

Smart Lighting 2D

Updated: 2021 / 11 / 29

Current Asset Version: [2021.11.5](#)

Latest Documentation: [Link](#)

Roadmap: [Link](#)

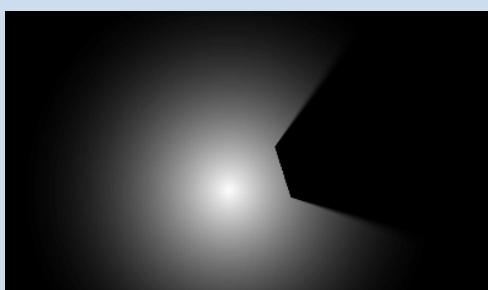
Forum Discussion: [Link](#)

Mail: simonas@kuzmickas.lt



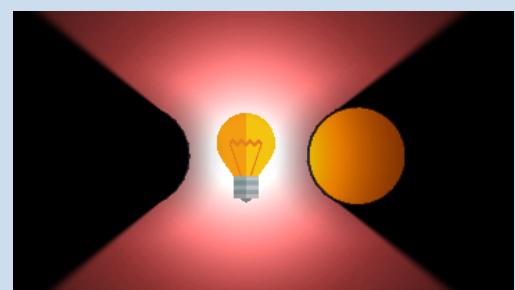
Discord Support: [Link](#)

User Manual Sections



How to Start?

A basic sample of how to set up a lighting system for a new scene.



What is Light Masking?

Explanation and Samples on how masking can be used to achieve certain results



Custom Physics Shape

What is Sprite's Custom Physics Shape?
How can we use it?



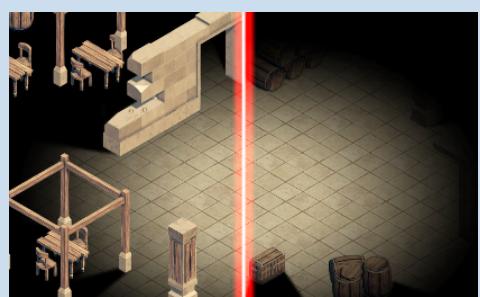
How to use Day Lighting?

A few steps to set up a lighting system for a new scene



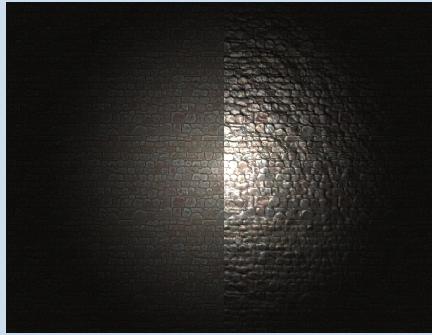
Super Tilemap Editor Support

Having issues with Super Tilemap Support?



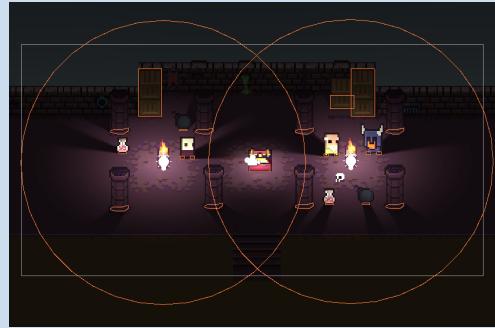
Overlay Sorting Layer

Having issues with sorting layers & order?



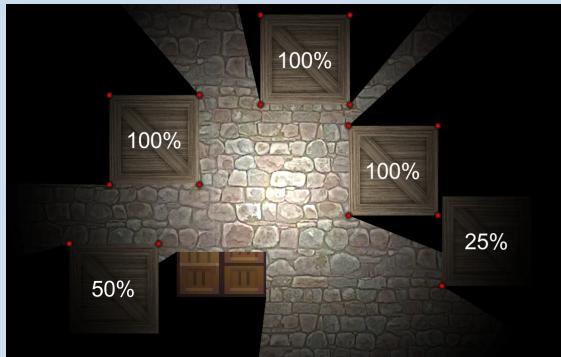
Internal Normal Maps

How to normal maps with 2D Lighting



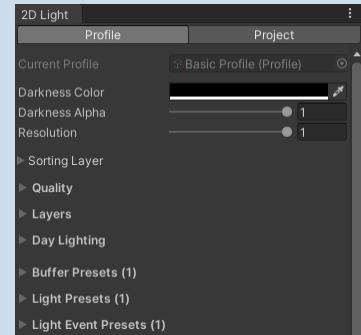
Overlay Scene View

How to setup use of 2D Lighting in Scene View



How to use Event Handling?

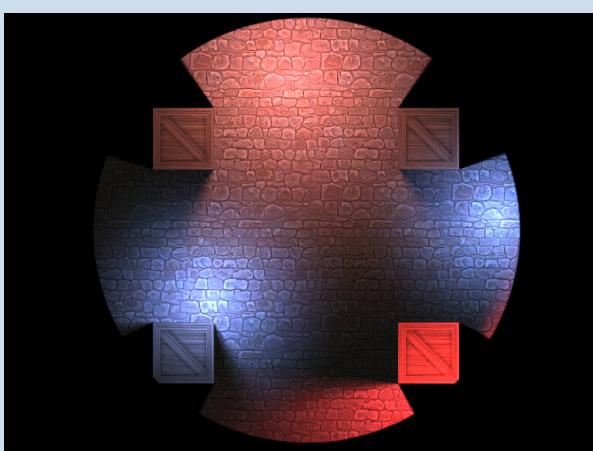
How to setup basic events



Lighting Profile (Basics)

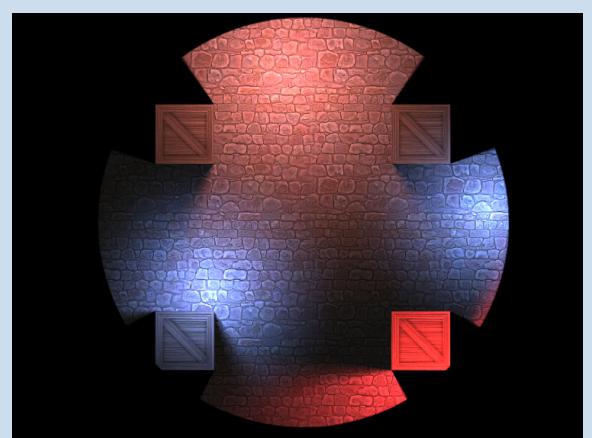
Settings asset object.

Fog of War (Simple)



Fog of War (Simple) - Part 1

Project & scene setup



Fog of War (Simple) - Part 2

Applying the setup to the game objects

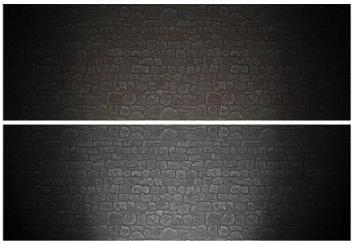
If there are any questions/feature requests or need help, feel free to contact!

I will answer the questions before you decide to use the asset.

Ask or find more information in [Discord!](#)

Documentation is in development, there will be updates!

Material System



Material Basics

Lightmaps can be used by materials



Fog of War Material

Using lightmaps with FOW material

Shadow Effects



Sprite Projection

W.I.P



Soft Shadows

W.I.P

Misc

► # <input checked="" type="checkbox"/> Light 2D (Script)	?	≠	:
► # <input checked="" type="checkbox"/> Light Sprite 2D (Script)	?	≠	:
► # <input checked="" type="checkbox"/> Light Collider 2D (Script)	?	≠	:
► # <input checked="" type="checkbox"/> Light Particle System 2D (Script)	?	≠	:
► # <input checked="" type="checkbox"/> Light Tilemap Collider 2D (Script)	?	≠	:

FAQ Common Issues (WIP)

Component Reference

A complete description of each component of the 2D Lighting System.

Shader Graph

Shader Graph API Nodes

File: SL2D_ShaderGraphAPI.hlsl

SL2D_Light (float2) => float4
SL2D_Depth (float2, float, float) => float4

SL2D_Light_1 (float2) => float4
SL2D_Light_2 (float2) => float4
SL2D_Light_3 (float2) => float4
SL2D_Light_4 (float2) => float4

SL2D_Depth_1 (float2, float, float) => float4
SL2D_Depth_2 (float2, float, float) => float4
SL2D_Depth_3 (float2, float, float) => float4
SL2D_Depth_4 (float2, float, float) => float4

SL2D_BBlend_Lit (float4, float) => float4
SL2D_BBlend_FogOfwar (float4, float4) => float4

If there are any questions/feature requests or need help, feel free to contact!

I will answer the questions before you decide to use the asset.

Ask or find more information in [Discord!](#)

Documentation is in development, there will be updates!

How to Start?

