Ultimate Decals

Version 1.1
15 September 2021

Latest online version:

Quick start

How to fix lighting in the demo scene

Lighting is not baked and auto baking is not enabled because of inconsistent behaviour across different Unity versions.

1. Open the demo scene (UltimateDecals_Demo) in a non-additive mode.
2. Open Lighting Settings (Window → Rendering → Lighting Settings).

For non-mobile built-in Deferred or Forward:
3. Enable "Realtime Global Illumunation" in Lighting Settings

For any mobile platform or LWRP/URP:
3. Enable "Baked Global Illumunation" in Lighting Settings
4. Set Mode of the Directional Light in the demo scene to Mixed.

Once everything is done, enable "Auto Generate" in Lighting Settings and let Unity bake it.

How to set things up

1. Create a new GameObject in your main scene and add UD_Manager component.

2. Add UD_Camera component to your main camera, and, if needed, to other cameras that should render decals.

How to create decals

1. From main or context menu: Create → 3D Object → Ultimate Decal

2. Select an existing decal and Shift+Click in Scene to clone it.

When creating new materials, set "Wrap Mode" to "Clamp" and check "Border Mip Maps" for all textures.

LWRP/URP

When importing UD to a LWRP/URP project, the following actions should be taken automatically:

1. UD_SRP/UD_LWRP/UD_URP added to Scripting Define Symbols in Player Settings.

2. UltimateDecalsFeature added to the active Forward Renderer in the current pipeline settings asset.

3. Demo materials are converted to the pipeline's default Lit shader.

4. Bearroll/UltimateDecals/Setup/UD_ScreenSpaceShadows.txt copied to Bearroll/UltimateDecals/Resources/Shaders/UD_ScreenSpaceShadows.shader

Notes
— If nothing is happening after the import, select UD_Setup.asset in the Project view to make Unity load it.
— Once everything is set up, you may delete the Bearroll/UltimateDecals/Setup folder.

Having troubles?
Join Discord for support!
https://discord.gg/5xPmvFw

About

Easy to use but powerful decal system for Unity.

Supported pipelines
— Built-in Deferred and Forward (2018.4 − 2020.3)
— URP 10.5 (2020.3)

Supported platforms:
— DirectX 11 and 12 (Windows)
— Vulkan (Desktop)
— Metal (Mac OS)
— OpenGL Core 4.1+ (Any OS)

Experimental support:
— OpenGL ES 3.0+ (Android)

The only supported Stereo mode is currently Single Pass.

Limitations

General

— Unity doesn't provide access to the original multisampled depth, decals can't utilize MSAA and will ignore it.
— Screen space decals don't support lightmaps and can't be lit by Baked lights.

Deferred

— Layer limiting is not compatible with light culling masks.
— If two or more decals overlap (even with transparent parts), only the last one rendered will output correct smoothness, while the first values will be ignored.

Forward, LWRP, URP

— Each decal can be lit by 1 directional light with realtime shadows and up to 3 per-pixel point or spot lights. Vertex lighting is not supported. There's a limit of 16 lights for all decals per frame.
— Limiting decals to layers without extra passes requires modification of object shaders.
— Not supported yet: light cookies, reflection probe blending.

Components

UltimateDecal

Mode
Screen Space is currently the only available mode.
More options will come in the future.

Control Object Name
If enabled, decal's game object name will be synced with its material name.

Type
— Dynamic: acts like regular MeshRenderer, can be moved/rotated/scaled/enabled/disabled at runtime.
— Static: can be enabled/disabled, but transform changes are not tracked. To move a static decal, disable it and re-enable after repositioning.
— Permanent Mark: once it's spawned, it's never deleted regardless of the state of the original object. Will be replaced with new decals when mark limit per material is reached.

Material
Material used to render this decal.
Must use a supported shader (UltimateDecals/Lit or UltimateDecals/Unlit).
Click "Show materials" to see all avaialble materials.

Render Queue Offset
Alters the "Render Queue" property of decal materials and allows to change rendering order of decals of the same type (Note that all Static decals are always rendered first, then Dynamic, then Marks).

In versions before 1.1 this setting was called "Order" and only affected decals of the same material.

UD_Manager

Max Permanent Marks
When the limit is reached, new marks will replace the oldest ones.
Note: this limit is per material.

Global Mip Bias
Makes all decals sharper (negative values) or more blurry (positive values).

Per Decal Light Probes
Enable if you have baked (or mixed) lights, light probes set up, and you want them to affect each decal separately.

Layer mask
Allows you to mark up to 3 layers to limit decal projection to these layers, or to everything but these layers (can be choosen per material).
Note: deferred only. To limit decal projection in other rendering paths, see 4.1

UD_Camera

A component that renders decals of the currently active UD_Manager to a specific camera. Scene view cameras are handled automatically.

Normals source
Allows you to switch to CameraDepthNormals texture which provides the best projection quality, but requires an additional render pass. Usually you don't want to enable it for decals only, but if you're already using other effects that require it, keep it on.
Built-in Forward only

Per pixel normals
Blends restored normals with GBuffer normals, smoothing polygon edges at the cost of worse projection angle limit handling.
Deferred only.

Debug View
Allows you to see internal textures used for decal rendering.

UD_LightProbeManager

Create a game object with this component in every scene which uses Realtime or Baked GI, and it will place a light probe on each decal whenever you bake lighting.

Layer Limits

Deferred

1. Pick up to 3 layers in UD_Manager's Layer Mask property

2. In a decal material, set Layer Limit to either Layer Mask (to project decals on the chosen layers) or Inverted Layer Mask (to project on everything but the chosen layers).

Note:
— Up to 3 layers can be used for masking.
— The method uses a deferred specific hack that's not available in other pipelines.

Other pipelines

The most reliable way that also doesn't affect peformance is to modify your opaque shaders to make them write to stencil (or an additional render target) to mark different types of objects.

Follow this guide:
https://docs.google.com/document/d/1jFPNDz_gUQYAKJ__WzC-b5S2dGx5jf6moU-u-O8sqCQ

Shaders

Common properties

Textures

Mask (R)

Atlas

Cols — the number of columns in the Atlas
Rows — the number of rows in the Atlas
Tiling X,Y — applied after atlasing

Other properties

Layer limit — allows to limit decal projection using Stencil buffer.
Angle Limit (0-90)
Angle Smoothing
Normal smoothing
Normal blending
Alpha Cutout
Alpha Smoothing
Colorization
Albedo Opacity

PBR-specific properties

Textures

Albedo (RGB) Opacity (A)
Normal map (RGB)
Occlusion (R)

Depends on selected Workflow:
Metallic (R) Smoothness (A)
Metallic (R) Roughness (A)
Specular (RGB) Smoothness (A)
Specular (RGB) Roughness (A)