Dynamic Decals

Daniel Dickinson

August 28, 2017

Chapter 1

Welcome to the documentation

To keep things from being too overwhelming I've only documented what's necessary to script with the system. For more advanced users, everything in the system is commented and built to be expanded on. Dig through the code to your hearts content. If you have any questions or get stuck at any stage I'm always available at Support@LlockhamIndustries.com.

Chapter 2

Namespace Index

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

Namespace Documentation

5.1 LlockhamIndustries Namespace Reference

Namespaces

5.2 LlockhamIndustries.Decals Namespace Reference

Classes

- class Additive
- class Base
- class CollisionPrinter
- class Cull
- class CursorPositioner
- class Deferred
- class DynamicDecals
- class Fade
- class Gloss
- class Metallic
- class Multiplicative
- class NineSprite
- class Normal
- class ParticleCollisionPrinter
- class Positioner
- class Printer
- class Projection
- class ProjectionBlocker
- class ProjectionPool
- class RandomScale
- class RayCollisionPrinter
- class RayPositioner
- class RayPrinter
- class SceneLayers
- class SheetAnimator
- class Specular
- class Unlit

5.3 LlockhamIndustries.ExtensionMethods Namespace Reference

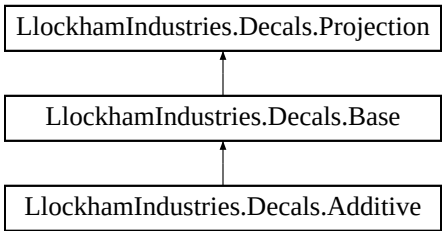
5.4 LlockhamIndustries.Misc Namespace Reference

Chapter 6

Class Documentation

6.1 LlockhamIndustries.Decals.Additive Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Additive:



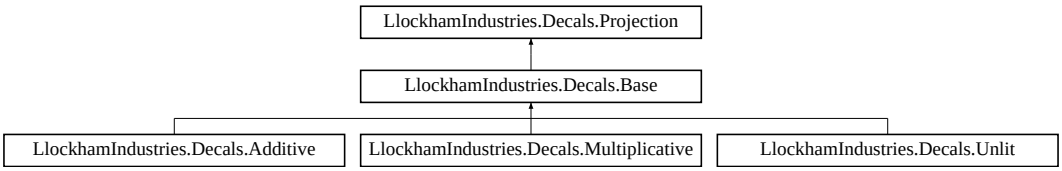
Additional Inherited Members

6.1.1 Detailed Description

Additive projection. Draws to the screen additively. If rendering in deferred, will be drawn in forward, after all other projections.

6.2 LlockhamIndustries.Decals.Base Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Base:



Public Attributes

- AlbedoPropertyGroup albedo

Additional Inherited Members

6.2.1 Detailed Description

The base of all unlit forward projections (Unlit, Additive, Multiplicative)

6.2.2 Member Data Documentation

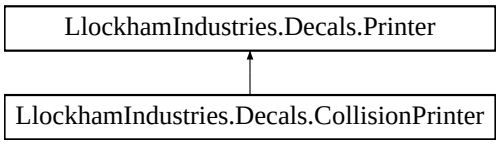
6.2.2.1 albedo

`AlbedoPropertyGroup LlockhamIndustries.Decals.Base.albedo`

The primary color details of your projection. The alpha channel of these properties is used to determine the projections transparency.

6.3 LlockhamIndustries.Decals.CollisionPrinter Class Reference

Inheritance diagram for LlockhamIndustries.Decals.CollisionPrinter:



Public Attributes

- RotationSource rotationSource
- CollisionCondition condition
- float conditionTime
- LayerMask layers

Additional Inherited Members

6.3.1 Detailed Description

The CollisionPrinter Component. Prints a projection under set conditions related to the collision of the object attached to this printer.

6.3.2 Member Data Documentation

6.3.2.1 rotationSource

```
RotationSource LlockhamIndustries.Decals.CollisionPrinter.rotationSource
```

Defines the orientation of the projection relative to the surface of the collision. Velocity will orient the projection as if its up is the direction the collision object is moving in. Random will orient the projection as if its up is random.

6.3.2.2 condition

```
CollisionCondition LlockhamIndustries.Decals.CollisionPrinter.condition
```

Defines the condition on which a projection is printed. Enter will print whenever a collision occurs. Delay will print the conditionTime seconds after a collision occurs. Constant will print every fixed update during a collision. Exit will print upon exiting a collision.

6.3.2.3 conditionTime

```
float LlockhamIndustries.Decals.CollisionPrinter.conditionTime
```

If the collision condition is set to delay, the conditionTime determines the length of that delay.

6.3.2.4 layers

```
LayerMask LlockhamIndustries.Decals.CollisionPrinter.layers
```

The layers that, when collided with, cause a print.

6.4 LlockhamIndustries.Decals.Cull Class Reference

Inherits MonoBehaviour.

Public Attributes

- float cullTime = 4
- float updateRate = 0.05f

6.4.1 Detailed Description

Culls attached projection (destroy or return to pool) once it's no longer visible by any cameras. Useful for cleaning up your scene without the player noticing. Designed to be printed with your projections. Attach to your prefab and enable print behaviours on your printer. You can turn this on or off by enabling and disbaling this component respectively.

6.4.2 Member Data Documentation

6.4.2.1 cullTime

```
float LlockhamIndustries.Decals.Cull.cullTime = 4
```

How long the projection has to be off screen before it's culled. 0 will cull the projection the second it's no longer visible.

