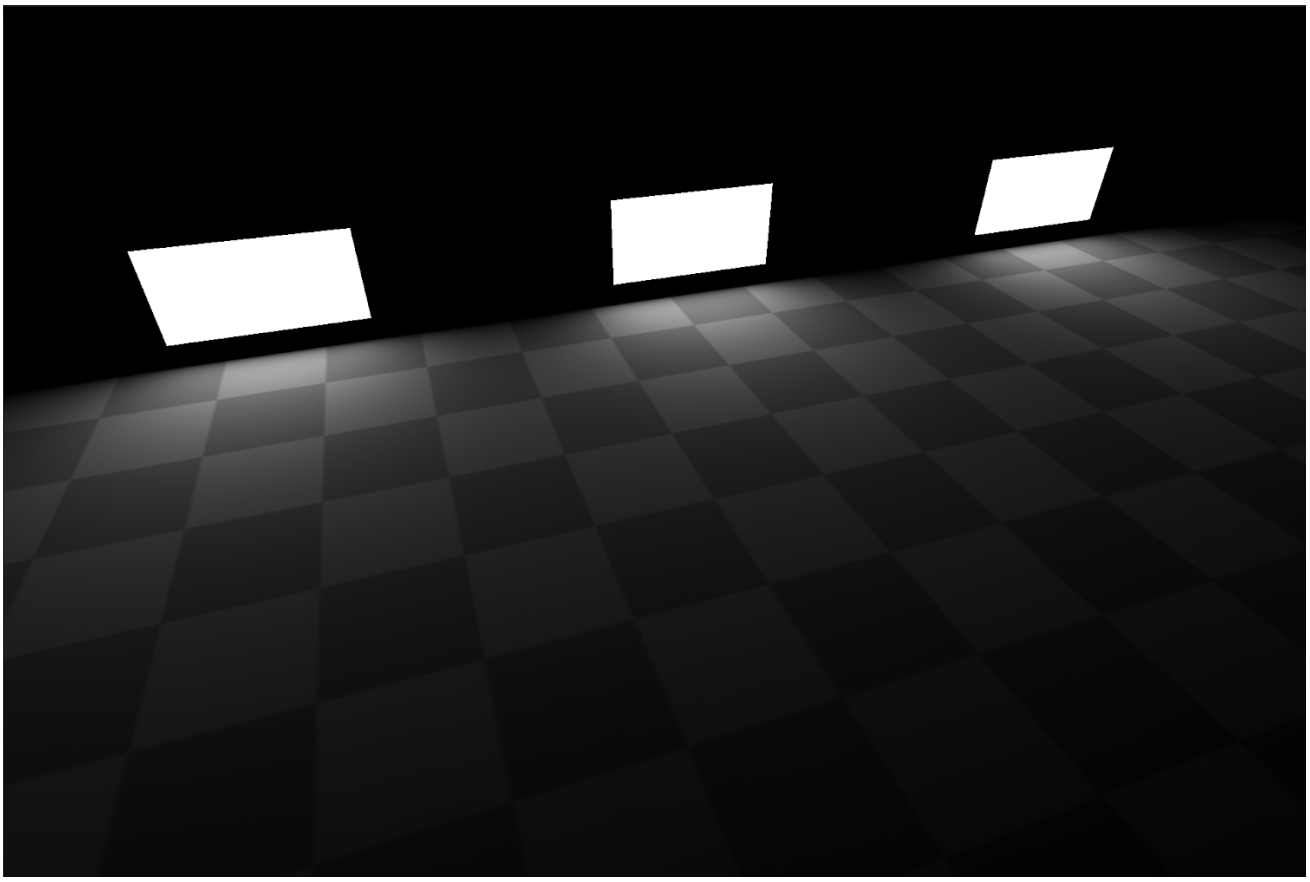
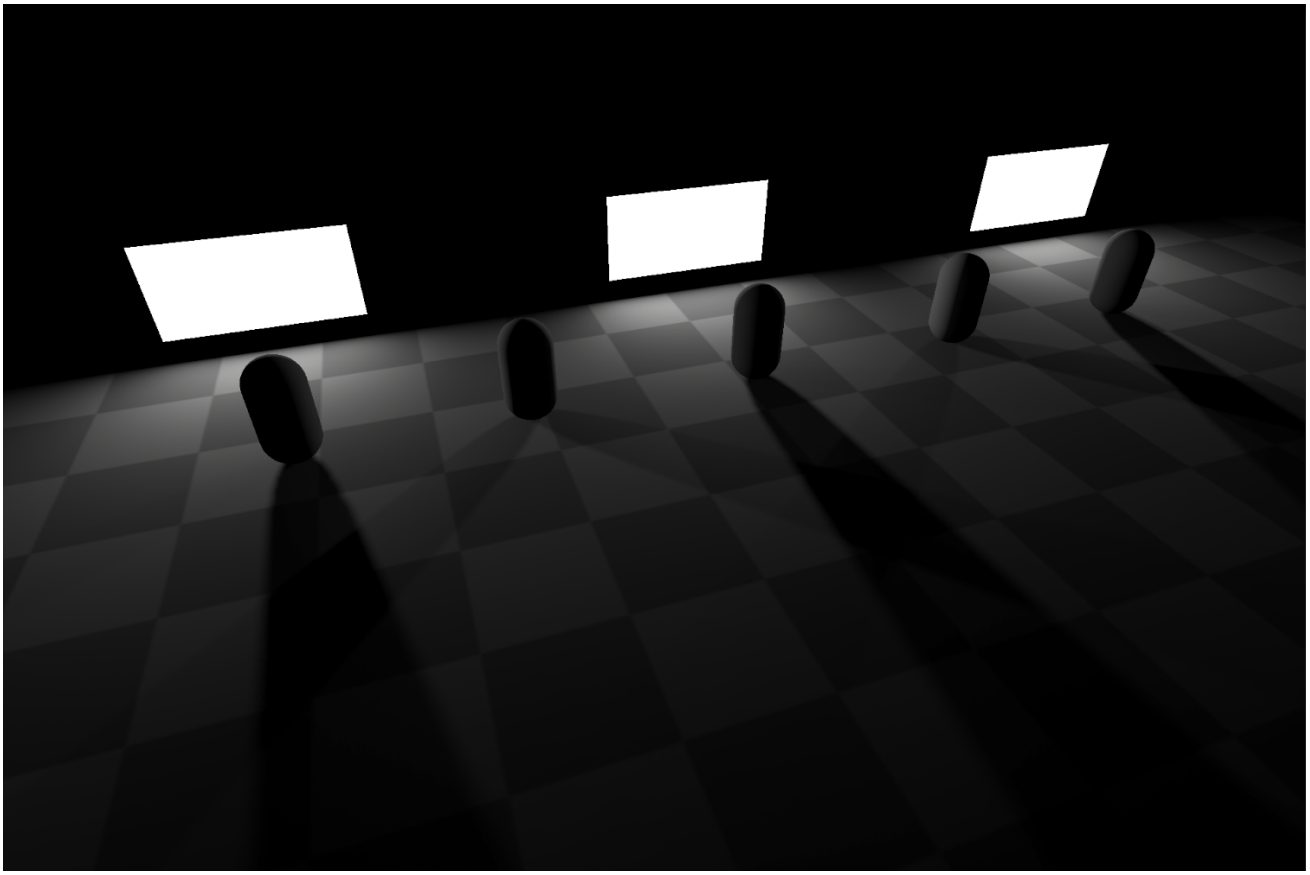
