



Contact : [jurjen@uhm-uhm.com](mailto:jurjen@uhm-uhm.com) Web: [www.uhm-uhm.com](http://www.uhm-uhm.com)

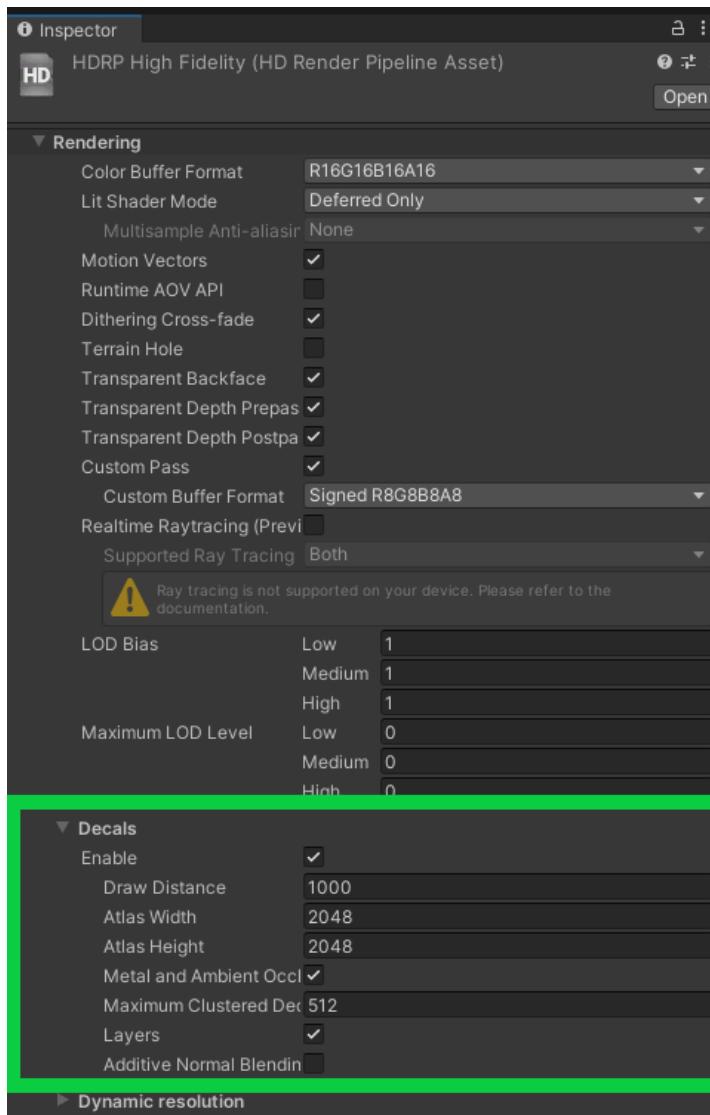
# Decal Splines User Manual 1.4.2

## Contents:

<b>HDRP Setup:</b>	2
<b>URP Setup:</b>	4
<b>Adding a Decal Spline:</b>	6
<b>Scene view:</b>	7
<b>Inspector:</b>	9
<b>Fade Paint:</b>	12
<b>Controls:</b>	13
<b>Appearance:</b>	14
<b>Asset menu:</b>	15
<b>Styles:</b>	16
<b>Themes:</b>	18
<b>Textures:</b>	18
<b>Shaders:</b>	20
<b>Rigging 3D models:</b>	23
<b>Script files:</b>	24
<b>Runtime support:</b>	25
<b>Compatibility:</b>	27

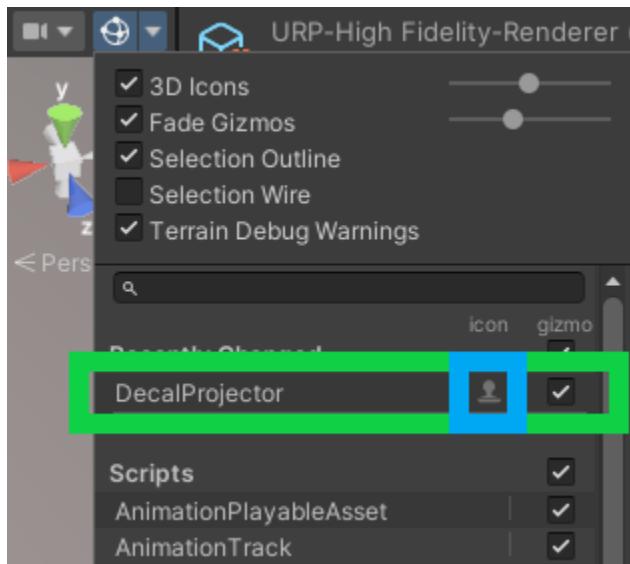
# HDRP Setup:

1. **Import** the **HDRP assets** by opening the “HDRP assets.unitypackage” file located in the main folder. All errors should now disappear.
2. Decal splines make use of the built-in Unity decal projectors, in order for them to function the decals need to be configured correctly. Open the inspector of your **HD Render Pipeline Asset** file. Here make sure Decals are Enabled.



3. Because of the extensive use of decal projectors the gizmo icons can get in the way. Disabling them is advised. On the top right of the scene editor window expand the Gizmos menu and find the decal projector item. Here you can disable the stamp icon by clicking the icon so that it toggles off. Don't fully

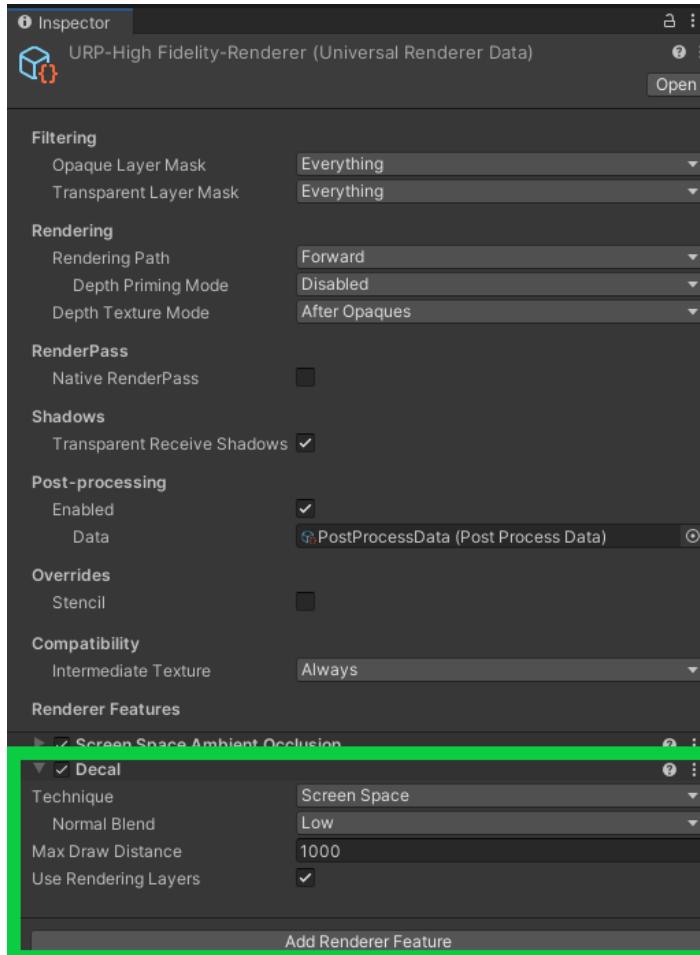
disable all gizmo's as doing this blocks Decal Splines's scene view functions.



