Plane**. Set count option to **One**.



Properties for geometric cutout effects are not visible inside material editor. They must to be updated using scripts. Package already includes such controller scripts.

Note, package included controller scripts are explained in chapter **Runtime API**.

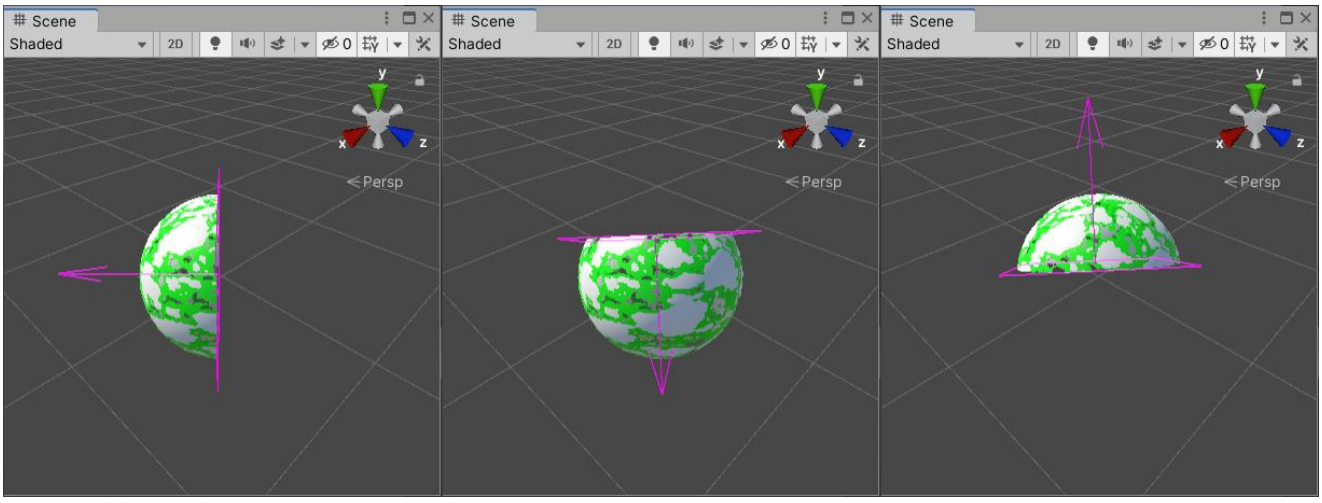
In our example, material editor displays warning to use controller script. Click on the **Create** button. This will create **AD Geometric Cutout Controller** game object with controller script attached to it.




**AdvancedDissolveGeometricCutoutController** script is already setup to work with **Plane** type cutout and **Plane #1** is using **Target (1)** game object.

**Plane** type geometric cutout effect needs *position* and *normal* for calculating dissolve area. Purpose of this controller script is: setup required material properties using user friendly interface. Script reads **Target (1)** object's *position* and *normal* values and sends them to the materials selected in the **Materials** array.

Inside scene select **Target (1)** game object. Move and rotate it. Pixels behind plane will be cutoff.

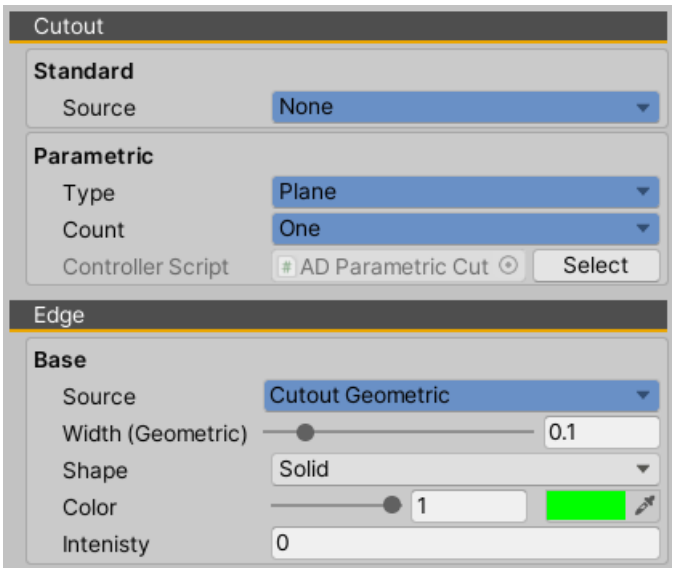


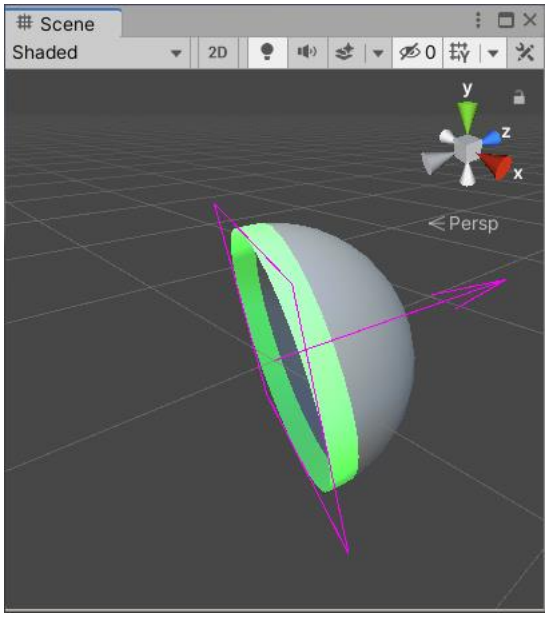
Advanced Dissolve shader includes several geometric cutout effects explained in more details in chapter **Geometric Cutout** and they all follow simple rules:

1. Geometric cutout **Type** and **Count** must be set inside material first (using material editor or by enabling appropriate keywords from script, explained later).
2. Geometric cutouts have no visible properties inside material editor, they all must be updated using script.
3. If using package included **AdvancedDissolveGeometricCutoutController** controller script, make sure geometric cutout **Type** and **Count** properties are the same as in the target materials, that are in the **Materials** array of this script.

As in this example material we are using both **Standard** and **Geometric** cutouts together, texture maps used in the **Standard** group will be used in calculating geometric **Noise** too, whose scale property is available inside **AdvancedDissolveGeometricCutoutController** script. Change it to see how it affects geometric cutout edge.

To use only **Geometric** cutout, set **Standard Source** to **None**.
Dissolve edges now will not be rendered as they were using source of the **Cutout Standard**, change it to the **Cutout Geometric**. Edge now will be rendered again.

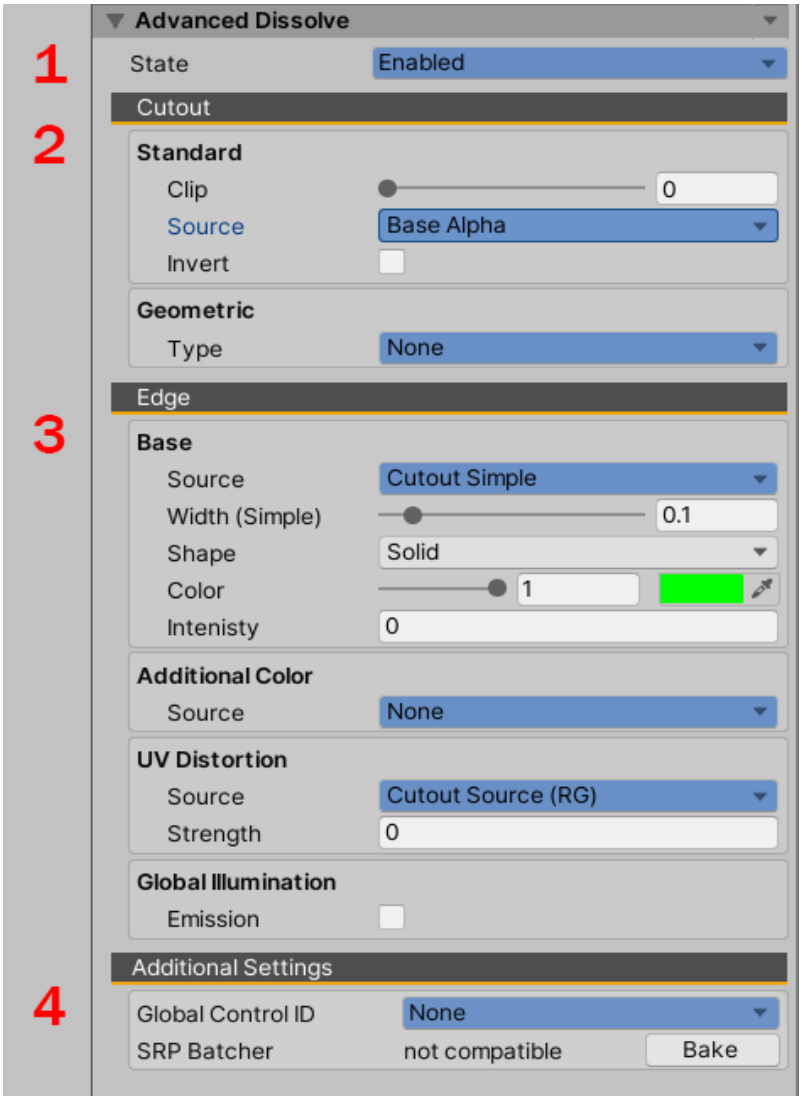



This is the end of the Quick Start chapter.
Advanced Dissolve shader features in more details are explained in chapters below and in example scenes.

# ADVANCED DISSOLVE SHADER PROPERTIES

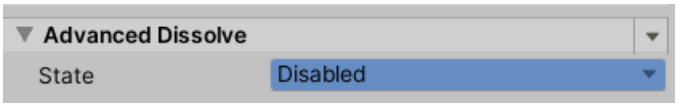
Advanced Dissolve options inside material editor are separated into 4 groups:




1. **State** - Activates or disables Advanced Dissolve effect for this selected material.
2. **Cutout** – Properties in this group define which part of a mesh will be cutoff and how.
3. **Edge** – Contains edge visual options.
4. **Additional Settings** – Includes shader global control settings.

## STATE

**State** - Activates or disables Advanced Dissolve effect calculation for selected material.



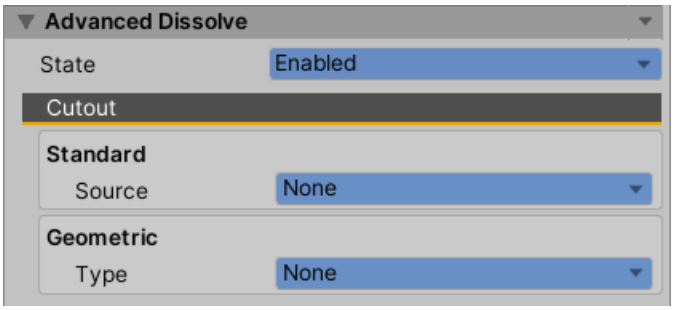

**State** is one one of the keywords that can be controlled from material editor or runtime scripts.
All Advanced Dissolve keywords have blue background inside material editor.

Advanced Dissolve keywords are explained in chapter **Keywords**.

Runtime scripts are explained in chapter **Runtime API**.