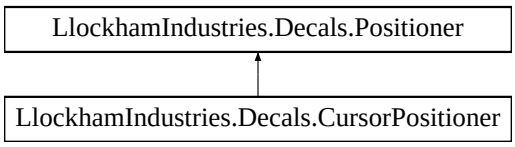
6.4.2.2 updateRate

```
float LlockhamIndustries.Decals.Cull.updateRate = 0.05f
```

How often we check if the projection is visible. There's no point checking visibility 60+ times a second. 0.05 will check 20 times a second, 0.5, twice a second and 2 once every 2 seconds.

6.5 LlockhamIndustries.Decals.CursorPositioner Class Reference

Inheritance diagram for LlockhamIndustries.Decals.CursorPositioner:



Public Attributes

- Camera projectionCamera

Additional Inherited Members

6.5.1 Detailed Description

The cursor positioner component. Positions a projection at the cursor position.

6.5.2 Member Data Documentation

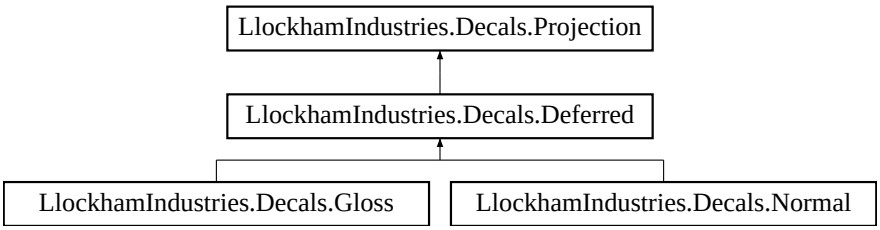
6.5.2.1 projectionCamera

```
Camera LlockhamIndustries.Decals.CursorPositioner.projectionCamera
```

The Camera used to intepret the mouse position. If null will default to the main camera.

6.6 LlockhamIndustries.Decals.Deferred Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Deferred:



Additional Inherited Members

6.6.1 Detailed Description

The base of all deferred only projections (Gloss, Normal)

6.7 LlockhamIndustries.Decals.DynamicDecals Class Reference

Inherits MonoBehaviour.

Public Member Functions

- ProjectionPool GetPool (string Title)
- ProjectionPool GetPool (int ID)

6.7.1 Detailed Description

The core class of the system, responsible for the majority of the systems functionality. For scripting purposes, it's almost entirely a black box, you should rarely need to access or modify anything within it. It's well stuctured and commented all the same though, so if your interested, open it up and have a look around.

6.7.2 Member Function Documentation

6.7.2.1 GetPool() [1/2]

```
ProjectionPool LlockhamIndustries.Decals.DynamicDecals.GetPool (
            string Title )
```

Returns a pool with the specified name, if it exists. If it doesn't, returns the default pool.

Parameters

Title	The title of the pool to be returned.

6.7.2.2 GetPool() [2/2]

```
ProjectionPool LlockhamIndustries.Decals.DynamicDecals.GetPool (
            int ID )
```

Returns a pool with the specified ID, if it exists. If it doesn't, returns the default pool.

Parameters

ID	The ID of the pool to be returned.

6.8 LlockhamIndustries.Decals.Fade Class Reference

Inherits MonoBehaviour.

Public Member Functions

- void InvokeFade ()
- void EndFade ()

Public Attributes

- FadeType type
- FadeMethod method
- FadeWrapMode wrapMode
- AnimationCurve fade
- float fadeLength = 1
- float updateRate = 0.05f

6.8.1 Detailed Description

Fades attached projection in and out over time. Designed to be printed with your projections. Attach to your prefab and enable print behaviours on your printer.

6.8.2 Member Function Documentation

6.8.2.1 InvokeFade()

```
void LlockhamIndustries.Decals.Fade.InvokeFade ( )
```

Used to begin a fade routine with a fade method set to OnInvoke.

6.8.2.2 EndFade()

```
void LlockhamIndustries.Decals.Fade.EndFade ( )
```

Used to stop a fade mid-way through it's routine.

6.8.3 Member Data Documentation

6.8.3.1 type

```
FadeType LlockhamIndustries.Decals.Fade.type
```

What's being used to fade in/out your projection. Alpha will adjust the alpha value, which can be used with blended transparency for a traditional fade, or cutout transparency for a dissolve. Scale will adjust the scale, allowing you to shrink your projection in and out. Both will adjust both the alpha value and scale of your projection

6.8.3.2 method

```
FadeMethod LlockhamIndustries.Decals.Fade.method
```

When should the fade routine begin. OnEnable will have the projection begin fading in/out immediately. OnInvoke will allow you to start the fade method via script later via the InvokeFade() method.

6.8.3.3 wrapMode

```
FadeWrapMode LlockhamIndustries.Decals.Fade.wrapMode
```

Determines what the fade method actually does. Are you fading the projection in, out, making it throb? Once is the traditional fade out. This will play through the fade curve once and then destroy the projection. Clamp is the traditional fade in. This will play through the fade curve once, then never touch your projection again. Loop can be used to make a projection appear to throb. This will play through the fade curve repeatedly. PingPong, can be used to make your projection appear to throb in a different manner. This will play through your fade curve, then through it backwards, then forwards again, repeating indefinitely.

6.8.3.4 fade

```
AnimationCurve LlockhamIndustries.Decals.Fade.fade
```

Determines the fade value over time. A curve starting at 0 and ending at 1 would have your projection fade in. A curve starting at 1 and ending at 0 would have your projection fade out. A curve starting and ending at 0, but peaking at 1 in the center will have your projection fade in and out.

6.8.3.5 fadeLength

```
float LlockhamIndustries.Decals.Fade.fadeLength = 1
```

Determines how long the fade takes, in seconds. Cannot be set to 0.