4. By default decal splines draw on every object, to exclude certain objects from receiving decals , a “Decal layer” can be used. For this feature to function, enable **Layers** in the **HD Render Pipeline Asset**.

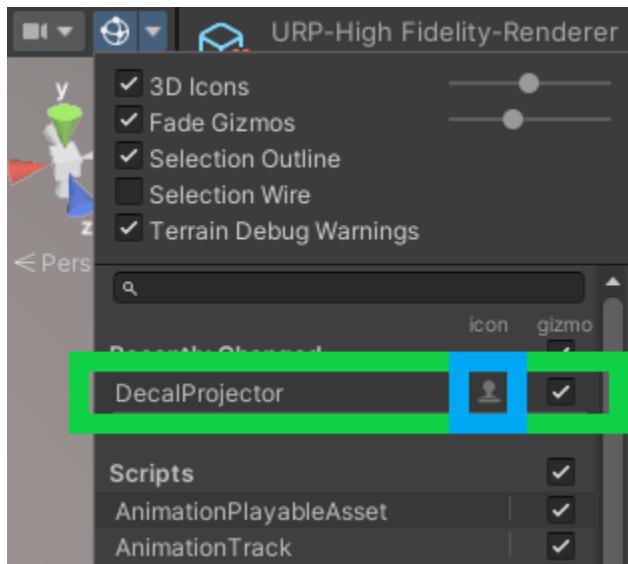
# URP Setup:

1. **Import** the **URP assets** by opening the “URP assets.unitypackage” file located in the main folder. All errors should now disappear.
2. Decal splines make use of the built-in Unity decal projectors, in order for them to function the decal projectors need to be configured correctly. Open the inspector of your **Universal Renderer Data** file and click “**Add Renderer Feature**”. Select “**Decal**” and add the component.



3. Under the component settings, set “**Technique**” to “**Screen Space**”.
4. Because of the extensive use of decal projectors the gizmo icons can get in the way. Disabling them is advised. On the top right of the scene editor window expand the Gizmos menu and find the decal projector item. Here you can disable the stamp icon by clicking the icon so that it toggles off. Don’t fully

disable all gizmo's as doing this blocks Decal Splines's scene view functions.



5. By default decal splines draw on every object, to exclude certain objects from receiving decals , a “Rendering layer” can be used. For this feature **enable “Use Rendering Layers”** A custom “Rendering layer” can be added by opening the **“Project Settings...”** under the edit menu. Find “URP Global Settings”. **Add** a new **rendering layer** and name it “decal splines”. This layer can now be used in the Decal Spline’s layer mask.

## Adding a Decal Spline:

After completing the setup you can add a Decal Spline to your scene. Right click the hierarchy window and under “**Uhm...Uhm..Games**” select “**Decal Spline**”.

A Decal spline object will be added to the scene.

Select the Decal spline object, so that the inspector window displays the object's properties.

The first step is to **rotate** the object so that the **white arrow** points in the direction of the surface you wish to paint on.

By default the decal spline is aimed towards the floor.

Next **assign a “Theme”** by pressing the circle icon located in the “Theme” section of the inspector right after “Active Theme”. After you selected a theme the style selection menu appears in the inspector.

**Select a style** by clicking the respective menu item in the style selection menu, the Decal Spline is now automatically set to place mode.

In **place mode** the scene view cursor turns into a circle. **Click in the scene view** to place a couple spline anchor points. Press the escape key to exit place mode. A continuous decal gets drawn along the spline.

**Edit** the spline by **clicking and dragging** the anchor points and handles. To place more points go back into place mode by pressing “Place” in the inspector menu. To **delete** a point, exit place mode and click an anchor point while holding down the **Left Control** key.

## Scene view:

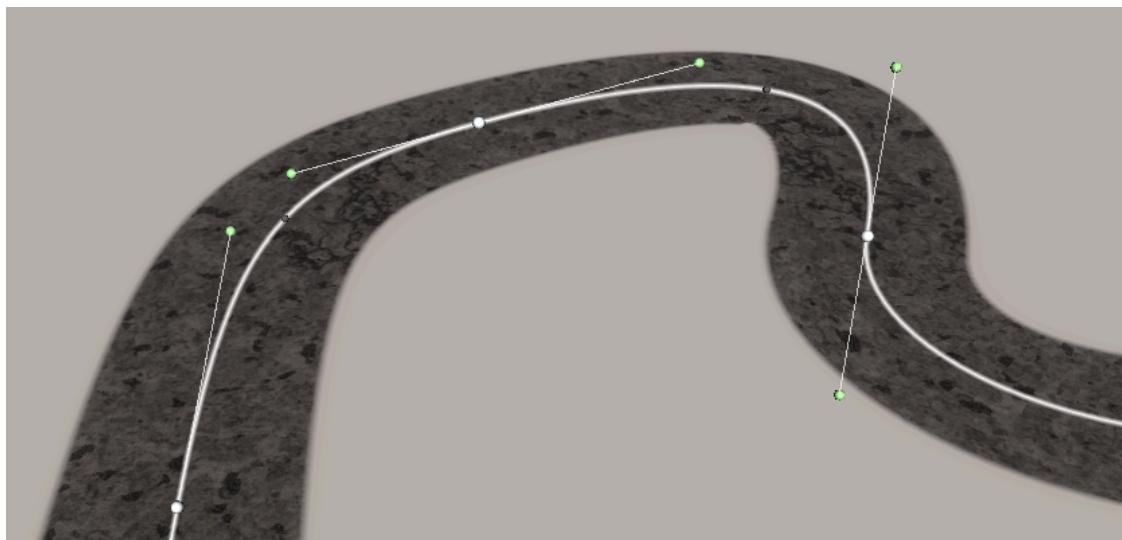
Splines can be edited directly in the scene view, when the “Decal Spline” object is selected the spline handles will appear in the scene view.

The different handle types have unique colors. By default the color assignment is as follows:

**Anchor points** are white.

**Curve handles** are Green.

**Insert handles** are Gray.



To move an anchor or curve point **click and drag** the respective handle and release at the desired position. Make sure to exit place mode before adjusting handle positions to prevent placing new segments on top.

The “**Auto snap**” setting determines the placement behavior of the handle.

With auto snap enabled the handle will stick to underlying objects, with this setting disabled the handle will be moved parallel to the scene camera. Clicking while holding the **left shift key** will temporarily **invert** the auto snap setting.

Anchor handles can get in the way when placing points, hold down **space** to **hide** the anchor handles.

Curve handles can be **locked and unlocked** by clicking them while holding down **left control**. This allows handles to be adjusted independently. By default the shade of green of an unlocked curve handle is a little darker.

To **remove** an anchor point, click the respective handle while holding down **left control**.

To **insert** an anchor point click on one of the insert handles of the spline.

Full **undo** functionality has been implemented, pressing **Control + Z** and **Control + Y** will undo/redo any action.