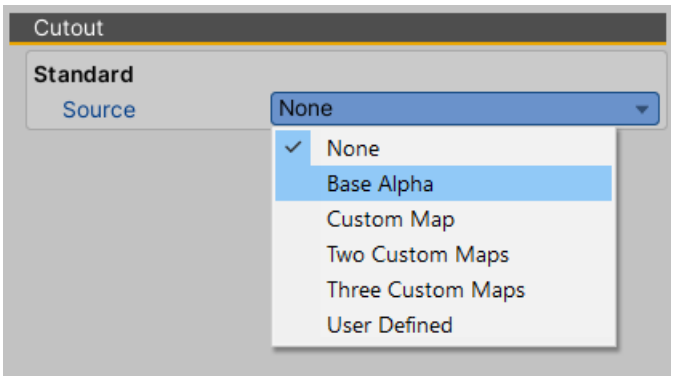
## CUTOUT

Options in **Cutout** section define dissolve area, its shape and amount.



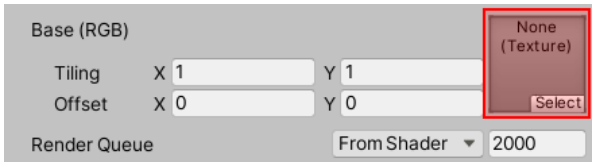

Dissolve effect can be calculated using **Standard** user defined texture maps or using **Geometric** equations. They can be used together or separately. There is no dissolve effect if both of them are disabled.

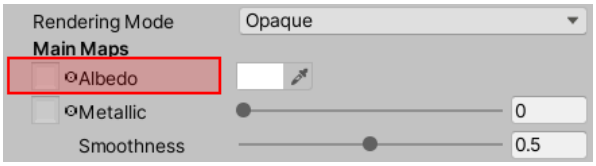
### STANDARD CUTOUT



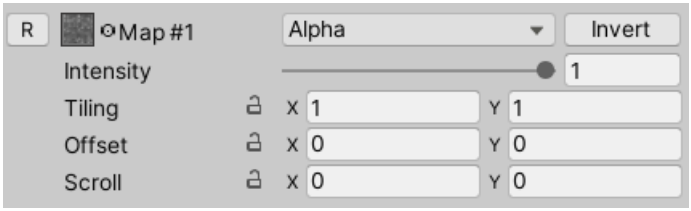
For **Standard** cutout effect are available following variations:

- **None** – Disables **Standard** cutout.
- **Base Alpha** – Dissolve value is read from material's default *BaseMap* or *AlbedoMap* texture's alpha channel.




- **Custom Maps** (One, Two, Three) – Dissolve value is calculating using one or by combining several textures.




  - R - Used for changing texture wrap mode. This is same setting as in the Texture Import Settings window.
  - **Channel** (Red, Green, Blue, Alpha) – Used texture channel.

    Note, custom maps Red and Green channels can be used by **UV Distortion** effect, explained in the **Edge** chapter below.

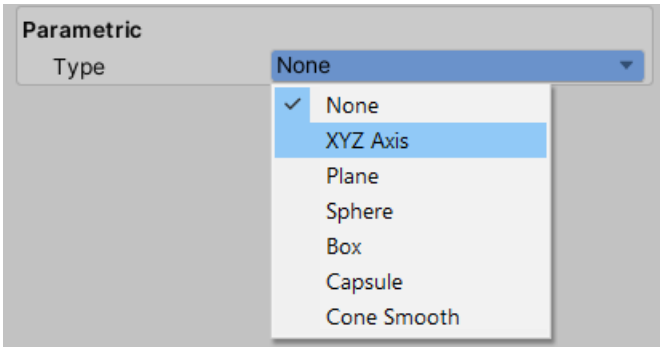
  - **Invert** – Inverts texture channel.
  - **Intensity** – Available only when using 2 or 3 custom maps. Controls texture weigh when blending multiple maps.
  - **Tiling/Offset**.
  - **Scroll** – Controls texture scroll animation. Requires texture with **Repeat** or **Mirror** wrap mode.
  - - When locked, all vector variables receive value from **x** property.

- **User Defined** – User defined cutout source is available only in shaders created using ShaderGraph. Explained in more details in chapter **Shader Graph**.

Dissolve effects created using **Standard** cutout maps are demonstrated in 1. Texture Cutout example scene. Open scene and enter game mode to enable materials **Clip** property animation, or modify it manually from material editor.

## GEOMETRIC CUTOUT

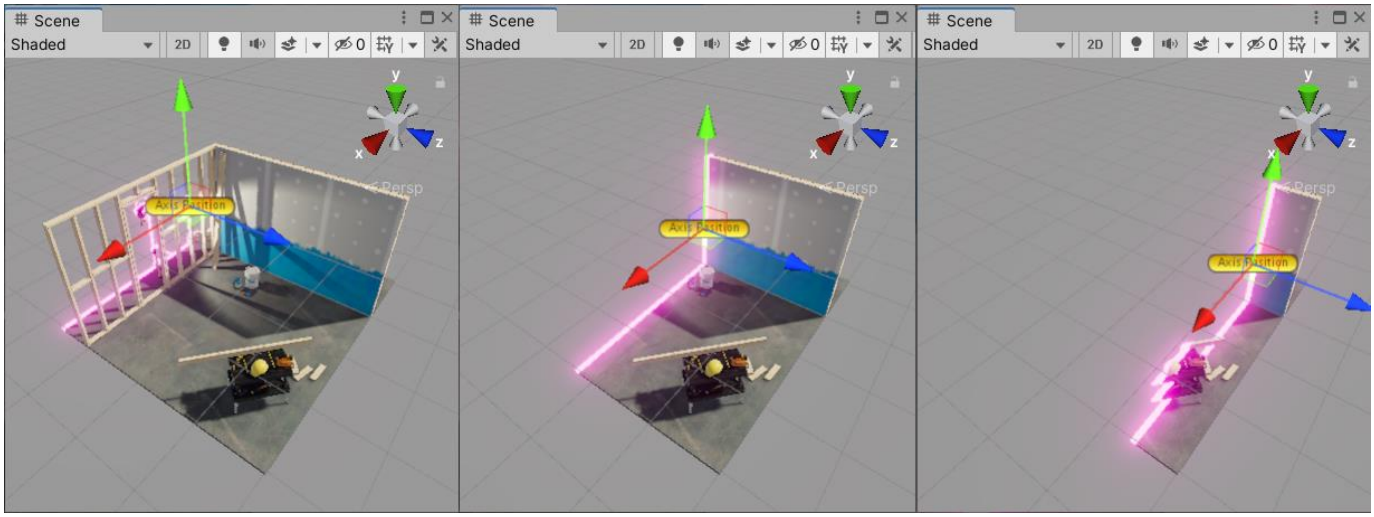
**Geometric** cutout is used for calculating geometrical shaped dissolve effects. It can be used separately from **Standard** cutout or in combination with it, in this case **Standard** cutout will be used for calculating geometric edge **Noise**.



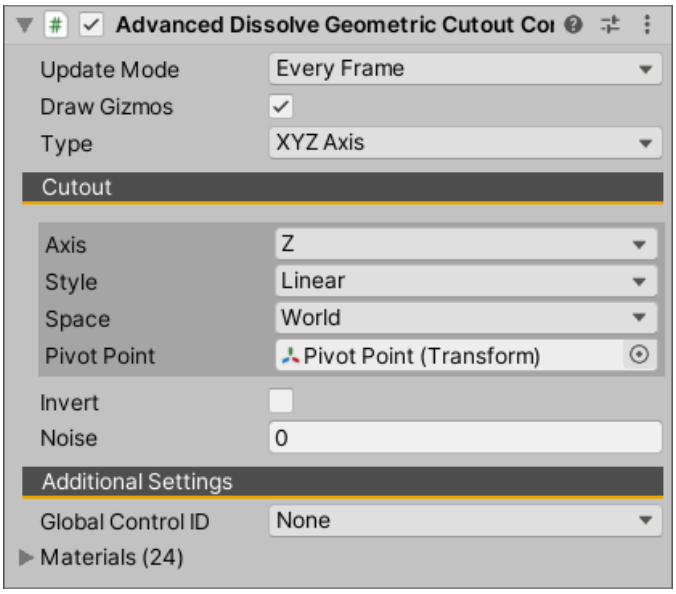

**Parametric** cutout effects have no visible controllers and options inside material editor and for updating their properties inside material, must be used **AdvancedDissolveParametricCutoutController** script or any other custom script.

Geometric Cutout > **XYZ Axis**

**XYZ Axis** cutout effect uses mesh vertex position to determine which pixels will be dissolved.
This effect is demonstrated in 2. XYZ example scene. Open it and move **Pivot Point** game object in the Scene view. All pixels whose position **Z** value are less than **Pivot Point** object's position **Z** value will be cutoff.

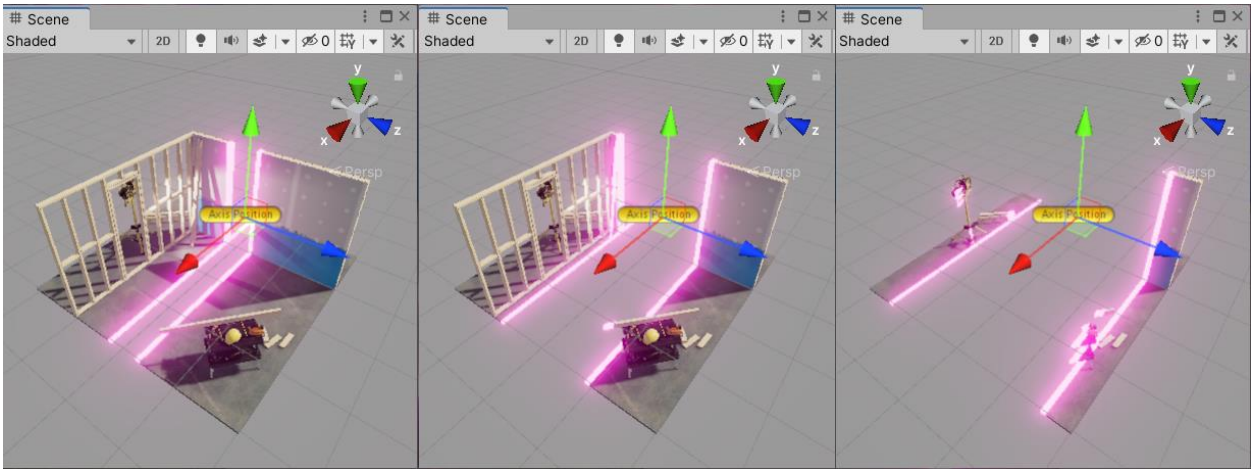


All geometric cutout effects properties are controlled using **AdvancedDissolveGeometric-CutoutController** script. Scene contains **AD Geometric Cutout Controller** game object with this script attached. Select it.




It is setup to use **Z** Axis with **Linear** style and **Pivot Point** is controlled in **World** space. All this data is sent to the materials selected in **Materials** array.

Reset **Pivot Point** object's scene position and change **Style** property in script from **Linear** to **Rollout.** Change **Axis** to **X**. Now based on the **Rollout** property mesh pixels will be dissolved symmetrically on the both side of the pivot point.

