

Smart Lighting 2D

Updated: 2021 / 01 / 18

Current Asset Version: [2021.2.0](#)

Latest Documentation: [Link](#)

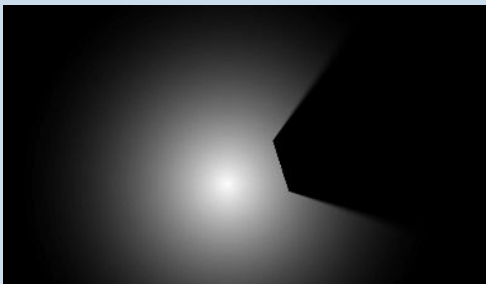
Roadmap: [Link](#)

Forum Discussion: [Link](#)

Mail: simonas@kuzmickas.lt

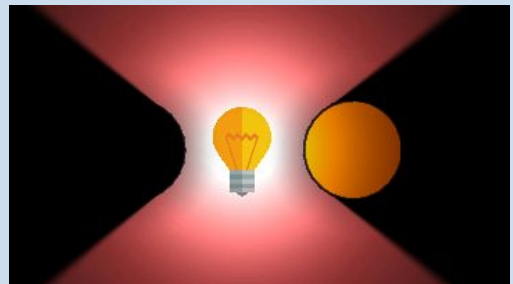
Discord Support: [Link](#) 

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?



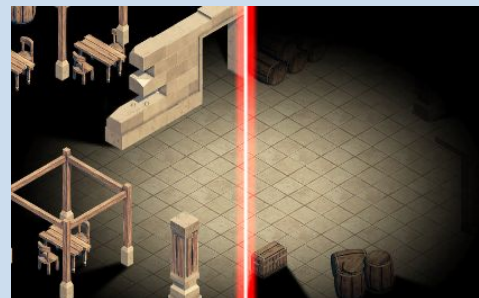
How to use Day Lighting?

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



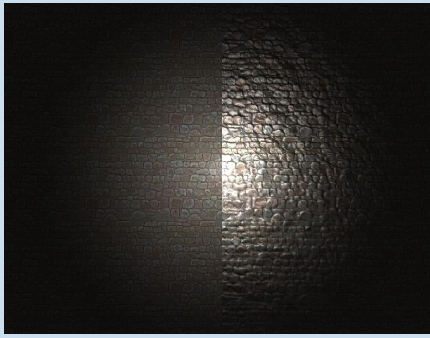
Super Tilemap Editor Support

Having issues with Super Tilemap Support?



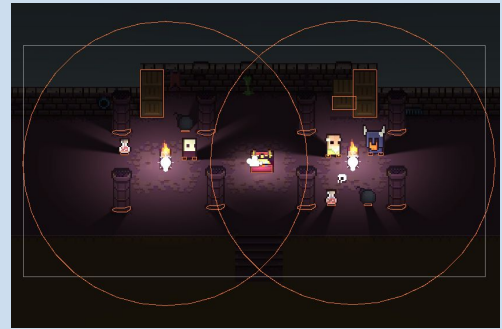
Sorting Layer

Having issues with sorting layer & order?



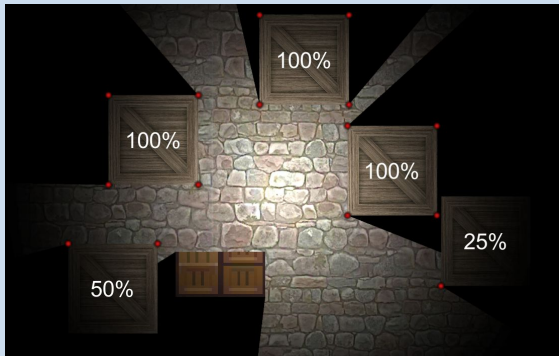
Normal Maps

How to normal maps with 2D Lighting



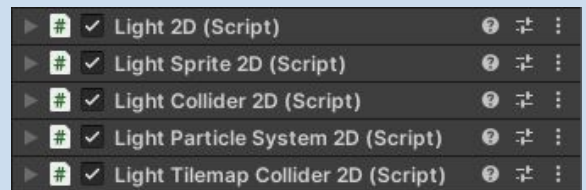
Scene View

How to setup use of 2D Lighting in Scene View



How to use Event Handling?

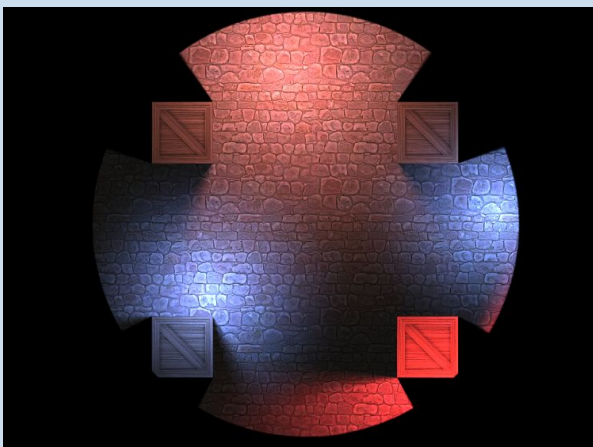
How to setup basic events



Component Reference

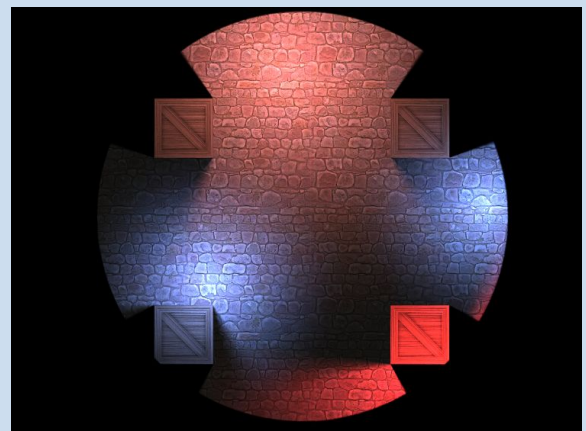
A complete description of each component of the 2D Lighting System.