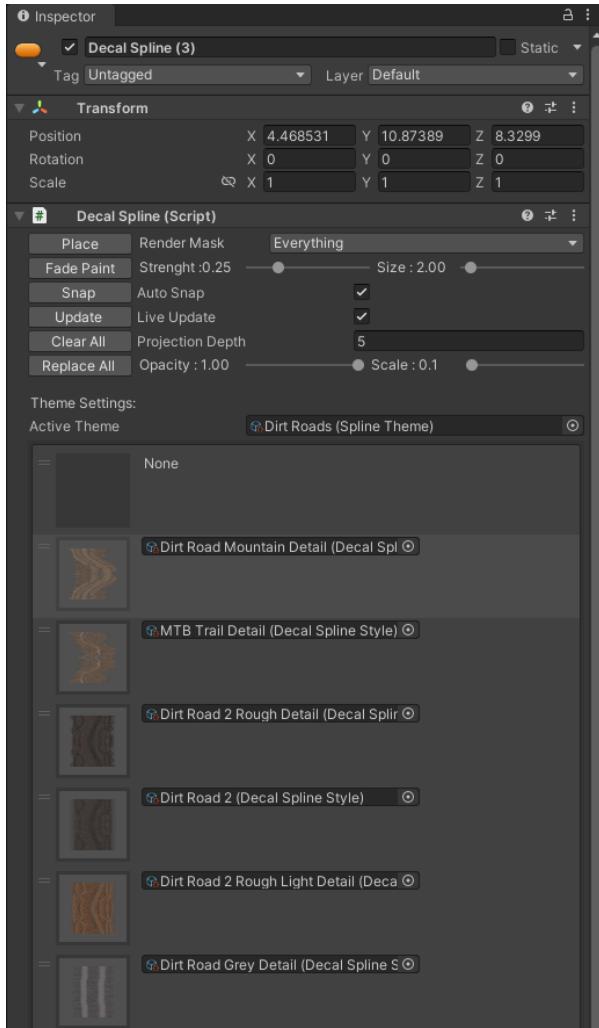
To place more spline segments, go into **Place mode** by clicking the place button in the inspector. And continue placing points in the scene view. Different decal styles can be combined in one single spline. There is also no restriction on using multiple themes. **Exit** place mode by pressing the **Escape key** while the scene view is selected.

If adjusting the spline is causing a frame rate drop, toggling “**Live update**” off will improve performance.

The **orientation** of the Decal Spline is displayed by the **white square** and **arrow** in the scene. These indicate the direction the decal will be projected and can be adjusted by rotating the Decal Spline’s transform.

# Inspector:

The Decal Spline is configured in the inspector window. Here you find settings regarding: **Placement**, **Style** and **Projector behavior**.



The “**Place**” button puts the Decal Spline into place mode.

The “**Fade Paint**” button puts the Decal Spline into fade paint mode.

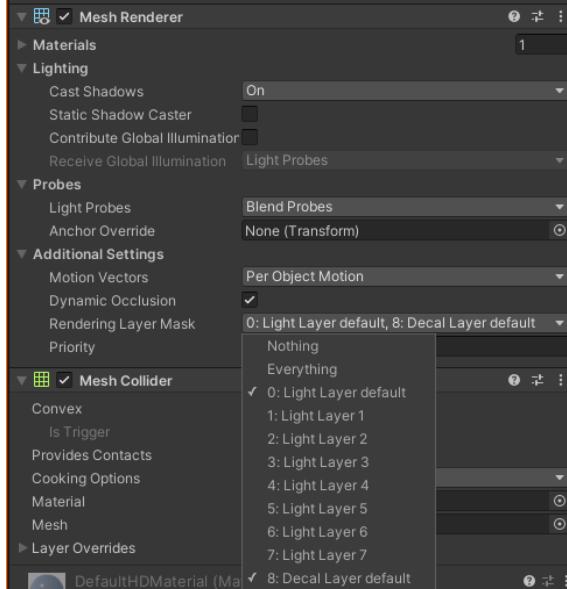
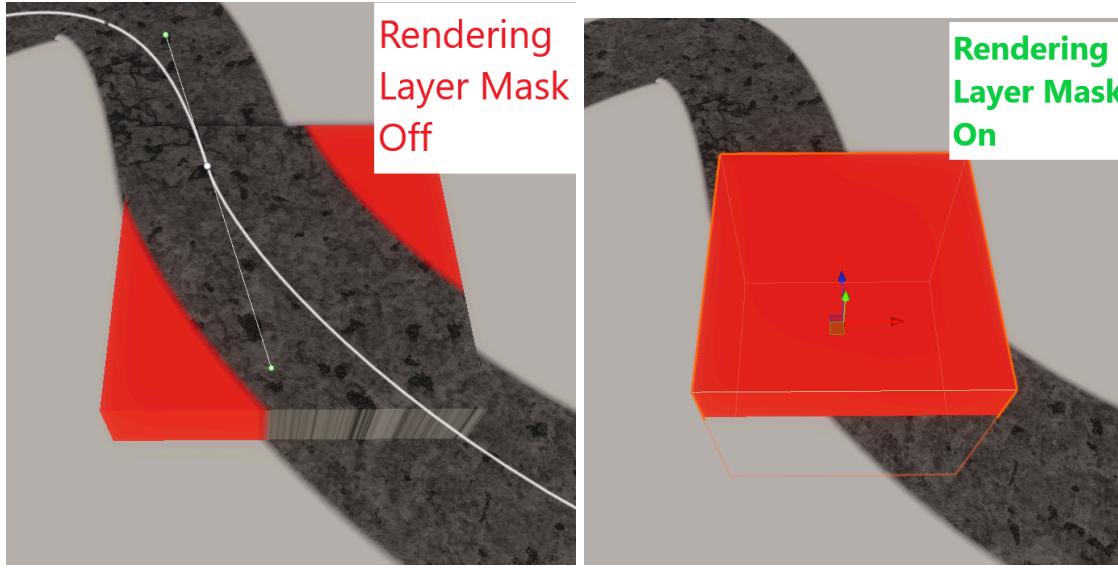
The “**Snap**” button will stick the spline down to the underlying objects.

The “**Update**” button updates all decal segments, this is used to force apply new settings in the case that a style has changed.

The “**Clear**” button completely wipes all placed spline segments. After pressing this button a user prompt pops up asking the user to confirm, press the “x” **close button** to **cancel** the clearing action.

The “**Replace All**” button changes the **entire Decal Spline** to the **currently selected style**.

The “**Render Mask**” controls the rendering layer mask used by the Decal Spline. When a mesh renderer’s render layer is not included in this mask, the decal will not be drawn on this object.



The “**Auto snap**” toggle will turn on and off the auto snap function.

The “**Live Update**” toggle will toggle the live update of the decal when adjusting the spline. When turned off the spline will only update after the mouse button is released.

The “**Projection Depth**” number determines the range the decal will be projected on. Any surface that is above or below the spline by less than the projection depth will receive the decal unless a layer mask is applied.

The “**Opacity**” controls the Decal Spline’s overall transparency. It scales from 0 to 1 in .01 increments giving precise control for the perfect blend.

The “**Scale**” slider works as a scalar for the width of the Decal Spline. It scales the width from 0.1 of the normal size to 5x the normal size.