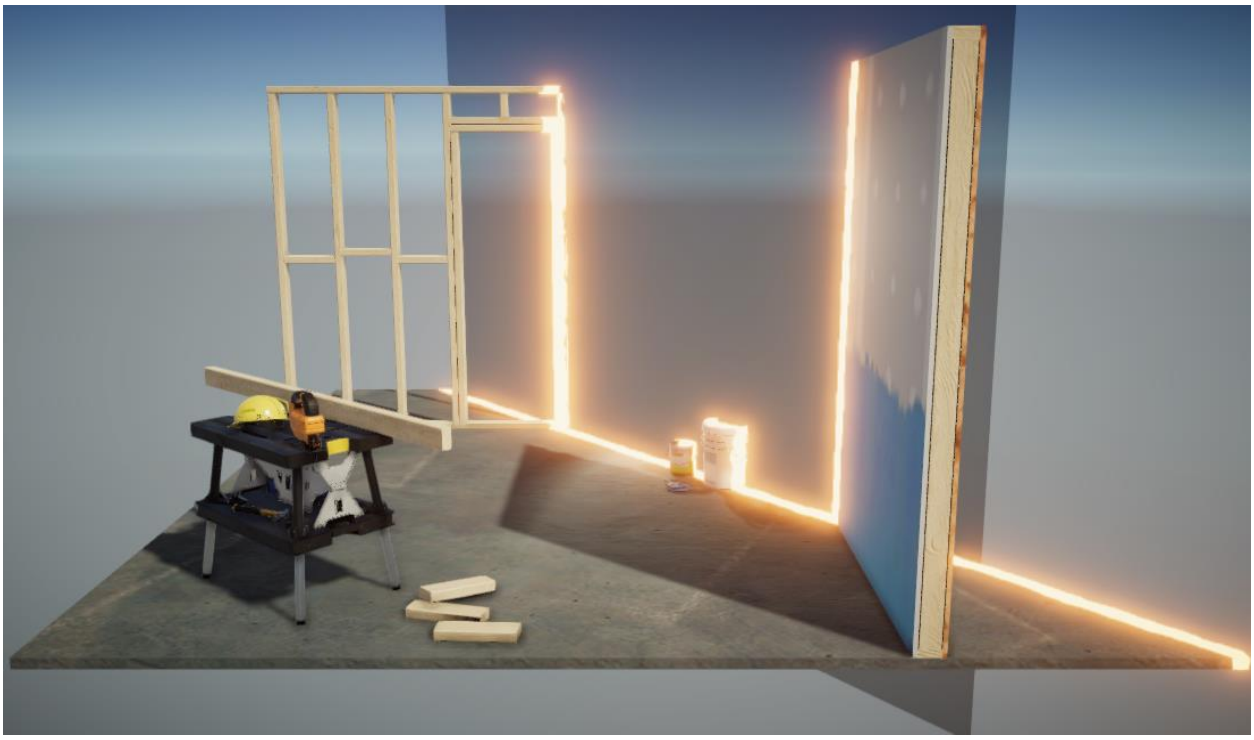



For **XYZ Axis** type cutout, controller script has following options:

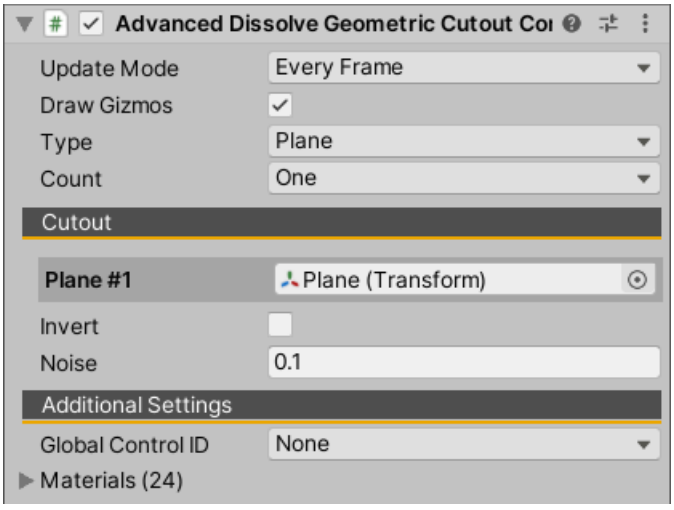
- **Axis** – Defines cutout axis. Only one of the X, Y, or Z axis can be used.
- **Style** – Dissolves pixels in **Linear** or **Rollout** Style.
- **Rollout** – Rollout size.
- **Space** - XYZ Axis cutout can be performed in vertex local or world coordinates.
- **Axis Position** – Cutout axis position. Based on the used **Axis**, only **x**, **y**, or **z** values are used.

Geometric Cutout > Plane

**Plane** cutout effect dissolves pixels behind parametric plane object.
This effect is demonstrated in 3. Plane example scene. Open it and move/rotate **Plane** game object. Pixels behind plane are cut off.

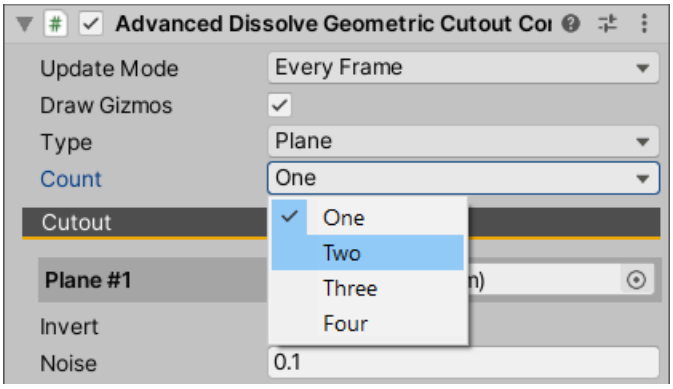


For calculation **Plane** type cutout effect shader requires *position* and *normal*. Those data to the material are sent using **AdvancedDissolveGeometricCutoutController** script attached to the **AD Geometric Cutout Controller** game object.




In this example scene materials are set up to use only one cutout plane. Advanced Dissolve shaders can use 4 cutout objects simultaneously, but only of the same type.

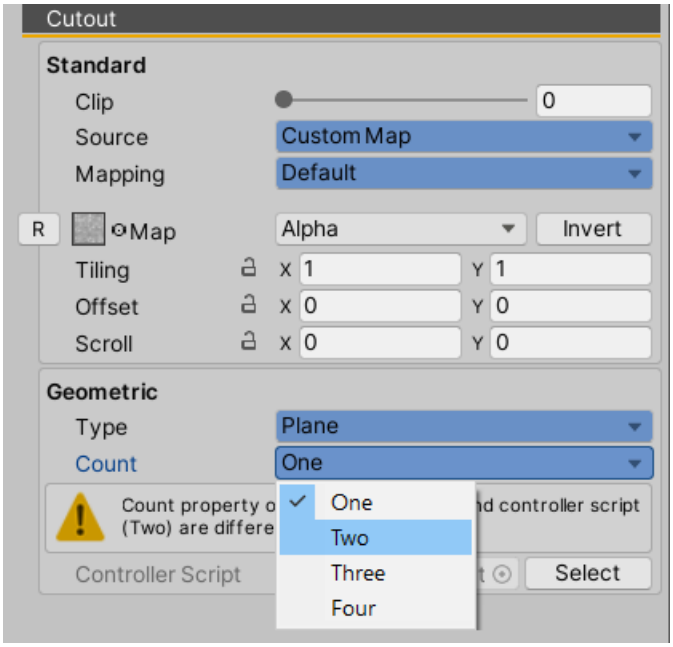
Inside controller script change **Count** property to **Two**.



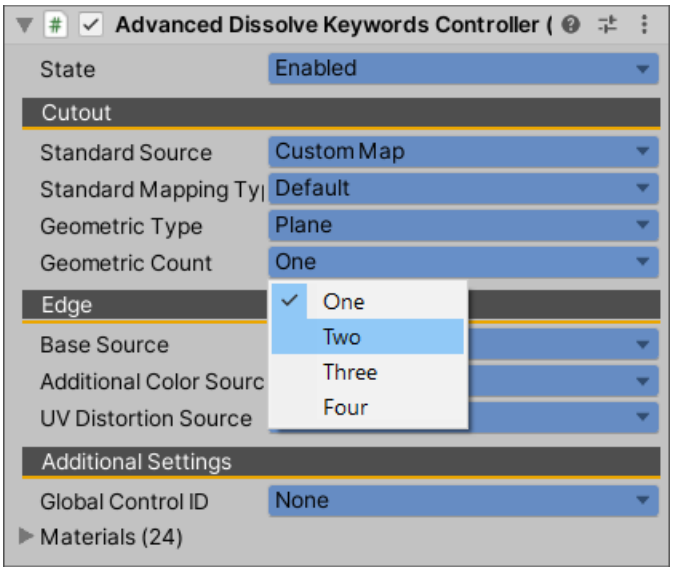

This will add **Plane #2** field and additional plane gizmo will be rendered in the scene view, visualizing new parametric plane's position and orientation.

Inside controller script it is possible to directly set *position* and *normal* values for plane equation, but in this case click on the **Create** button. This will create child transform object and it will be used by **Plane #2** field. Position and normal of this transform will be sent to the materials.

But currently scene materials do not expect second plane data, there are setup to use only one cutout plane. To enable cutout effect using two planes each material must be updated from their material editors.




To speed up working with Advanced Dissolve keywords for the big list of materials, package includes **AdvancedDissolveKeywordsController** script and in the scene it is attached to the **AD Keywords Controller** game object. Select it and change **Geometric Count** property from there.




All materials in the script **Materials** list will be updated. Now moving and rotating **Plane #2** target game object will affect materials too.

Note, geometric cutout **Count** is the keyword, explained in chapter **Keywords**.

Note, all package included scripts are explained in chapter **Runtime API**.

Package includes 4. Plane (Two Worlds) example scene using **Plane** cutout effect, demonstrating how one meshes can be dissolved in the same place where others are rendered. All scene materials here have exactly the same dissolve properties and keywords and are cutout using one plane, but for one meshes cutout effect is inverted in **AdvancedDissolveGeometricCutoutController** script. Open scene and move/rotate **Plane** game object.

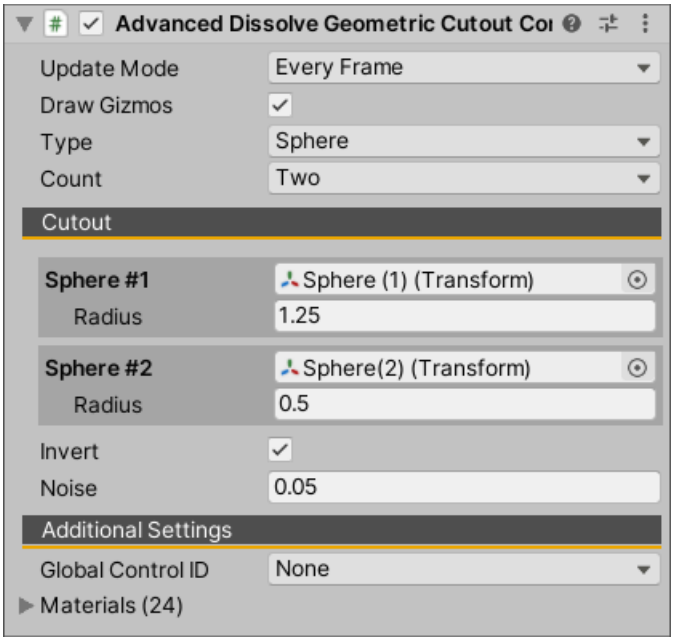
Geometric Cutout

## Sphere

**Sphere** cutout effect dissolves all pixels outside its radius.

This effect is demonstrated in 5. Sphere example scene. Open scene and move/scale **Sphere (1)** and **Sphere (2)** game objects.