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

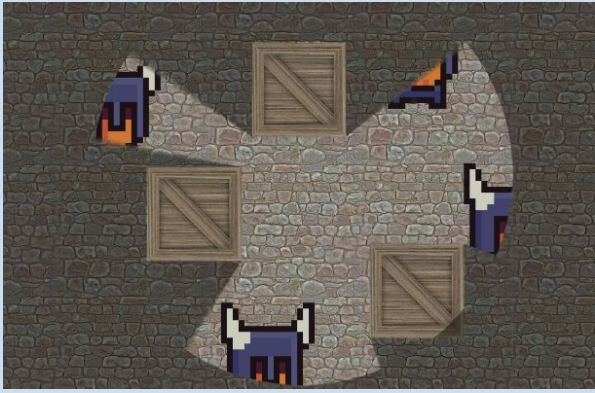
If there are any questions/feature requests or need help, feel free to contact!

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

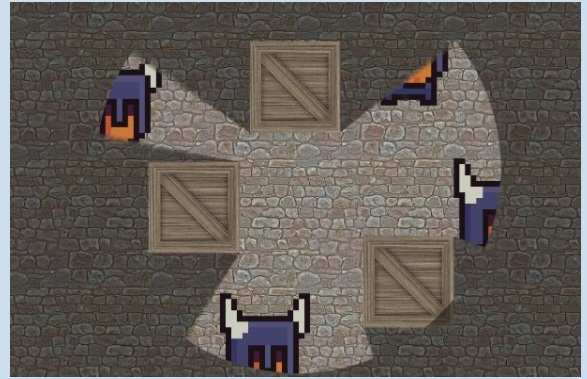
Ask or find more information in [Discord!](#)

Documentation is the development, there will be updates!

Fog of War (Render Texture)



Fog of War (RT) - Part 1
Scene setup



Fog of War (RT) - Part 2
Fog of War & lighting settings

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

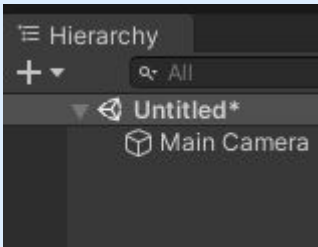
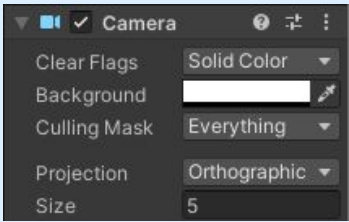
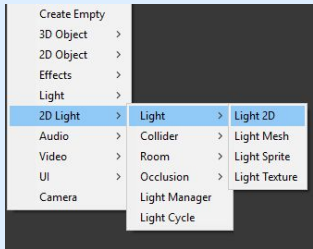
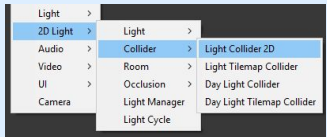
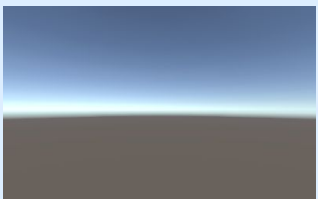
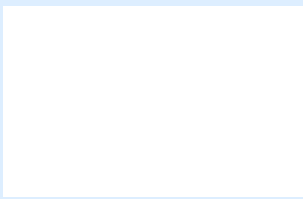


I will answer the questions before you decide to use the asset.

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

Documentation is the development, there will be updates!

How to Start?

Instructions		
Step 1	Creating a new Scene	Create a new scene
Step 2	Camera Setup	Make sure to have orthographic mode set for the camera. Set the scene background to be quite bright. If you'll have black background, your default setup lights & shadows won't be visible.
Step 3	Creating a Light	Create a light in the tab "GameObject/2D Light/Light" .
Step 4	Creating Light Manager	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.
Step 5	Creating Light Collider	Create a collider in tab "GameObject/2D Light/Light Collider" After adding this object into the scene, you should be able to see shadows from the collider. The shadow should be visible, the collider object should be black. For making collider visuals visible read more in "What is Masking" .

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	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	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 To Sprite	Attaching “LightCollider2D” component to the already existing “GameObject” with sprite.
Step 5	Setup Light Collider	Make sure Mask Type is “Sprite” . So the shape of the “SpriteRenderer” sprite will be masked and visible for the light source. Also make sure Collider Type is “Sprite Physics Shape” . In that case you don't need to attach any collider components for the object to cast shadows.