The “**Theme**” settings directly come from the selected theme. Here you select the **active theme** and **style** that are used in place mode. Themes can be edited from here as well as directly from the theme inspector, for more info read “Themes”.

Selecting “**None**” under any theme will allow you to place an empty decal segment that can be used for **spacing**.

The **Strength slider** sets the **fade paint** strength.

The **Size slider** controls the size of the **fade paint area**.

## Fade Paint:

Fade paint mode is used to paint the Decal Spline's transparency for blending purposes. **Click** near an anchor point with the paint circle to **increase** its **transparency**. **CTRL click** to make it mode **opaque**.

The closer a point is to the mouse cursor the stronger the applied effect.

Used the **Strength** and **Size** sliders in the inspector to control the paint cursor strength and size respectively.



## Controls:

**Escape** - exit edit mode.

**Left shift** - flip auto snap while holding.

**Left control** - Remove anchor point / unlock curve handle/Increase opacity in Fade Paint.

**Space** - Hide anchor points.

## Appearance:

The look to the spline can be customized in the “Tool style” file found in the main folder.

Here you can change the **size** and **colors** of handles and the size and color of the lines used in the scene view.

## Asset menu:

The asset menu contains a “**Decal Splines**” tab, here you find the menu options used to create the components used to extend the default styles and themes.

The “**2D Decal**” option adds a 2D Decal Style asset to the project, this file type is used to define a 2D decal and configure its settings

The “**3D Decal**” option adds a 3D Decal Style asset to the folder, this file type is used to define a 3D decal and configure its settings.

The “**Material**” option adds a decal material to the project. This material is used to configure the looks of a decal style.

The “**Theme**” option adds a new Theme file to the project, here you define a new selection of styles used to organize them for easy selection during spline editing.

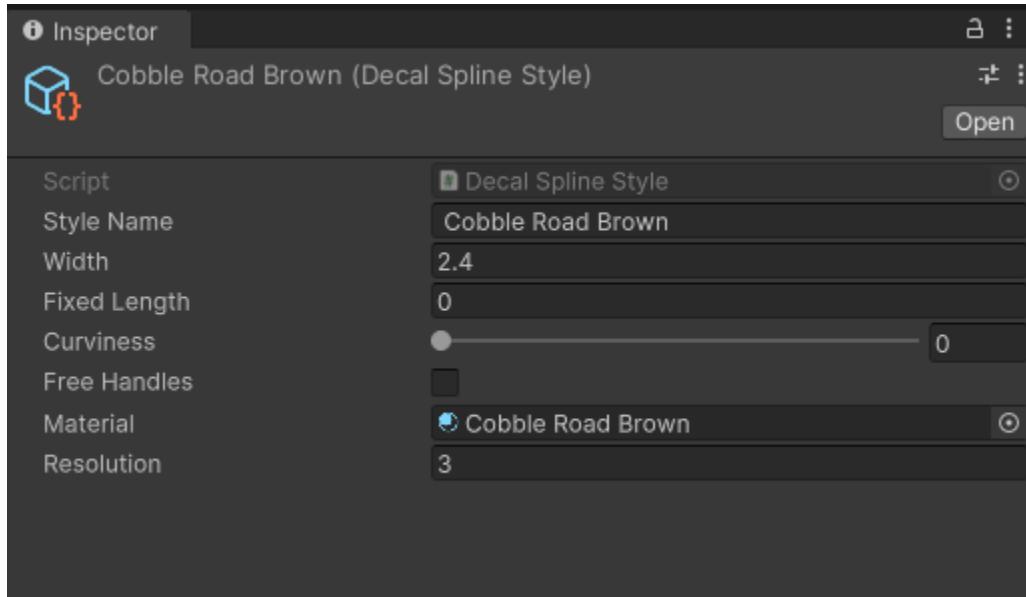
# Styles:

Style files are used to define the way a Decal looks. Two style types exist 2D and as a bonus 3D.

To use a style, they have to be added to a “Theme”, a single style can be added to multiple themes.

## 2D Decal:

In the inspector you'll find the following settings:



“**Style Name**” is the name of the decal style.

“**Width**” is the unit width of the Decal when placed in the scene.

“**Fixed length**” is used to define the length of the Decal segment. When set to a number **greater than zero** the decal’s anchor points will always be placed the fixed length apart. When left at zero the anchor points are separated by variable distances.

“**Curviness**” describes how strong the spline bends when placing down anchor points.

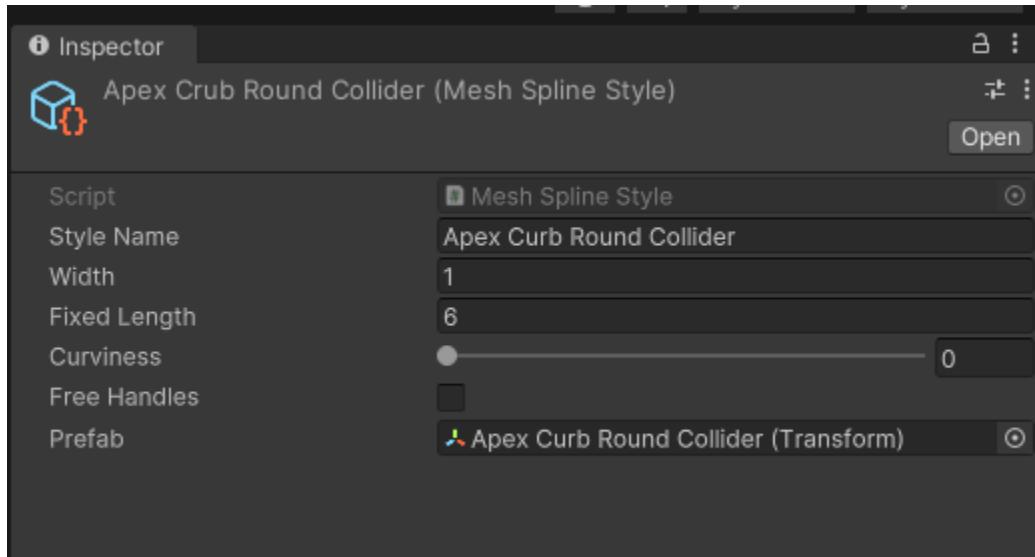
“**Free Handles**” Unlocks the curve handles by default.

“**Material**” assigns a material file to style. Only use materials suited for decal projector use, preferably a material created using the Decal Spline Asset Menu.

**“Resolution”** describes how many segments per unit length the Decal Spline spawns. Increasing this number improves blending and smoothness of the decal’s segment transitions.

### 3D Decal:

In the inspector you’ll find the following settings:



**“Style Name”** is the name of the decal style.

**“Width”** is the unit width of the Decal when placed in the scene.

**“Fixed length”** is used to define the length of the Decal segment. When set to a number **greater than zero** the decal’s anchor points will always be placed the fixed length apart. When left at zero the anchor points are separated by variable distances.