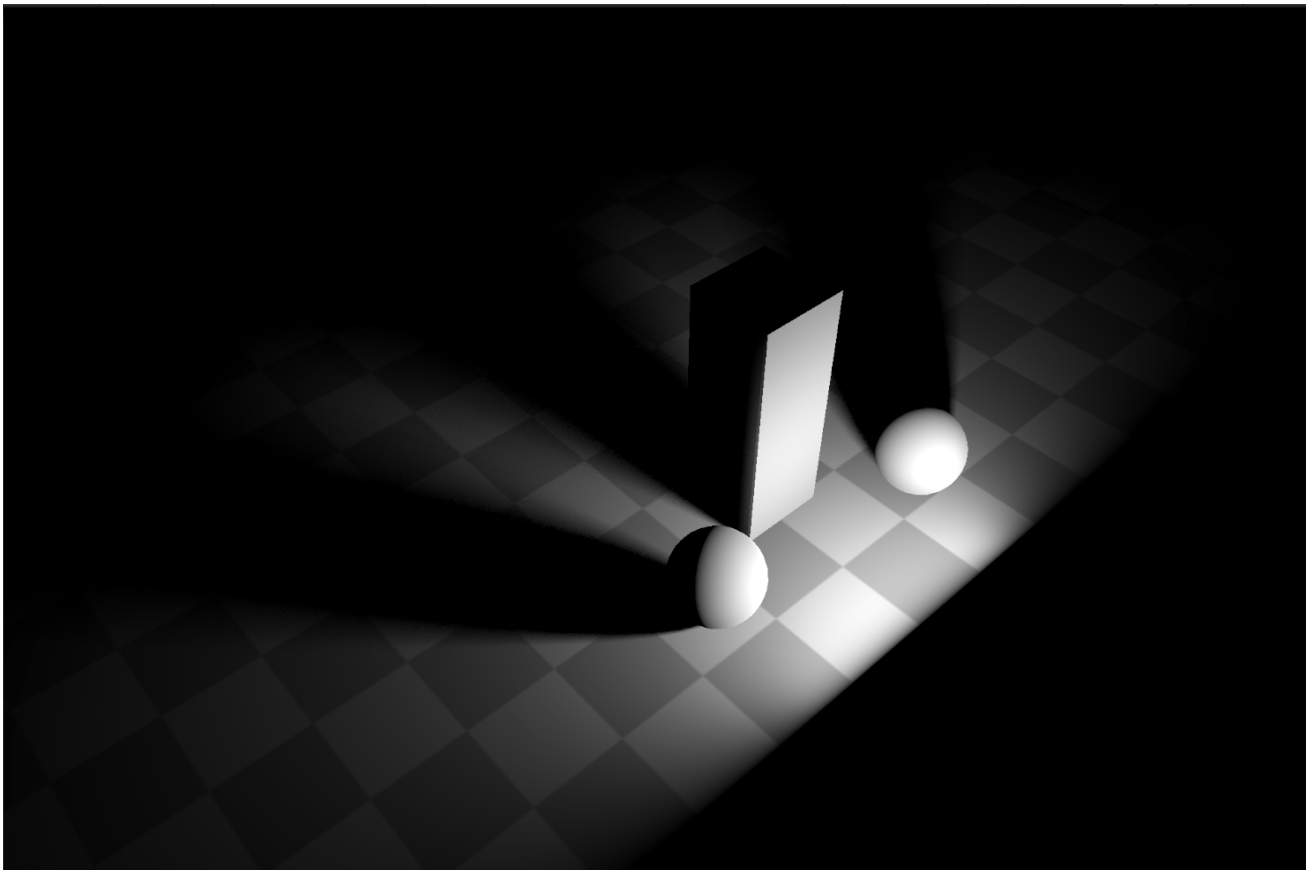
Realtime Area Light for URP

