Standard Deferred Decals

Standard Deferred Decals is a material for Unity that allows you to add mesh-based non-projected decals into your projects. This means that objects, that have semi-transparent albedo textures (or have values in the alpha channel of their albedo textures that are between 0 and 1) will start smoothly blending with the rest of the environment, instead of having sharp transitions like with the Cutout shader. For all intents and purposes SDD decals use the very same Standard shader, so you can be sure that they will look consistent with the rest of the world. However, there are a few limitations that should be noted when using them:

1. SDD material renders every decal twice which can be tasking on your CPU if many decals are present in the scene.
2. Render queue matter when working with multiple decals that overlap. Please make sure that you've set the higher render queue for decals that should be on top of other decals.
3. By default decals don't write depth into depth buffer to prevent light discontinuities between decals and decal recipient surfaces, however, there are a few rare cases when decals should actually be rendering depth. This is when decals are mostly opaque and change the silhouette of the recipient model (see the rocks example in the demo scene). Be careful though, the SDD shader is not a magic bullet that introduces lit transparent objects into the deferred pipeline. If you position your decal that writes depth far away from the recipient model, you will most likely see lighting glitches under certain lighting conditions.
4. Keep in mind, that while decals do not render depth into the depth texture, they still do depth checks to ensure they don't appear on top of dynamic objects (like players character walking on top of decals).
5. Decals should be located as close to their recipient surface as possible to prevent parallax effect that can break immersion.

6. This decal solution does not support projected decals, because projected decals cannot be marked as static and participate in baked lighting, which would make them inconsistent with objects that use the Standard shader.
7. There is no way to specify which objects should be clear of decals at all times.
8. Unlike the Standard shader, SDD shader has override alpha values that allow you to specify how much would you want a specific decal's channel to be written on top of the base layer. So, for instance, if you want your decal to only write the diffuse texture – you can set all the override alphas but the albedo one to zero. On the other hand, if you want your base normal map to partially show underneath your decal – try setting Normal Map alpha to 0.5.

The documentation does not have many screenshots and it is intentional, it is instead advised to open the demo scene and play with various objects in there to get a good understanding of how the shader works. The demo also has a number of text boxes near points of interest that showcase the strengths and weaknesses of the system in detail.

I hope you enjoy the shader, feel free to message me at cubrman@gmail.com if you have any questions.