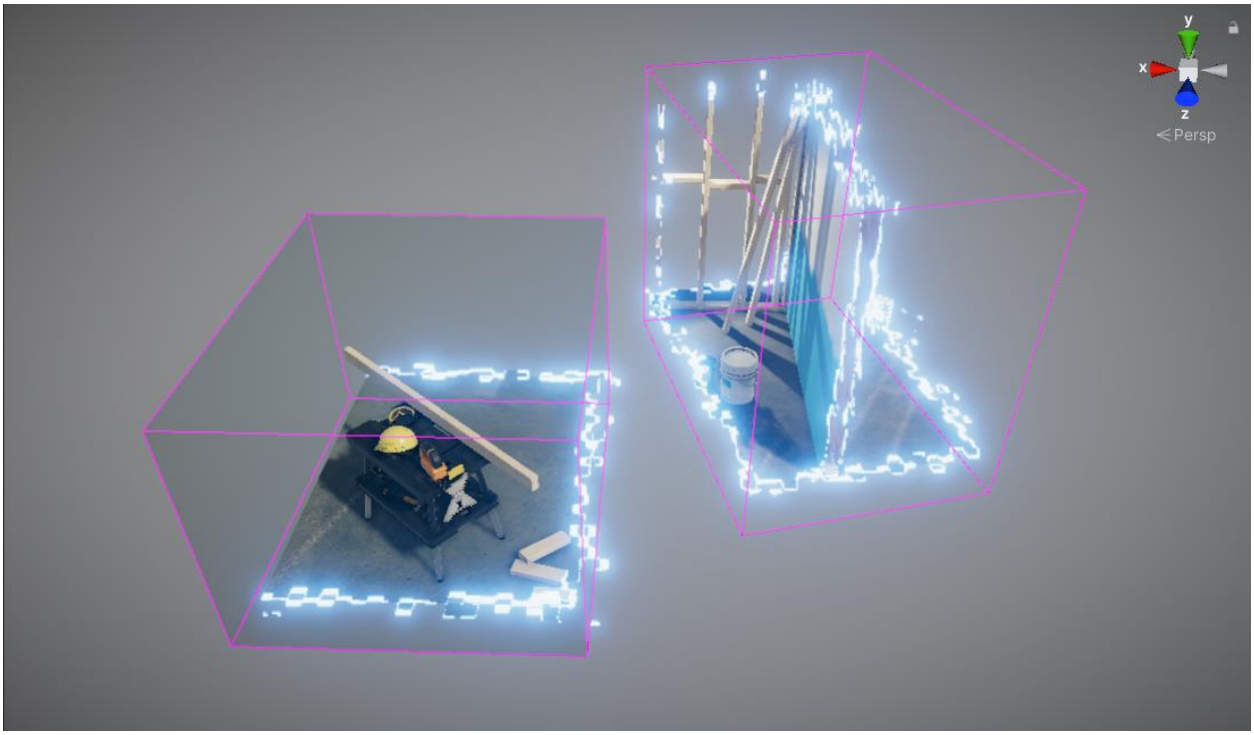
For calculation **Sphere** type cutout effect shader requires *position* and *radius*. Those data to the material are sent using **AdvancedDissolveGeometricCutoutController** attached to the **AD Geometric Cutout Controller** game object.



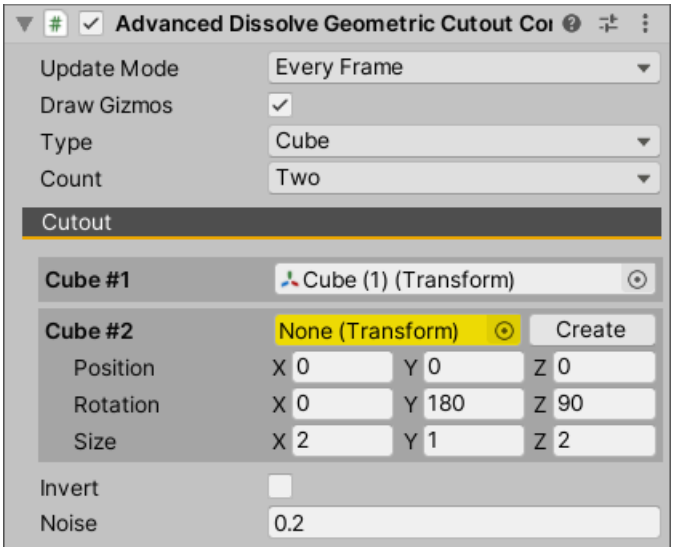
In this example scene, **Radius** properties in the **AdvancedDissolveGeometricCutoutController** script are updated separately from helper script attached to the sphere game objects which converts their transform scale into radius.

## Geometric Cutout > Cube

**Cube** cutout effect dissolves all pixels outside its bounds.
This effect is demonstrated in 6. Cube example scene. Open scene and move/rotate/scale **Cube (1)** game object.



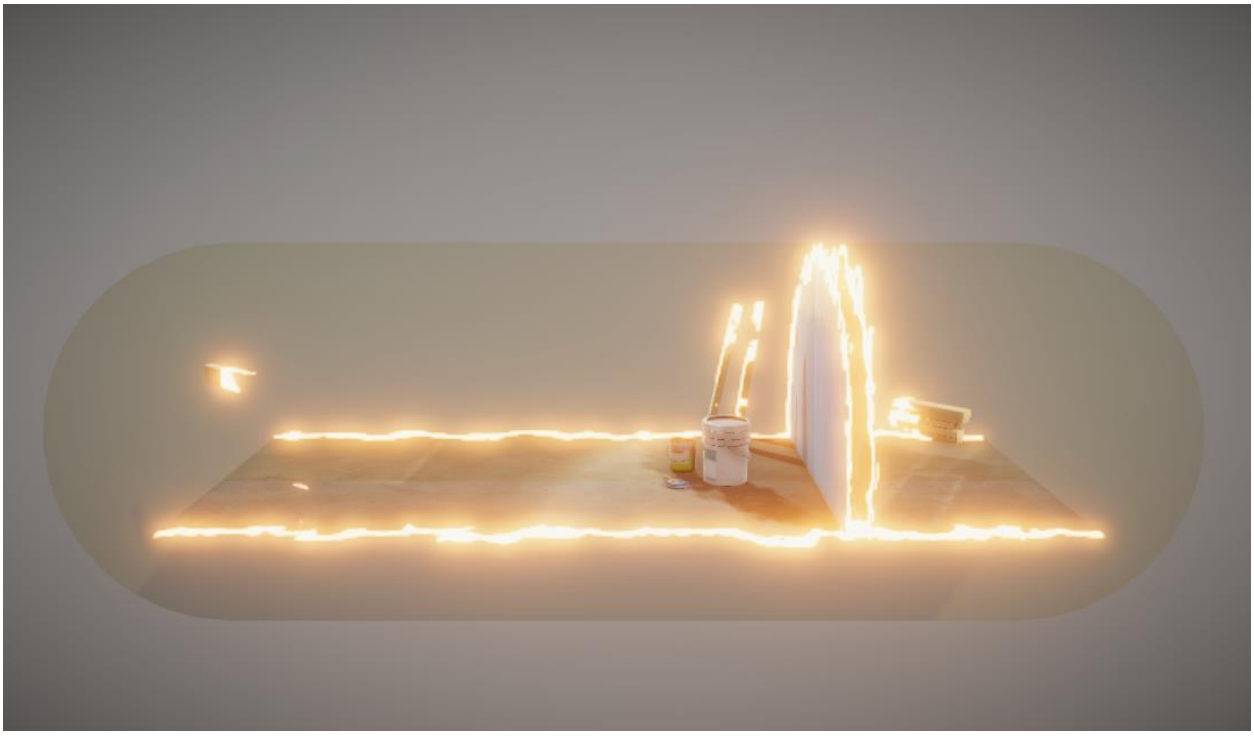

For calculation **Cube** type cutout effect shader requires *position*, *rotation* and *size*. Those data to the materials are sent using **AdvancedDissolveGeometricCutoutController** attached to the **AD Geometric Cutout Controller** game object.



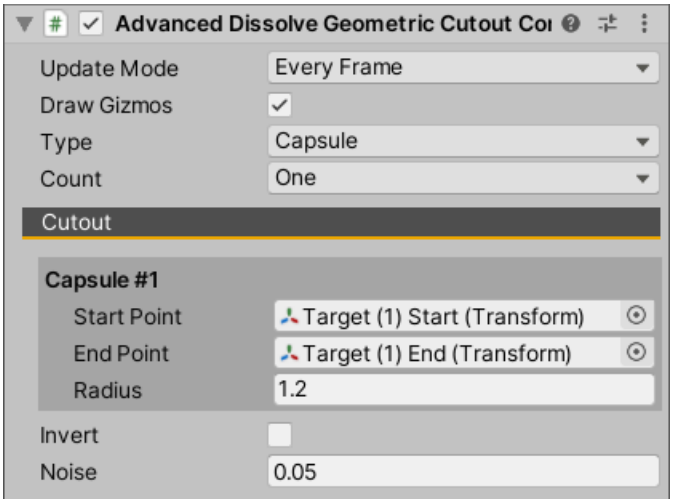
In this example scene **Cube #1** field reads *position*, *rotation* and *size* data from **Cube (1)** game object. **Cube #2** does not use any objects and allows to assign all requires data by hand.

## Geometric Cutout > Capsule

**Capsule** cutout effect dissolves all pixels outside its volume.
This effect is demonstrated in 7. Capsule example scene. Open scene and move **Point (1) Start** and **Point (1) End** game objects.



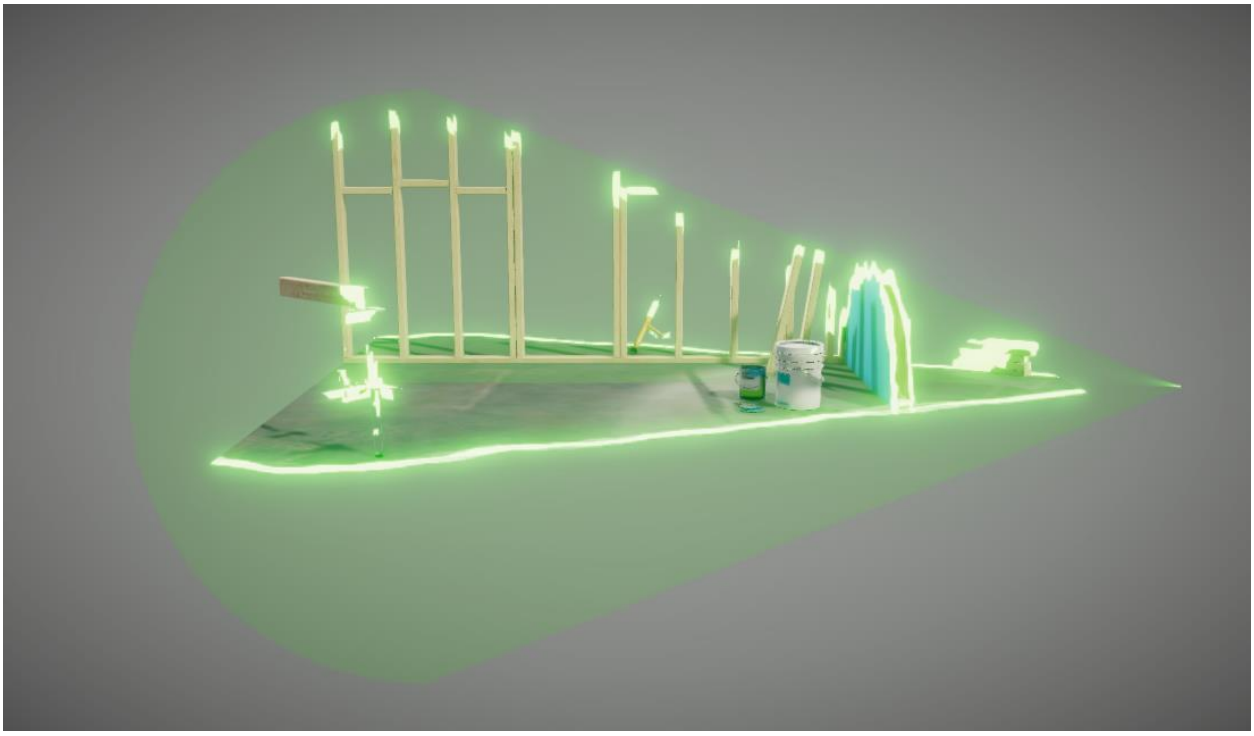
For calculation **Capsule** type cutout effect shader requires two point's *position* (start and end) and *radius*. Those data to the materials are sent using **AdvancedDissolveGeometricCutoutController** attached to the **AD Geometric Cutout Controller** game object.