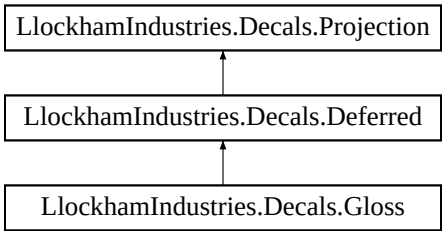
6.8.3.6 updateRate

```
float LlockhamIndustries.Decals.Fade.updateRate = 0.05f
```

How often we update the fade of our projection. The lower this value the smoother the fade, though slow gradual fades won't need to be updated frequently. 0.05 will update 20 times a second, 0.5, twice a second and 2 once every 2 seconds.

6.9 LlockhamIndustries.Decals.Gloss Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Gloss:



Public Attributes

- GlossPropertyGroup gloss

Properties

- GlossType GlossType `[get, set]`

6.9.1 Detailed Description

Deferred Only gloss projection. Only affects the gloss channel of the deferred buffers. Useful for making things wetter or rougher.

6.9.2 Member Data Documentation

6.9.2.1 gloss

```
GlossPropertyGroup LlockhamIndustries.Decals.Gloss.gloss
```

The primary color details of your projection. The alpha channel of these properties is used to determine the projections transparency.

6.9.3 Property Documentation

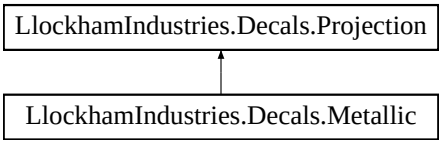
6.9.3.1 GlossType

```
GlossType LlockhamIndustries.Decals.Gloss.GlossType  [get], [set]
```

Defines how the gloss modifcation affects the surface. Shine will have the decal shine the surface it's applied too. Great for making surfaces appear wet. Dull will have the decal dull the surface it's applied too. Great for making surfaces appear worn or weathered.

6.10 LlockhamIndustries.Decals.Metallic Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Metallic:



Public Attributes

- AlbedoPropertyGroup albedo
- MetallicPropertyGroup metallic
- NormalPropertyGroup normal
- EmissivePropertyGroup emissive

Additional Inherited Members

6.10.1 Detailed Description

Standard Shader - metallic setup.

6.10.2 Member Data Documentation

6.10.2.1 albedo

`AlbedoPropertyGroup LlockhamIndustries.Decals.Metallic.albedo`

The primary color details of your projection. The alpha channel of these properties is used to determine the projections transparency.

6.10.2.2 metallic

`MetallicPropertyGroup LlockhamIndustries.Decals.Metallic.metallic`

The metallic texture, with a multiplier. Determines how metallic the surface of the decal appears. black will make the decal surface appear like plastic. white will make the decal surface appear metallic. Only the R channel of the texture is used.

6.10.2.3 normal

`NormalPropertyGroup LlockhamIndustries.Decals.Metallic.normal`

The normal texture of your decal, multiplied by the normal strength. Normals determine how the surface of your decal interacts with lights.

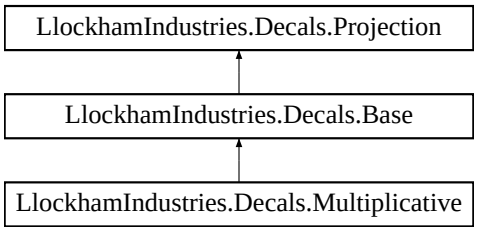
6.10.2.4 emissive

`EmissivePropertyGroup LlockhamIndustries.Decals.Metallic.emissive`

The emission texture of your projection, multiplied by the emission color and intensity. Emission allows us to make a decal appear as if it's emitting light. Supports HDR.

6.11 LlockhamIndustries.Decals.Multiplicative Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Multiplicative:



Additional Inherited Members

6.11.1 Detailed Description

Multiplicative projection. Multiplies result with whats already on the screen. If rendering in deferred, will be drawn in forward, after all other projections.

6.12 LlockhamIndustries.Decals.NineSprite Class Reference

Inherits MonoBehaviour.

Public Member Functions

- void UpdateProperties ()
- void UpdateTransforms ()

Properties

- ProjectionRenderer Sprite `[get, set]`
- float BorderPixelSize `[get, set]`
- float BorderWorldSize `[get, set]`

6.12.1 Detailed Description

This component allows you to stretch a decal by dividing it into nine different decals (each corner, edge and the center) and stretching the individual components. This is useful for UI elements like borders or box selections.

6.12.2 Member Function Documentation

6.12.2.1 UpdateProperties()

```
void LlockhamIndustries.Decals.NineSprite.UpdateProperties ( )
```

Updates all nine decals with updated properties from the original projection renderer. This should be called whenever the original projection renderer is modified and you want these changes to be reflected by the nine sprite.

6.12.2.2 UpdateTransforms()

```
void LlockhamIndustries.Decals.NineSprite.UpdateTransforms ( )
```

Updates all nine decals to account for a change in scale. This should be called when you stretch (scale) the nine-sprite, to have it rescale it's sprite pieces to match the new scale.

6.12.3 Property Documentation

6.12.3.1 Sprite

```
ProjectionRenderer LlockhamIndustries.Decals.NineSprite.Sprite  [get], [set]
```

The decal we want to use as a base. Nine copies of this will be made and used to represent your sprite. This should almost always be a prefab.

6.12.3.2 BorderPixelSize

```
float LlockhamIndustries.Decals.NineSprite.BorderPixelSize  [get], [set]
```

How large each corner / how thick each edge should be in pixels. This will adjust how we sample from the original decal.