**“Curviness”** describes how strong the spline bends when placing down anchor points.

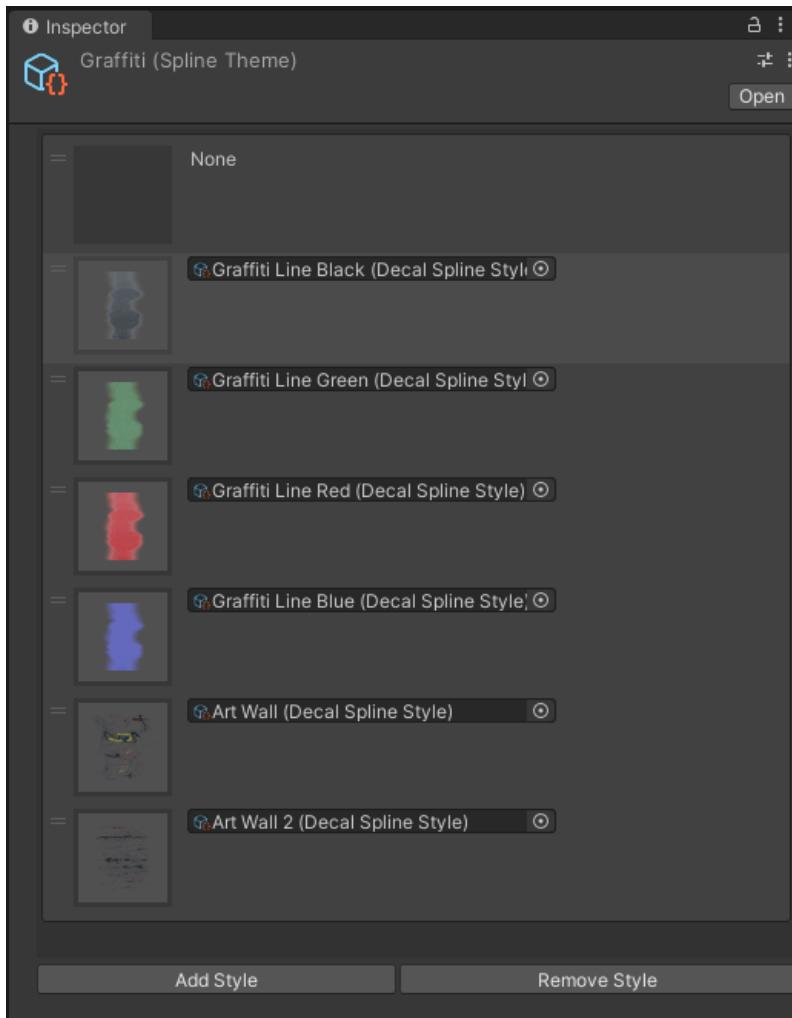
**“Free Handles”** Unlocks the curve handles by default.

**“Prefab”** assigns a prefab to the style. Only correctly **rigged 3D models** should be used. If the prefab mesh object contains a **mesh collider**, the Decal Spline will automatically generate and assign a matching mesh to the collider.

# Themes:

Theme files define a collection of styles with the goal of organizing and categorizing the styles.

The inspector consists of a list view and a couple buttons.



The **Style list** view displays all the styles in the theme. The top item always displays the “None” option. This selection is used to place an empty segment used for spacing.

The **selected** style is **highlighted**.

The “**Add style**” button adds a style to the theme.

The “**Remove Style**” button removes the **selected** style from the theme.

## Textures:

Decal splines work by projecting textures onto 3D models in the scene. These textures should have the following properties for optimal results:

The textures should be **vertically seamless**, meaning that the texture can be scrolled continuously in the height direction.

It's best if the texture has a **gradient transparency** at the **horizontal edges**, meaning that it has soft edges that taper to full transparently. This gives a nice blended look.

If the texture is in **gray scale** it can be assigned a **color filter** in the material.

An accompanying **bump texture** can give a bit of structure to the decal.

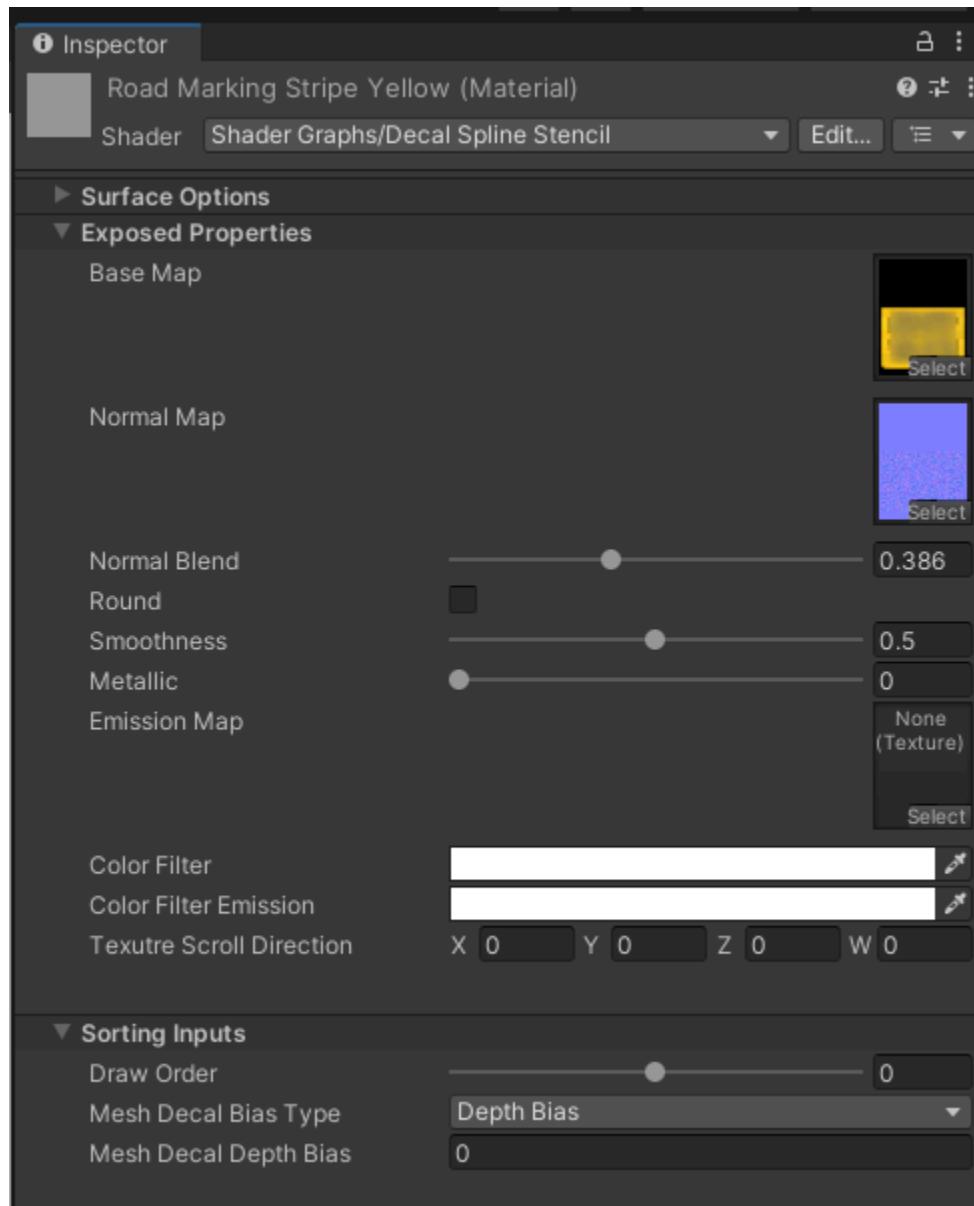
On **Mac/Metal** builds it is advised to **disable “Generate Mipmaps”**

## Shaders:

Decal materials have the option to use one of two shaders provided. Both shaders are essentially the same except for the way they handle **transparency**.