Check 8. Capsule (Camera See-Through) example scene using same cutout affect. The only difference is **AdvancedDissolveGeometricCutoutController** script is using scene's Main Camera for **Start Point** and **End Point** is custom mesh object in scene that camera is looking at and rotates around it. This creates "*see-through*" effect when target object is always visible to the camera.

## Geometric Cutout > Cone Smooth

**Cone Smooth** cutout effect works exactly the same way as the **Capsule** and requires same *position* and *radius* data. The only difference is that **Cone Smooth** radius is 0 at the *start* point and grows to the full size at the *end* point position. This effect is demonstrated in 9. Cone Smooth example scene. Open scene and move/rotate **Spot Light** game object.

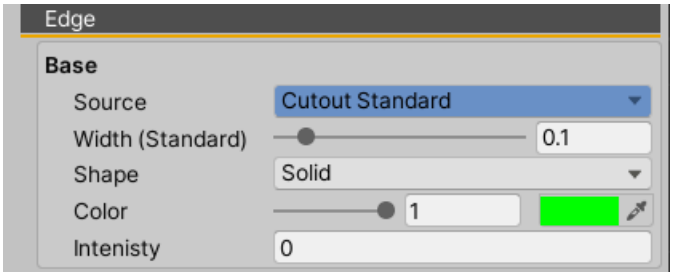


In this example, **AdvancedDissolveGeometricCutoutController** data is updated separately using helper script attached to the **Spot Light** game object**,** which converts spot lights properties (transform position, spot light range and angle) into **Cone Smooth** properties: *start position*, *end position* and *radius*.

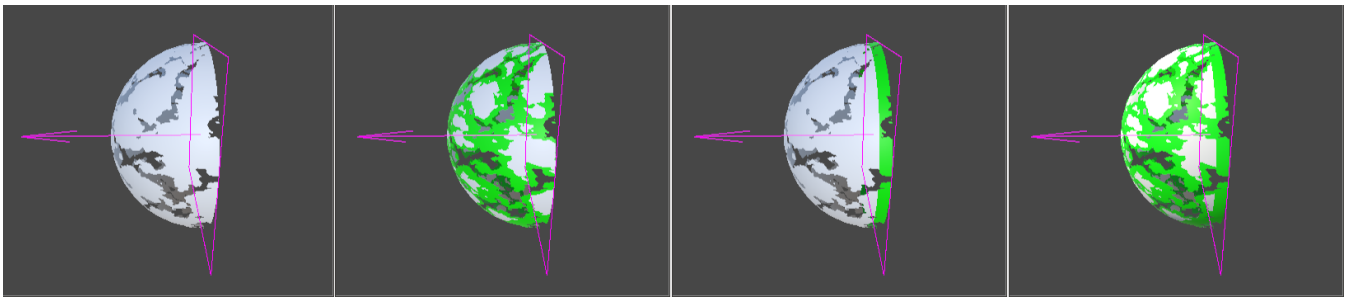
## EDGE

Edge section contains dissolve edge rendering options, divide into several sub-groups.

### BASE

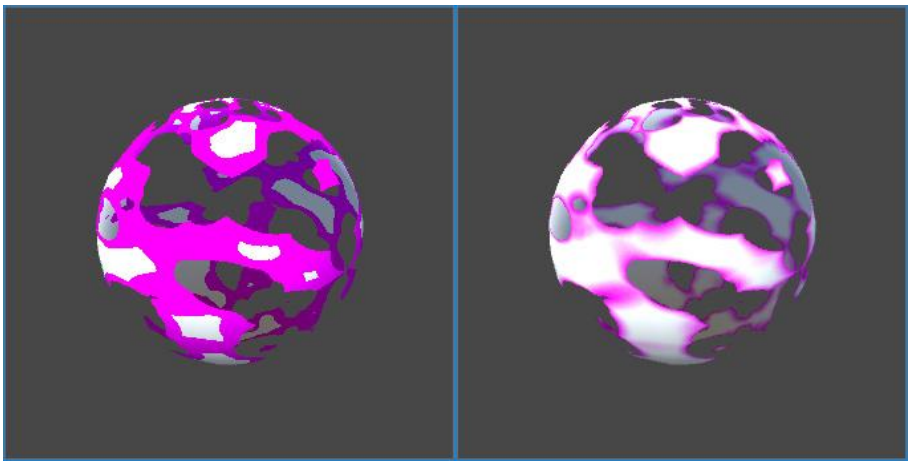



- **Source** – Rendered edge is calculated based on **Standard** or **Geometric** cutout data:



(None) (Standard) (Geometric) (All)

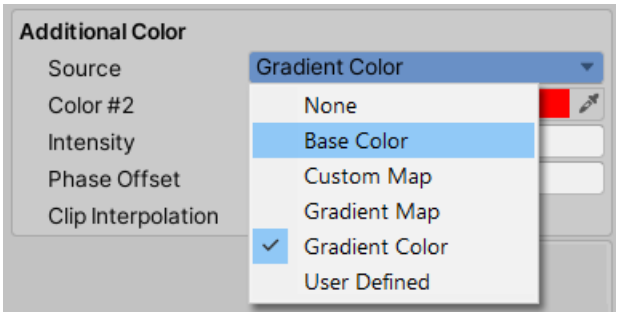
- **Width** – Rendered edge width.
- **Shape** – Defines how edge blends with background: Solid or Smooth.



(Solid) (Smooth)

- **Color** – Edge color value (RGB) and its transparency.
- **Intensity** – Color intensity. Controls color emission strength used in post processing (Bloom) effects.

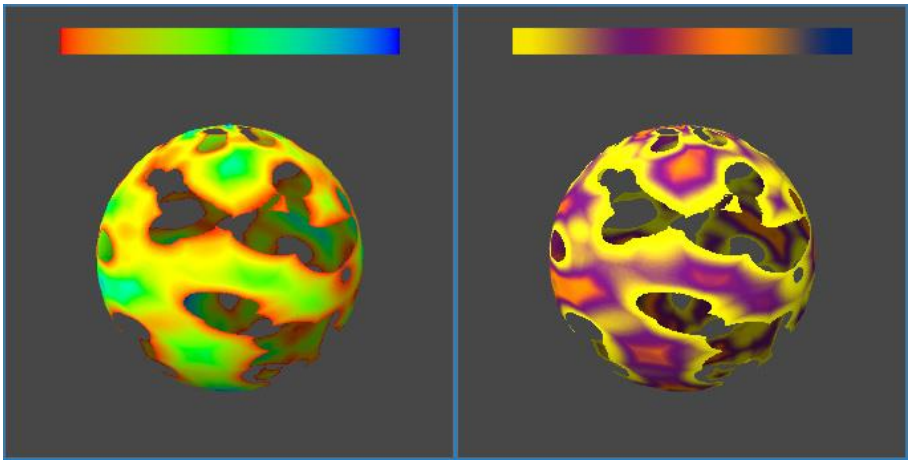
## ADDITIONAL COLOR




Allows using additional color or texture for creating more colorful edges.

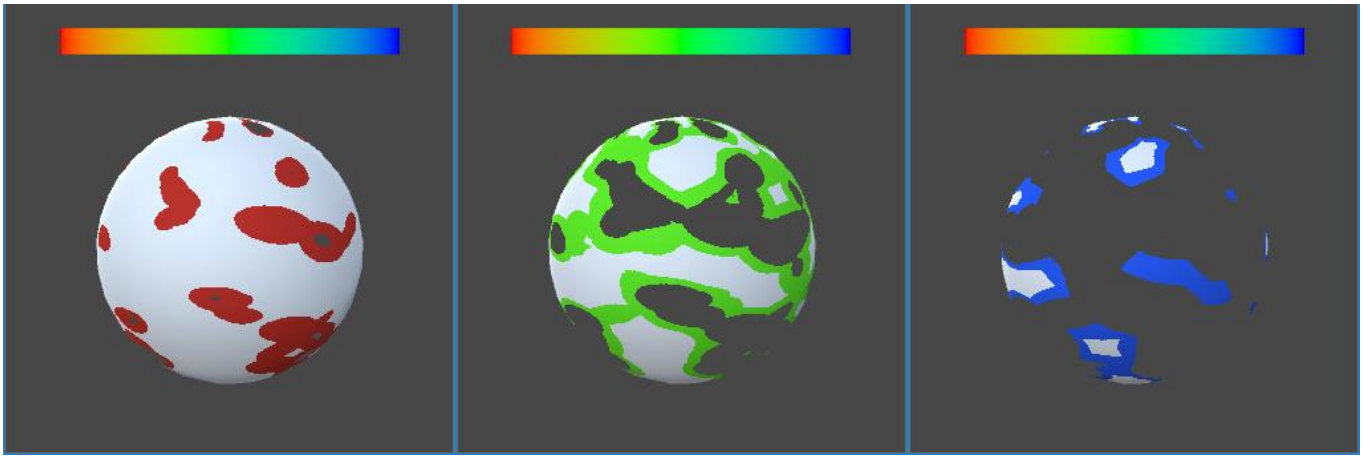
Note, take into account that results of the **Additional Color** are multiplied by **Color** property from the Edge Base sub-group.

- **Base Color** – Edge color is the same as material's default base color (Albedo).
- **Custom Map** – Common 2D texture read using mesh UV0 coordinate. Alpha value affects edge transparency that can be adjusted using **Alpha Offset** slider.
- **Gradient Map** – Used for creating gradient edge color. Results can be animated using **Scroll** property.



(Example of using gradient ramp textures)

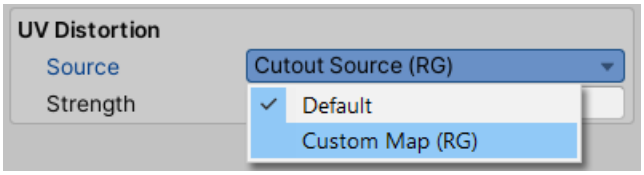
If **Clip Interpolation** is enabled (available only with **Standard** cutout) then gradient color interpolation value depends on the **Clip** property from **Standard** cutout.