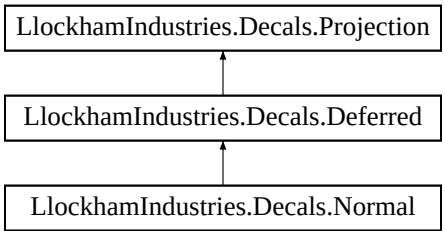
6.12.3.3 BorderWorldSize

```
float LlockhamIndustries.Decals.NineSprite.BorderWorldSize  [get], [set]
```

How large each corner / how thick each edge should be in units (world space). This will adjust how we represent the original decal in the world.

6.13 LlockhamIndustries.Decals.Normal Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Normal:



Public Attributes

- NormalPropertyGroup normal

Additional Inherited Members

6.13.1 Detailed Description

Deferred Only normal projection. Only affects the normal buffer. Useful for adding cracks or normal details to tiled surfaces.

6.13.2 Member Data Documentation

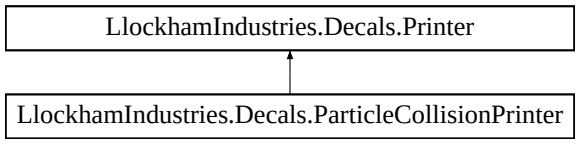
6.13.2.1 normal

```
NormalPropertyGroup LlockhamIndustries.Decals.Normal.normal
```

The primary color details of your projection. The alpha channel of these properties is used to determine the projections transparency.

6.14 LlockhamIndustries.Decals.ParticleCollisionPrinter Class Reference

Inheritance diagram for LlockhamIndustries.Decals.ParticleCollisionPrinter:



Public Attributes

- RotationSource rotationSource
- float ratio = 1

Additional Inherited Members

6.14.1 Detailed Description

The CollisionPrinter Component. Prints a projection under set conditions related to the collision of the object attached to this printer.

6.14.2 Member Data Documentation

6.14.2.1 rotationSource

```
RotationSource LlockhamIndustries.Decals.ParticleCollisionPrinter.rotationSource
```

Defines the orientation of the projection relative to the surface of the collision. Velocity will orient the projection as if its up is the direction the collision object is moving in. Random will orient the projection as if its up is random.

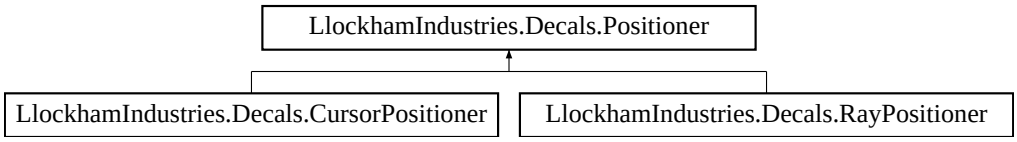
6.14.2.2 ratio

```
float LlockhamIndustries.Decals.ParticleCollisionPrinter.ratio = 1
```

Defines the percentage of particles that print projections. At 0, no particles will print, at 1, all will.

6.15 LlockhamIndustries.Decals.Positioner Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Positioner:



Public Attributes

- ProjectionRenderer projection
- LayerMask layers
- bool alwaysVisible

Properties

- ProjectionRenderer Active `[get]`

6.15.1 Detailed Description

The base abstract class all other positioners inherit from. Positioners manage the position and rotation of a projection under different circumstances.

6.15.2 Member Data Documentation

6.15.2.1 projection

`ProjectionRenderer LlockhamIndustries.Decals.Positioner.projection`

The projection we want to position. This should usually be a prefab.

6.15.2.2 layers

`LayerMask LlockhamIndustries.Decals.Positioner.layers`

The layers we want to position onto. All positioners are based on raycasts, which layers should those rays collide with.

6.15.2.3 alwaysVisible

`bool LlockhamIndustries.Decals.Positioner.alwaysVisible`

If enabled the projection will not be hidden when a raycast fails. It will simply be left where it was last. If disabled the projection will be hidden when a raycats fails.

6.15.3 Property Documentation

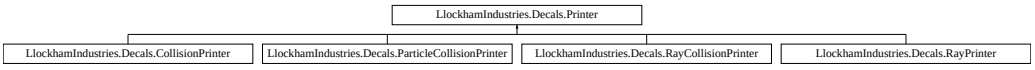
6.15.3.1 Active

`ProjectionRenderer LlockhamIndustries.Decals.Positioner.Active [get]`

The instance of the projection that we are currently positioning. If you seek to modify the positioners current projection, modify this. Cannot be set, may be active or inactive.

6.16 LlockhamIndustries.Decals.Printer Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Printer:



Public Member Functions

- void Print (Vector3 Position, Quaternion Rotation, Transform Surface, int Layer=0)

Public Attributes

- ProjectionRenderer [] prints = new ProjectionRenderer[1]
- LayerMask [] printLayers
- string [] printTags
- PrintSelection printMethod
- PrintParent parent
- bool printBehaviours
- bool destroyOnPrint
- float frequencyTime
- float frequencyDistance

Properties

- ProjectionPool Pool `[get, set]`

6.16.1 Detailed Description

The abstract base of all printers. Prints copies of projections given a position & rotation.

6.16.2 Member Function Documentation

6.16.2.1 Print()

```
void LlockhamIndustries.Decals.Printer.Print (
            Vector3 Position,
            Quaternion Rotation,
            Transform Surface,
            int Layer = 0 )
```

The simplest method of printing available. Use this when you know exactly where you want to print. Printing is still subject to frequency & interection checks, and any fading or culling specified will still be applied.

Parameters

Position	The position to print the projection at, in world space.
Rotation	The orientation of the printed projection, in world space.
Surface	The transform the projection will be childed to. Will be ignored unless printer has Print Parent set to surface (Not default).