Switching to the **non stencil** shader will allow for overlapping decals and improved compatibility, this comes however at the cost of potential **artifacts**.

Therefore the default shader is the “**Decal Spline Stencil**” shader. This shader used the stencil buffer(Bit #8) to prevent artifacts in the decal.



“**Base Map**” is the main texture used by de Decal.

“**Normal Map**” is the Bump texture used by de Decal.

“**Normal Blend**” sets the strength of the normal map.

**"Round"** Deprecated.

**"Smoothness"** sets the shine the decal receives.

**"Metallic"** makes the decal appear metallic.

**"Emission Map"** can be used to make a decal emissive in certain places.

**"Color Filter"** Changes the color of the decal.

**"Color Filter"** Emission Changes the emission color.

**"Texture Scroll Direction"** makes the decal scroll in the set direction, the amount determines the scrolling speed.

### **Detail Mapped Shader:**

For **added detail** up close or to save texture space, you can opt to use the **"Decal Spline Detail Stencil"** shader.

This shader comes with two layers of detail maps consisting of an **Overlay Map** and a **Normal Map**.

The **Detail Normal Blend slider** controls the normal strength and the **tiling X and Y** are used to scale the UV coordinates of the respective detail layer.

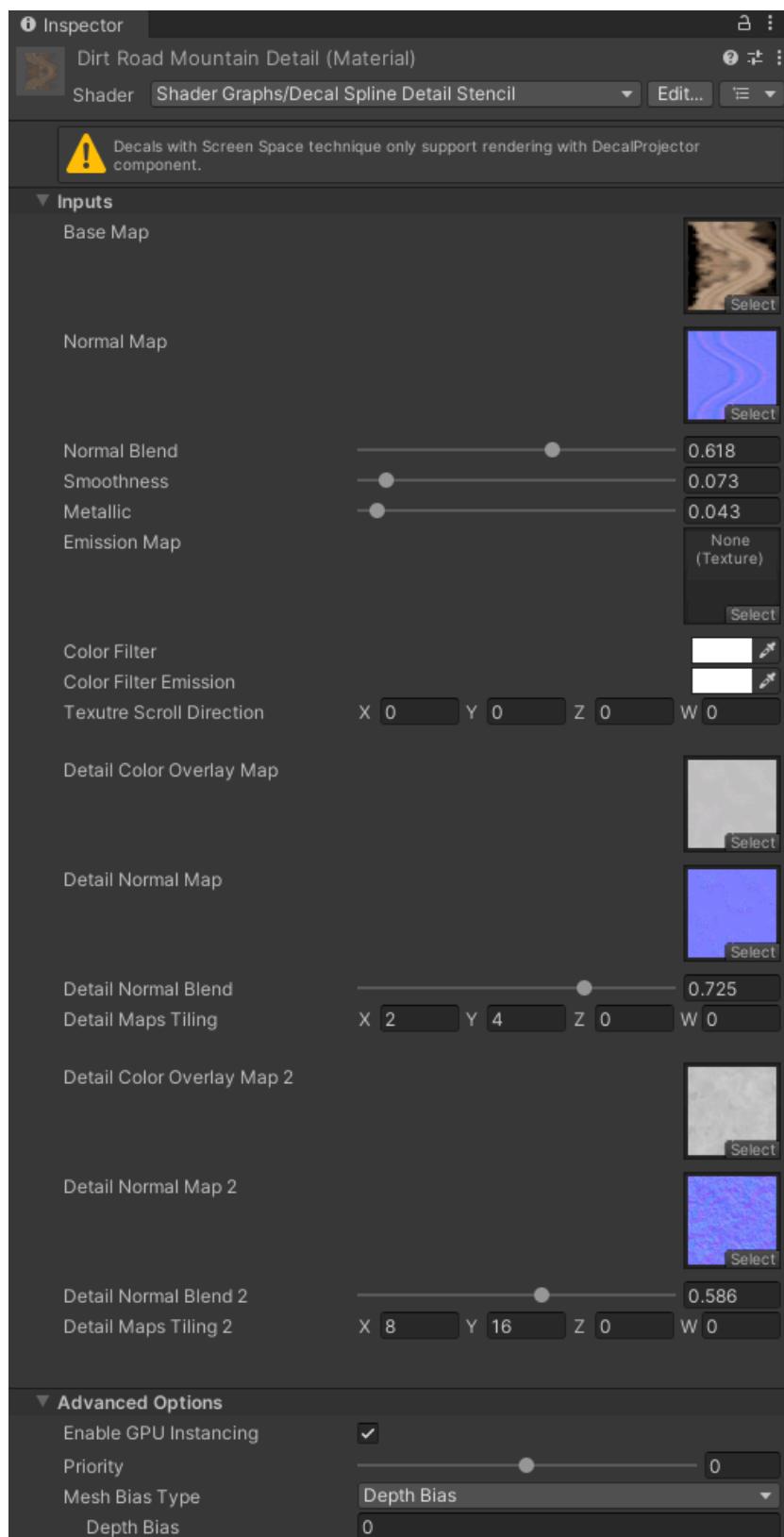
The package **includes detail map pairs** in the Detail Maps folder for use on your custom styles.

### **Advanced:**

A custom shader can be used as long as it has Decal Projector as the output node.

In order to convert the custom shader to a stencil buffer type shader, a couple lines of code need to be added to the compiled shader code.

URP:



Add the following block of code to the subshader code right after the Tags{} block.

```
Stencil{
    Ref 128
    Comp Greater
    ReadMask 128
    WriteMask 128
    Pass Replace
}
```

Then add the following block of code to the SurfaceDescriptionFunction code of the "DecalScreenSpaceProjector" shader code.

```
if(surface.Alpha == 0)
{
    discard;
}
```

HDRP:

Add the following block of code to the subshader code right after the Tags{} block.

```
Stencil{
    Ref 128
    Comp Greater
    ReadMask 128
    WriteMask 128
    Pass Replace
}
```

Remove the stencil block of the "DBufferProjector" shader.

