UVPaint

Unlimited Decals on your characters from iRobi

Thank you so much for your support! If you have any questions, suggestions, or you have found some bugs, you are welcome to Unity Forum:

https://forum.unity3d.com/threads/uvpaint-skinned-mesh-decal-system-released.401458

Follow us:

https://www.facebook.com/groups/iRobiDev/

https://twitter.com/iR0biDev

Introduction:

UVPaint creates decals and immediately bakes them in needed ShaderProperty texture on 3D objects. Thus, you can endlessly impose a variety of images directly into the model without losing performance and video memory. You can also return to the original texture state using Restore Tool (Erase).

New **Flexi** mode allowing to work with any shaders (including PBR) just putting ShaderPropertyName in call settings of the plugin is now available in UVPaint. It is also possible to change Normal Map, Occlusion Map, Height Map and anything else you want!

You need 3D model with the UNWrapped UV (for example character model) for correct plugin work.

We suggest you to work without overlapping UV islands on each others on UV map for the best effect and result.

How to use:

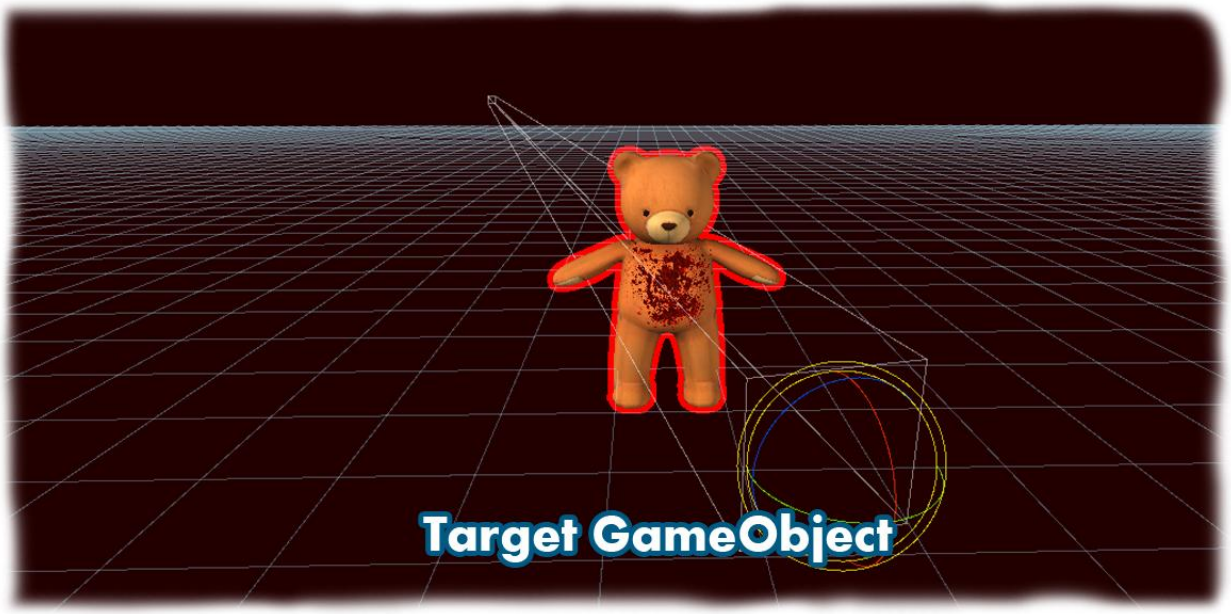
1. To get started with the plugin, you need to declare the namespace **iRobi** at the beginning of the script as in this example:

```
1 using UnityEngine;
2 using System.Collections;
3 using iRobi;
4
5 public class Paint_ball : MonoBehaviour {
6     int frame;
7     Vector3 Direct VECTOR;
```