Instructions		
Step 1	<i>Creating a new Scene</i>	<i>Create a new scene</i>
Step 2	<i>Camera Setup</i>	<i>Make sure to have an orthographic mode set for the game camera. Set the scene background to be bright. If you have black background, your default setup lights & shadows won't be visible.</i>
Step 3	<i>Creating a Light</i>	<i>Create a light in the tab “GameObject/2D Light/Light”.</i>
Step 4	<i>Creating Light Manager</i>	<i>After creating the light, the Lighting Manager should be generated automatically. At this step, you should not do anything, except to check if Lighting Manager 2D is in the root of the hierarchy. If not, try to start/stop the scene.</i>
Step 5	<i>Creating Light Collider</i>	<i>Create a collider in the tab “GameObject/2D Light/Light Collider” Now you should be able to see shadows casted from the collider. Set the “Mask Lit” option to “Unlit” to make the shadow effect itself. The collider object should appear black. For making collider visuals visible, read more in “What is Masking”.</i>

Step 1	Step 2	Step 3	Step 4
Visual Explanation			
Game View			

What is Masking?

Introduction

Masking feature allows your objects to appear above the shadows.

Instructions

Step 1	<i>Setting Up Scene & Camera</i>	<p><i>Create and Setup a new scene for this sample. Do not forget to use an orthographic camera and white background for the scene.</i></p>
Step 2	<i>Creating a Light</i>	<p><i>Create a light in the tab “GameObject/2D Light/Light Source”.</i></p>
Step 3	<i>Creating a Sprite</i>	<p><i>Creating a new “GameObject” and attaching a “Sprite Renderer” component to it.</i></p>
Step 4	<i>Attach Light Collider To Sprite GameObject</i>	<p><i>Attaching the “LightCollider2D” component to the already existing “GameObject” with sprite.</i></p>
Step 5	<i>Setup Light Collider</i>	<p><i>Make sure the Mask Type is “Sprite”. So the shape of the “SpriteRenderer” sprite will be masked and visible for the light source.</i></p> <p><i>Also make sure the Collider Type is “Sprite Physics Shape”. In that case you don't need to attach any collider components for the object to cast shadows.</i></p>

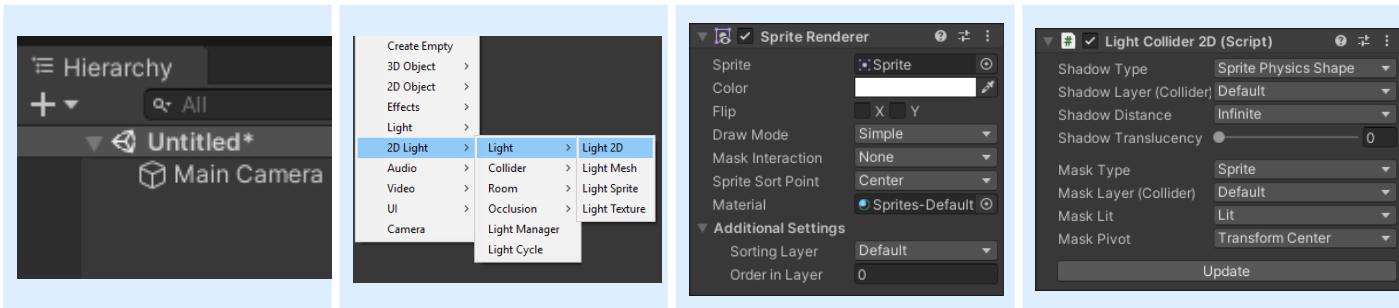
Step 1

Step 2

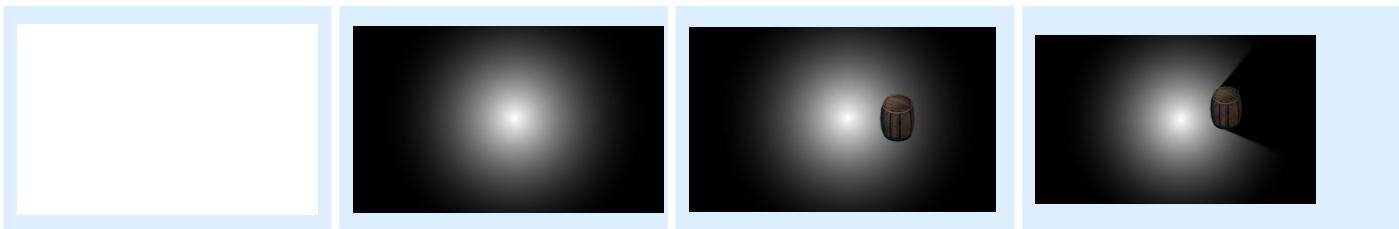
Step 3

Step 4 & 5

Visual Explanation



Game View



Custom Physics Shape

Introduction

The Sprite Editor's Custom Physics Shape allows you to edit a Sprite's Physics Shape. You can use this specific information from the sprite to cast shadows instead of using the Collider component attached.

Unity Documentation: [Link](#)

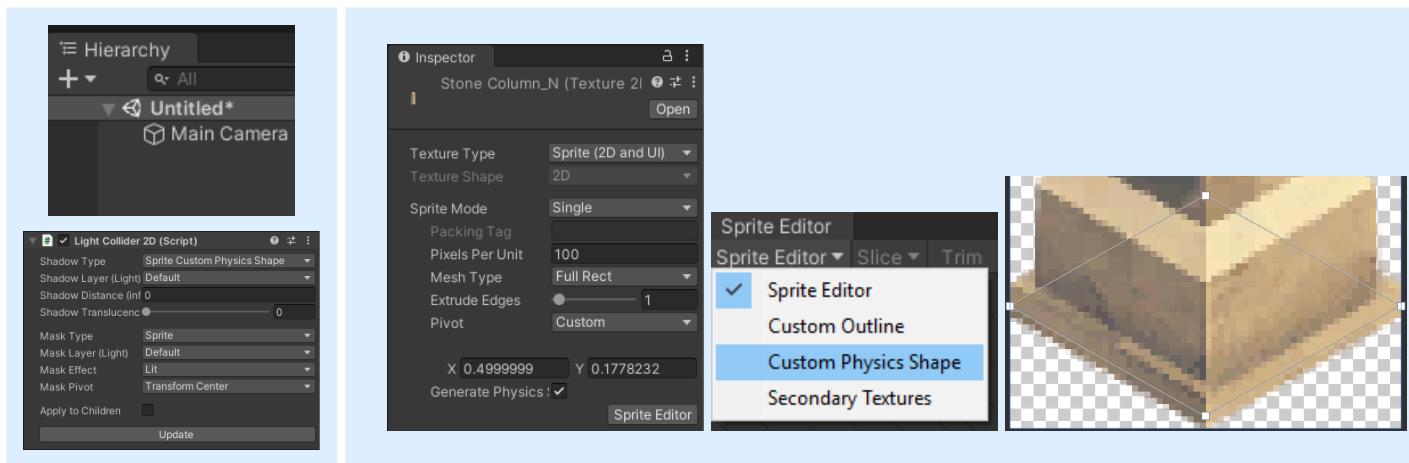
Instructions

Step 1	Setting Up Scene & Camera	Create and Setup a new scene for this sample. Do not forget to use an orthographic camera and white background for the scene.
Step 2	Creating a Light Source	Create a light in the tab “GameObject/2D Light/Light Source”.
Step 3	Creating a Sprite	Creating a new “GameObject” and attaching a “Sprite Renderer” component to it.
Step 4	Attach Light Collider	Attaching the “LightCollider2D” component to the already existing “GameObject” with sprite. Make sure the Mask Type is “Sprite”, so the shape of the “SpriteRenderer” sprite will be masked and not affected by the shadow. Also make sure you are using Collider Type “Sprite Physics Shape”.
Step 5	Setup Custom Physics Shape	Go to the Sprite Import Inspector and press the “Sprite Editor” button. Then switch to Custom Physics Shape mode. There you can add and edit vertices of the shadow casting collider. Do not forget to press “Apply” after finishing to edit the shape.

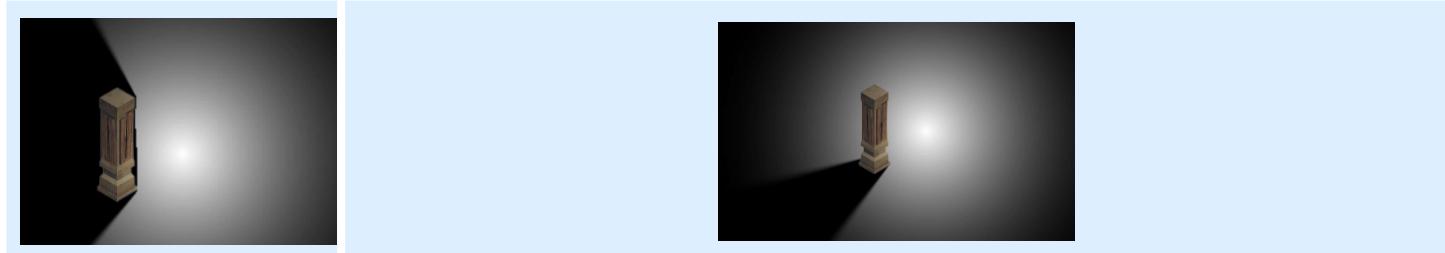
Step 1 & 2 & 3 & 4

Step 5

Visual Explanation



Game View



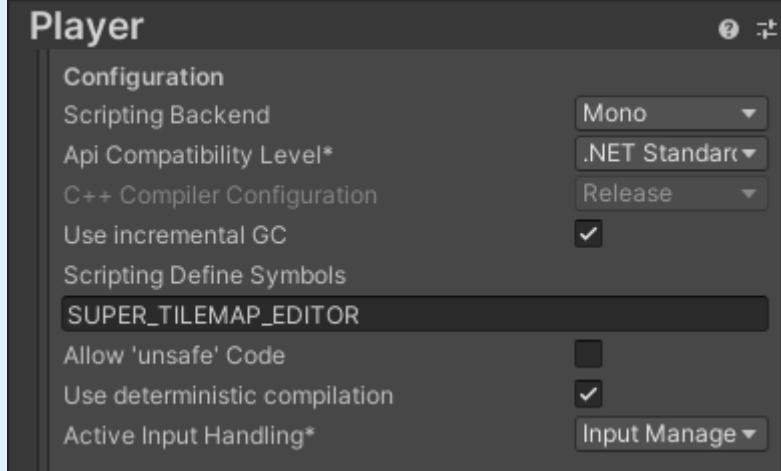
Super Tilemap Editor Support

Introduction

Super Tilemap Editor is a powerful and easy to use tile editor with everything you need to create any game based on tiles. Use it not only to create tilemaps but also as a powerful level editor placing prefabs as if they were tiles.