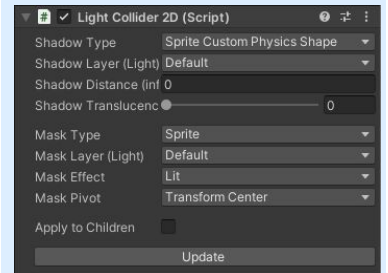
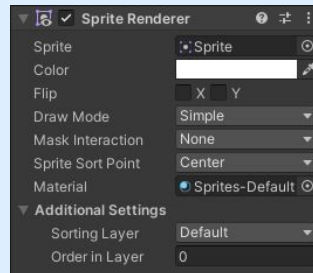
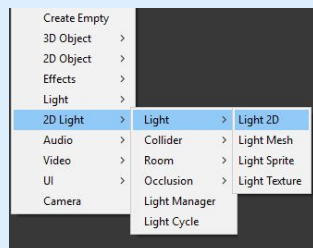
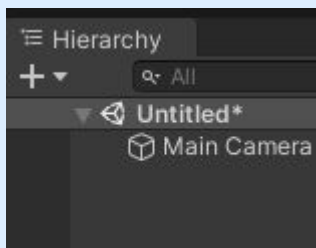
Step 1

Step 2

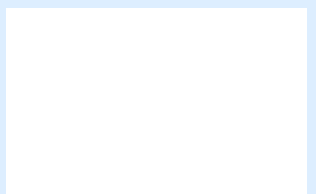
Step 3

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](#)

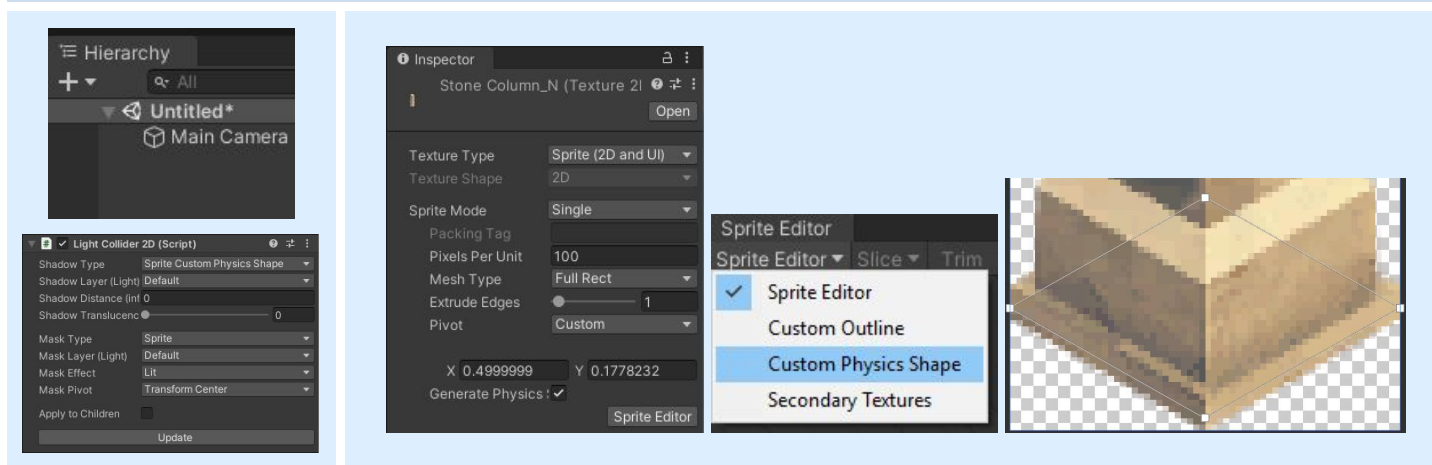
Instructions

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 " LightCollider2D " component to the already existing " GameObject " with sprite. Make sure the Mask Type is " Sprite ", so the shape of " SpriteRenderer " sprite will be masked and visible for the light source. 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 " Sprite Editor " button. Then switch to Custom Physics Shape mode. There you can add and edit vertices of 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](#)

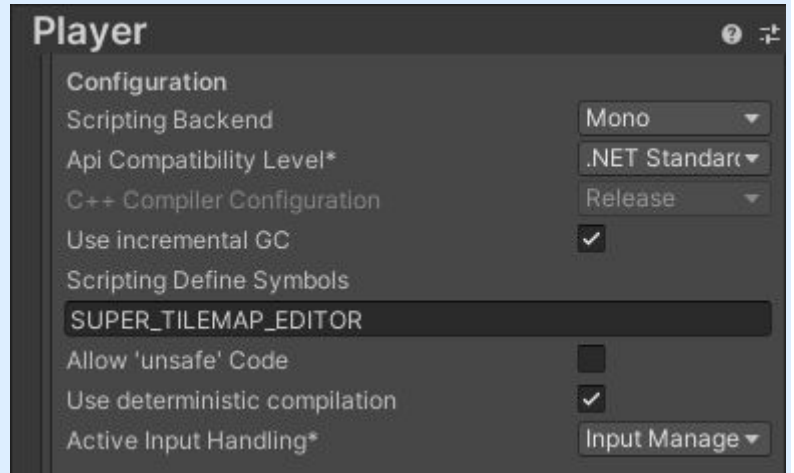
How To Enable Support?

