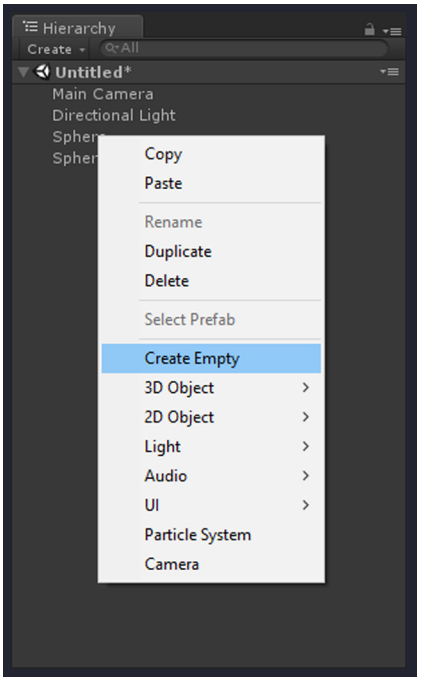
Getting Started

This guide assumes you have just installed the package & have no knowledge of the system, but at least some knowledge of Unity. If your worried about setup, don't be, there is none to speak of, you can start right away.



We'll start by creating a simple decal and run through using it. It's important to note that all projections within this package, project onto other geometry. So if your starting in one of your own scenes, great! If your in an empty scene, you'll need to populate it with a few primitives so we have something to work with.

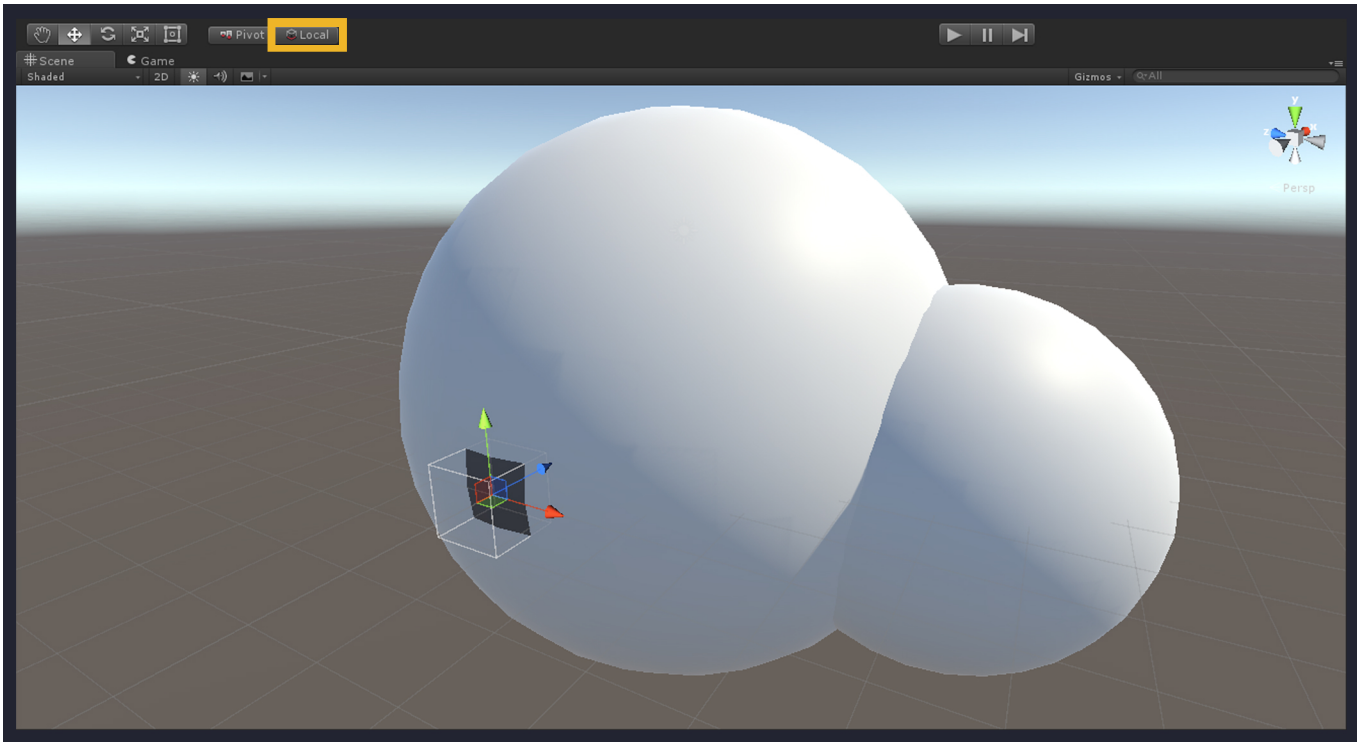
Okay, now that we have a scene to work with, lets create a decal. This is really simple, just create an empty gameObject (Right click in an empty part of the heirarchy tab, Create Empty) & add a Projection Renderer component to it.