[Asset Store Link](#)

How To Enable Support?

Step 1	<i>Open Player Settings</i>	
Step 2	<i>In the category “Other Settings” Make sure Scripting Define Symbols include “SUPER_TILEMAP_EDITOR”</i>	
Step 3	<i>Enjoy SuperTilemap Support</i>	:)

Sorting Layer

Introduction

Sorting Layers and Order in Layer are used to determine the render order of the lighting buffer in a scene.

Unity Documentation: [Link](#)

Instructions

Step 1	Setting Up Scene & Camera	Create and Setup a new scene for this sample. Do not forget to use an orthographic camera and white background for the scene.
Step 2	Creating a Light Source	Create a light in the tab “GameObject/2D Light/Light/Light2D”.
Step 3	Creating a Sprite	Creating a new “GameObject” and attaching a “Sprite Renderer” component to it.
Step 4	Create Sorting Layers	Create a new sorting layer in the tab “Edit/Project Settings/Tags and Layers”. Call the first layer “My Sprites” Call the second layer “My Lighting”
Step 5	Assign Sorting Layer To Sprite	Go to the game object with Sprite Renderer, and apply the “My Sprite” layer in the “Sorting Layer” dropdown menu.
Step 6	Assign Sorting Layer To Lighting	Go to the “Lighting Manager 2D” game object. Camera 1->Lightmap 1 field Set Sorting Layer Name to “My Lighting”

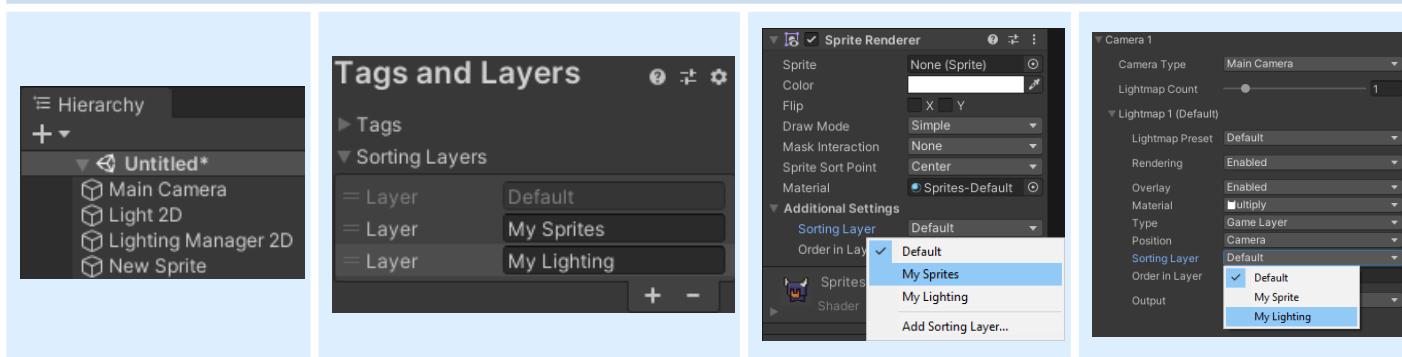
Step 1 & 2 & 3

Step 4

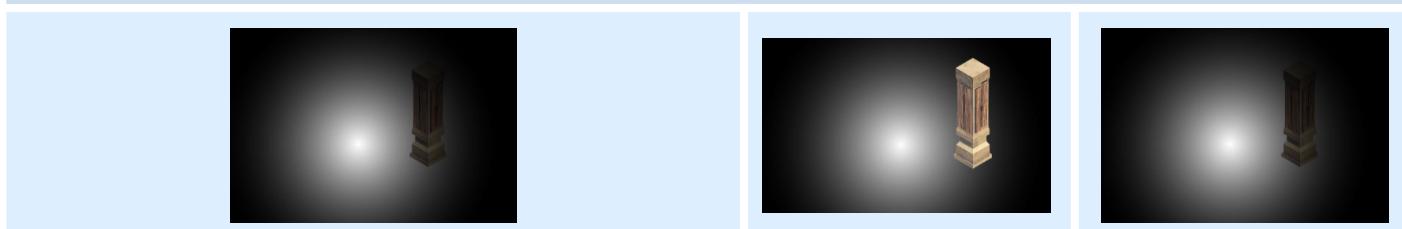
Step 5

Step 6

Visual Explanation



Game View



Normal Maps

Introduction

Lighting 2D has integrated and optimized 2D normal map support.

Instructions

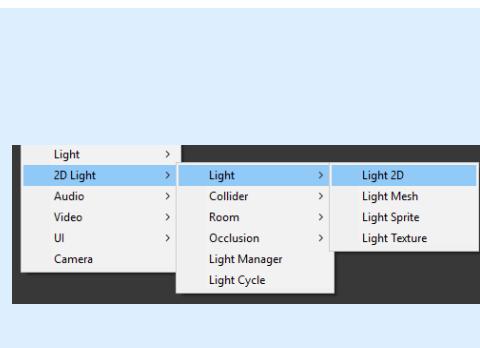
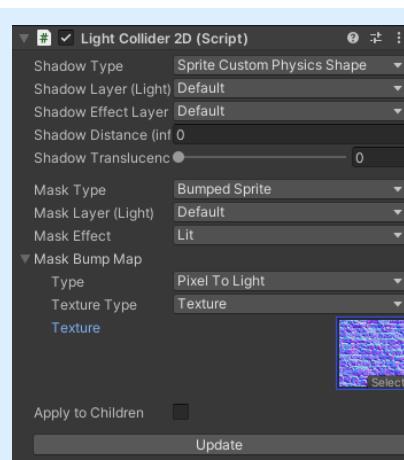
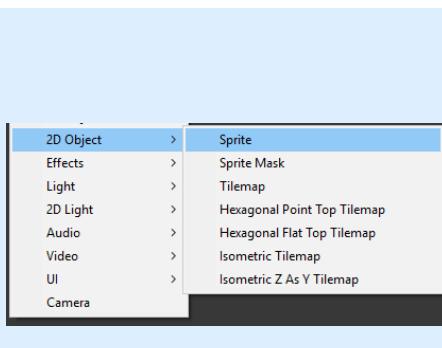
Step 1	2D Sprite	Add a 2D sprite to the scene.
Step 2	Light Collider 2D	Attach the Light Collider 2D Component to the sprite.
Step 3	Light Collider 2D Setup	Mask Type: Bumped Sprite Drag the normal map texture into the “ Mask Bump Map ” field.
Step 4	Light Source 2D	Add Light Source 2D to the scene.

Step 1

Step 2 & 3

Step 4

Visual Explanation



Game View



Scene View

Introduction

Specific editor setup must be used to have a proper scene view of 2D Lighting.