Step 1

Open Player Settings

Step 2

*In category "Other Settings"
Make sure Scripting Define
Symbols include
"SUPER_TILEMAP_EDITOR"*



Step 3

Enjoy SuperTilemap Support

:)

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 object with Sprite Renderer, apply the "My Sprite" layer in "Sorting Layer" dropdown menu.
Step 6	Assign Sorting Layer To Lighting	Go to "Tools/2D Light" window that you can find in toolbar Set Sorting Layer Name to "My Lighting"

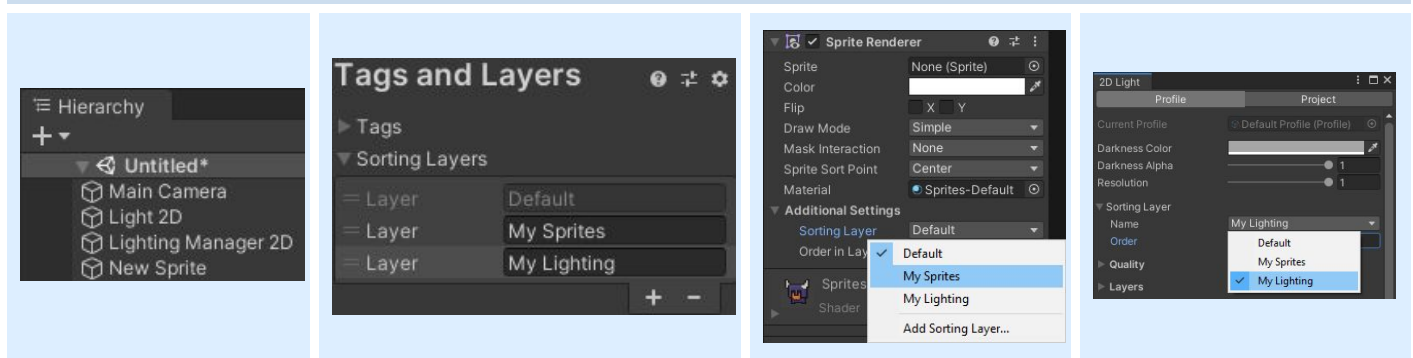
Step 1 & 2 & 3

Step 4

Step 5

Step 6

Visual Explanation



Game View



Normal Maps

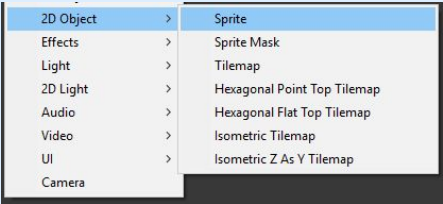
Introduction		
Lighting 2D has integrated and optimized 2D normal map support.		
Instructions		
Step 1	2D Sprite	Add a 2D sprite to the scene.
Step 2	Light Collider 2D	Attach Light Collider 2D Component to the sprite.
Step 3	Light Collider 2D Setup	Mask Type: Bumped Sprite Drag normal map texture into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