- **Introduction**
  - HDRP-style Area Lights for URP. Realtime performance with soft shadows. Fully optimized for PC and mobile.

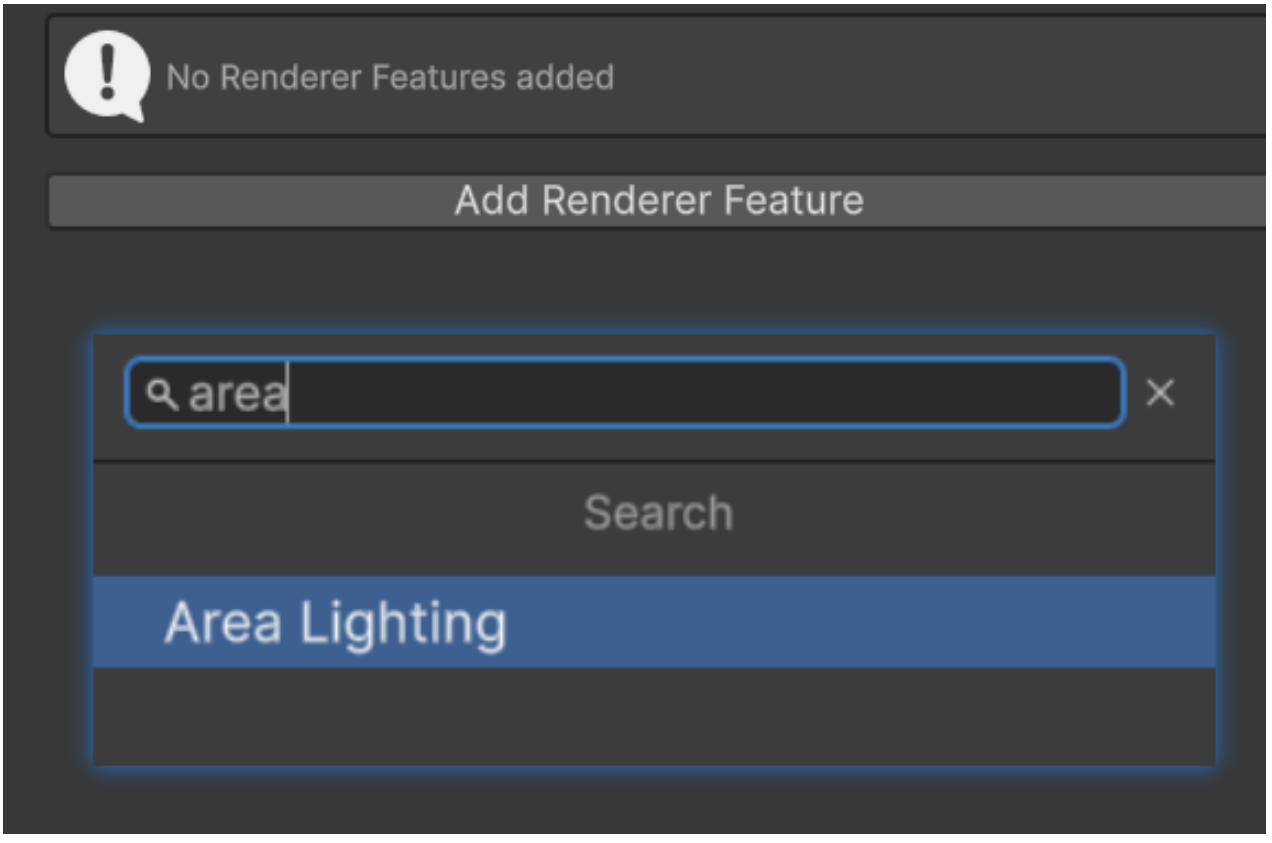




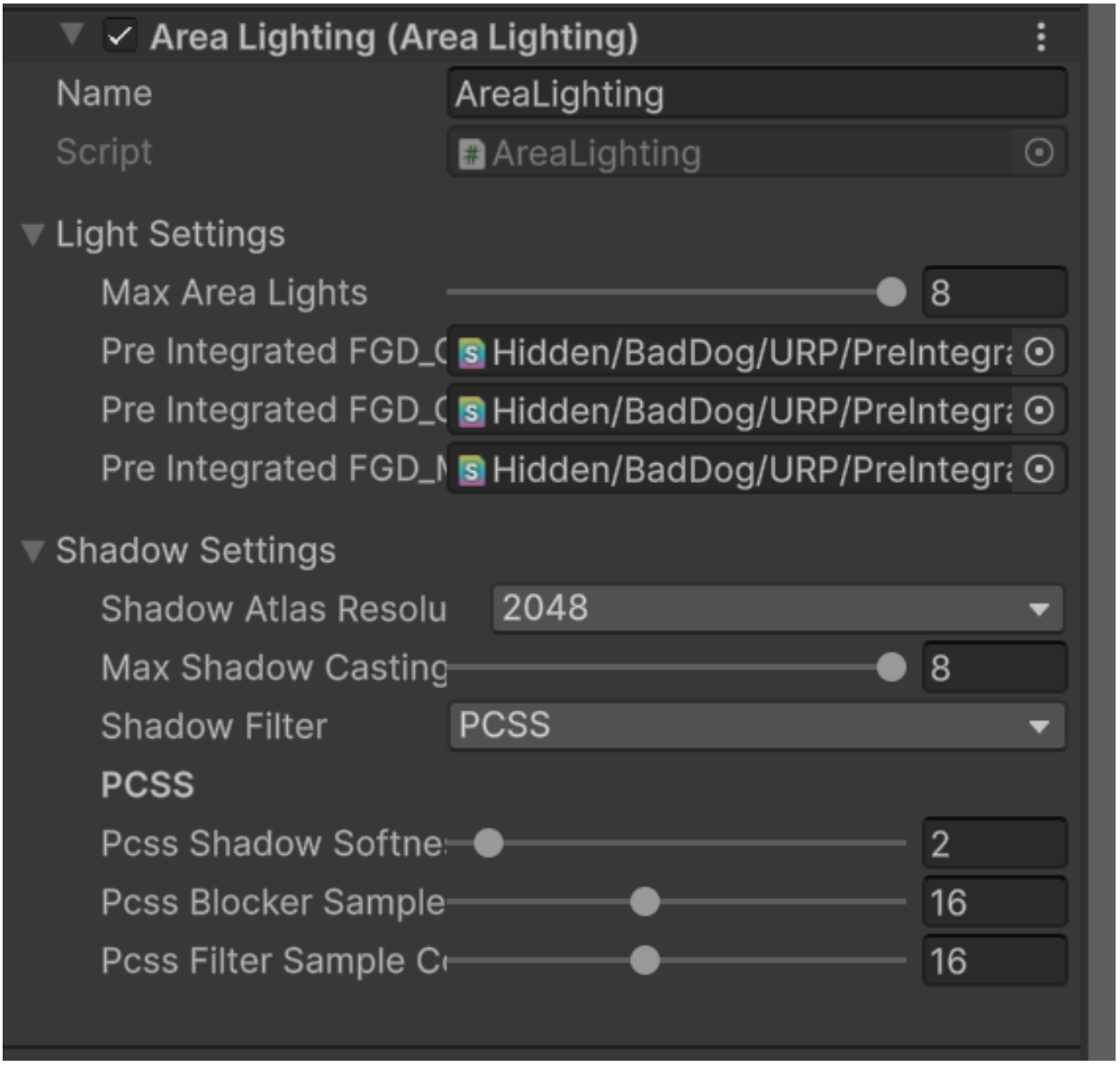


- **Core Features**
  - **HDRP-Grade Area Lights**
    - Matches HDRP's quality with real-time area lights and soft shadows.
  - **8 Concurrent Lights & Shadows**