2. To apply decals you need to call just one function with parameters :

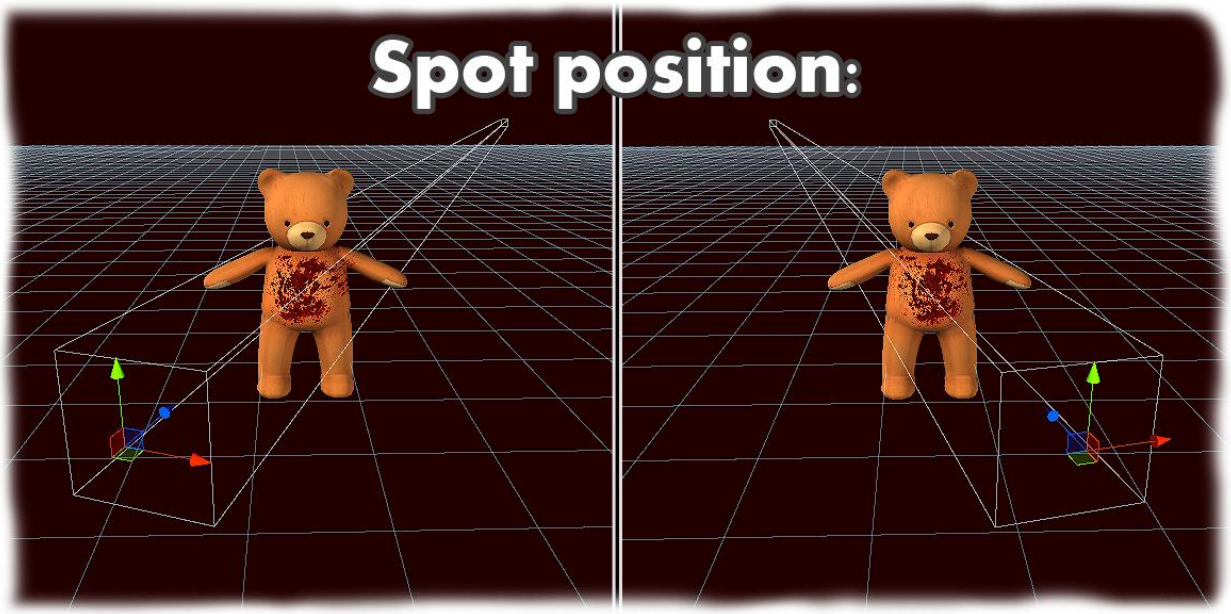
UVPaint.Create(**TargetGameObject**, SpotPosition, SpotRotation, Decal, UVPaint_Options[]**)**;

TargetGameObject: UnityEngine.GameObject Type.



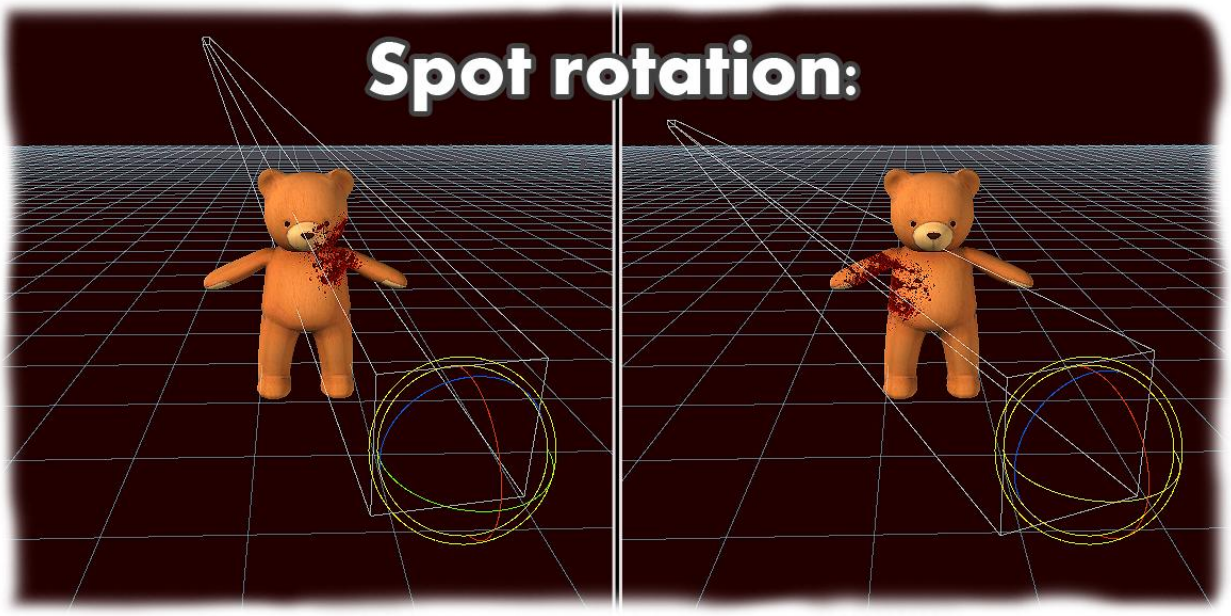
UVPaint.Create(TargetGameObject, **SpotPosition**, SpotRotation, Decal, UVPaint_Options[]**)**;

SpotPosition: UnityEngine.Vector3 Type.