Instructions

Step 1	Scene with Lighting Manager 2D	<i>Create or Load a scene that is using 2D Lighting.</i>
Step 2	Creating Unity Layers	<i>Create new layers in “Edit/Project Settings/Tags and Layers”, layer section. Add “Lighting Internal (Game)” Add “Lighting Internal (Scene)”</i>
Step 3	Disable Unity Layer	<i>In the top-right of the editor, set the “Lighting Internal (Game)” layer invisible for the editor.</i>
Step 4	Set the layer for Lighting Manager 2D	<i>Tools/Light2D/Project Tab/Editor View Set “Game Layer” to “Lighting Internal (Game)” Set “Scene Layer” to “Lighting Internal (Scene)”</i>
Step 5	Scene Camera	<i>In Lighting Manager 2D Set “Scene View=Enabled” for the lightmap under the camera. Make sure Scene View “2D” and “Lighting Icon” are enabled.</i>
Step 6	Camera Settings	<i>In your game camera “culling mask” list: disable “Lighting Internal (Scene)”</i>

Step 2

Step 3

Step 4

Step 5

Step 6

Visual Explanation

Tags and Layers	Layers	2D Light Inspector	Camera Inspector																											
<ul style="list-style-type: none"> ► Tags ► Sorting Layers ▼ Layers <table border="1"> <tr><td>Builtin Layer 0</td><td>Default</td></tr> <tr><td>Builtin Layer 1</td><td>TransparentFX</td></tr> <tr><td>Builtin Layer 2</td><td>Ignore Raycast</td></tr> <tr><td>Builtin Layer 3</td><td></td></tr> <tr><td>Builtin Layer 4</td><td>Water</td></tr> <tr><td>Builtin Layer 5</td><td>UI</td></tr> <tr><td>Builtin Layer 6</td><td></td></tr> <tr><td>Builtin Layer 7</td><td></td></tr> <tr><td>User Layer 8</td><td>Lighting Internal (G)</td></tr> <tr><td>User Layer 9</td><td>Lighting Internal (S)</td></tr> </table>	Builtin Layer 0	Default	Builtin Layer 1	TransparentFX	Builtin Layer 2	Ignore Raycast	Builtin Layer 3		Builtin Layer 4	Water	Builtin Layer 5	UI	Builtin Layer 6		Builtin Layer 7		User Layer 8	Lighting Internal (G)	User Layer 9	Lighting Internal (S)	Layers <ul style="list-style-type: none"> Everything Nothing 0: Default 1: TransparentFX 2: Ignore Raycast 4: Water 5: UI 8: Lighting Internal (G) 9: Lighting Internal (S) 	2D Light Inspector <ul style="list-style-type: none"> Profile: Basic Profile (Profile) Rendering Mode: On Render Color Space: Gamma Manager Instance: Static Manager Internal: Hide In Hierarchy Max Light Size: 10 Editor View: Draw Gizmos, Gizmos Bounds Game Layer (Default): Lighting Internal (Game) Scene Layer (Default): Lighting Internal (Scene) FOW Game Layer: FOW Scene Layer Chunks <table border="1"> <tr><td>0: Default</td></tr> <tr><td>1: TransparentFX</td></tr> <tr><td>2: Ignore Raycast</td></tr> <tr><td>4: Water</td></tr> <tr><td>5: UI</td></tr> <tr><td>8: Lighting Internal (Game)</td></tr> <tr><td>9: Lighting Internal (Scene)</td></tr> </table>	0: Default	1: TransparentFX	2: Ignore Raycast	4: Water	5: UI	8: Lighting Internal (Game)	9: Lighting Internal (Scene)	Camera Inspector <ul style="list-style-type: none"> Clear Flags: Solid Color Background: Mixed... Culling Mask: Nothing, Everything, Default, TransparentFX, Ignore Raycast, Water, UI, Lighting Internal (Game), Lighting Internal (Scene) Projection: Orthographic Size: 10 Clipping Planes: 0 Viewport Rect: 0, 0, 1000, 1000 Depth: 0 Rendering Path: Deferred Target Texture: Main Camera Indirect Culling: Enabled
Builtin Layer 0	Default																													
Builtin Layer 1	TransparentFX																													
Builtin Layer 2	Ignore Raycast																													
Builtin Layer 3																														
Builtin Layer 4	Water																													
Builtin Layer 5	UI																													
Builtin Layer 6																														
Builtin Layer 7																														
User Layer 8	Lighting Internal (G)																													
User Layer 9	Lighting Internal (S)																													
0: Default																														
1: TransparentFX																														
2: Ignore Raycast																														
4: Water																														
5: UI																														
8: Lighting Internal (Game)																														
9: Lighting Internal (Scene)																														
Game View																														



How to use Event Handling

Introduction

Light 2D - invoking events. Light Collider 2D is a receiver.

In this example we will use the “Light Event Listener” component to receive “visibility” of the “Light Collider 2D” object.

Instructions

Step 1	Basic Scene with 2D Light	<i>Let's start with the scene that already includes light 2D</i>
Step 2	Light Event Preset	<i>Event Handling logic is set up in “Light Event Presets”. These settings can be found in the Tools/Light2D window. Let's set the name of the first preset “Basic Events”. The preset can be used in Light 2D components.</i>
Step 3	Adding Receiver Collider	<i>Add Light collider 2D to your event receiver object.</i>
Step 4	Adding Events Listener	<i>Add “Light Event Listener” to the receiver object. This script would store the public “visibility” variable. Add “Light Event Listener GUI” script to the object. This script displays the visibility value in game view and is used only for convenience/debugging.</i>
Step 5	Adding Shadow Collider	<i>Create an “Light Collider 2D” object with default settings.</i>

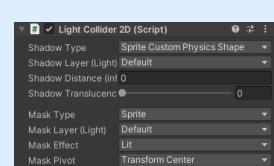
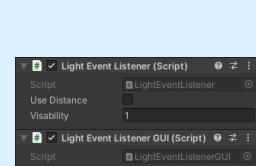
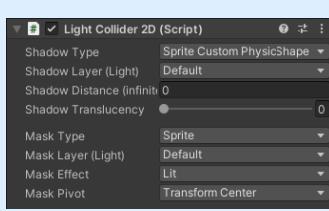
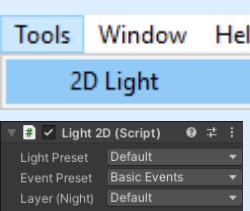
Step 2

Step 3

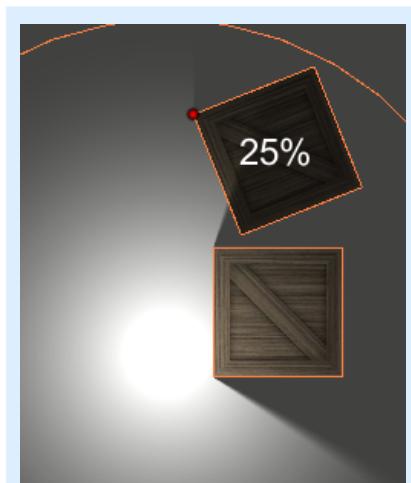
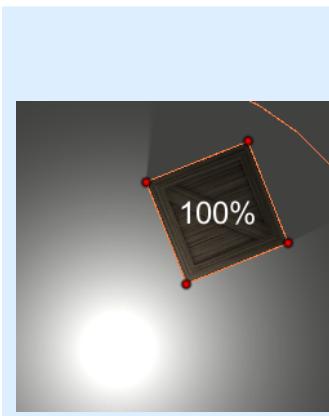
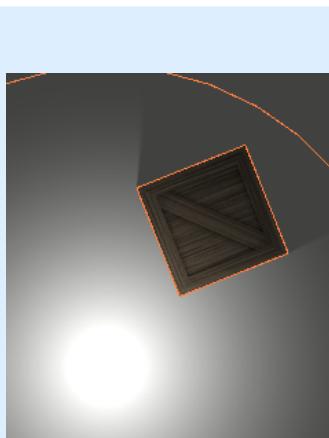
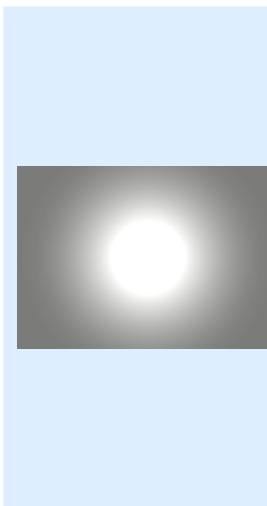
Step 4

Step 5

Visual Explanation



Game View



How to use Day Lighting

Introduction

Day Lighting is a separate system to work with specific lighting effects to generate shadows created by directional light.

Instructions

Step 1	Setting Up Scene & Camera	Create and Setup a new scene for this sample. Do not forget to use an orthographic camera and white background for the scene.
Step 2	Setting Up Darkness Color	Open “Tools/Lighting 2D” window, Set darkness color to white and it’s alpha to 0.
Step 3	Adding a Sprite	Add 2D sprite into the scene. Apply a sprite image to it.
Step 4	Adding Daylight Collider to Sprite	Add the “ Day Light Collider 2D ” component to the sprite game object.
Step 5	Setting Up Daylight Collider	Collision type: Sprite Custom Physics Shape Mask Type : Sprite

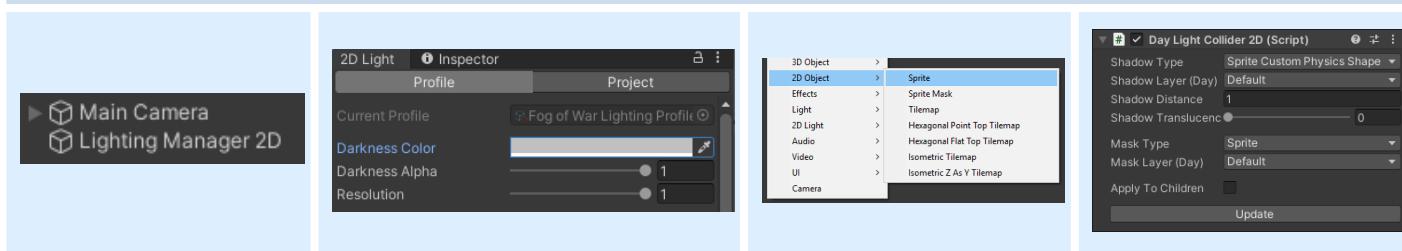
Step 1

Step 2

Step 3

Step 4 & 5

Visual Explanation



Game View



If there are any questions/feature requests or need help, feel free to contact!