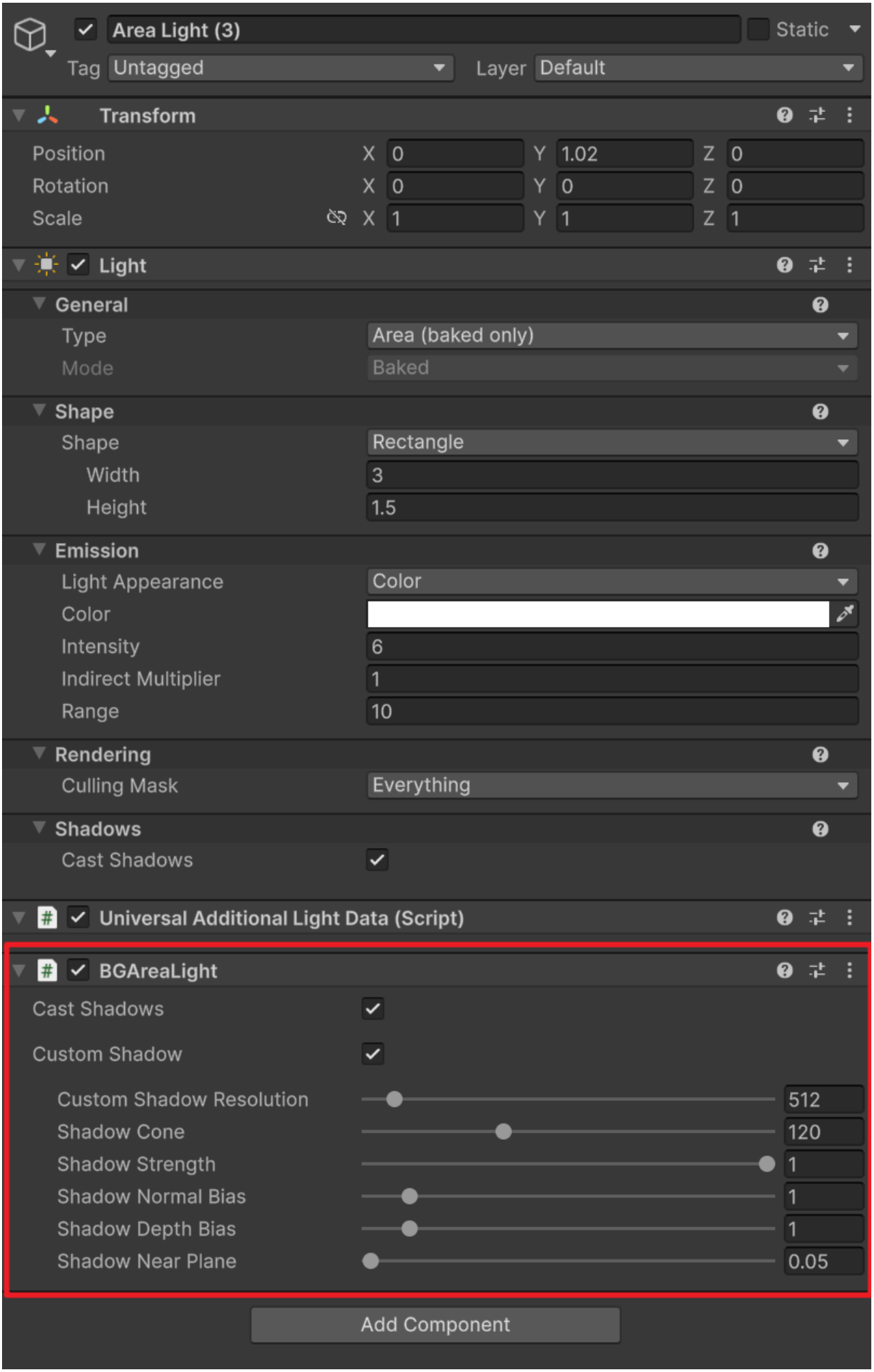
        - Supports up to 8 real-time area lights, all capable of casting dynamic shadows.
    - **Smart Shadow Management**
        - Unified shadow atlas with per-light settings and automatic priority sorting.
    - **Cross-Platform Soft Shadows**
        - Optimized algorithms: PCF for mobile performance, PCSS for PC realism.
    - **Extended Shader Support**
        - Custom-built versions of Lit, SimpleLit, and TerrainLit. Easily extendable.
    - **Plug-and-Play Integration**
        - URP Renderer Feature compatible with all render paths. Full RenderGraph and legacy mode support.
- **Install**
    - Prerequisite: Ensure your project is using Universal Render Pipeline (URP) and has a URP Renderer configured.
    - Import the `Realtime Area Light for URP` package into your project.
    - Locate your URP Renderer asset, In its Inspector window, click the `Add Renderer Feature` button, and select `Area Lighting` from the dropdown menu.