UVPaint.Create(TargetGameObject, SpotPosition, **SpotRotation**, Decal, UVPaint_Options[]**);**

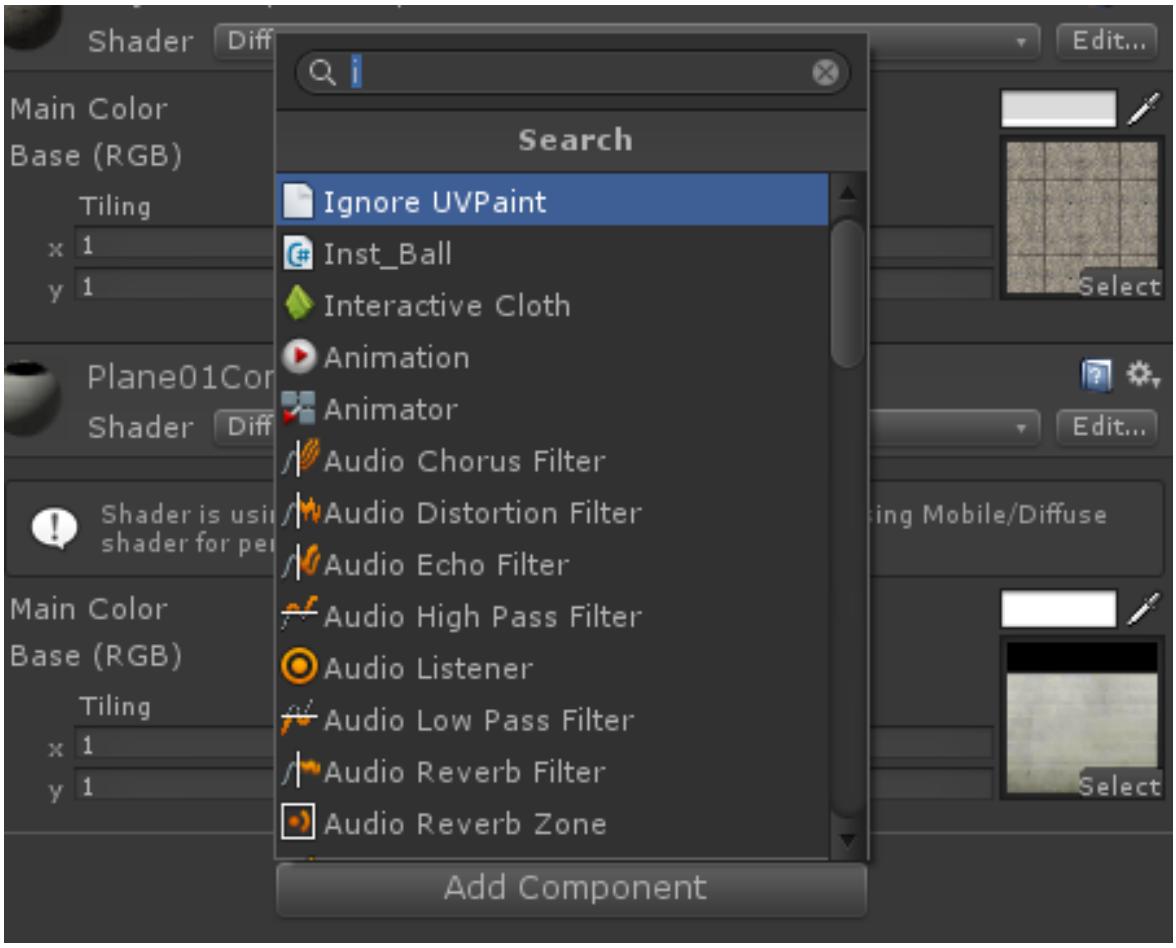
SpotRotation: UnityEngine.Quaternion Type.



UVPaint.Create(TargetGameObject, SpotPosition, SpotRotation, **Decal**, UVPaint_Options[]**);**

Decal: UnityEngine.Texture2D Type.