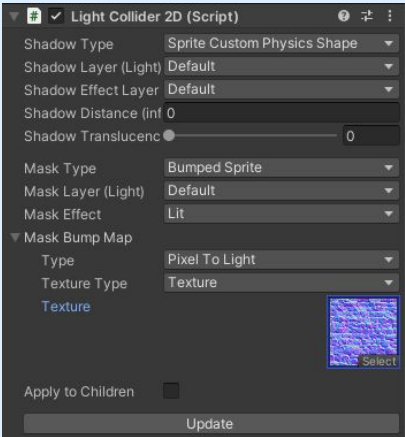
Step 1

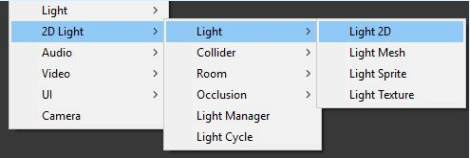
Step 2 & 3

Step 4


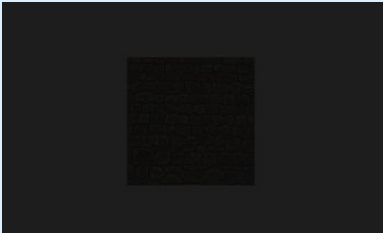

Visual Explanation







Game View



Scene View

Introduction

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

Instructions

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

Step 2

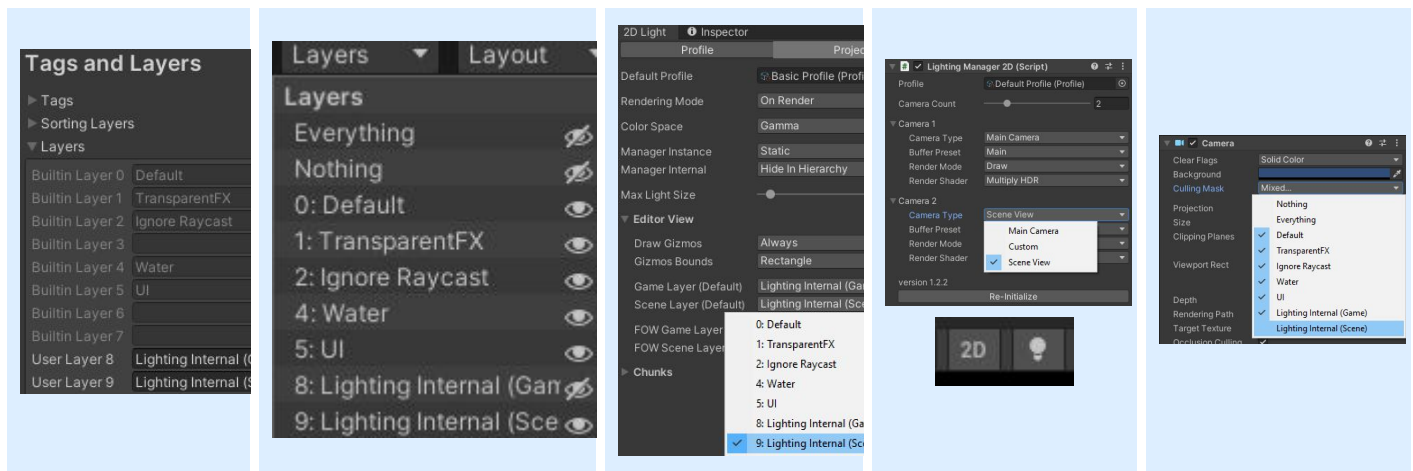
Step 3

Step 4

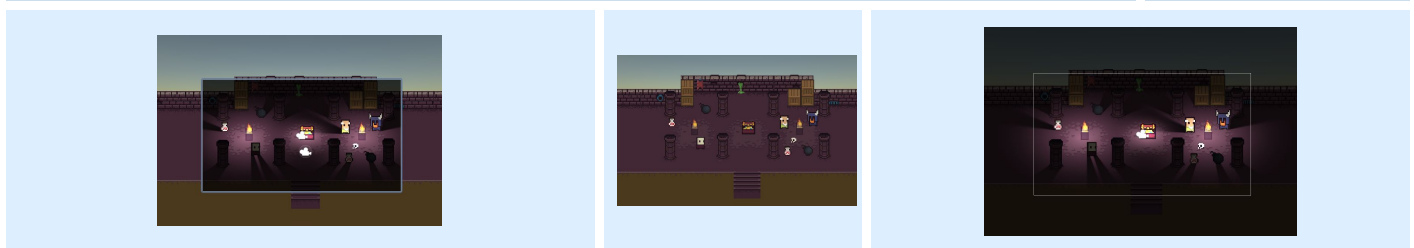
Step 5

Step 6

Visual Explanation



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	Let's start with the scene that already includes light 2D
Step 2	Light Event Preset	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.
Step 3	Adding Receiver Collider	Add Light collider 2D to your event receiver object.
Step 4	Adding Events Listener	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 visibility value in game view and used only for convenience/debug.
Step 5	Adding Shadow Collider	Create an “Light Collider 2D” object with default settings.

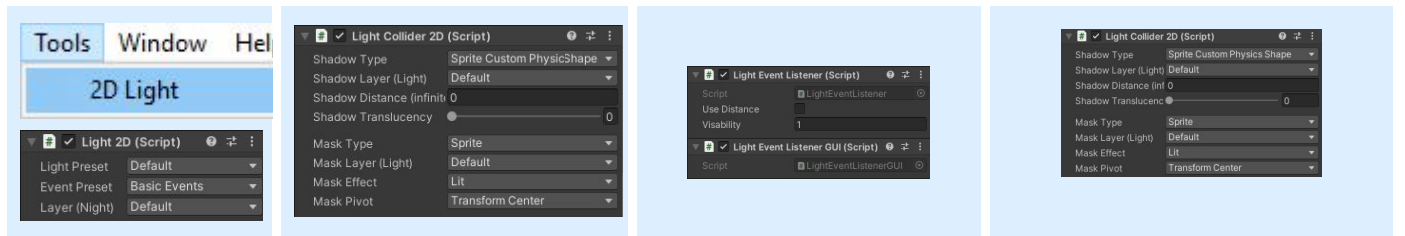
Step 2

Step 3

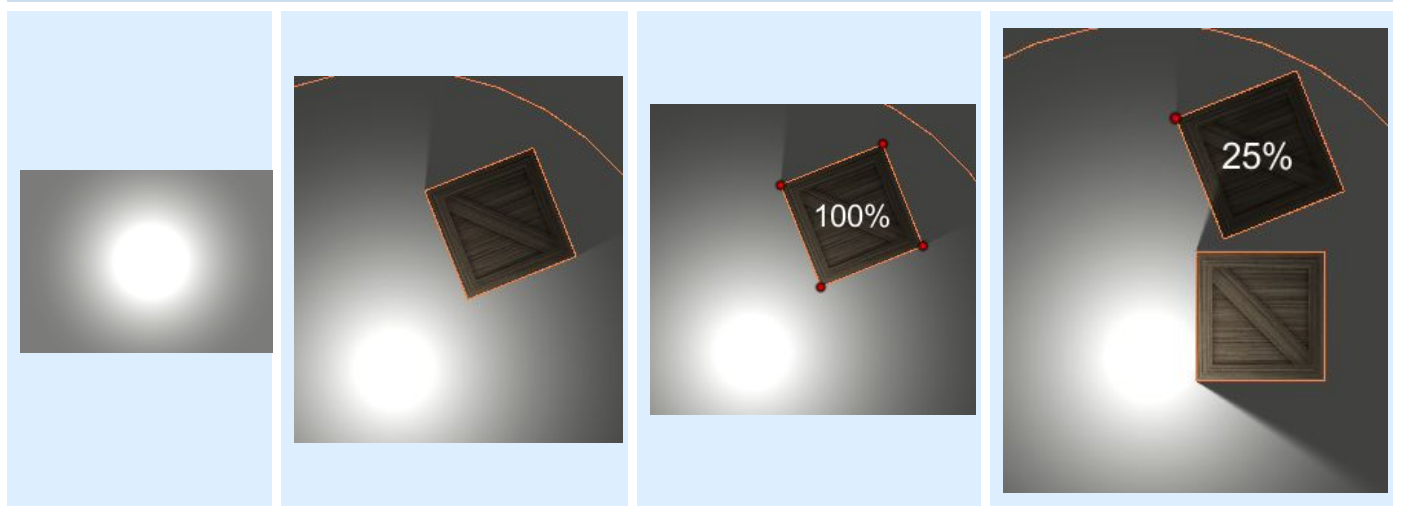
Step 4

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/2D Lighting window, Set darkness color to white or it's alpha to 0.
Step 3	Adding a Sprite	Add 2D Object/Sprite into the scene. Apply a sprite image to it.
Step 4	Adding Day Light Collider to Sprite	Add the “DayLight Collider 2D” component to the sprite game object.
Step 5	Setting Up Day Light Collider	Collision type: Sprite Custom Physics Shape Mask Type : Sprite