Once we have our renderer, we'll need a projection to render. Projections are essentially materials specifially for projection renderers. Multiple renderers can share the same Projection, so once you've setup one you can reuse on multiple renderers. There are a few projection types, each for different purposes. For now pick "Metallic". This is the equivalent to Unity's Standard Shader.

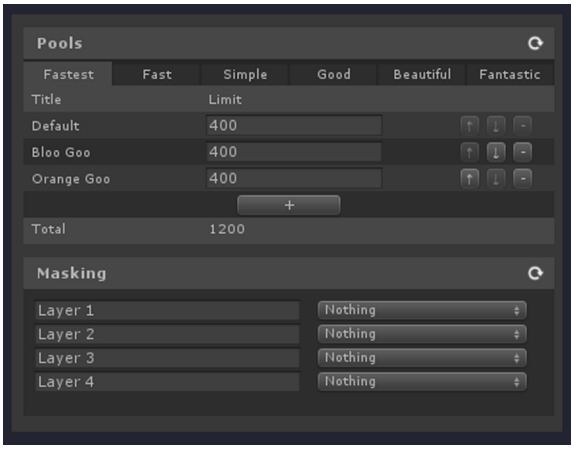
We'll need to position it so that its bounds are colliding with another object. If you can't see the decal bounds, try setting it's position to origin (0,0,0) then looking towards the centre of your scene. Also make sure the Projection Renderer component has not been collapsed, as this will hide the gizmo cube required to see it.
Now position it so its intersecting with, and pointing towards, your geometry
(The decal projects in the forward Z axis (local blue arrow))

You should now see your decal projecting onto another object! It's just a grey square at this stage, but within the component there's a ton of options to customise whats being projected. If you want to move, rotate, resize or stretch the decal, manipulate it with the transform component like you would any other object. And if you want to create copies, make a prefab and instantiate. It's that simple.

There are a ton of additional features like printers, positioners and pooling included in the system to help manage these things for you, but these are entriely optional, and decoupled from the core system. If you want to strike out on your own and build your own systems to manage instantiating, pooling and positioning decals at run-time you are free to do so.



The Settings



The settings menu allows you to tune the system to your specific needs. It can be found under:

Window > Dynamic Decals > Settings

The top section of the window is dedicated to setting up pools. Here you can add, rename, reorder and remove pools as you please. It's recommended to use different pools for different purposes, ie. if you use the system for both blood decals and bullet-hole decals, they should have separate pools. You can also adjust the pool limit for each pool, per quality level. To change which quality level your adjusting, use the tabs at the top of the window. This allows you to tweak the system to the get best performance on a range of devices.

The lower section of the window contains your mask layers. Here you can name each layer and setup which Unity layers compose each mask layer. For performance reasons, we can only have four layers, but each layer can consist of as many or few Unity layers as you please. These mask layers are used throughout the system to mask which objects can be projected onto by decals. You might, for example, have your terrain on one layer and characters on another. You could then have ootprint decals that only print on the terrain, while having bullethole decals that project onto everything but your characters.

Optimization

If you have multiple cameras in your scene and you only need projections on specific cameras (ie. not required on MiniMap) you can use the ProjectionBlocker component to tell specified camera's not to draw projections. Simply attach the component to the camera(s) not requiring projections and all shader replacement and rendering will be ignored on those cameras.

The system will automatically detect and optimize for your platform / rendering path behind the scenes for you. Choosing the cameras you wish to ignore projections is all you need to worry about.

Other Resources

Once your familiar with the Projection Renderer and the settings, you should start to familiarize yourself with the system as a whole.

Included within the package is a showcase scene and a few demo scenes. The showcase scene demonstrates the different projection types, as well as basic applications for printers and positioners. The other scenes aim to showcase practical applications of the system. Check these out at your leisure.

Included within the package is also a full scripting reference, in pdf format. If you have access to the internet though, there is a more inuitive version hosted online at -
http://www.llockhamindustries.com/documentation/dynamic-decals/