LIGHT2D

The definitive 2D Lighting Solution

Version 1.1

Thank you for purchasing Light2D! Although ease of use was one of the key goals of the plugin's design, all software needs documentation these days. This brief PDF will get you started on making cool things with Light2D.

Creating a 2D Light

New lights are created via the GameObject menu. Specifically in:

GameObject > Light > 2D

From there you'll find the two options for basic lights: Additive and Alpha. Either option will create a new GameObject with a basic Light2D setup. The key difference will be the material's shader either performing an additive color blend or performing an alpha blend (more info on blending at https://en.wikipedia.org/wiki/Blend_modes).

Customizing a 2D Light

If you've followed the above step, you've already seen the various editor control handles and properties of a new light. Accessible in the editor and through code, these are:

Range
The radius of the light in standard Unity units.
Light Color
The color of the light. Uses vertex coloring to save batches.
Intensity
A scalar multiple for the Color property. Allows for color values >1, and therefore appearing more bright in the scene.
Angle
The angle in degrees that the conical shape covers. Note that the center of the "seam" of the cone is the direction of `-Transform.up`.
Shadow Mask
A LayerMask that governs which objects cast shadows. Only objects on the layers included in this mask will be considered when building the Light2D mesh. Other objects with colliders should be kept off these layers.

Sorting Layer
The sorting layer used. Works just as `SpriteRenderer` and `ParticleSystem` do.
Order in Layer
The order in the sprite layer. Again, works as you would expect.

Runtime Objects

While the editor is not in play-mode, Light2D constantly finds colliders to use for shadow casting. However, at runtime, only the colliders already placed in the scene are used. This is an optimization and avoids unnecessary memory allocations.

This means if you instantiate an object with a 2D Collider that you want to cast shadows, you need to tell Light2D about the object. This is done using the static method:

```
Light2D.RegisterCollider(GameObject go);
```

All Light2D objects in the scene will be notified and will now be aware of the GameObject. You do not have to deregister objects that are destroyed; Light2D will do this automatically when references to those objects become null.

Notes:

- If your GameObject is a compound collider with multiple children colliders, you will have to register each child individually
- If a new scene is loaded and a Light2D object is marked with DontDestroyOnLoad, it will automatically refresh and be aware of all colliders in the new scene
- You can manually call Refresh() per Light2D, though this is generally not recommended versus using the Light2D.RegisterCollider method as the latter is much more performant than the former

Circle Colliders

Light2D does not handle shadow casting from Circle Colliders out-of-box. Because of how Physics2D ray casting is used, CircleCollider2D objects generally cause mesh generation errors (jumpy/jittery shadows). To facilitate circular object’s shadow casting, a helpful wizard is included that will convert a GameObject’s CircleCollider2D to a PolygonCollider2D with a variable resolution.

Demo Scenes

Included with the package are various demo scenes to get you started, they include:

Simple Setup
A basic setup of several colored lights in a scene with several polygon objects.
Soft Light Setup
An example using the Blur (Optimized) Image Effect to achieve soft lighting.
Advanced Setup
Lights are rendered onto a black texture, blurred, and then blended overtop of the scene to create real darkness, where unlit areas are completely occluded by shadow.
Runtime Objects
An example of how runtime instantiated objects can register themselves with Light2D and cast shadows.

Support

Email: dan@danjohnmoran.com
Website: http://danjohnmoran.com
Asset Store: https://www.assetstore.unity3d.com/en/#!/content/48539

Thanks!

Thank you again for purchasing Light2D! Your purchase helps me make new plugins and upkeep existing ones like this. Feel free to email me or leave a review. I will attempt to respond in a timely fashion (1-2 business days) and resolve any issues you're having.

Cheers,
Dan