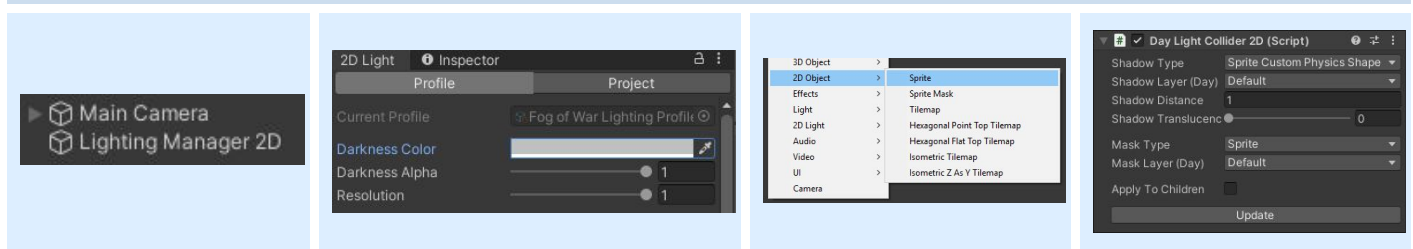
Step 1

Step 2

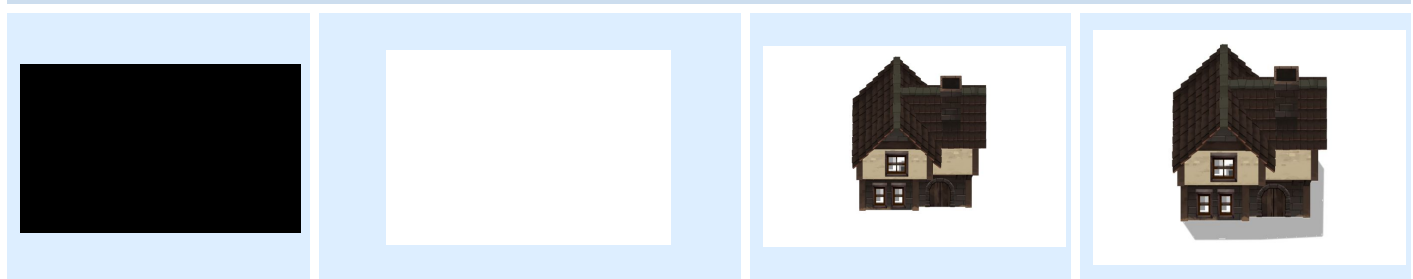
Step 3

Step 4 & 5

Visual Explanation



Game View

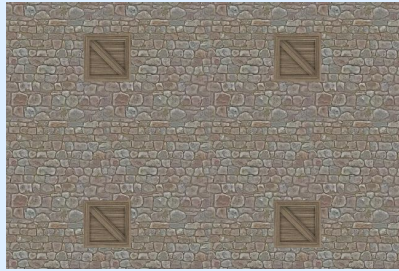


If there are any questions/feature requests or need help, feel free to contact!
I will answer the questions before you decide to use the asset.
Ask or find more information in [Discord!](#)
Documentation is the development, there will be updates!

Fog Of War (Simple) Part 1

Step 1

Scene

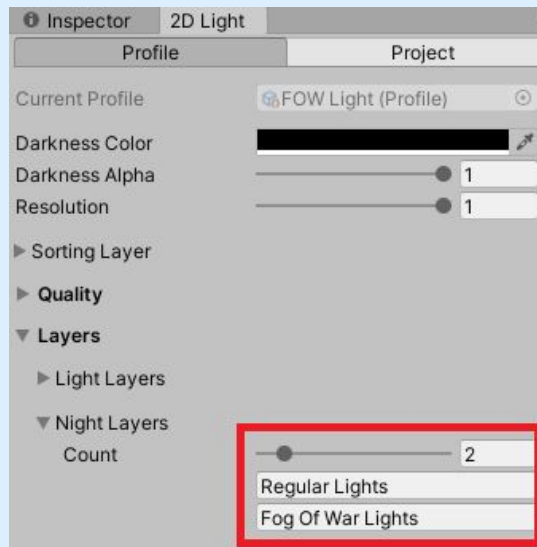


Starting scene includes

*Orthographic Camera
Floor (Sprite Renderer)
Boxes (Sprite Renderer)*

Step 2

Night Layers



*Search for “Night Layers” In
Tools/Light 2D, Layer section.*

*For this setup you will need two different
Night layers.*

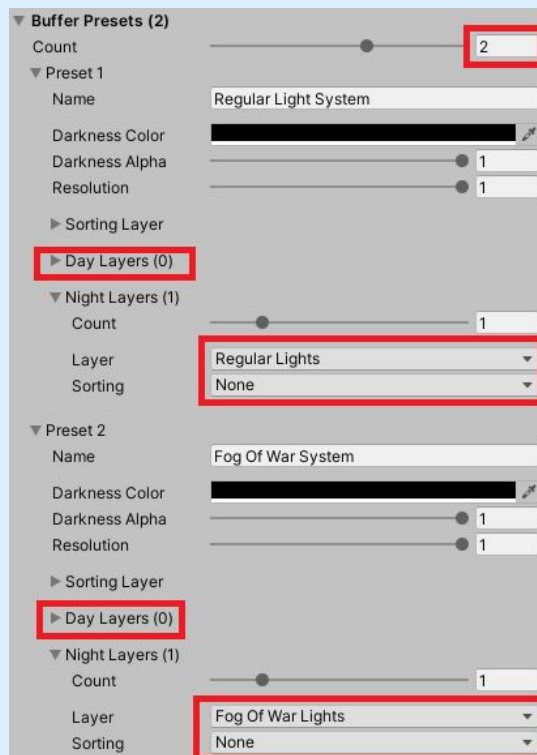
*1 - **Regular Lights** (behind the fog of war)*