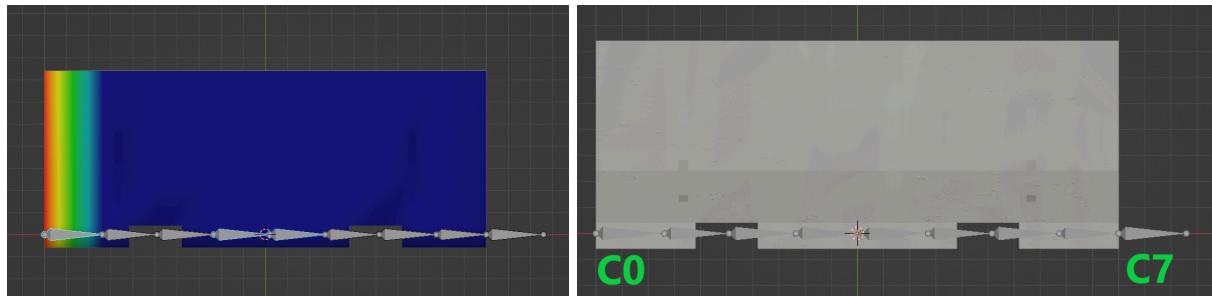
Then add the following block of code to the SurfaceDescriptionFunction code of the "DBufferProjector" shader code.

```
if(surface.Alpha == 0)
{
    discard;
}
```

# Rigging 3D models:

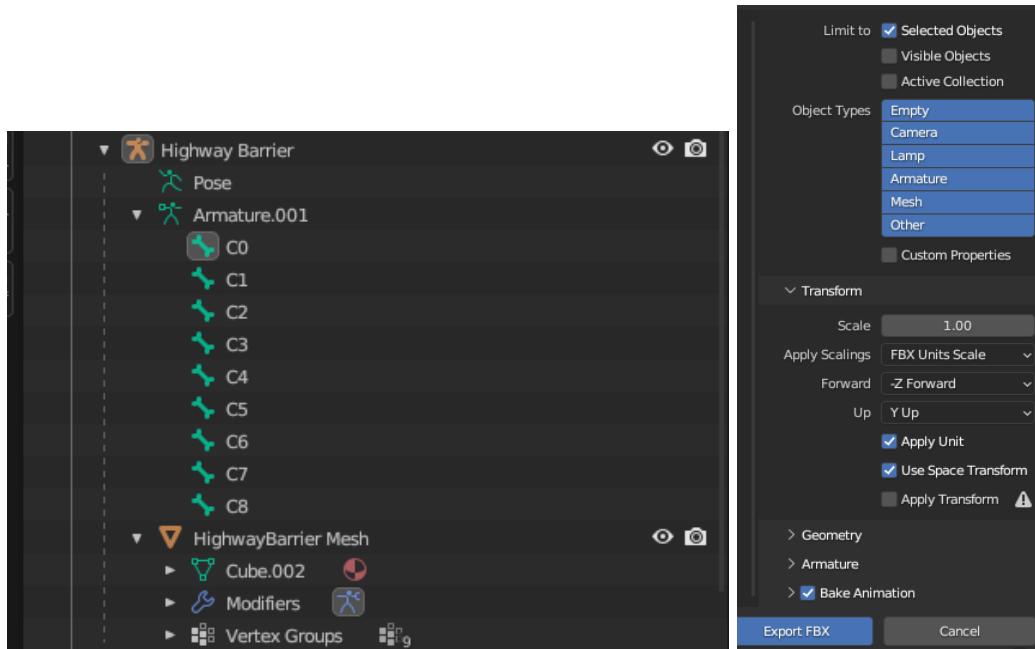
A bonus feature of the tool is 3D Decals. To use this feature with your custom models, rigging needs to be done in a specific way.

The model will be deformed along the spline using its bones.



The rig needs to have the following properties:

- Bones are **placed along the X axis** from  $x=0$  to  $x > 0$  (left to right). The origin of each bone will snap to the spline.
- Bones are **named** left to right **C0** to **Cn** where  $n = \text{number of bones} - 1$ .
- Bones are **not parented** to each other.
- All vertices have weights, and are assigned to their corresponding bones.



Blender exports can be confusing, rotating mesh and rig in relation to each other can help with proper alignment. Testing in engine is the best way to find out how to rotate your model.

## Script files:

The Decal Spline package comes with full source code supplied. Additional functionality or changes to the Decal Spline behavior can be achieved by extending the source files.

The function of the most important classes are as follows.

**DecalSpline:** the main monobehaviour that controls the Decal Spline game object.

**DecalSplineEditor:** the editor extension class for the Decal Spline game object. Here the custom inspector is defined as well as behavior such as scene view editing.

**SegmentManager:** this class manages the decal segments and their hierarchy. The most important variable here is the firstSegment, this variable contains the first node in the chain of decal segments.

**ISplineSegment:** this is the interface class for the 2D/3D and None segment classes. Important functions such as node management functions, position and handle functions and interface functions live here.

**DecalSplineSegment:** this class is used for 2D decal segments, and holds functions unique to the 2D decal variant.

**ProjectorManager:** this monobehaviour holds functions regarding the decal projectors used by the 2D decal variants. Such as spawn and despawn functions and functions used to properly size and rotate the projectors.

**MeshSplineSegment:** this class is used by 3D decal segments, it defines functions unique to 3D decal variants.

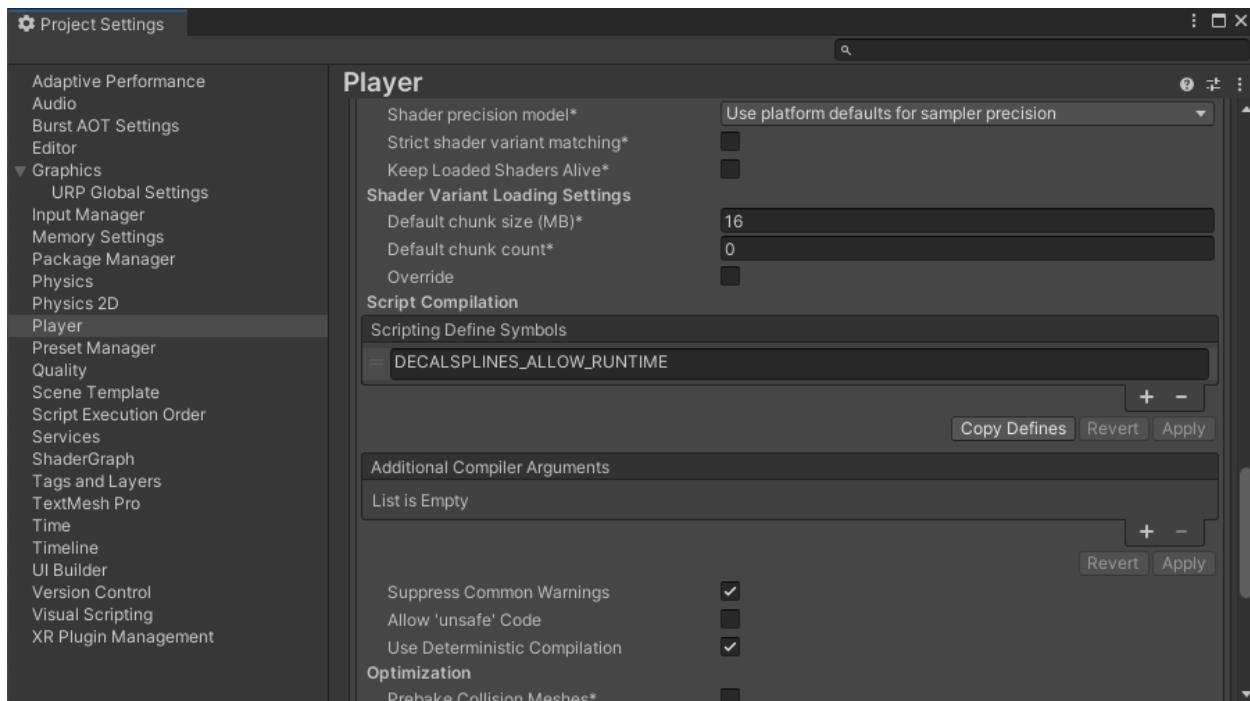
**ModelManager:** this monobehaviour contains functions that manage mesh deformation and placement used by the 3D decal variants.