(Clip: 0.1) (Clip: 0.5) (Clip: 0.9)

- **Gradient Color** – Same as Gradient Map effect, but with color property instead of a texture.
- **User defined** – Option available only for shaders created using ShaderGraphs and allows to use any user defend color node. Explained in chapter **ShaderGraph**.

## UV DISTORTION




Distorts material's main UV0 coordinate used by *MainMap*, *BumpMap* and other textures. By default uses Red and Green channels from **Standard** cutout.

## GLOBAL ILLUMINATION

When this option is enabled, edge color becomes visible to the Unity's lightmap baking system (both baked and real-time GI). Baked color intensity can be controlled using **Meta Pass Multiplier** property.

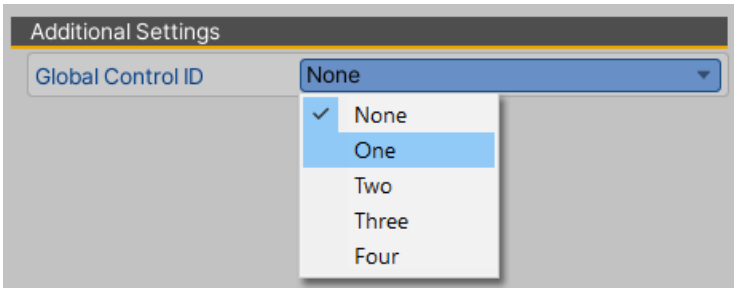
## GLOBAL CONTROL

Beside material editor, Advanced Dissolve shader properties can be controlled from scripts too and for some properties this is the only way to be updated (for example for geometric cutouts).

In example scenes dissolve properties of all materials are updated using two script:

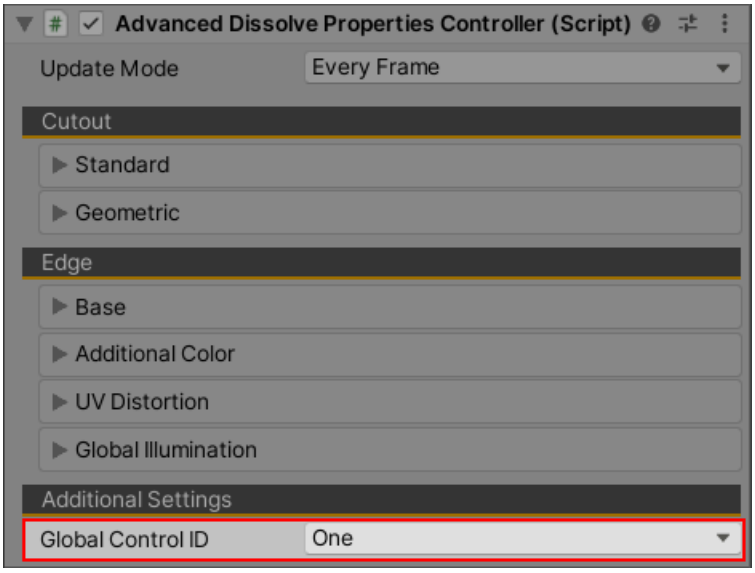
- **AdvancedDissolvePropertiesController** – Updates edge visual properties.
- **AdvancedDissolveGeometricCutoutController** – Updates geometric cutout properties.

Both scripts use simple ***for loop*** to iterate through materials array and update them. For scenes with just a few dissolve materials this has no impact on performance, but in the case of big amount of materials needed to be updated every frame, or if those materials share the same properties (for example geometric cutouts), **Global Control** becomes very useful. As it updates only one shader property (global property) and all materials automatically use it.
When **Global Control** is enabled material editor becomes inactive and it does not allow property changes from there (except keywords). Instead all dissolve properties must be updated from script.




**Global Control** usage is demonstrated in 10. Cone Smooth (Global Control) example scene. It is exact copy of the 9. Cone Smooth example scene, but with **Global Control** enabled.

Check **AD Properties Controller** and **AD Geometric Cutout Controller** game objects with attached controller scripts there. None of them has materials assigned in **Materials** array, instead they are updating global properties defined by **ID One**.




All scene materials are setup to use Global Control with **ID One**.

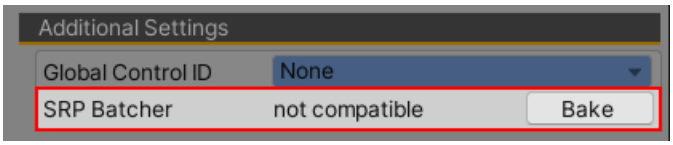
Advanced Dissolve has 4 global control IDs. It allows to have 4 independent groups of global materials updated separately from each other.

Note, global control ID is on of the keywords, explained in chapter Keywords.

## SRP BATCHER

Option is available only for scriptable render pipeline (Universal and High Definition) shaders and displays selected shaders compatibility with SRP Batcher. By default it is not compatible.

To make shader SRP Batcher compatible it is necessary to remove all Advanced Dissolve keywords and shader variants from it. This is done by clicking on **Bake** button.



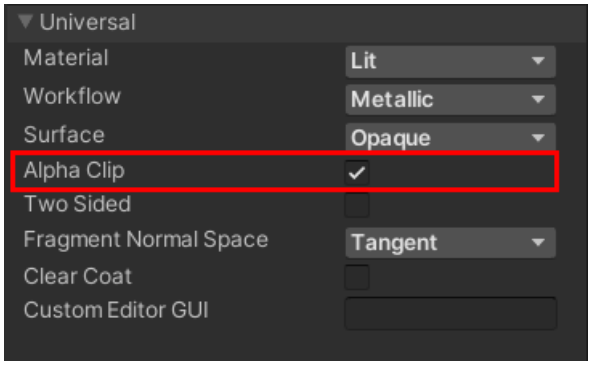

This generates new shader file that is an exact copy of selected shader, with currently used Advanced Dissolve keywords baked inside – all unused keywords and shader variants are removed. For baked shaders it is not possible in the future to change or assign new Advanced Dissolve keywords, appropriate fields inside material editor also becomes inactive.

# SHADERGRAPH

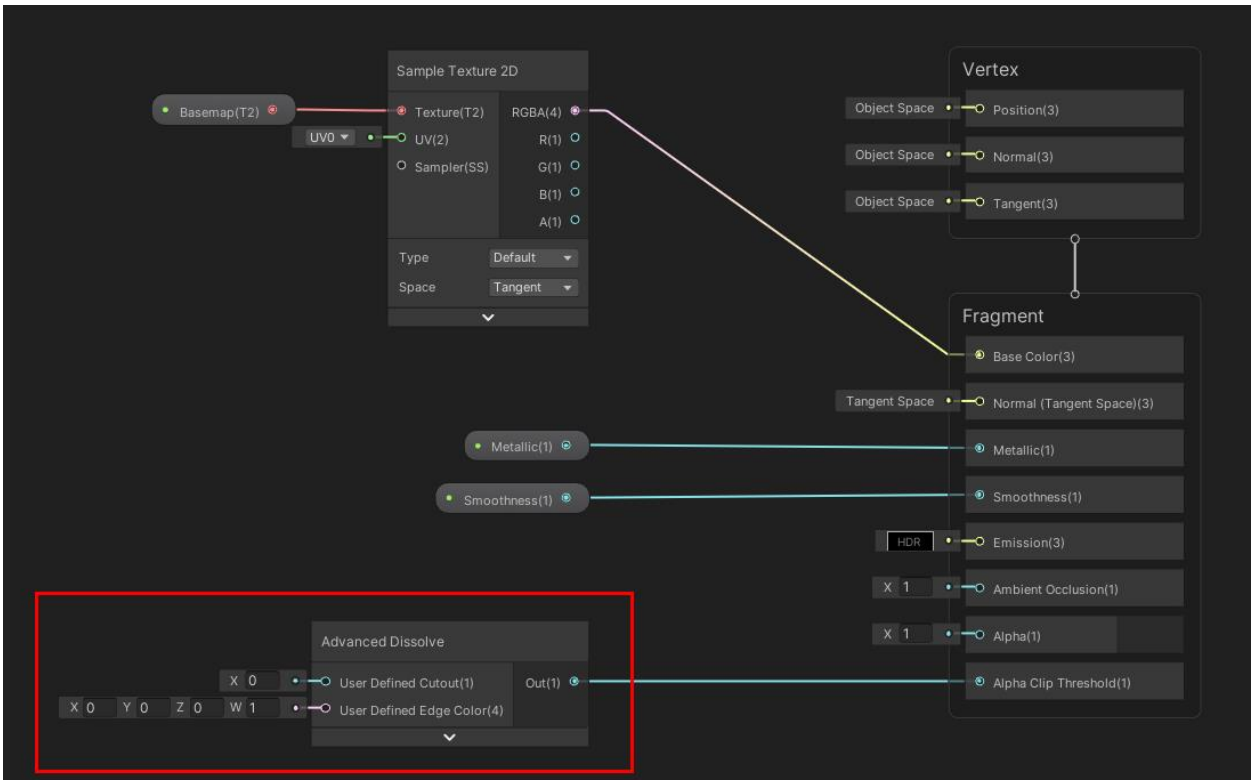
Advanced Dissolve effect cannot be created inside Unity ShadeGraph using just a node, however it can be integrated into HSLS shader file generated from ShaderGraph.

Here are all required steps:

1. (This step is only for Unity 2020.1 and newer versions) Inside ShaderGraph settings make sure **Alpha Clip** checkbox is turned on. This will add **Alpha Clip Threshold** filed to the master node.



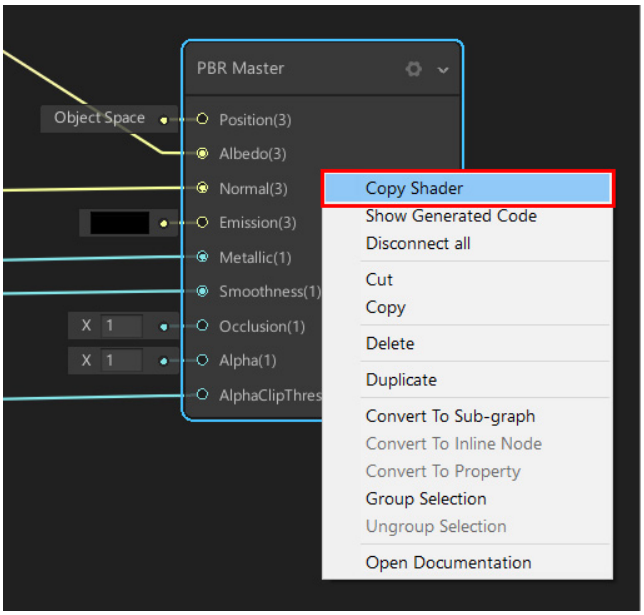

2. Add **Advanced Dissolve** node and connect its output to the **Alpha Clip Threshold** filed.



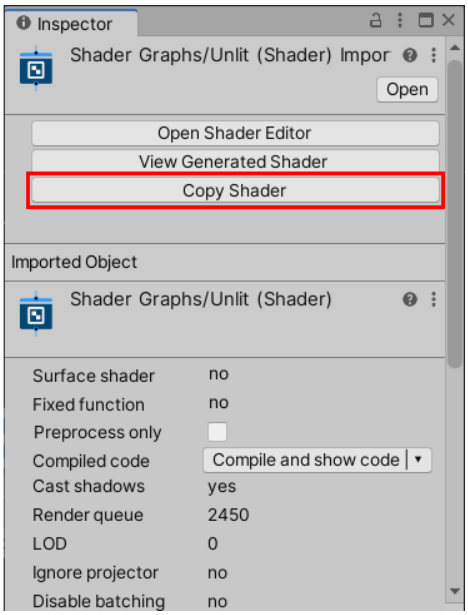

**Advanced Dissolve** node itself does nothing here. Its output value is always 0. This node serves as a marker for HLSL shader generator.