*2 - **Fog of War Lights***

*Before adding the names, increase the
layer count.*

Step 3

Buffer Presets



*Search for the “Buffer Presets” section In
Tools/Light 2D.*

We need to initialize 2 buffer presets.

*1 - **Regular Light System** for the **Regular Lights** Night Layer.*

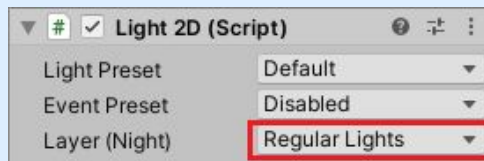
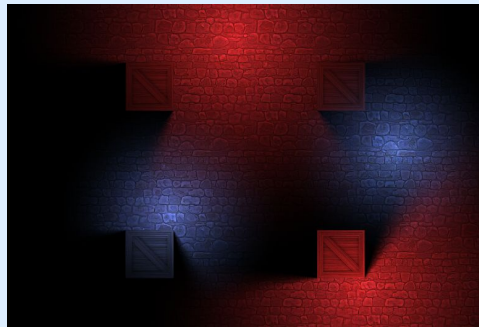
*2 - **Fog Of War System** for the **Fog Of War Lights** Night Layer.*

*Preferably set “Day Layers” count to 0 for
both of the presets.*

Fog Of War (Simple) Part 2

Step 4

Regular Lights & Colliders



Attach “Light Collider 2D” components for the boxes.

Light Collide 2D:

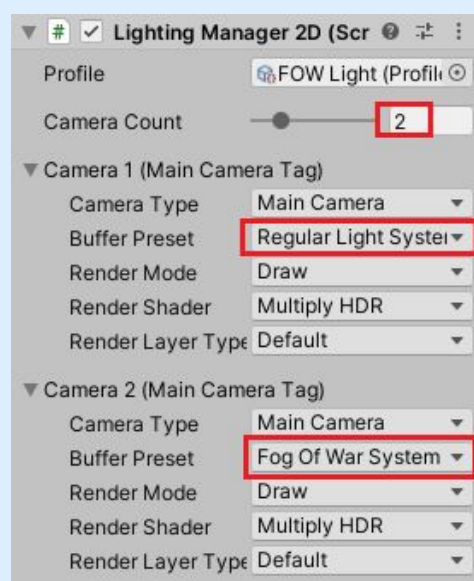
Shadow Type = Sprite Physics Shape

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 did set up **Buffer Presets**.

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

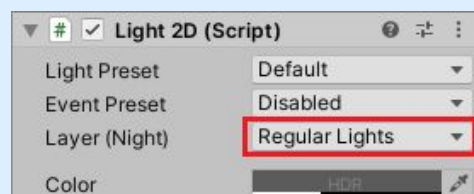
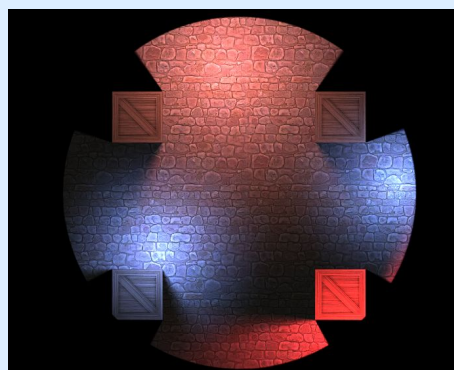
Beforehand, set camera count to 2.

Camera 1 should use **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 “**gfx_fogofwar**” sprite for the texture of the light.

-

Ideally, the light should be white.

Both presets are mixed with Multiply blends.

Fog Of War (Render Texture) Part 1

Step 1

Scene



Starting scene includes

Orthographic Camera

Floor (Sprite Renderer)

Box (Sprite Renderer)

Players (Sprite Renderer)