6.16.3 Member Data Documentation

6.16.3.1 prints

```
ProjectionRenderer [] LlockhamIndustries.Decals.Printer.prints = new ProjectionRenderer[1]
```

The projections to be printed. Multiple can be queued at once and are printed based on the print method.

6.16.3.2 printLayers

```
LayerMask [] LlockhamIndustries.Decals.Printer.printLayers
```

The layers associated with each of the projections to be printed. Used when multiple prints are availble and the print method is set to layer. Each print will be printed when it's associated layer is being printed on.

6.16.3.3 printTags

```
string [] LlockhamIndustries.Decals.Printer.printTags
```

The tags associated with each of the projections to be printed. Used when multiple prints are availble and the print method is set to tag. Each print will be printed when the surface being printed on has the given tag.

6.16.3.4 printMethod

`PrintSelection LlockhamIndustries.Decals.Printer.printMethod`

Determines which projections from the queue are printed. Layer, will switch prints based on the layer hit. Random will select a different projection at random each print and print it. All will print all of the projections every print.

6.16.3.5 parent

`PrintParent LlockhamIndustries.Decals.Printer.parent`

The print parent determines where within the scenes heirarchy the printed projections will be placed. Default will place them in the default pooling heriarchy. Surface will parent them to whatever they are printed upon, which can be useful when you want projections to attach and move with the surface they're printer upon.

6.16.3.6 printBehaviours

`bool LlockhamIndustries.Decals.Printer.printBehaviours`

Printing with behaviours allows any behaviours attached to the printed renderer to be transfered to the prints. This allows us to attach behaviours that, say, fade a projection out over time, or destroy a projection once we are 500 units away from it.

6.16.3.7 destroyOnPrint

`bool LlockhamIndustries.Decals.Printer.destroyOnPrint`

If enabled, the printer and the attached gameobject will be destroyed after the printer prints. Useful for single print items like projectiles.

6.16.3.8 frequencyTime

`float LlockhamIndustries.Decals.Printer.frequencyTime`

Frequency time restricts how often the printer can print projections (in seconds). ie a value of 0.1f will prevent the printer from printing more than once every 0.1 seconds.

6.16.3.9 frequencyDistance

`float LlockhamIndustries.Decals.Printer.frequencyDistance`

Frequency distance restricts how close a printer can print a projection to its previous print. ie. a value of 0.1f will prevent the printer from printing if it's previous print was within 0.1 units of the print location.

6.16.4 Property Documentation

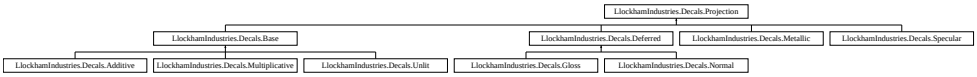
6.16.4.1 Pool

`ProjectionPool LlockhamIndustries.Decals.Printer.Pool [get], [set]`

The pool determines which projection pool the printed projections will belong to.

6.17 LlockhamIndustries.Decals.Projection Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Projection:



Properties

- ProjectionType ProjectionType `[get, set]`
- int Priority `[get, set]`
- TransparencyType TransparencyType `[get, set]`
- float Cutoff `[get, set]`
- Vector2 Tiling `[get, set]`
- Vector2 Offset `[get, set]`
- MaskMethod MaskMethod `[get, set]`
- bool MaskLayer1 `[get, set]`
- bool MaskLayer2 `[get, set]`
- bool MaskLayer3 `[get, set]`
- bool MaskLayer4 `[get, set]`
- float ProjectionLimit `[get, set]`
- bool Instanced `[get, set]`
- bool ForceForward `[get, set]`

6.17.1 Detailed Description

The abstract projection class. All projections inherit from this singular class. If you want to make your own projection types, inherit from this class. The system's UI should automatically detect and allow you to create projections of your custom type.

6.17.2 Property Documentation

6.17.2.1 ProjectionType

`ProjectionType LlockhamIndustries.Decals.Projection.ProjectionType [get], [set]`

Defines how the projection is projected. Decals project from a plane in a single direction. This allows complex detail to be projected accurately. OmniDecals project from a point in all directions. This samples a gradient based on how far from the point a surface being projected on is.

6.17.2.2 Priority

```
int LlockhamIndustries.Decals.Projection.Priority  [get], [set]
```

Defines whether this projection appears above or below other projections. Higher priority projections will appear above lower priority projections. ie. a priority 10 projection will appear to overlap a priority 5 projection. Values should be positive and less than 100.

6.17.2.3 TransparencyType

```
TransparencyType LlockhamIndustries.Decals.Projection.TransparencyType  [get], [set]
```

Defines the transparency method. Cutout will simply cull any pixels under a certain alpha value and is the cheaper method. Blend will blend the projection with the surface it's drawing on based on the alpha value.

6.17.2.4 Cutoff

```
float LlockhamIndustries.Decals.Projection.Cutoff  [get], [set]
```

The alpha cutoff of the projection. Any pixels with an alpha value below this value will not be rendered.

6.17.2.5 Tiling

```
Vector2 LlockhamIndustries.Decals.Projection.Tiling  [get], [set]
```

The tiling of all textures applied to your decal. Higher values will cause your texture to repeat while lower values will cause it to stretch.

6.17.2.6 Offset

```
Vector2 LlockhamIndustries.Decals.Projection.Offset  [get], [set]
```

The offset of all textures applied to your decal. Allows you to adjust the position of your texture, useful for scrolling effects, atlassing or to tweak tiling textures.

6.17.2.7 MaskMethod

```
MaskMethod LlockhamIndustries.Decals.Projection.MaskMethod  [get], [set]
```

Defines which masking method we should apply to this projection. Either "DrawOnEverythingExcept" or "OnlyDrawOn". Draw On Everything Except - will draw on all surface except those in the selected mask layers. Only Draw On - will only draw on surfaces that are part of the selected mask layers.