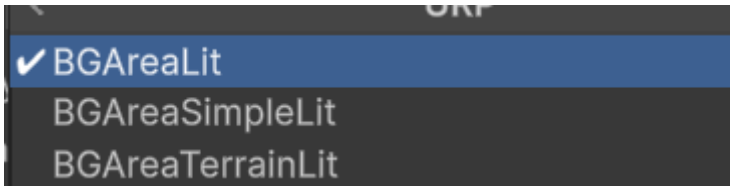
    - The Area Lighting global parameter can be adjusted here



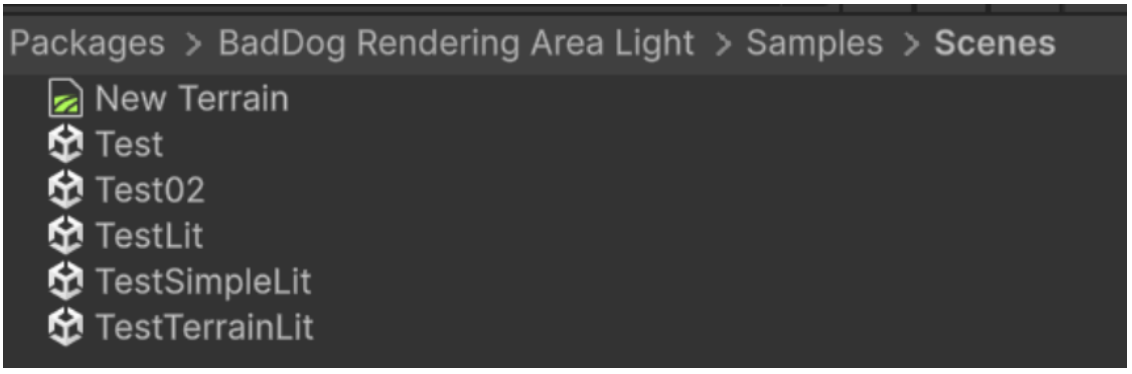
- **Next:** Create a built-in Unity Area Light and attach **BGAreaLight** script to it.

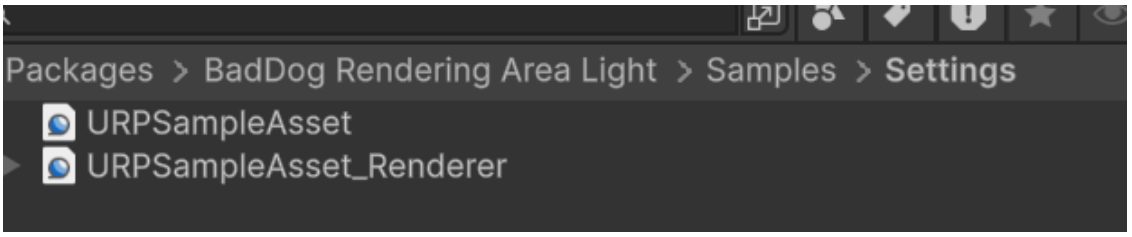

Area Light (3)
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 1.02 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Light
General
Type Area (baked only)
Mode Baked
Shape
Shape Rectangle
Width 3
Height 1.5
Emission
Light Appearance Color
Color
Intensity 6
Indirect Multiplier 1
Range 10
Rendering
Culling Mask Everything
Shadows
Cast Shadows
Universal Additional Light Data (Script)
BGAreaLight
Cast Shadows
Custom Shadow
Custom Shadow Resolution 512
Shadow Cone 120
Shadow Strength 1
Shadow Normal Bias 1
Shadow Depth Bias 1
Shadow Near Plane 0.05
Add Component

- The area light is now active. To see the complete effect, apply our pre-configured area light shader to your models; the area light and its shadows will then be fully functional.
    - For URP's built-in Lit, SimpleLit, and TerrainLit shaders, I have provided dedicated versions optimized for area lights.

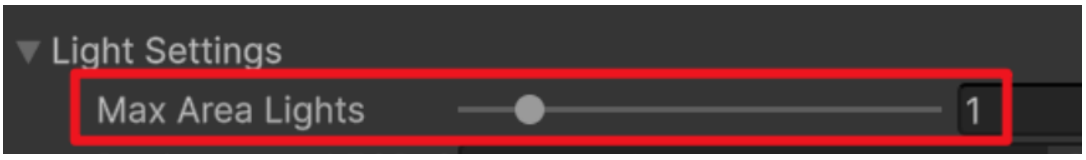


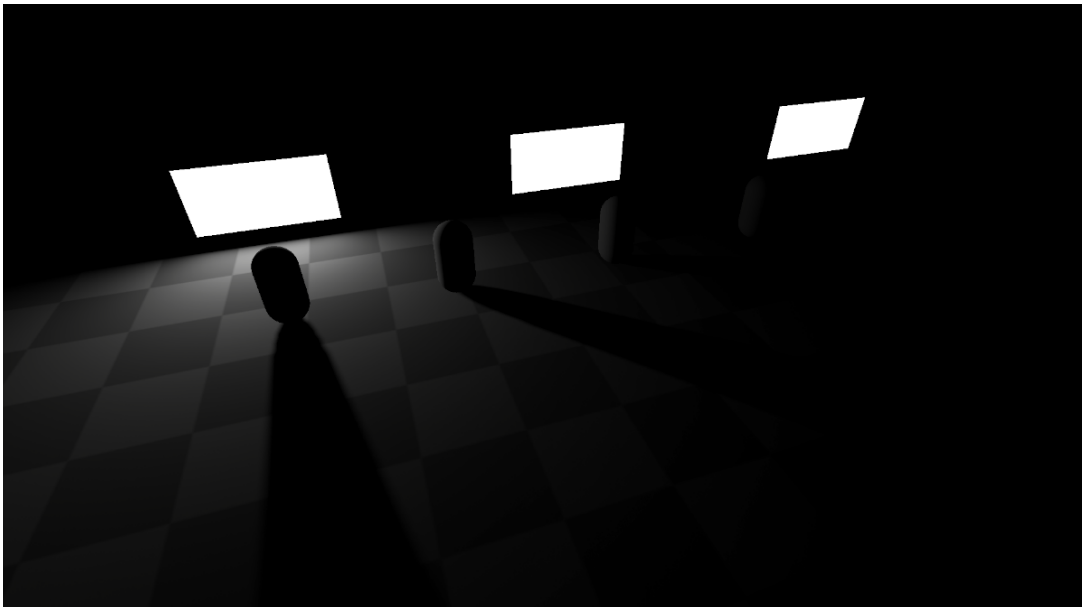
    - If you need to extend support to your own custom shaders, the process is straightforward. We'll cover this in detail in the following sections.
- You can also open the pre-configured test scene I've prepared to see it in action. `To ensure the sample scene works correctly, use the pre-configured URPSampleAsset as shown in the example below.`