**SplineUtility:** this static class contains functions used to render handles in the scene view and to manage the adjustment of the spline.

# Runtime support:

In order to expose the **Runtime Decal Splines API** it is required to define a global compiler symbol. This tells the compiler to include the necessary Decal Splines script in the final build.

Simply add the symbol **DECALSPLINES\_ALLOW\_RUNTIME** to the “Scripting Define Symbols” list under “Player->Other Settings”.



## API Functions:

The main API function needed to spawn a Decal Spline are:

**DecalSpline.AddSegment(Vector3 Position, ISplineStyle Style);**

This function spawn a spline segment at **Position** in world space that has style **Style**.

**DecalSpline.UpdateDecalSpline();**

This function should be called after changing the spline in order to update and display the changes made.

**DecalSpline.SegmentManager.FirstSegment;**

This variable is a reference to the first node in the Decal Spline. It is of type **ISplineSegment**.

Use its **next** and **prev** properties to traverse the spline.

### **ISplineSegment:**

Each node in the spline is of type ISplineSegment.

Use full properties include:

**Position** use this to set the node's position in local space.

**h1** and **h2** represent the two spline handles for this spline segment and control the curve.

### **Code Snippet:**

```
using DecalSplines;
using Unity.VisualScripting;
using UnityEngine;

public class RuntimeDecal : MonoBehaviour
{
    [SerializeField] DecalSplineStyle style;
    private DecalSpline decalSpline;
    //Add the Decalspline to the game object and spawn three spline nodes around the
    world origin.
    void Start()
    {
        decalSpline = this.AddComponent<DecalSpline>();
        decalSpline.AddSegment(Vector3.zero, style);
        decalSpline.AddSegment(new Vector3(2,0,0), style);
        decalSpline.AddSegment(new Vector3(3, 0, 6), style);
        decalSpline.UpdateDecalSpline();
    }

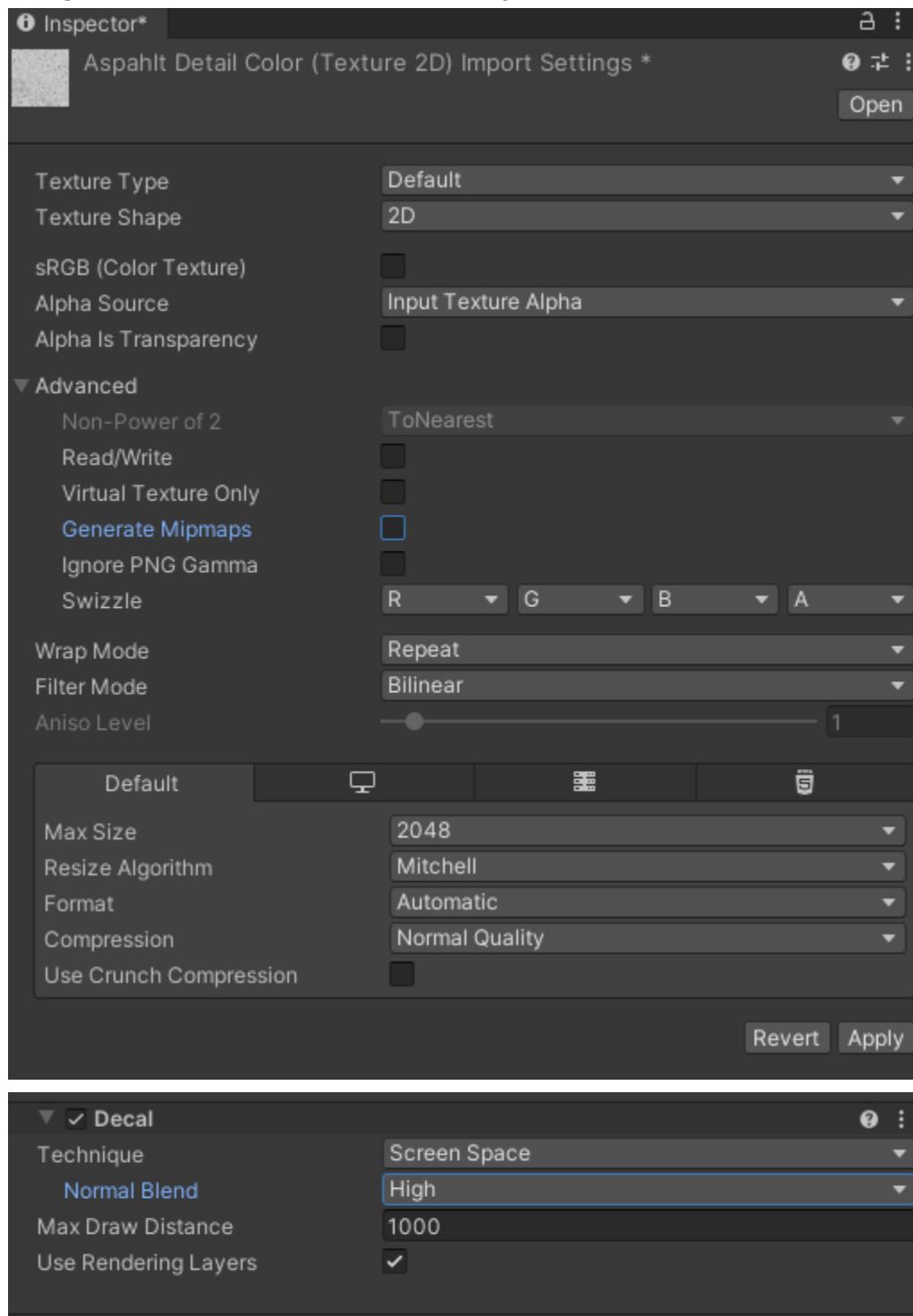
    //Move the FirstSegment by updating its Position each frame.
    private void Update()
    {
        Vector3 pos = decalSpline.SegmentManager.FirstSegment.Position;
        Vector3 newPos = new Vector3(pos.x - 0.01f, pos.y, pos.z);
        decalSpline.SegmentManager.FirstSegment.Position = newPos;

        Vector3 h1 = decalSpline.SegmentManager.FirstSegment.h1;
        Vector3 newh1 = new Vector3(h1.x + 0.004f, h1.y, h1.z + 0.0015f);
        decalSpline.SegmentManager.FirstSegment.h1 = newh1;
        decalSpline.UpdateDecalSpline();
    }
}
```

# Compatibility:

## Mac/Metal:

On Mac/Metal URP builds **artifacts** can show up due to bugs in this Unity Pipeline. To solve this **disable “Generate Mipmaps”** on any texture used with Decal Splines and set **Normal Blend** to **High** in the Decal Render Feature settings.



### **Updating DecalSplines:**

After importing an update for the Decal Spline package via the package manager, make sure you manually delete the URP/HDRP folder found in the main Decal Splines folder, and then reimport the URP/HDRP Asset as described in the setup section.