I will answer the questions before you decide to use the asset.

Ask or find more information in [Discord!](#)

Documentation is in development, there will be updates!

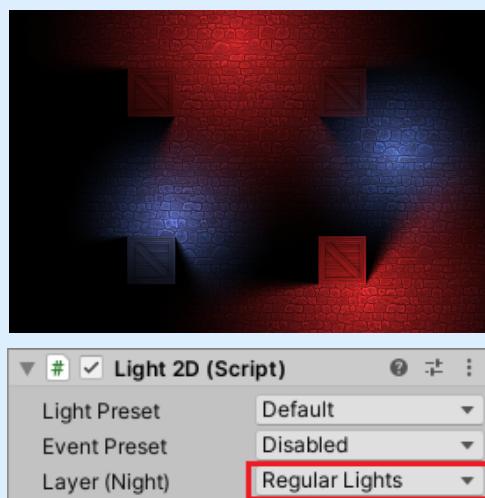
Fog Of War (Simple) Part 1

Step 1 Scene		<p><i>Scene includes</i></p> <p><i>Orthographic Camera</i> <i>Floor (Sprite Renderer)</i> <i>Boxes (Sprite Renderer)</i></p>
Step 2 Night Layers		<p><i>Search for "Light Layers" In Tools/Light 2D, Layer section.</i></p> <p><i>For this setup you will need two different Light layers.</i></p> <p>1 - Regular Lights (behind the fog of war)</p> <p>2 - Fog of War Lights</p> <p><i>Before adding the names, increase the layer count.</i></p>
Step 3 Buffer Presets		<p><i>Search for the "Buffer Presets" section In Tools/Light 2D.</i></p> <p><i>We need to initialise 2 buffer presets.</i></p> <p>1 - Regular Light System for the Regular Lights Light Layer.</p> <p>2 - Fog Of War System for the Fog Of War Lights Light Layer.</p> <p><i>Preferably set "Day Layers" count to 0 for both of the presets.</i></p>

Fog Of War (Simple) Part 2

Step 4

Regular Lights & Colliders



Attach "Light Collider 2D" components for the boxes.

Light Collide 2D:

Shadow Type = Sprite Physics Shape
Mask Type = Sprite

Create a few "Light 2D" game objects in the scene. Make sure "Layer (Night)" is set to **Regular Lights**.

Step 5

Lighting Manager



In step 3 we set up **Buffer Presets**.

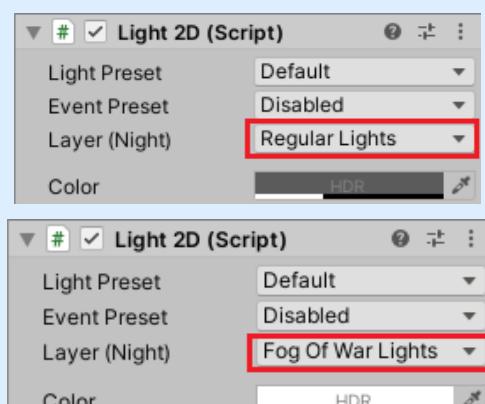
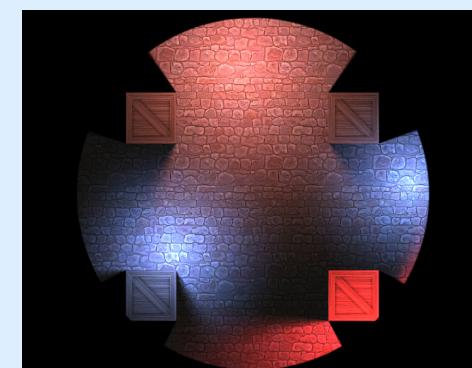
Buffer Presets are used in **Lighting Manager** to output Lighting into the camera with light logic you have included in the Presets.

Beforehand, set the camera count to 2. Camera 1 should use the **Regular Light System** Preset. Camera 2 - **Fog Of War System**.

After this step is complete, the game view should be **black** because there is **no fog of war light** in the scene yet.

Step 6

Fog of War Lights



To create a Fog of War light source, you will need to create two separate lights.

1 - Light up a **Regular Lights System**

2 - Light up **Fog of War System**

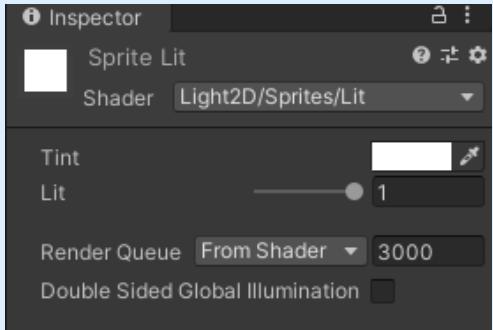
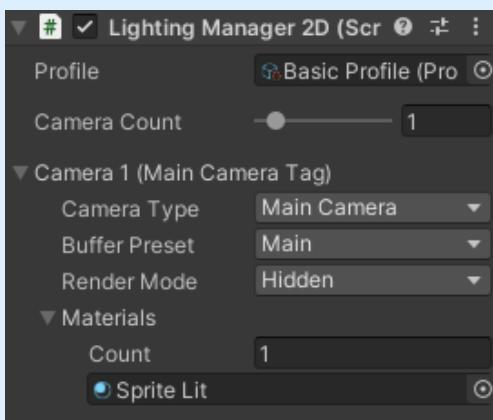
You can set these layers in the Light 2D component "Layer (Night)" field.

Use the "**gfx_fogofwar**" sprite for the texture of the light.

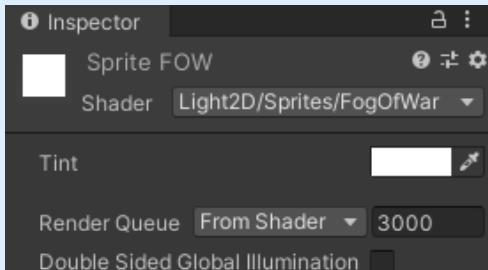
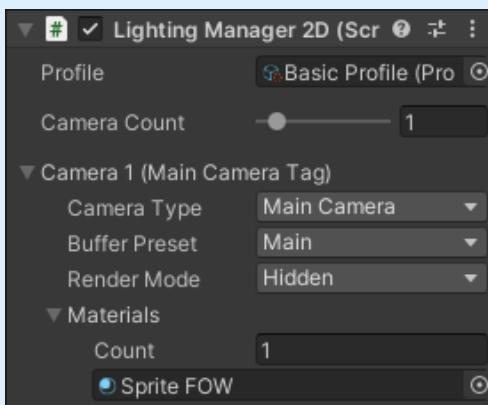
Ideally, the light should be white.

Both presets are mixed with Multiply blends.

Material Basics

Step 1		<p>Scene includes</p> <p><i>Orthographic Camera Building (Sprite Renderer) Light 2D gameobject</i></p>
Step 2		<p><i>Create a new material with “Light 2D/Sprites/LitDefault” shader.</i></p>
Step 3		<p><i>In Lighting Manager 2D:</i></p> <p><i>set “Render Mode” to “Hidden”</i></p> <p><i>Add Sprite Lit material to the materials list in Lighting Manager 2D.</i></p>
Step 4		<p><i>Attach Sprite Lit material to the building Sprite Renderer in the scene.</i></p>

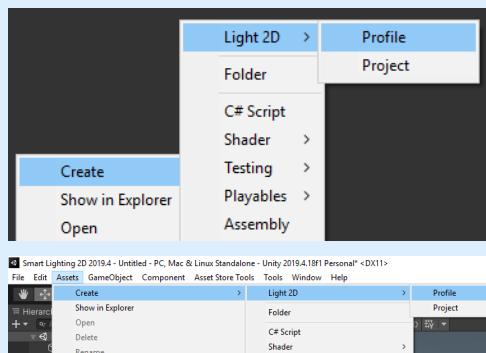
Fog of War Material

Step 1	Scene		<p><i>Scene includes</i></p> <p><i>Orthographic Camera</i> <i>Floor (Sprite Renderer)</i> <i>Building (Sprite Renderer)</i> <i>Light 2D gameobject</i></p>
Step 2	Sprite Material		<p><i>Create a new material with “Light 2D/Sprites/FogOfWar” shader.</i></p>
Step 3	Manager		<p><i>In Lighting Manager 2D:</i></p> <p><i>set “Render Mode” to “Hidden”</i></p> <p><i>Add Sprite FOW material to the materials list in Lighting Manager 2D.</i></p>
Step 4	SpriteRenderer with Material		<p><i>Attach Sprite FOW material to the building Sprite Renderer in the scene.</i></p>

Lighting Profile (Basics)

Step 1

Create Profile

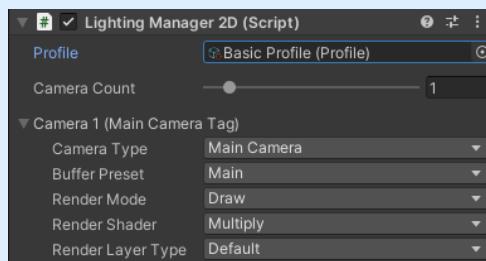


Create a *Lighting Profile* asset file

Assets/Create/Light 2D/Profile

Step 2

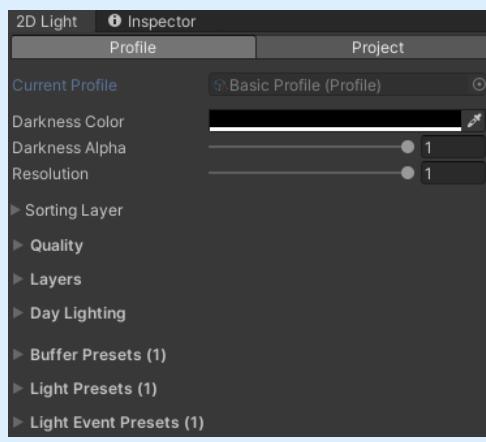
Select Profile



Drag the new *Lighting Profile* into the *Lighting Manager* in the scene.