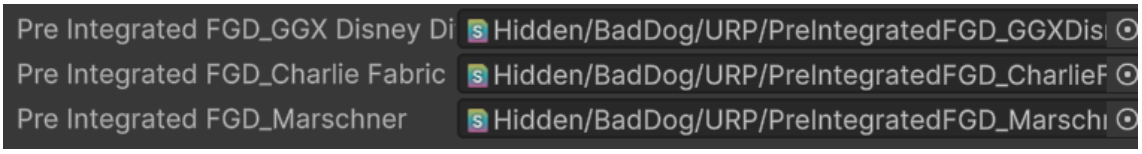


- **Parameter Description**
    - Area Lighting Global Settings
        - Light Settings
            - Max Area Lights
                - Maximum amount of area lights.
                    - Example

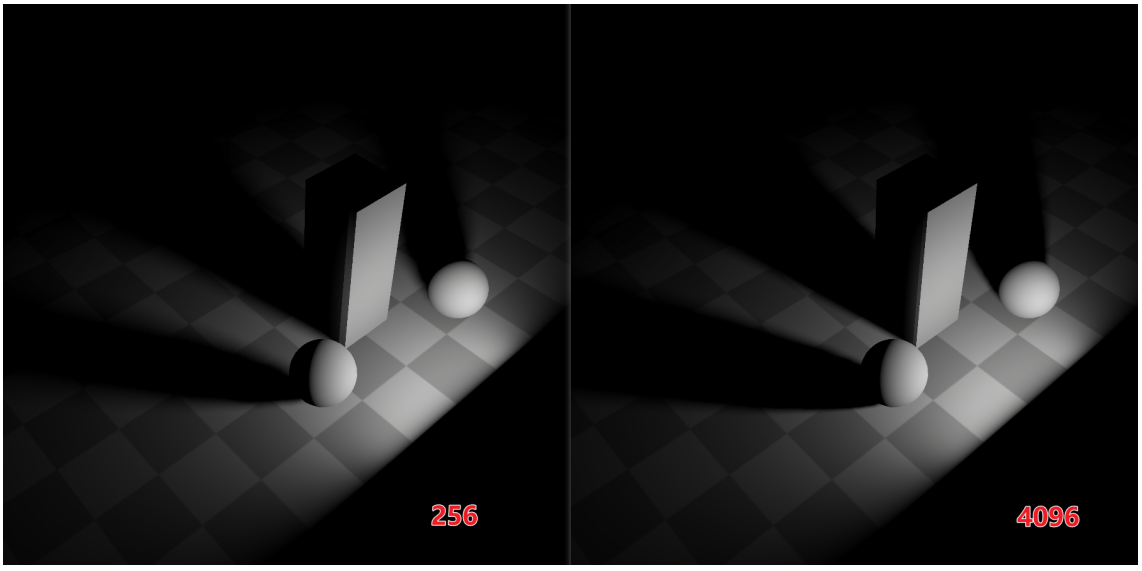




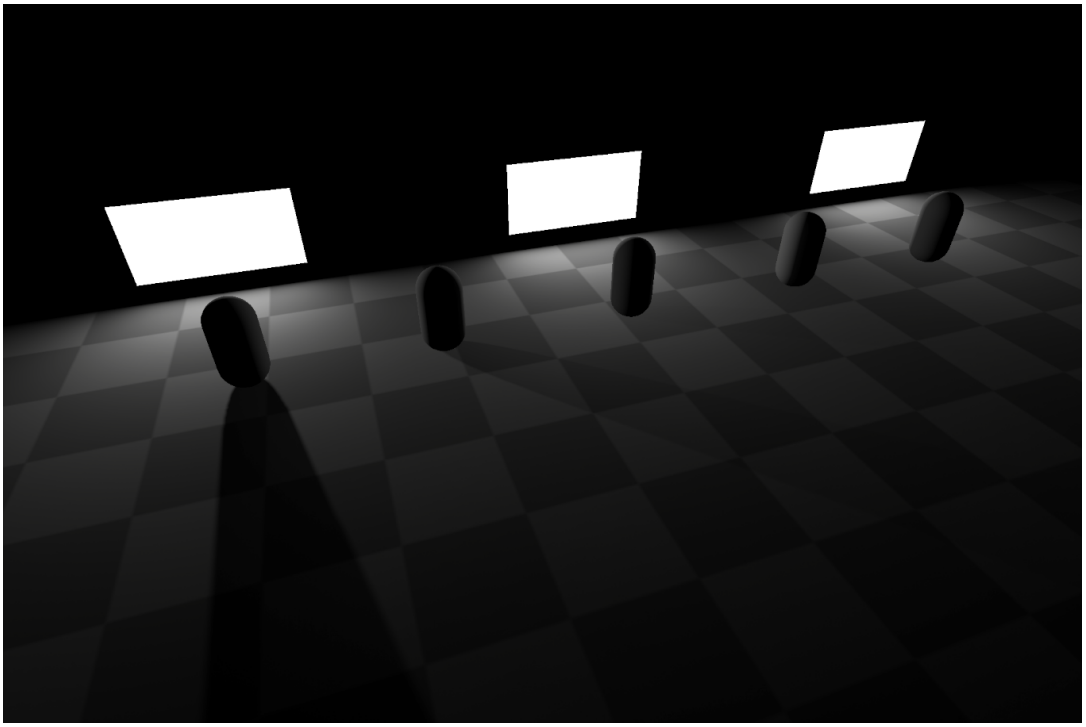
- Pre Integrated FGD_xxxx Shaders
  - These shaders are for pre-computation and are bound automatically, requiring no user attention.



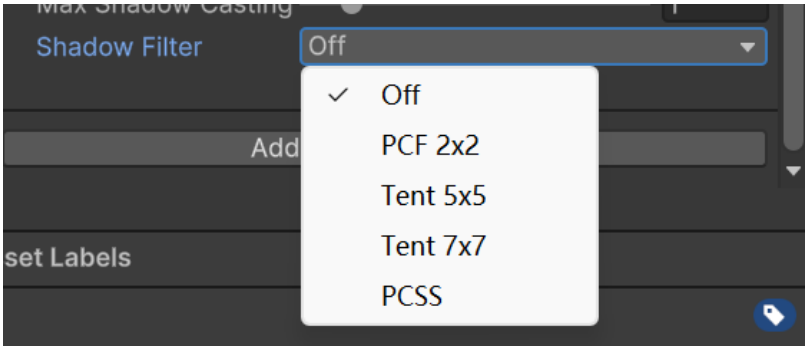
- Shadow Settings
  - Shadow Atals Resolution
    - Shadow Atlas Resolution: Powers of two, max 8192.