If you want to protect a specific object from dealing of decals, just add Component **Ignore UVPaint**:



Also you can save changed Textures by filePath:

UVPaint.FIX(Renderer REN, string PathToSaveImage **)**

REN: Renderer with decaled Textures.

PathToSaveImage: Only folder. Texture names will generate automatic.

UVPaint Options:

UVPaint.Create(TargetGameObject, SpotPosition, SpotRotation, Decal, **UVPaint_Options[])**;

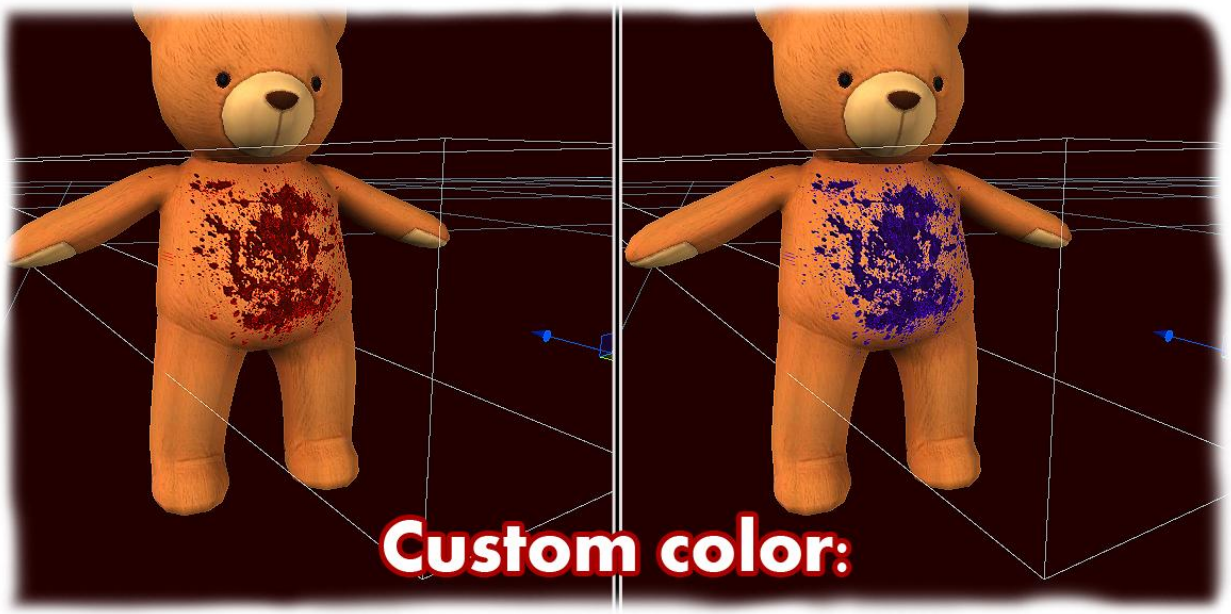
UVPaint_Options[]: Options

The following options allow you to customize the applying of decals according to your liking and desire with their maximal variety.