Step 3

Use Profile



Under the "Tools" tab you can open the *Lighting 2D* setting window.

All properties are modified within the **Lighting Profile** that is assigned to the **Lighting Manager**.

Note:

Changes applied in runtime won't be saved.

Soft Shadow

Step 1 Scene		<i>Scene includes</i> <i>Orthographic Camera</i>
Step 2 Sprite Material		<i>Create a new material</i>
Step 3 Manager		<i>In Lighting Manager 2D:</i>
Step 4 SpriteRenderer with Material		<i>Attach Sprite FOW material to the building Sprite Renderer in the scene.</i>

Unity Lighting 2D Components

Components

LightingManager2D	<p>Previously used for lighting settings, now all settings are moved to “Tools/Lighting 2D” via the <i>Settings Profile</i> (scriptable object). Now used as root for all generated Lighting 2D effects</p>
Light2D	<p>Versatile light emitting source, this component emits light. Can use custom texture to set up a unique light emitting look.</p>
LightCollider2D	<p>Versatile light collider component. Can be used to set up shadow casting for sprites & solo collider components.</p>
LightTilemapCollider2D	<p>Tilemap Light Collider component can be used to set up shadow casting for standard unity Tilemap components and the Super Tilemap Editor system.</p>
DayLightCollider2D	
DayLightTilemapCollider2D	
LightSprite2D	<p>The Lightning Sprite Renderer component can draw images with different blending modes straight into the light buffer to light up objects, particles and other scene entities. This component is very efficient to make lights without shadow casting. (Very Mobile Friendly)</p>
LightTexture2D	
LightParticleSystem2D	
LightRoom2D	<p>This component can be used to darken the area in the daylight. For example you might need to have a dark room in brightly lit scenes with daylight shadows.</p>
LightTilemapRoom2D	<p>This component can be used to darken the area in the daylight. Can be used to mask a tilemap to be affected by lights.</p>
LightOcclusion2D	
LightTilemapOcclusion2D	
FogOfWarSprite	
FogOfWarTilemap	

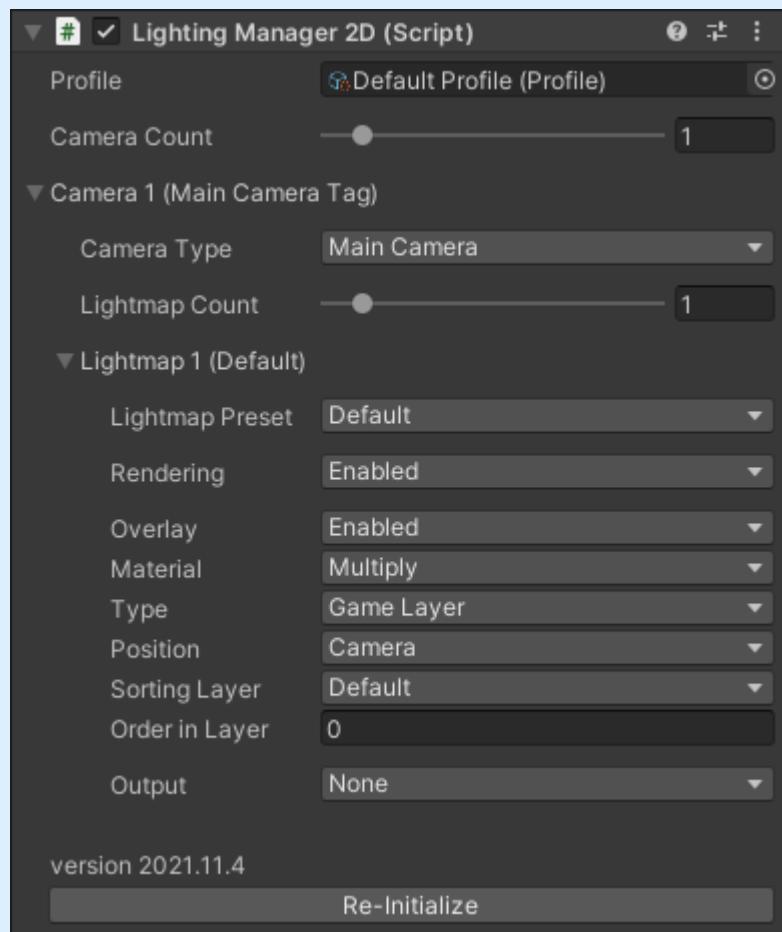
Bonus Components

ColliderLineRenderer2D	<i>Creates an outline for Collider components with selected color</i>
Mesh2D	<i>Creates a mesh from Collider components and attaches it to the mesh renderer. Mostly used to display basic demo scenes without any images.</i>

Component Reference

LightingManager2D

appearance



description

*Only one Lighting Manager 2D is allowed per scene.
This component is automatically generated on the first 2D lighting API call.*

information

Profile

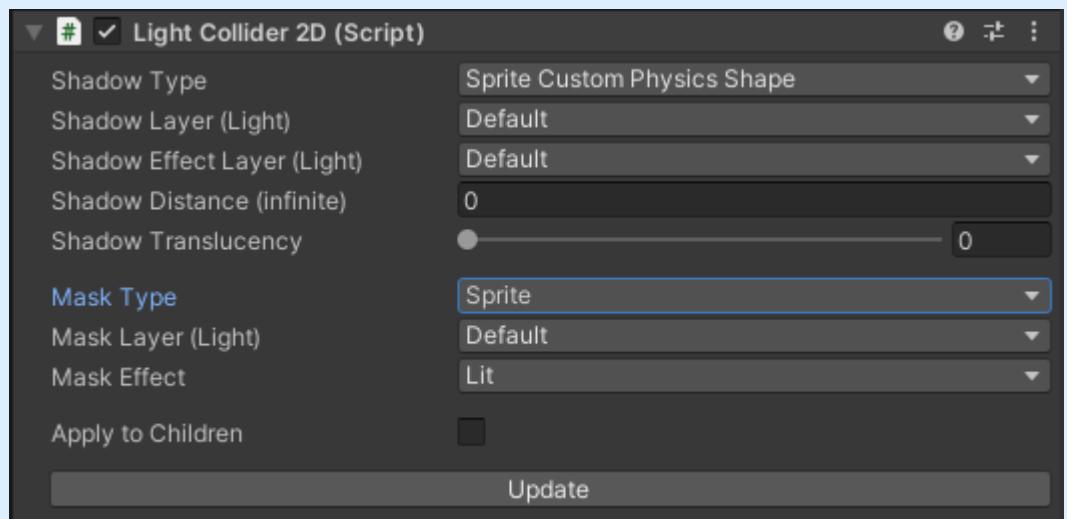
Cameras

Light2D

appearance			
description	<i>This component emits light. Can use custom textures to set up a unique light look.</i>		
information	Color	24 Bit Color	<i>The color of light. The darker the color, the less visible it will appear. Black color is not visible at all.</i>
	Alpha	Float [0 - 1]	<i>Transparency of light. The higher the alpha value, the more visible light appears to be.</i>
	Size	Int [0 - Unidentified]	<i>The size of light, keep in mind that increasing the size of light does not automatically increase its "buffer size", very large lights require larger buffer size, otherwise pixelated artifacts appear.</i>
	Buffer Size	Enumerator	<i>The resolution of the light buffer. Larger buffer leads to more crispy shadow details, however it costs more performance.</i>
	Light Sprite	Default	<i>Default texture which is being applied to the light.</i>
		Custom	<i>Enables a custom sprite texture to be used for the light.</i>
	Sprite	Sprite	<i>After enabling LightSprite/Custom, you are able to select your custom sprite texture for the light.</i>
	Apply Rotation	Boolean	<i>Enable object transform rotation for the light</i>
	Apply Light Inside Collider	Boolean	<i>By default, once light appears in the collider, no collisions are generated with that particular object. Once this option is enabled, you can put light inside the objects and light will still collide with their walls.</i>