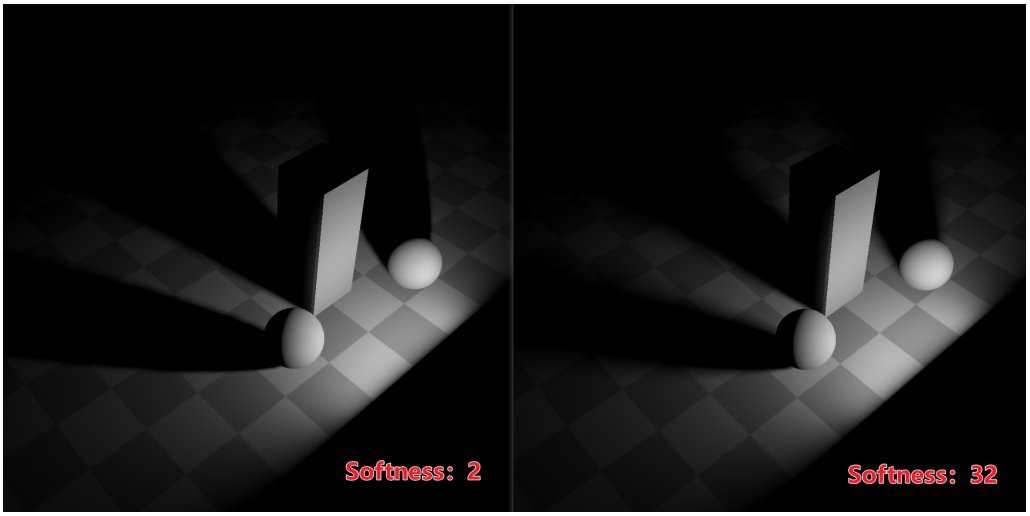
  - Max Shadow Castering
    - Maximum number of area lights that can cast shadows simultaneously
      - Example



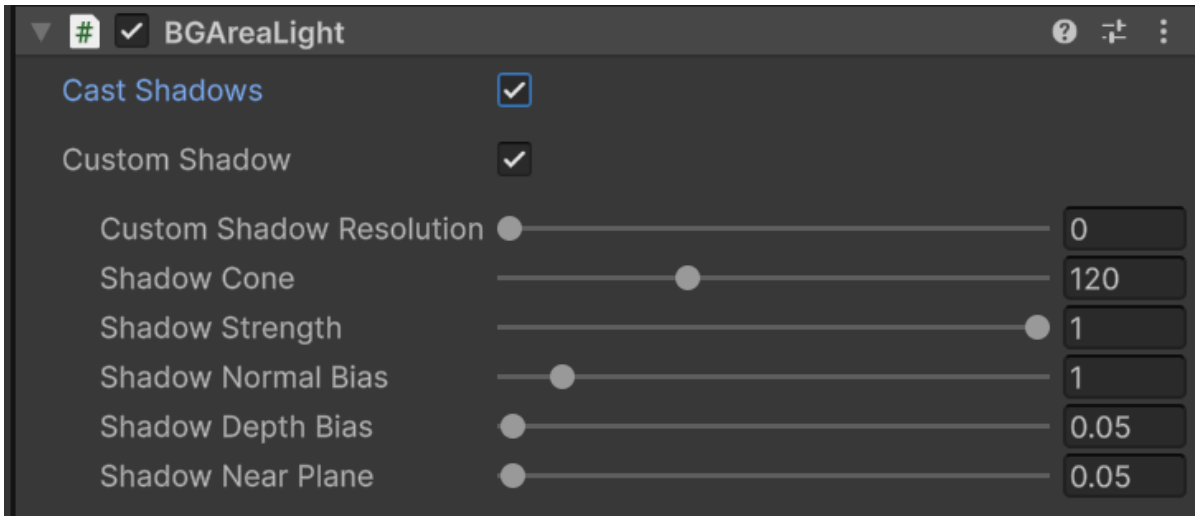
- Shadow Filter
  - Supports all URP built-in methods (PCF 2x2, Tent 5x5/7x7), plus a PCSS implementation designed for high-end visual fidelity.