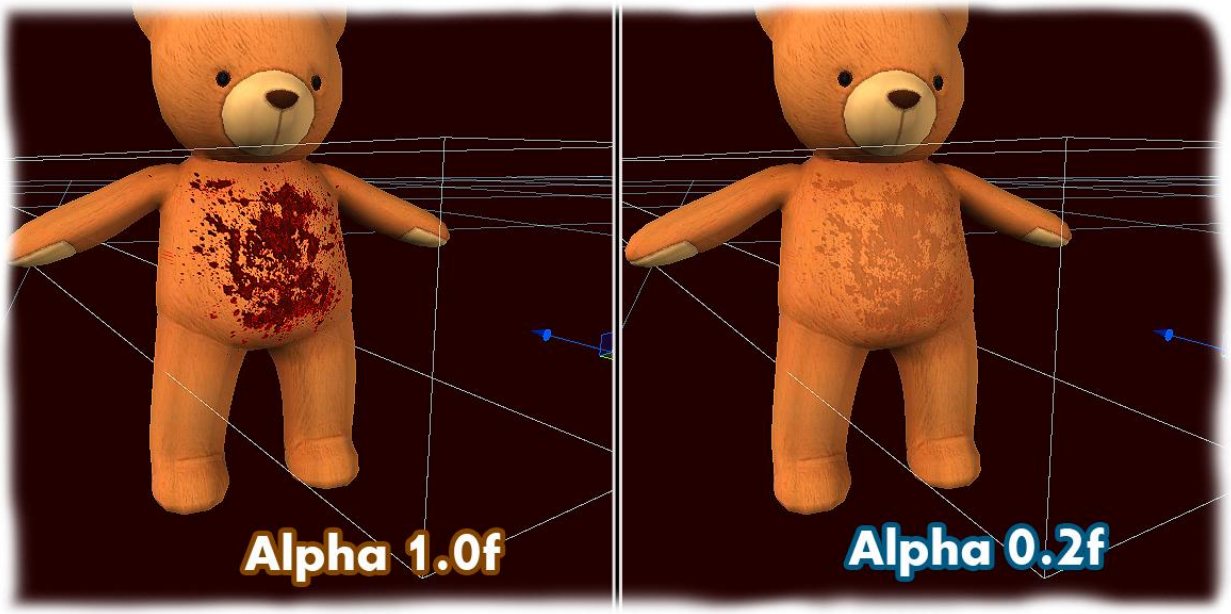
There are two ways of writing options right now. The first is before calling the main function, and the second is directly calling within the function:

```
ContactPoint contact = collision.contacts[0];//Directin of applying Decal are starting from this point.
trans.forward=-trans.forward;//Mirroring Decal Direction, because this Transform are looking back.

UVPaint.Normalized_DeepAlpha (1f);//Also you can you options like this. Before UVPaint.Create();
UVPaint.Create(gameObject,contact.point,trans.rotation,f1,UVPaint.DecalColor(color),UVPaint.DecalSize(Random.Range(1.5f,3f)));

Destroy(gameObject);//Destroy after applying Decal.
```

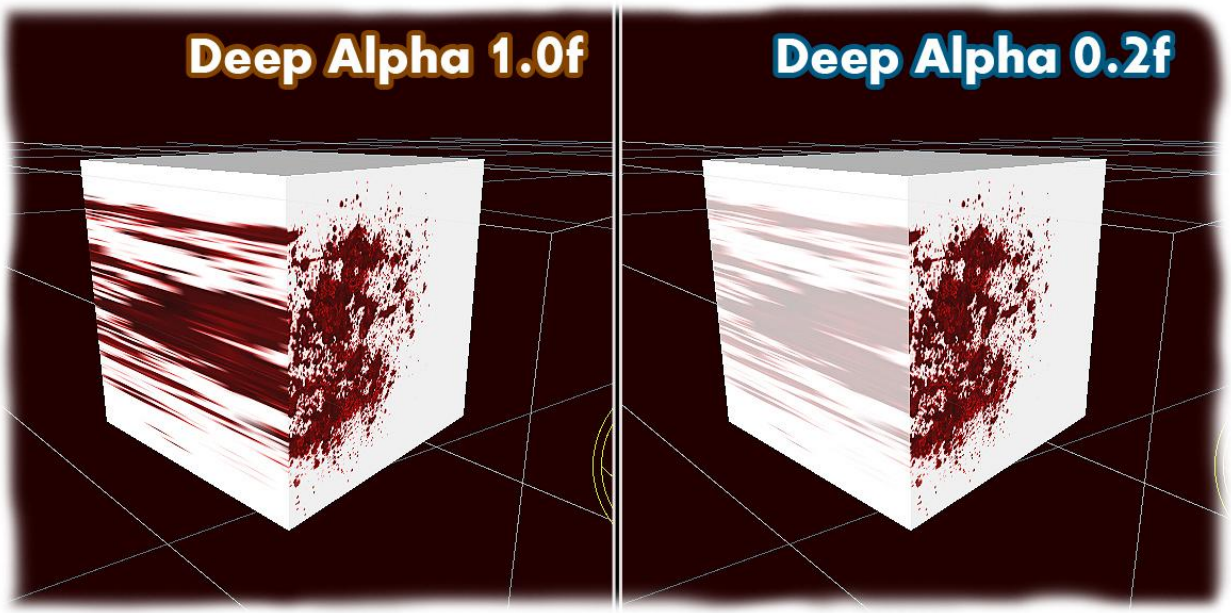
UVPaint.DecalColor(Type UnityEngine.Color). **White** by default.