6.17.2.8 MaskLayer1

```
bool LlockhamIndustries.Decals.Projection.MaskLayer1  [get], [set]
```

Defines whether this projection is affected by the first masking layer. To add surfaces to this mask layer add a Mask component to a renderable gameObject and toggle on the appropriate mask layer.

6.17.2.9 MaskLayer2

```
bool LlockhamIndustries.Decals.Projection.MaskLayer2 [get], [set]
```

Defines whether this projection is affected by the second masking layer. To add surfaces to this mask layer add a Mask component to a renderable gameObject and toggle on the appropriate mask layer.

6.17.2.10 MaskLayer3

```
bool LlockhamIndustries.Decals.Projection.MaskLayer3 [get], [set]
```

Defines whether this projection is affected by the third masking layer. To add surfaces to this mask layer add a Mask component to a renderable gameObject and toggle on the appropriate mask layer.

6.17.2.11 MaskLayer4

```
bool LlockhamIndustries.Decals.Projection.MaskLayer4 [get], [set]
```

Defines whether this projection is affected by the fourth masking layer. To add surfaces to this mask layer add a Mask component to a renderable gameObject and toggle on the appropriate mask layer.

6.17.2.12 ProjectionLimit

```
float LlockhamIndustries.Decals.Projection.ProjectionLimit [get], [set]
```

The normal cutoff angle of the decal. If the angle between the surface and the inverse direction of projection is beyond this limit, the pixel will not be rendered. This is designed to prevent your decals from stretching when they project onto near parralel surfaces, or surfaces in which they would appear streched. Setting this to 180 will render all pixels.

6.17.2.13 Instanced

```
bool LlockhamIndustries.Decals.Projection.Instanced [get], [set]
```

Defines whether this projection should be instanced. Disable this if you need your projections to be drawn specifically in the order they where created, instead of instanced together and drawn in a pseudo random order. Whether instanced or not, the priority of the projection will be respected, as this is set projection wide.

6.17.2.14 ForceForward

```
bool LlockhamIndustries.Decals.Projection.ForceForward [get], [set]
```

Defines whether this projection should be forced to render in a forward renderloop. This only affects those using deferred rendering. This is useful when you need to draw decals on objects that are rendered in the forward rendering loop, usually objects with shaders that don't support deferred rendering.

6.18 LlockhamIndustries.Decals.ProjectionBlocker Class Reference

Inherits MonoBehaviour.

6.18.1 Detailed Description

This component blocks a camera from rendering decals. Useful if you only want some of your cameras to render decals in the scene.

6.19 LlockhamIndustries.Decals.ProjectionPool Class Reference

Public Member Functions

- bool CheckIntersecting (Vector3 Point, float intersectionStrength)
- ProjectionRenderer Request (ProjectionRenderer Renderer=null, bool IncludeBehaviours=false)

Static Public Member Functions

- static ProjectionPool GetPool (string Title)
- static ProjectionPool GetPool (int ID)

6.19.1 Detailed Description

In-built projection pooling class. Use ProjectionPool.GetPool() to get a reference to a pool instance or use ProjectionPool.Default to get a reference to the default pool. You can the request projections from the pool as you see fit. Once you are done with them, instead of deleting them, use the Return method to return them back to the pool.

6.19.2 Member Function Documentation

6.19.2.1 GetPool() [1/2]

```
static ProjectionPool LlockhamIndustries.Decals.ProjectionPool.GetPool (
            string Title )  [static]
```

Returns a pool with the specified name, if it exists. If it doesn't, returns the default pool.

Parameters

Title	The title of the pool to be returned.

6.19.2.2 GetPool() [2/2]

```
static ProjectionPool LlockhamIndustries.Decals.ProjectionPool.GetPool (
            int ID )  [static]
```

Returns a pool with the specified ID, if it exists. If it doesn't, returns the default pool.

Parameters

ID	The ID of the pool to be returned.

6.19.2.3 CheckIntersecting()

```
bool LlockhamIndustries.Decals.ProjectionPool.CheckIntersecting (
            Vector3 Point,
            float intersectionStrength )
```

Checks to see if a point is intersecting with any of the projections in the pool. Returns true if an intersecting projection is found, otherwise returns false.

Parameters

Point	The type of projection being requested.
intersectionStrength	How far within the bounds of the projection the point must be before it's considered an intersection. 0 will consider a point anywhere within a projections bounds as an intersections. 1 will only a point as intersecting if it is perfectly at the center of a projections bounds.

6.19.2.4 Request()

```
ProjectionRenderer LlockhamIndustries.Decals.ProjectionPool.Request (
            ProjectionRenderer Renderer = null,
            bool IncludeBehaviours = false )
```

Returns a projection of the specified type from the pool. Projection will be enabled and ready to use. Use the return method once your done with it, do not delete it.

Parameters

Renderer	Optional - The renderer to copy from. In 90% of use cases this should be a prefab.
IncludeScripts	Optional - Should the renderer being copied have it's scripts copied as well?

6.20 LlockhamIndustries.Decals.RandomScale Class Reference

Inherits MonoBehaviour.

Public Attributes

- float minSize = 0.5f
- float maxSize = 0.8f

6.20.1 Detailed Description

Randomizes the initial scale of your projection. Designed to be printed with your projections. Attach to your prefab and enable print behaviours on your printer.

6.20.2 Member Data Documentation

6.20.2.1 minSize

```
float LlockhamIndustries.Decals.RandomScale.minSize = 0.5f
```

The minimum range of the randomized scale (in units).