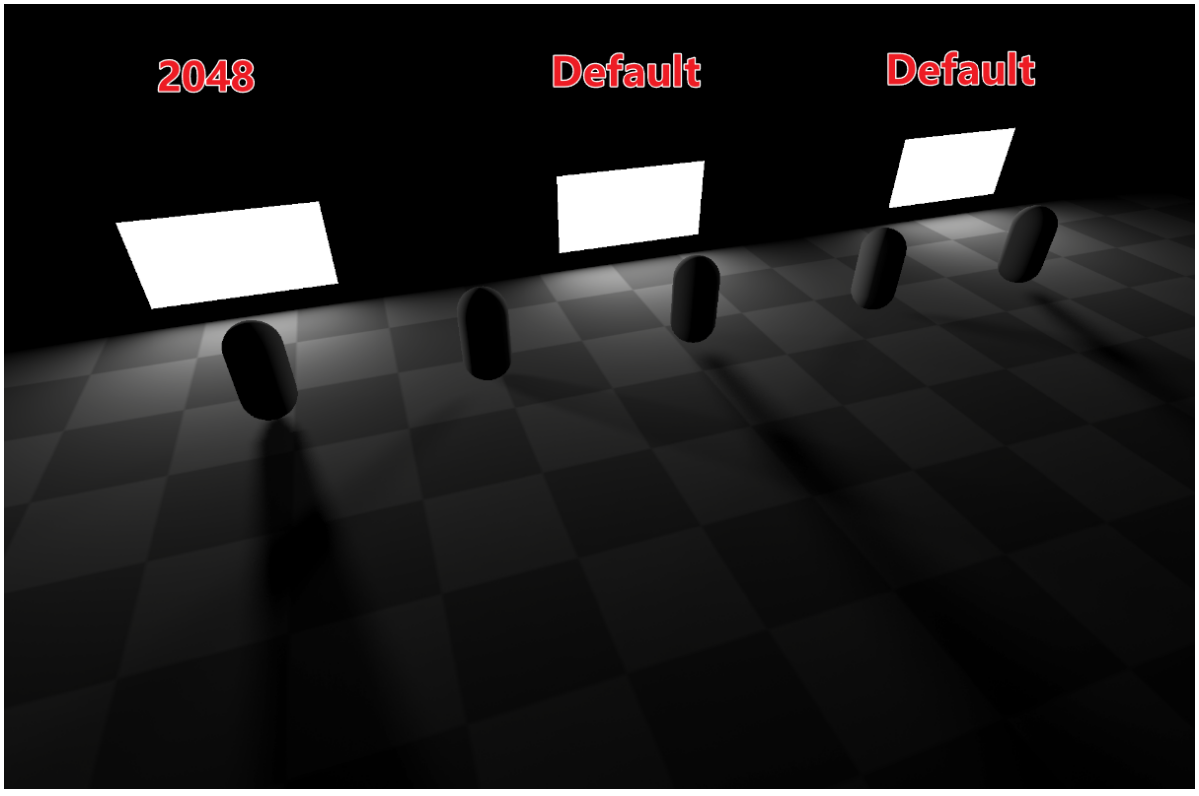
  - PCSS
    - Pcss Shadow Softness
      - Increasing this parameter produces more realistic, softer penumbrae by simulating a larger light source.



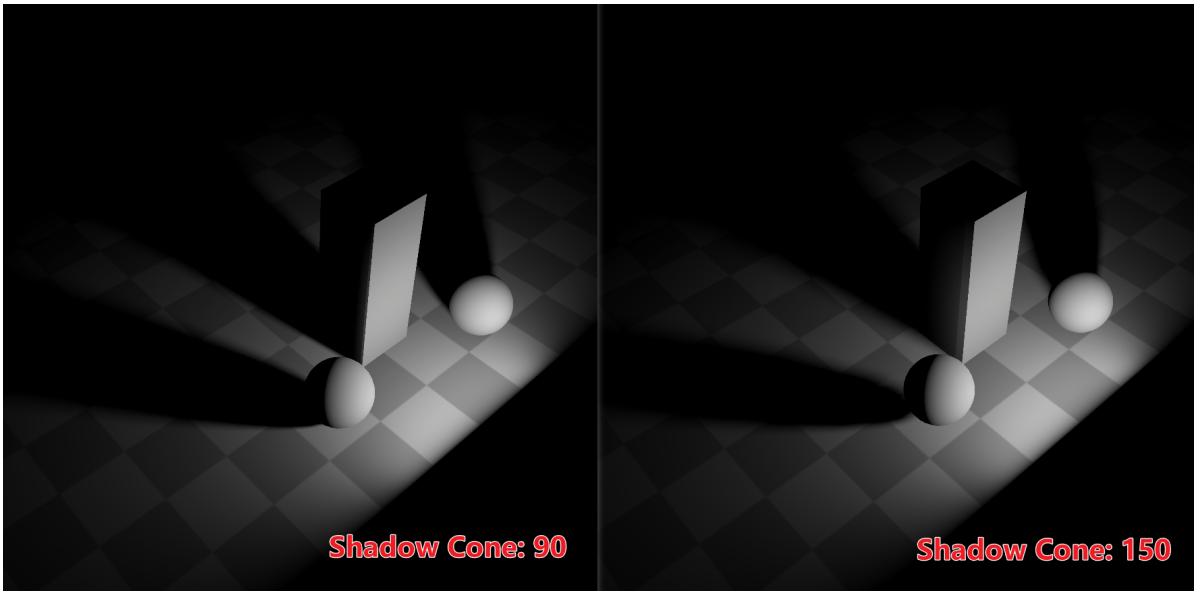
        - Pcss Blocker Sample Count/Pcss Filter Sample Count
            - Higher values yield better shadow quality at the cost of increased performance overhead.
- BGAreaLight
    - Adjust shadow quality individually for each light.
        - Inspector



    - Custom Shadow Resolution
        - You can set the shadow map resolution for each light individually.



    - Shadow Cone
        - Aperture of the cone used for shadowing the area light.



- Shadow Strength/Shadow Normal Bias/Shadow Depth Bias/Shadow Near Plane
  - Literal meaning, same as the built-in parameter in URP.

- **Quality and Performance tips**
  - The lighting for area lights uses the LTC method. The system is capped at 8 lights, but aside from the light count limit, the computation is the same.
  - The current solution supports up to 8 area light shadows, utilizing an Atlas scheme similar to URP that automatically calculates priority. However, the shadow filtering mode significantly impacts performance. Generally, PCSS should only be used on PC, while PCF is recommended for mobile platforms.
  - Also, shadow count is a major performance hit. Tune it based on the target device.
  - Finally, the current solution does not support URP's **_CLUSTER_LIGHT_LOOP**. If you require peak performance, the clustered lighting approach is superior, but this would require modifications to the URP source code, which is not provided with this plugin.
- **Adding Area Lights to Your Shader**
  - First, include the **BGAreaLighting.hlsl** header file in your shader:

```
// Include Area Light support
#include "Packages/com.baddog.rendering.arealight/Shaders/Include/BGAreaLighting.hlsl"
```

  - Then, invoke the **EvaluateAreaLight** function within your fragment shader. This function supports multiple parameter overloads — just open BGAreaLighting.hlsl to see all available signatures.

```
// Additional Area Light (only when enabled)
#if defined(_ENABLE_BG_AREA_LIGHTING)
color.rgb += EvaluateAreaLight(inputData, surfaceData);
#endif
```

  - That's all!
- **Compatibility Note**

- Compatible with all major Graphics APIs, including GLES 3. Mobile-ready.
- Support All Rendering Paths in Unity 6.
- Support RenderGraph and Compatibility Mode.