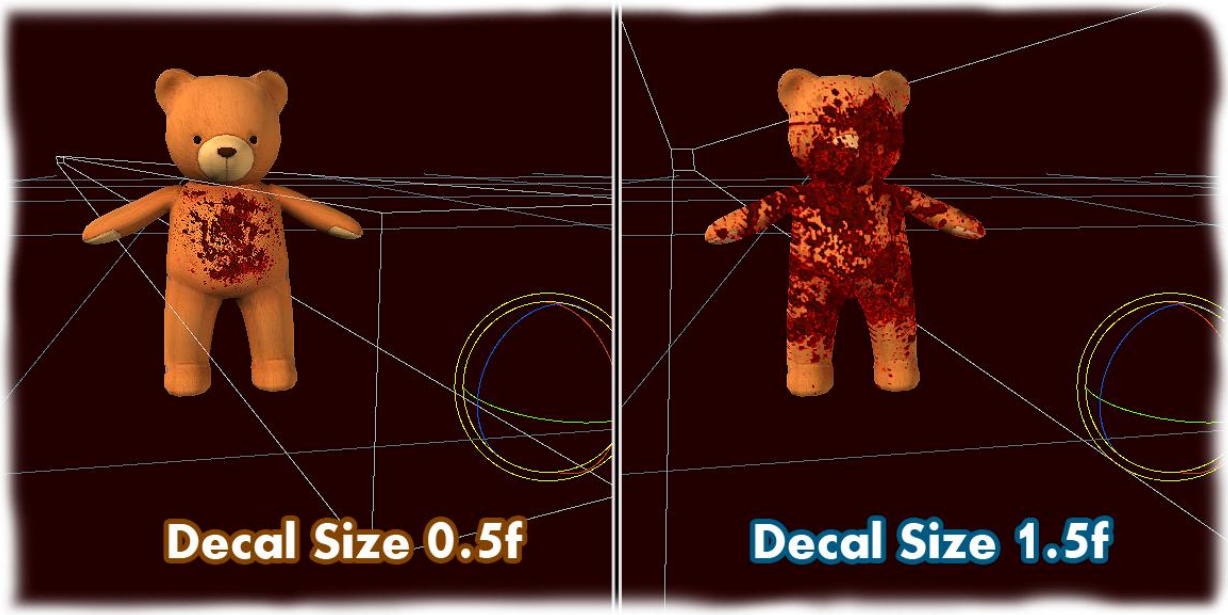
UVPaint.NormalizedAlpha(Type float). **1f** by default.



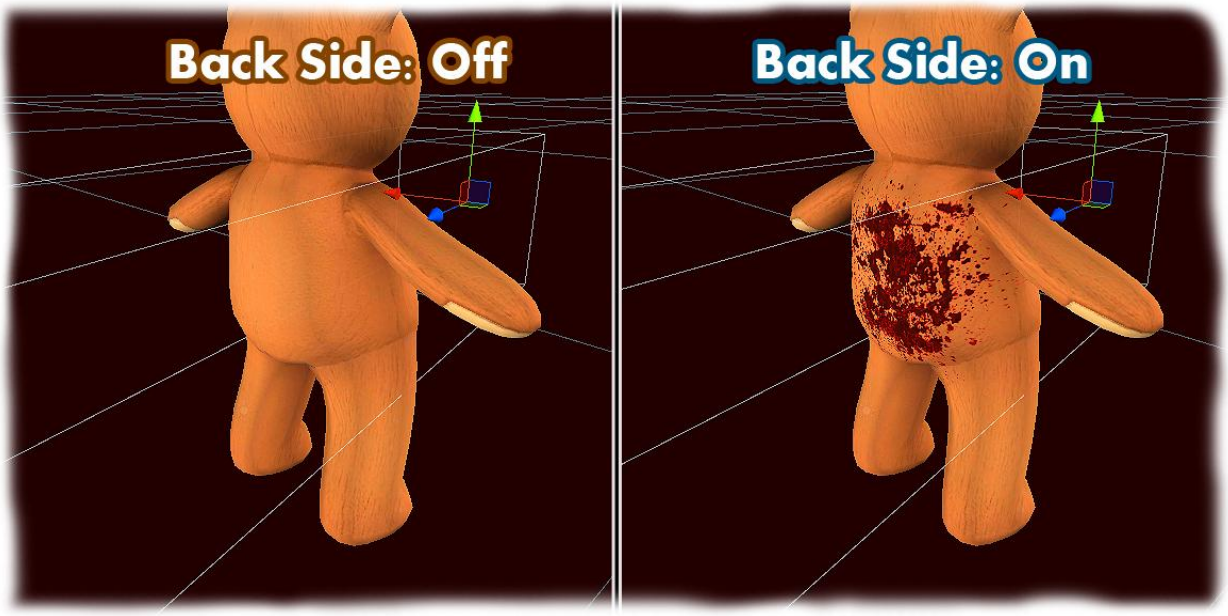
UVPaint.Normalized_DeepAlpha(Type float). **1f** by default.