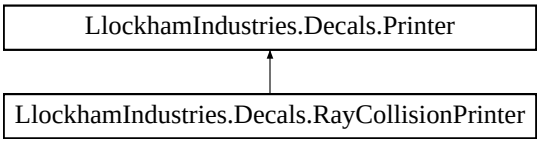
6.20.2.2 maxSize

```
float LlockhamIndustries.Decals.RandomScale.maxSize = 0.8f
```

The maximum range of the randomized scale (in units).

6.21 LlockhamIndustries.Decals.RayCollisionPrinter Class Reference

Inheritance diagram for LlockhamIndustries.Decals.RayCollisionPrinter:



Public Attributes

- CollisionCondition condition
- float conditionTime = 1
- LayerMask layers
- CastMethod method
- Transform castCenter
- Vector2 castDimensions
- Vector3 positionOffset
- Vector3 rotationOffset
- float castLength = 1
- QueryTriggerInteraction hitTriggers = QueryTriggerInteraction.UseGlobal

Additional Inherited Members

6.21.1 Detailed Description

The RayCollisionPrinter Component. Given a transform, it projects a ray that starts at the transforms position and casts in the transforms forward direction. It then prints a projection under set conditions relating to that raycast.

6.21.2 Member Data Documentation

6.21.2.1 condition

```
CollisionCondition LlockhamIndustries.Decals.RayCollisionPrinter.condition
```

Defines the condition on which a projection is printed. Enter will print whenever a ray-collision occurs. Delay will print the conditionTime seconds after a ray-collision occurs. Constant will print every fixed update during a ray-collision. Exit will print upon exiting a ray-collision.

6.21.2.2 conditionTime

```
float LlockhamIndustries.Decals.RayCollisionPrinter.conditionTime = 1
```

If the collision condition is set to delay, the conditionTime determines the length of that delay.

6.21.2.3 layers

```
LayerMask LlockhamIndustries.Decals.RayCollisionPrinter.layers
```

The layers that, when hit by a ray with, cause a print.

6.21.2.4 method

```
CastMethod LlockhamIndustries.Decals.RayCollisionPrinter.method
```

Should you cast from a single point, or randomly within an area.

6.21.2.5 castCenter

```
Transform LlockhamIndustries.Decals.RayCollisionPrinter.castCenter
```

The transform that defines the collision ray. If left null will default to the attached transform. The transforms position will be used as a base for the rays starting position & it's forward direction will be used as a base for the rays direction.

6.21.2.6 castDimensions

```
Vector2 LlockhamIndustries.Decals.RayCollisionPrinter.castDimensions
```

The dimensions of the cast area. Only applicable if cast method is set to area.

6.21.2.7 positionOffset

```
Vector3 LlockhamIndustries.Decals.RayCollisionPrinter.positionOffset
```

The position offset is applied to the castPoint to get the starting point of the collision ray. This essentially allows you to offset the rays starting position.

6.21.2.8 rotationOffset

```
Vector3 LlockhamIndustries.Decals.RayCollisionPrinter.rotationOffset
```

The rotation offset is applied to the castPoint transforms forward direction to get the direction of the collision ray. This essentially allows you to offset the rays direction.

6.21.2.9 castLength

```
float LlockhamIndustries.Decals.RayCollisionPrinter.castLength = 1
```

The length of the ray thats cast.

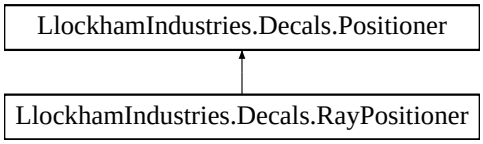
6.21.2.10 hitTriggers

```
QueryTriggerInteraction LlockhamIndustries.Decals.RayCollisionPrinter.hitTriggers = Query↩
TriggerInteraction.UseGlobal
```

Should the raycasts hit triggers.

6.22 LlockhamIndustries.Decals.RayPositioner Class Reference

Inheritance diagram for LlockhamIndustries.Decals.RayPositioner:



Public Attributes

- Transform rayTransform
- Vector3 positionOffset
- Vector3 rotationOffset
- float castLength = 100

Additional Inherited Members

6.22.1 Detailed Description

The ray positioner component. Positions a projection at the hit point of a raycast. The ray is created starting at a transforms position and casts in the transforms forward direction.

6.22.2 Member Data Documentation

6.22.2.1 rayTransform

```
Transform LlockhamIndustries.Decals.RayPositioner.rayTransform
```

The transform that acts as the base of the raycast. If null will this objects transform

6.22.2.2 positionOffset

```
Vector3 LlockhamIndustries.Decals.RayPositioner.positionOffset
```

A position offset applied to the base of the transform to get the starting position of the ray.

6.22.2.3 rotationOffset

```
Vector3 LlockhamIndustries.Decals.RayPositioner.rotationOffset
```

A rotation offset applied to the transforms forward direction to get the direction of the ray.

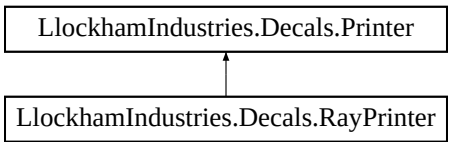
6.22.2.4 castLength

```
float LlockhamIndustries.Decals.RayPositioner.castLength = 100
```

The cast length of the ray.

6.23 LlockhamIndustries.Decals.RayPrinter Class Reference

Inheritance diagram for LlockhamIndustries.Decals.RayPrinter:



Public Member Functions

- void PrintOnRay (Ray Ray, float RayLength, Vector3 DecalUp=default(Vector3))

Public Attributes

- LayerMask layers

Additional Inherited Members

6.23.1 Detailed Description

The RayPrinter Component. Given a ray, prints a copy of a projection at the ray collision point.