3. When shader is ready and **Advanced Dissolve** node has been added, save it and after that copy shader code.

   In Unity 2019.4 this can be done by selecting Master node inside Shader Graph and choosing **Copy Shader** from context menu.
   In Unity 2020.1 and later versions by selecting ***.shadergraph*** file inside Project window and in the Inspector window clicking on the **Copy Shader** button.



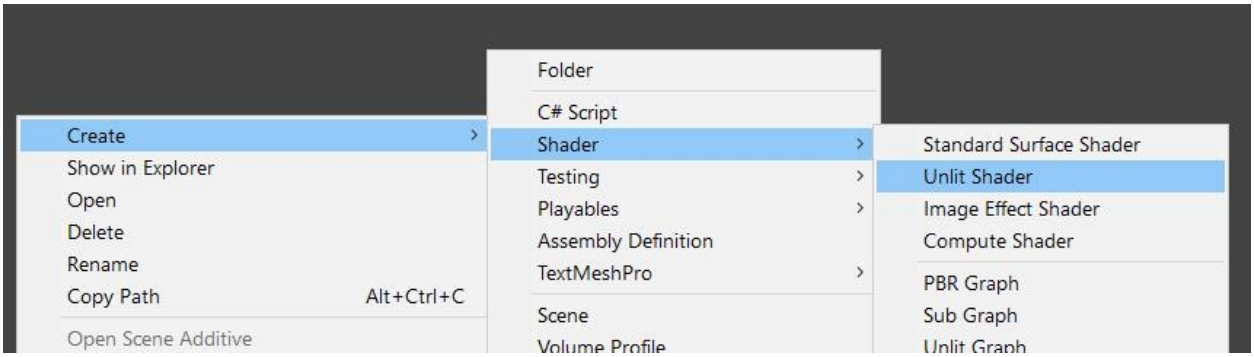

(Unity 2019.4)



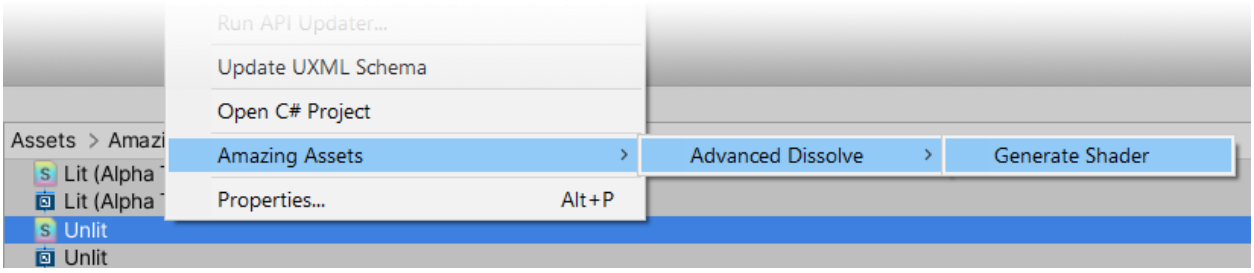

(Unity 2020.2)

   This copies ShaderGraph code into the system memory.

4. Inside Project window create basic Unlit shader (not ShaderGraph Unlit) and rename it to have the same name as ***.shaderGraph*** file.



5. Select just created shader file and from the context menu choose **Amazing Assets/Advanced Dissolve/Generate Shader**.



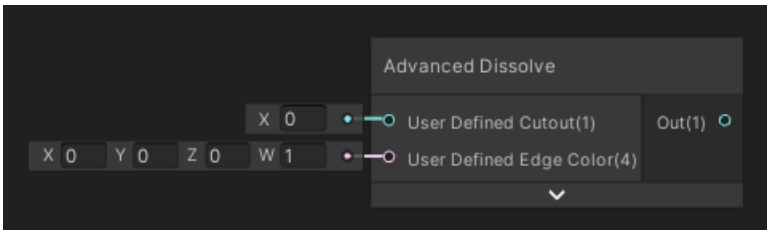

This will generate HLSL shader exactly the same as ShaderGraph file, but with integrated Advanced Dissolve features. Shader is ready to be used in materials.

Keep original ***.shaderGraph*** file and if in the future some changes are required, just repeat steps 3 and 5 (creating new Unlit shader file is not necessary, as copied shader code can be generated into already existing and used shader file).

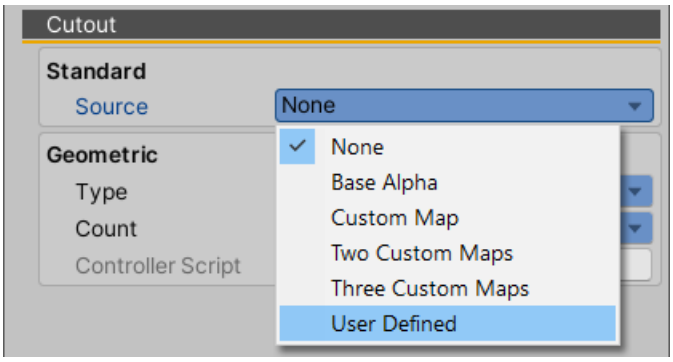
Note, ignore shader warnings, they are related to the Unity core rendering files and nothing can be fixed there.

Note, **Generate Shader** option in the context menu is inactive if ShaderGraph code is not copied into the system memory.

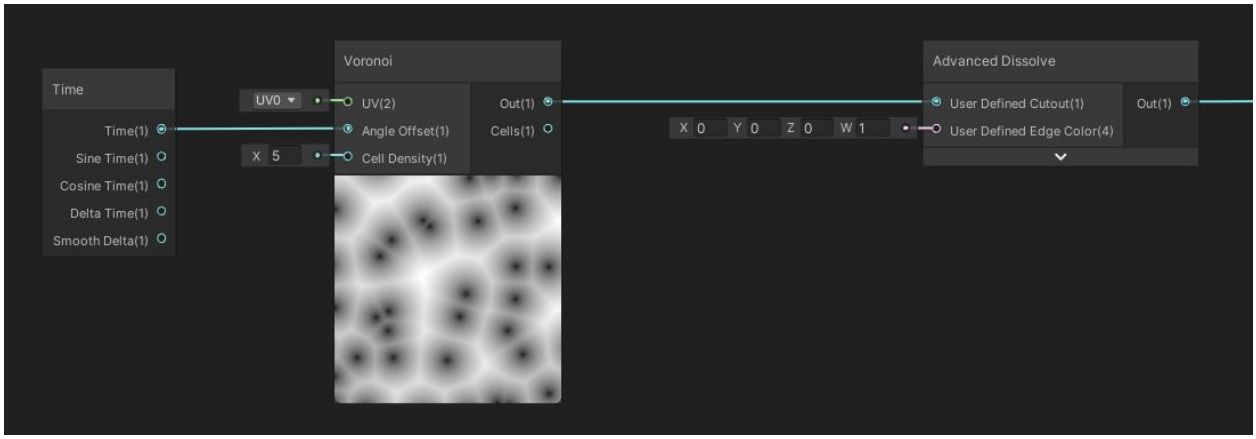
Advanced Dissolve node has two optional input:



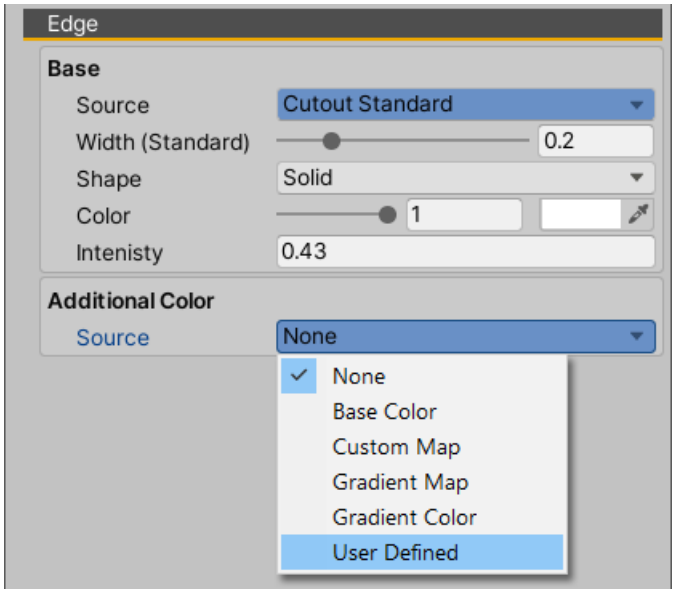

- **User Defined Cutout** – Advanced Dissolve shaders calculate cutout effect from **Standard** maps or using **Geometric** equations. With this option users can define their own cutout effect and make it active from **Cutout Standard** group by enabling **User Defined** option.



Example of self-animated dissolve effect using Voronori node:



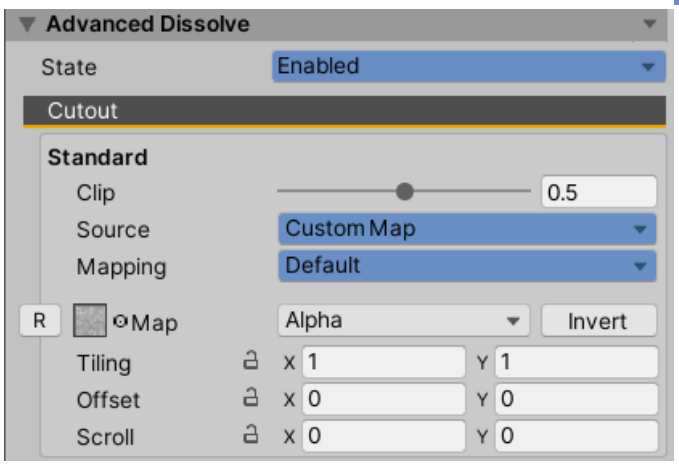

- **User Defined Edge Color** – Allows to use custom user defined color for edge visualization. If node uses this input, it can be made active in generated shader by enabling **User Defined** option in the **Edge Additional Color** group.



# KEYWORDS

Advanced Dissolve is not just a shader, it is a collection of various multiple effects that can be turned on/off any time, mixed together or used separately. Each of those effects is controlled using its own individual shader keyword.

All Advanced Dissolve keywords in material editor have blue background.




Advanced Dissolve shader has 9 groups of effects controlled by 39 keywords.

Note, in Unity 2019.4 and later versions all keywords are **local** and they do not contribute towards Unity's global keyword count limit (256).
In Unity 2018.4 there is no definition of **local** keywords and all Advanced Dissolve keywords are part of the project keywords.