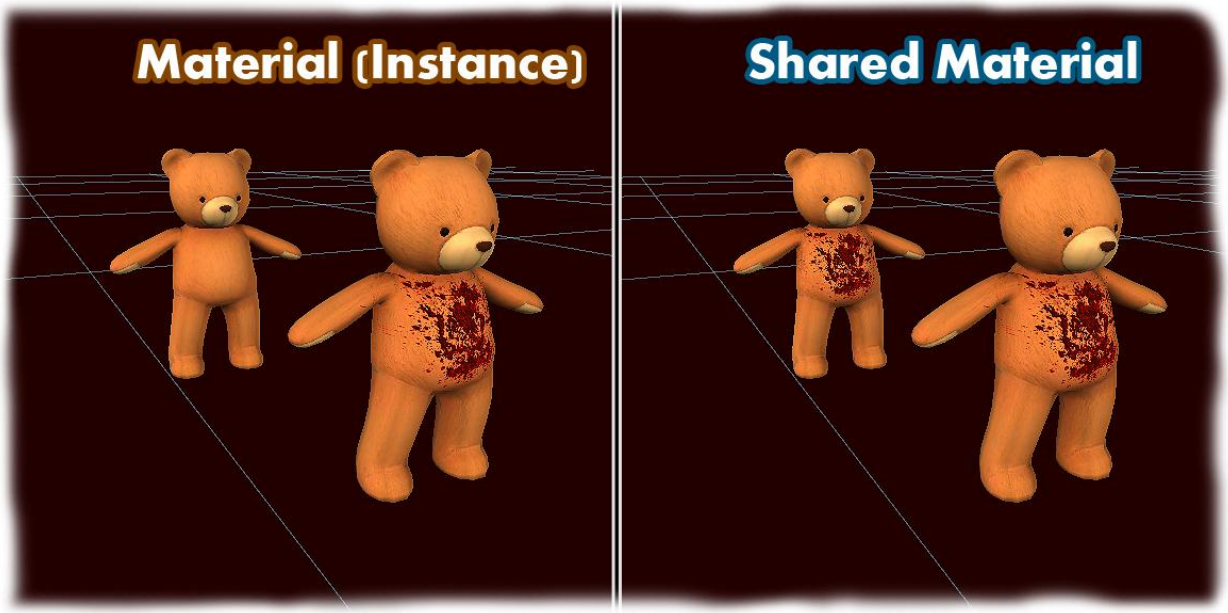
UVPaint. DecalSize(Type float). 1f by default.



UVPaint. isTwoSided (Type bool). false by default.



UVPaint. EditAsSharedMaterial (Type bool). **false** by default. It does't work on **Instaced** materials. Once using the **True**, the material will become **Instance**.



UVPaint. ShaderPropertyName (Type string). Texture Property of the Shader with which UVPaint must work. For Example *"_BumpMap".*

"_MainTex" by default.

UVPaint. AngleInDegrees (Type float). This option allows you to change the rotation relative the **Z** axis. From 0f to 360f. **0f** by default.

Restore Tool:

With the Restore Tool you can erase decals from 3D models (in this case, decal's texture will act as an eraser).

UVPaint. Restore (TargetGameObject, SpotPosition, SpotRotation, Decal, UVPaint_Options[]**)**

Also, you can instantly return 3D model's texture to its original state using the Instant Restore:

UVPaint.FullRestore(TargetGameObject, UVPaint_Options[]**)**

Save State Tool is necessary for saving the restore point, but you have to understand that it's impossible to return to the original texture before the restart of scene or material:

UVPaint.SaveStateForRestore(TargetGameObject, UVPaint_Options[]**)**

The example of saving Restore State is in Color Play scene (Teddy).

That is all. We will do our best to try to add new features. Thank you once again for your support. iRobi.

Forum:

https://forum.unity3d.com/threads/uvpaint-skinned-mesh-decal-system-released.401458

Follow us:

https://www.facebook.com/groups/iRobiDev/

https://twitter.com/iR0biDev