6.23.2 Member Function Documentation

6.23.2.1 PrintOnRay()

```
void LlockhamIndustries.Decals.RayPrinter.PrintOnRay (
            Ray Ray,
            float RayLength,
            Vector3 DecalUp = default(Vector3) )
```

Prints immediately at the hit position of a raycast, with a rotation relative to the normal of the surface hit. This should be used in situations where you want to define when and how you print. Bullet-holes or laser-burns for a hitscan weapon would be perfect use-cases for this printer.

Parameters

Ray	The ray to be cast, to determine the print position and rotation.
RayLength	The length of the ray being cast.
DecalUp	The rotation of the decal relative to the surface. By default, points upwards.

6.23.3 Member Data Documentation

6.23.3.1 layers

`LayerMask LlockhamIndustries.Decals.RayPrinter.layers`

The layers the raycast collide with, this should be set in editor or once at initialization.

6.24 LlockhamIndustries.Decals.SceneLayers Class Reference

Inherits MonoBehaviour.

6.24.1 Detailed Description

Allows a single scene to have unique masking layers seperate to the rest of the project. These masking layers will take effect when the component is enabled (on scene load) and the original layers returned when the component is disabled (on scene end).

6.25 LlockhamIndustries.Decals.SheetAnimator Class Reference

Inherits MonoBehaviour.

Public Member Functions

- void Play ()
- void Pause ()
- void Stop ()

Public Attributes

- int collumns
- int rows
- float speed
- int skipFirst
- int skipLast
- bool invertY
- bool destroyOnComplete

6.25.1 Detailed Description

This component allows you to animate your projections. Attach to a projection renderer with a projection set up as a sprite sheet. Designed to be printed with your projections. Attach to your prefab and enable print behaviours on your printer.

6.25.2 Member Function Documentation

6.25.2.1 Play()

```
void LlockhamIndustries.Decals.SheetAnimator.Play ( )
```

Plays the sprite animation.

6.25.2.2 Pause()

```
void LlockhamIndustries.Decals.SheetAnimator.Pause ( )
```

Pauses the sprite animation. Calling Play() will begin the animation again from the current position.

6.25.2.3 Stop()

```
void LlockhamIndustries.Decals.SheetAnimator.Stop ( )
```

Stops the sprite animation. Calling Play() will begin the animation from the begining.

6.25.3 Member Data Documentation

6.25.3.1 collumns

```
int LlockhamIndustries.Decals.SheetAnimator.collumns
```

The number of collumns in the sprite sheet being sampled.

6.25.3.2 rows

```
int LlockhamIndustries.Decals.SheetAnimator.rows
```

The number of rows in the sprite sheet being sampled.

6.25.3.3 speed

```
float LlockhamIndustries.Decals.SheetAnimator.speed
```

The playback speed, in frames per second.

6.25.3.4 skipFirst

```
int LlockhamIndustries.Decals.SheetAnimator.skipFirst
```

Skip the first x frames of the animation.

6.25.3.5 skipLast

```
int LlockhamIndustries.Decals.SheetAnimator.skipLast
```

Skip the last x frames of the animation.

6.25.3.6 invertY

```
bool LlockhamIndustries.Decals.SheetAnimator.invertY
```

Sample frames from the bottom instead of the top.

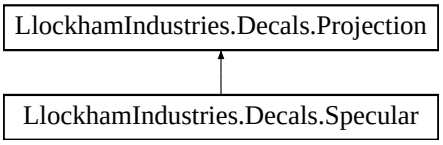
6.25.3.7 destroyOnComplete

```
bool LlockhamIndustries.Decals.SheetAnimator.destroyOnComplete
```

Destroy the projection when the animator has finished its first loop.

6.26 LlockhamIndustries.Decals.Specular Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Specular:



Public Attributes

- AlbedoPropertyGroup albedo
- SpecularPropertyGroup specular
- NormalPropertyGroup normal
- EmissivePropertyGroup emissive

Additional Inherited Members

6.26.1 Detailed Description

Standard Shader - specular setup.

6.26.2 Member Data Documentation

6.26.2.1 albedo

`AlbedoPropertyGroup LlockhamIndustries.Decals.Specular.albedo`

The primary color details of your projection. The alpha channel of these properties is used to determine the projections transparency.

6.26.2.2 specular

`SpecularPropertyGroup LlockhamIndustries.Decals.Specular.specular`

The specular texture, with a color multiplier. Determines how the light bouncing off the surface of the decal appears. The color defines the color of the light that reflects of your surface. The alpha defines how glossy your surface appears.

6.26.2.3 normal

`NormalPropertyGroup LlockhamIndustries.Decals.Specular.normal`

The normal texture of your decal, multiplied by the normal strength. Normals determine how the surface of your decal interacts with lights.

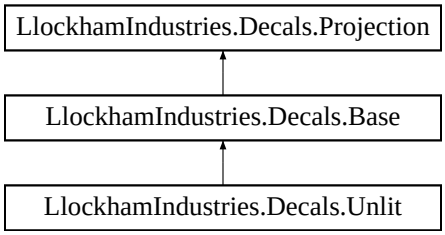
6.26.2.4 emissive

`EmissivePropertyGroup LlockhamIndustries.Decals.Specular.emissive`

The emission texture of your projection, multiplied by the emission color and intensity. Emission allows us to make a decal appear as if it's emitting light. Supports HDR.

6.27 LlockhamIndustries.Decals.Unlit Class Reference

Inheritance diagram for LlockhamIndustries.Decals.Unlit:



Additional Inherited Members

6.27.1 Detailed Description

Unlit projection. Draws a flat color to the screen. Useful for projected UI elements. If rendering in deferred, will be drawn in forward, after all other projections.