LightCollider2D

appearance

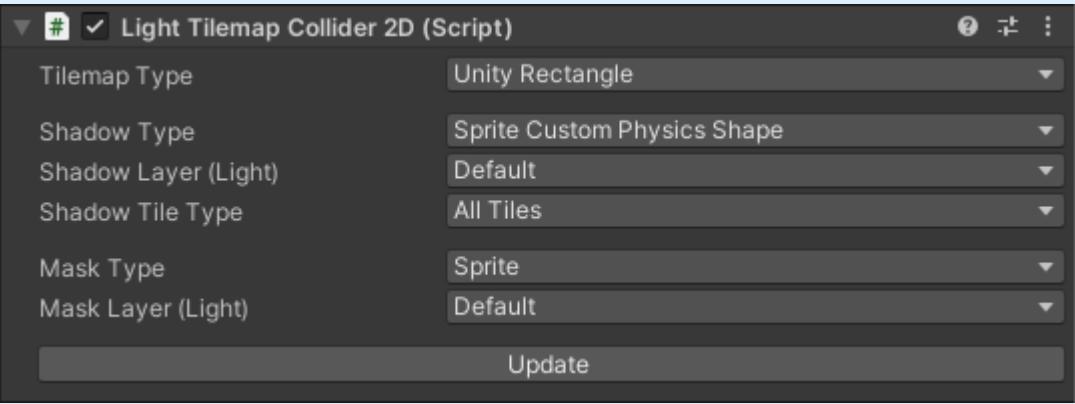


description

Versatile light collider component. Can be used to set up shadow casting for sprites & solo collider components.

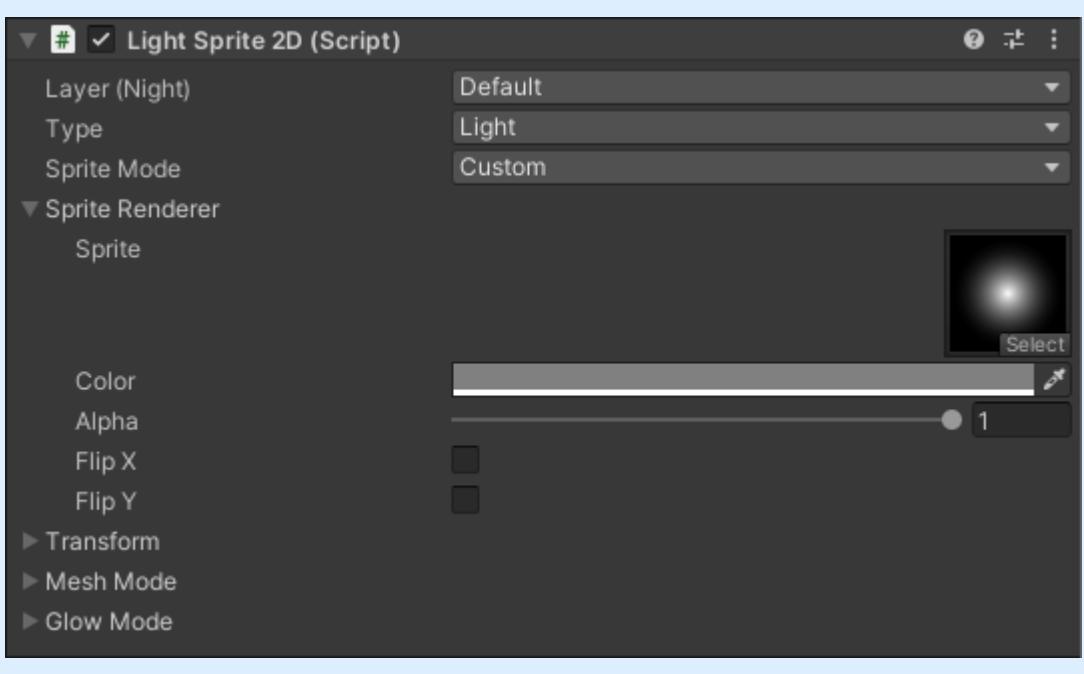
information	Collision Type	None	<i>Disables all shadow casting for this object.</i>
		Collider	<i>Use 2D collider for geometry of shadow casting. (Box2D, Circle2D, CapsuleCollider2D, Polygon2D, Edge2D)</i>
		Sprite Custom Physics Shape	<i>Use Sprite's custom physics shape which can be accessed with the Unity Sprite Editor.</i>
		Mesh	<i>Uses Mesh Filter Mesh to cast shadows.</i>
	Collision Layer	Int [0 - 31]	<i>Lighting layer of the object, this layer should be included in the lighting source layer list.</i>
	Mask Type	None	<i>Disables all masking for this object.</i>
		Sprite	<i>Uses a sprite from the sprite renderer of this object for the mask.</i>
		Collider	<i>Uses 2D Collider geometry for the mask.</i>
		Sprite Custom Physics Shape	<i>Use Sprite Custom Physics Shape geometry to mask the object.</i>
	Mask Layer	Int [0 - 31]	<i>Lighting layer of the object, this layer should be included in the lighting source layer list.</i>
	Update	Editor Button	<i>Press this object to re-initialize the geometry of the collider. This is a workaround for performance reasons because geometry is not updated in real time. For example, this should be triggered after changing polygon collider geometry (editor run time). Keep in mind that after going into play mode everything is applied automatically.;</i>

LightTilemapCollider2D

appearance			
description	<p><i>Tilemap Light Collider component can be used to setup shadow casting for standard unity Tilemap component and Super Tilemap Editor system.</i></p>		
information	Tilemap Type	Unity Engine Tilemap	<i>Use a standard 2D Tilemap for shadow casting. No additional collider components are needed for</i>

		<i>this. Lighting system will take sprites used in the tileset and apply their selected properties for collisions.</i>
	Super Tilemap Editor	<i>Use the Super Tilemap Editor for the shadow casting.</i>
Collision Type	None	<i>Disable shadow casting for this tilemap object.</i>
	Rectangle	<i>All tiles are treated like rectangles</i>
	Sprite Custom Physics Shape	<i>Try using the sprite's custom physics shape for shadow casting.</i>
Collision Layer	Layer Enumerator [0 - 31]	<i>Lighting layer of the object, this layer should be included in the lighting source layer list.</i>
Mask Type	None	<i>Disable masking for this object.</i>
	Sprite	<i>Uses tile sprites for masking.</i>
	Rectangle	<i>Use a rectangle shape for masking this tilemap.</i>
	Sprite Custom Physics Shape	<i>Use tile sprite custom physics shapes.</i>
Mask Layer	Layer Enumerator [0 - 31]	<i>Lighting layer of the object, this layer should be included in the lighting source layer list.</i>
Day Height & Size	Enumerator Float [0 - Undenified]	<i>Enable sun's shadow casting in the daylighting system.</i>
Batch Sprite Masking	Boolean	<i>This is optimization. Enable this option when the whole tile palette consists of the same texture file. This should improve batch calls from the lighting system.</i>
Update Collisions	Editor Button	<i>Press this object to re-initialize the geometry of the collider. This is a workaround for performance reasons because geometry is not updated in real time. For example, this should be triggered after changing polygon collider geometry (editor run time). Keep in mind that after going into play mode everything is applied automatically.;</i>