1. **State** (2):
   - `_AD_STATE_DISABLED`
   - `_AD_STATE_ENABLED`

2. **Cutout Standard Source** (6):
   - `_AD_CUTOUT_STANDARD_SOURCE_NONE`
   - `_AD_CUTOUT_STANDARD_SOURCE_BASE_ALPHA`
   - `_AD_CUTOUT_STANDARD_SOURCE_CUSTOM_MAP`
   - `_AD_CUTOUT_STANDARD_SOURCE_TWO_CUSTOM_MAPS`
   - `_AD_CUTOUT_STANDARD_SOURCE_THREE_CUSTOM_MAPS`
   - `_AD_CUTOUT_STANDARD_SOURCE_USER_DEFINED`

3. **Cutout Standard Source Maps Mapping Type** (3):
   - `_AD_CUTOUT_STANDARD_SOURCE_MAPS_MAPPING_TYPE_DEFAULT`
   - `_AD_CUTOUT_STANDARD_SOURCE_MAPS_MAPPING_TYPE_TRIPLANAR`
   - `_AD_CUTOUT_STANDARD_SOURCE_MAPS_MAPPING_TYPE_SCREEN_SPACE`

4. **Cutout Geometric Type** (7):
   - _AD_CUTOUT_GEOMETRIC_TYPE_NONE
   - _AD_CUTOUT_GEOMETRIC_TYPE_XYZ
   - _AD_CUTOUT_GEOMETRIC_TYPE_PLANE
   - _AD_CUTOUT_GEOMETRIC_TYPE_SPHERE
   - _AD_CUTOUT_GEOMETRIC_TYPE_CUBE
   - _AD_CUTOUT_GEOMETRIC_TYPE_CAPSULE
5. **Cutout Geometric Count** (4):
   - _AD_CUTOUT_GEOMETRIC_COUNT_ONE
   - _AD_CUTOUT_GEOMETRIC_COUNT_TWO
   - _AD_CUTOUT_GEOMETRIC_COUNT_THREE
   - _AD_CUTOUT_GEOMETRIC_COUNT_FOUR

6. **Edge Base Source** (4):
   - _AD_EDGE_BASE_SOURCE_NONE
   - _AD_EDGE_BASE_SOURCE_CUTOUT_STANDARD
   - _AD_EDGE_BASE_SOURCE_CUTOUT_GEOMETRIC
   - _AD_EDGE_BASE_SOURCE_ALL

7. **Edge Additional Color Source** (6):
   - _AD_EDGE_ADDITIONAL_COLOR_NONE
   - _AD_EDGE_ADDITIONAL_COLOR_BASE_COLOR
   - _AD_EDGE_ADDITIONAL_COLOR_CUSTOM_MAP
   - _AD_EDGE_ADDITIONAL_COLOR_GRADIENT_MAP
   - _AD_EDGE_ADDITIONAL_COLOR_GRADIENT_COLOR
   - _AD_EDGE_ADDITIONAL_COLOR_USER_DEFINED

8. **Edge UV Distortion Source** (2):
   - _AD_EDGE_UV_DISTORTION_SOURCE_DEFAULT
   - _AD_EDGE_UV_DISTORTION_SOURCE_CUSTOM_MAP

9. **Global Control ID** (5):
   - _AD_GLOBAL_CONTROL_ID_NONE
   - _AD_GLOBAL_CONTROL_ID_ONE
   - _AD_GLOBAL_CONTROL_ID_TWO
   - _AD_GLOBAL_CONTROL_ID_THREE
   - _AD_GLOBAL_CONTROL_ID_FOUR

Note, keywords in shaders can be controlled from scripts using Unity Material.EnableKeyword and Material.DisableKeyword methods, or using Advanced Dissolve controller script explained in chapter **Runtime API**.

Note, SRP Batcher compatible shaders are keywords free. All used keywords are baked inside shader and unused ones are removed.

# RUNTIME API

Advanced Dissolve package includes 2 files describing core classes and 3 controller scripts. They can be brought into scope with `using AmazingAssets.AdvancedDissolve;` directive.

## CORE CLASSES

**AdvancedDissolveKeywords** – This class is designed for working with Advanced Dissolve shader keywords for a material.

```
static public void SetKeyword(Material material,    AdvancedDissolve.Keyword keyword, bool enable)
static public void SetKeyword(Material[] materials, AdvancedDissolve.Keyword keyword, bool enable)
```

Sets Advanced Dissolve keyword for selected material(s).

- **material(s)** – Target material whose keyword is being changed.
- **keyword** – Advanced Dissolve shader keyword *enum* identifier. (AdvancedDissolveKeywords.cs file lines: 11-19)
- **enable** – Value of *true* enables selected keyword, *false* – disables.

Example of enabling **State** keyword:

```
AdvancedDissolveKeywords.SetKeyword(material, AdvancedDissolveKeywords.State.Enabled, true);
```

Example of disabling **State** keyword (turns off Advanced Dissolve effect):

```
AdvancedDissolveKeywords.SetKeyword(material, AdvancedDissolveKeywords.State.Enabled, false);
```

Example of enabling **Two Custom Maps** for Standard Cutout:

```
AdvancedDissolveKeywords.SetKeyword(material, AdvancedDissolveKeywords.CutoutStandardSource.TwoCustomMaps, true);
```

Example of enabling **4 Spheres** for Geometric cutout:

```
AdvancedDissolveKeywords.SetKeyword(material, AdvancedDissolveKeywords.CutoutGeometricType.Sphere, true);
AdvancedDissolveKeywords.SetKeyword(material, AdvancedDissolveKeywords.CutoutGeometricCount.Four, true);
```

```
static public void GetKeyword(Material material, AdvancedDissolve.Keyword keyword, bool enable)
```

Retrieves Advanced Dissolve keyword from material.

```
static public void RemoveAll(Material material, bool ignoreState)
```

Removes all Advanced Dissolve keywords from material.

**AdvancedDissolveProperties** - This class exposes all Advanced Dissolve properties from a material, allowing you to animate them.

```
static public void UpdateLocalProperty(Material material, Property property, object value)
```

Updates selected material's property.

- **Material** – Target material whose property is being changed.
- **Property** – Shader property *enum* identifier.
- **Value** – Property value in *float*, *int*, *color*, *texture* etc. format.

Example of updating **Clip** value in Standard Cutout group:

```
AdvancedDissolveProperties.Cutout.Standard.UpdateLocalProperty
(
	material,
	AdvancedDissolveProperties.Cutout.Standard.Property.Clip,
	#VALUE#
);
```

Example of updating **Color** value in Edge Base group:

```
AdvancedDissolveProperties.Edge.Base.UpdateLocalProperty
(
	material,
	AdvancedDissolveProperties.Edge.Base.Property.Color,
	#VALUE#
);
```

Example of updating **Color Intensity** value in Edge Additional Color group:

```
AdvancedDissolveProperties.Edge.AdditionalColor.UpdateLocalProperty
(
	material,
	AdvancedDissolveProperties.Edge.AdditionalColor.Property.ColorIntensity,
	#VALUE#
);
```

In all three examples above method names have following structure:
AdvancedDissolveProperties.***Section***.***Group***.***Scope***()

- ***Section*** – Shader section name same as in the material editor. Can be **Cutout** or **Edge**.
- ***Group*** – group name in selected section.
  For Cutout section it can be: **Standard** or **Geometric**.
  For Edge section it can be: **Base**, **AdditionalColor**, **UVDistortion** or **GlobalIllumination**.
- ***Scope*** – If updating material's local property this is `UpdateLocalProperty`. In the case of updating global property – `UpdateGlobalProperty`.

```
static public void UpdateGlobalProperty(GlobalControlID ID, Property property, object value)
```

Updates global property.

- **ID** – AdvancedDissolveKeywords.GlobalControlID *enum*.
- **Property** – Shader property *enum* identifier. Depends on used ***Section*** and ***Group***.
- **Value** – Property value in *float*, *int*, *color*, *texture* etc. format.

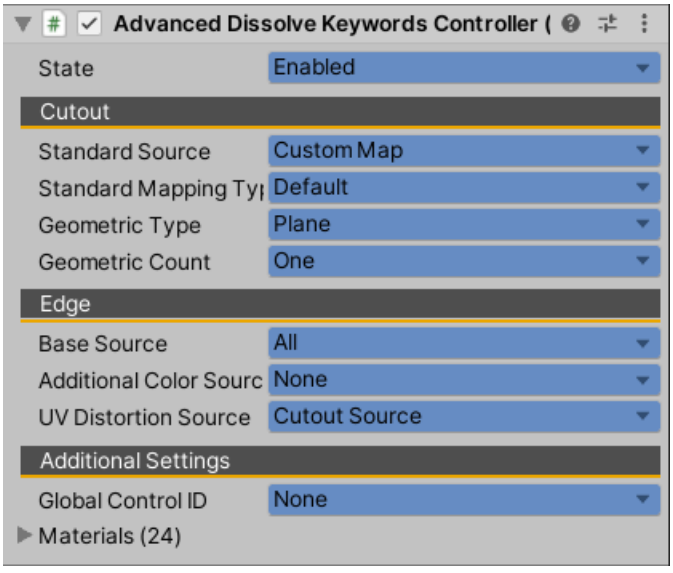
Example of updating **Clip** value in Standard Cutout group for materials with global **ID One**.

```
AdvancedDissolveProperties.Cutout.Standard.UpdateGloabalProperty
(
	AdvancedDissolveKeywords.GlobalControlID.One,
	AdvancedDissolveProperties.Cutout.Standard.Property.Clip,
	#VALUE#
);
```

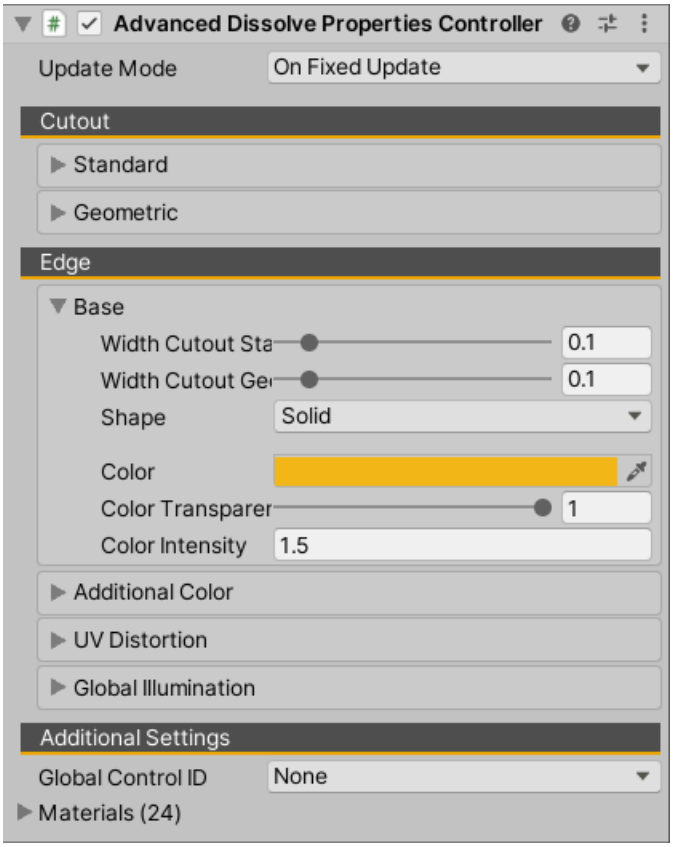
## MONOBEHAVIOUR CONTROLLER CLASSES

Advanced Dissolve package includes 3 `MonoBehaviour` controller scripts.

**AdvancedDissolveKeywordsController** – Used for setting up Advanced Dissolve keywords for materials. This script is used in all cutout example scenes for assigning proper keywords to the materials used in those scenes.



**AdvancedDissolvePropertiesController** – Used for setting up Advanced Dissolve properties.




**AdvancedDissolveGeometricCutoutController** – Used for updating Geometric cutout effect.