LightSprite2D

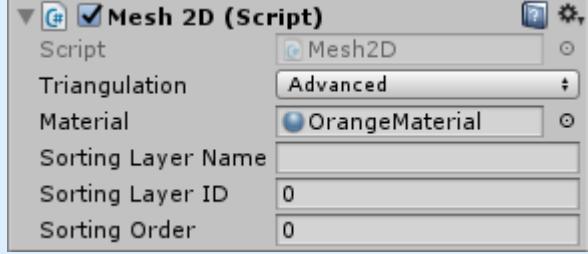
appearance																																														
description	<p>The Light Sprite 2D component can draw images with different blending modes straight into the light buffer to light up objects, particles and other scene entities. This component is very efficient to make lights without shadow casting. (Very Mobile Friendly)</p>																																													
information	<table border="1"> <tbody> <tr> <td data-bbox="327 1001 637 1057"></td><td data-bbox="637 1001 860 1057">Particle</td><td data-bbox="860 1001 1443 1057">Additive shader effect for this component.</td></tr> <tr> <td data-bbox="327 1057 637 1226" style="text-align: center;">Type</td><td data-bbox="637 1057 860 1226">White Mask</td><td data-bbox="860 1057 1443 1226">Applying a white mask for this object, the object is always fully visible and over the lighting buffer. However, this can be also achieved using sorting order which is higher than lighting buffer sorting order.</td></tr> <tr> <td data-bbox="327 1226 637 1282"></td><td data-bbox="637 1226 860 1282">Black Mask</td><td data-bbox="860 1226 1443 1282">Apply a black mask for the object, and everything underneath is completely not visible.</td></tr> <tr> <td data-bbox="327 1282 637 1450" style="text-align: center;">Sprite Mode</td><td data-bbox="637 1282 860 1361">Custom</td><td data-bbox="860 1282 1443 1361">Select your own sprite for this component.</td></tr> <tr> <td data-bbox="327 1361 637 1450"></td><td data-bbox="637 1361 860 1450">Sprite Renderer</td><td data-bbox="860 1361 1443 1450">Synchronize a sprite variable with the Sprite Renderer component attached to the same game object.</td></tr> <tr> <td data-bbox="327 1450 637 1574" style="text-align: center;">Color</td><td data-bbox="637 1450 860 1574">24 Bit Color</td><td data-bbox="860 1450 1443 1574">Color of the effect. This does not take any effect when using white mask or black mask.</td></tr> <tr> <td data-bbox="327 1574 637 1697" style="text-align: center;">Alpha</td><td data-bbox="637 1574 860 1697">Float [0 - 1]</td><td data-bbox="860 1574 1443 1697">Transparency of this effect. This does not take any effect when using white mask or black mask.</td></tr> <tr> <td data-bbox="327 1697 637 1754" style="text-align: center;">Flip X</td><td data-bbox="637 1697 860 1754">Boolean</td><td data-bbox="860 1697 1443 1754">Flips the sprite on the X axis.</td></tr> <tr> <td data-bbox="327 1754 637 1810" style="text-align: center;">Flip Y</td><td data-bbox="637 1754 860 1810">Boolean</td><td data-bbox="860 1754 1443 1810">Flips the sprite on the Y axis.</td></tr> <tr> <td data-bbox="327 1810 637 1866" style="text-align: center;">Offset Position</td><td data-bbox="637 1810 860 1866">Vector 2</td><td data-bbox="860 1810 1443 1866">Offset sprite's position.</td></tr> <tr> <td data-bbox="327 1866 637 1922" style="text-align: center;">Offset Scale</td><td data-bbox="637 1866 860 1922">Vector 2</td><td data-bbox="860 1866 1443 1922">Additional scale offset for the sprite.</td></tr> <tr> <td data-bbox="327 1922 637 1978" style="text-align: center;">Offset Rotation</td><td data-bbox="637 1922 860 1978">Degrees</td><td data-bbox="860 1922 1443 1978">Additional rotational offset for the sprite.</td></tr> <tr> <td data-bbox="327 1978 637 2034" style="text-align: center;">Blur Size</td><td data-bbox="637 1978 860 2034">Int [1 - 10]</td><td data-bbox="860 1978 1443 2034">When blur is enabled, you may choose its strength.</td></tr> <tr> <td data-bbox="327 2034 637 2091" style="text-align: center;">Blur Iterations</td><td data-bbox="637 2034 860 2091">Int [1 - 10]</td><td data-bbox="860 2034 1443 2091">The time blur algorithm is being applied.</td></tr> <tr> <td data-bbox="327 2091 637 2221" style="text-align: center;">Apply Blur</td><td data-bbox="637 2091 860 2221">Boolean</td><td data-bbox="860 2091 1443 2221">For this option to be used, you need to enable the sprite write/read setting.</td></tr> </tbody> </table>		Particle	Additive shader effect for this component.	Type	White Mask	Applying a white mask for this object, the object is always fully visible and over the lighting buffer. However, this can be also achieved using sorting order which is higher than lighting buffer sorting order.		Black Mask	Apply a black mask for the object, and everything underneath is completely not visible.	Sprite Mode	Custom	Select your own sprite for this component.		Sprite Renderer	Synchronize a sprite variable with the Sprite Renderer component attached to the same game object.	Color	24 Bit Color	Color of the effect. This does not take any effect when using white mask or black mask.	Alpha	Float [0 - 1]	Transparency of this effect. This does not take any effect when using white mask or black mask.	Flip X	Boolean	Flips the sprite on the X axis.	Flip Y	Boolean	Flips the sprite on the Y axis.	Offset Position	Vector 2	Offset sprite's position.	Offset Scale	Vector 2	Additional scale offset for the sprite.	Offset Rotation	Degrees	Additional rotational offset for the sprite.	Blur Size	Int [1 - 10]	When blur is enabled, you may choose its strength.	Blur Iterations	Int [1 - 10]	The time blur algorithm is being applied.	Apply Blur	Boolean	For this option to be used, you need to enable the sprite write/read setting.
	Particle	Additive shader effect for this component.																																												
Type	White Mask	Applying a white mask for this object, the object is always fully visible and over the lighting buffer. However, this can be also achieved using sorting order which is higher than lighting buffer sorting order.																																												
	Black Mask	Apply a black mask for the object, and everything underneath is completely not visible.																																												
Sprite Mode	Custom	Select your own sprite for this component.																																												
	Sprite Renderer	Synchronize a sprite variable with the Sprite Renderer component attached to the same game object.																																												
Color	24 Bit Color	Color of the effect. This does not take any effect when using white mask or black mask.																																												
Alpha	Float [0 - 1]	Transparency of this effect. This does not take any effect when using white mask or black mask.																																												
Flip X	Boolean	Flips the sprite on the X axis.																																												
Flip Y	Boolean	Flips the sprite on the Y axis.																																												
Offset Position	Vector 2	Offset sprite's position.																																												
Offset Scale	Vector 2	Additional scale offset for the sprite.																																												
Offset Rotation	Degrees	Additional rotational offset for the sprite.																																												
Blur Size	Int [1 - 10]	When blur is enabled, you may choose its strength.																																												
Blur Iterations	Int [1 - 10]	The time blur algorithm is being applied.																																												
Apply Blur	Boolean	For this option to be used, you need to enable the sprite write/read setting.																																												

	<i>Apply Additive</i>	Boolean	<i>Apply additive shader for the lighting sprite renderer.</i>
	<i>Apply Transform Rotation</i>	Boolean	<i>Enable transform offset for the sprite.</i>

LightRoom2D

appearance			
description	<i>This component can be used to darken the area in the daylight. For example you might need to have a dark room in a brightly lit scene with daylighting shadows.</i>		
information	<i>Color</i>	24 Bit Color	<i>The color of the room</i>

Bonus Component Reference

Mesh2D		
appearance		
description	<i>Creates a mesh from already attached Collider2D</i>	
information	Triangulation	<i>Triangulation method to be used when generating 2D mesh.</i>
	Material	<i>The material of the object.</i>

Tools/Light2D

appearance			
description	<p><i>Only one Lighting Manager 2D is allowed per scene.</i> <i>This component is automatically generated on the first 2D lighting API call.</i></p>		
information	Rendering Mode	On Render	<i>An Additional Sorting Order option will appear for this setting. You can set a specific sorting order for the lighting buffer.</i>
		Pre Render	<i>Game objects with sorting order will appear above the lighting buffer. Game objects with lower sorting order ID will appear below the lighting buffer.</i>
		Post Render	<i>Not recommended to use. Lighting buffer is drawn on a post process rendering loop. It seems to have many issues if you want to use post-processing effects.</i>
	Darkness Color	24 Bit Color	<i>The darker the color, the darker the scene will be drawn.</i> <i>For day lighting effects you should set up a dark color that is very bright.</i> <i>Can be used to get "tint" which could represent dusk or dawn.</i>
	Shadow Darkness	Float [0 - 1]	<i>The darkness of day lighting shadows.</i> <i>0 - not visible, 1 - opaque.</i>
	Sun Rotation	Radians	<i>Sun rotation affects all lighting colliders with day shadows.</i> <i>This variable can be manipulated in real time to achieve a day & night cycle.</i>
	Draw Additive Lights	Boolean	<i>Enable additive lights drawing. When disabled, it will skip all checks for additive light drawing. If you are not using this feature in any of the lights, it is suggested to disable this.</i>
	Draw Rooms	Boolean	<i>Enable rooms feature, mostly used for scenes with day lighting effects.</i>
	Draw Occlusion	Boolean	<i>Enable occlusion drawing, currently not recommended to use, this feature is going to be improved in 1.0.6 and 1.0.7 together with day lighting.</i>
	Draw Day Shadows	Boolean	<i>Enables day shadow casting for lighting colliders. It is recommended to disable this if you are not using day lighting effects.</i>
	Draw Main Buffer	Boolean	<i>When disabled, it hides the lighting buffer from the main camera. This setting is similar to "Disable Engine" except all lighting calculations will still be performed, but not drawn.</i>
	Draw Scene Buffer	Boolean	<i>When enabled, lighting sources can be seen in scene view.</i>
	Lighting Resolution	Float [0.125 - 1]	<i>The resolution of Lighting buffer. The higher resolution is, the more detailed the lighting is, however it also impacts the performance. For very high resolution, it's recommended to reduce lighting resolution because additional crispy details are not very noticeable in higher resolutions than 720x1280. It is recommended to set this setting 0.5 for mobile device build.</i>
	Fixed Light Buffer & Size	Boolean Enumerator	<i>This option enables a better poll system for light source systems. Improves performance and is recommended for mobile build. When enabled, custom light buffer size is not available, all lights will have the same texture size.</i>
	Batch Collider Mask	Bool	<i>When enabled, lighting will use the same texture as a source for sprite masks. It won't work unless all sprites are included in the same texture. Use this to improve</i>

			<p><i>performance.</i></p>
	Debug	Bool	<p><i>When enabled, additional lighting information will be displayed in the game view. This helps to benchmark lighting performance. Mostly used by Smart Lighting 2D Developers.</i></p>