Step 2

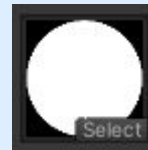
Collider & Light



*Attach **Light Collider 2D** to the box sprite.*

*Add **Light 2D** gameobject into the scene.*

*Use the “**gfx_fogofwar**” texture for the light’s sprite.*



Step 3

Fog Of War Sprites

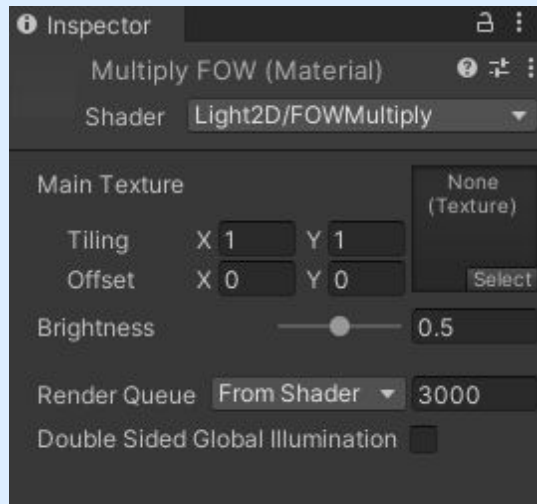


*Attach “**FogOfWarSprite**” to the player gameobjects.*

Fog Of War (Render Texture) Part 2

Step 4

Material

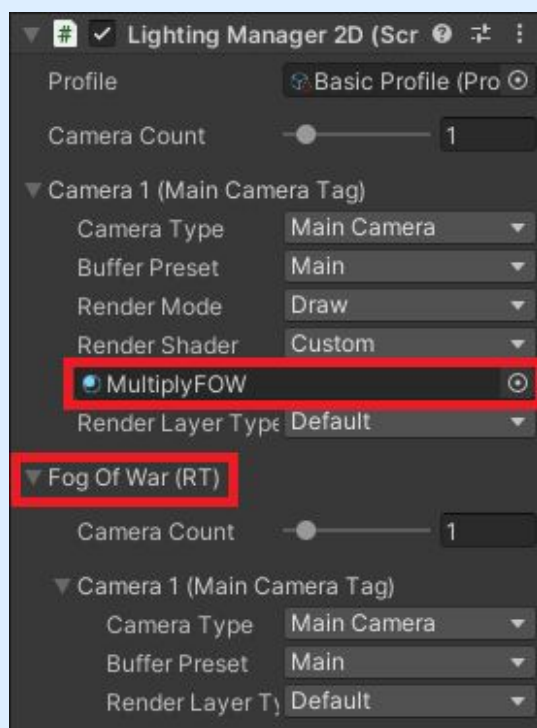


Create a material using shader
"Light2D/FOWMultiply"

Brightness would determine how bright the
shadows appear to be.

Step 5

Lighting Manager



Attach created material as
Render Shader for camera 1.

Also set camera count 1
to Fog Of War (RT).



Unity Lighting 2D Components

Components

LightingManager2D	<i>Previously used for lighting settings, now all settings are moved to “Tools/Lighting 2D” via Settings Profile (scriptable object). Now used as root for all generated Lighting 2D effects</i>
Light2D	<i>Versatile light emitting source, this component emits light. Can use custom texture to set up a unique light emitting look.</i>
LightCollider2D	<i>Versatile light collider component. Can be used to set up shadow casting for sprites & solo collider components.</i>
LightTilemapCollider2D	<i>Tilemap Light Collider component can be used to set up shadow casting for standard unity Tilemap component and Super Tilemap Editor system.</i>
DayLightCollider2D	
DayLightTilemapCollider2D	
LightSprite2D	<i>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)</i>
LightTexture2D	
LightParticleSystem2D	
LightRoom2D	<i>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 day light shadows.</i>
LightTilemapRoom2D	<i>This component can be used to darken the area in the daylight. Can be used to mask tilemap to be affected by lights.</i>
LightOcclusion2D	
LightTilemapOcclusion2D	
FogOfWarSprite	
FogOfWarTilemap	

Bonus Components

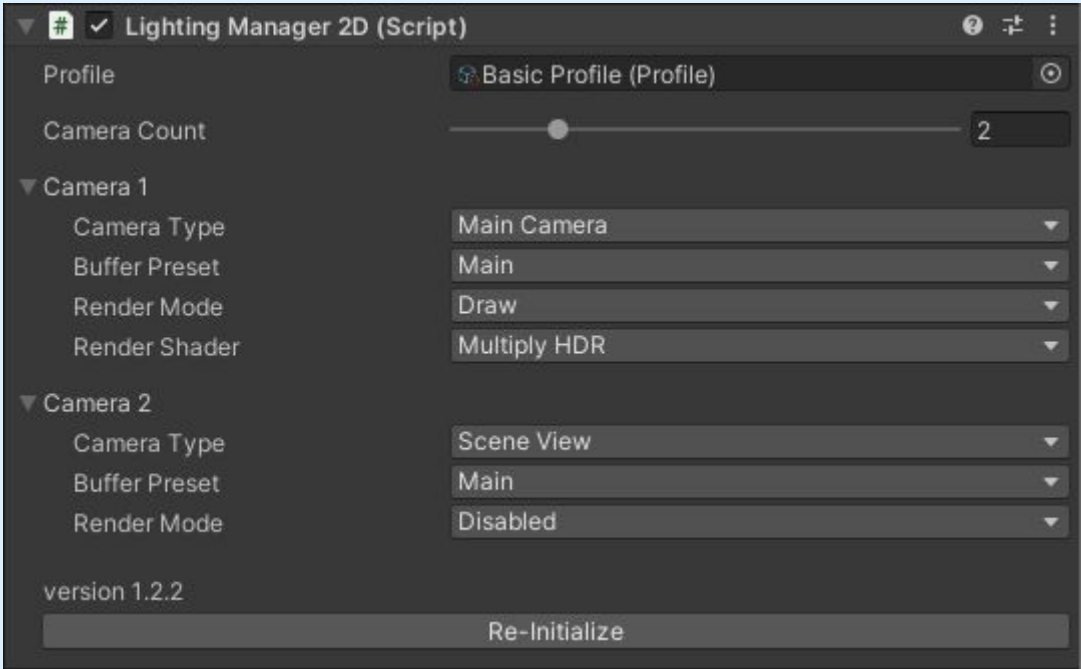
[ColliderLineRenderer2D](#)

Creates an outline for Collider components with selected color

[Mesh2D](#)

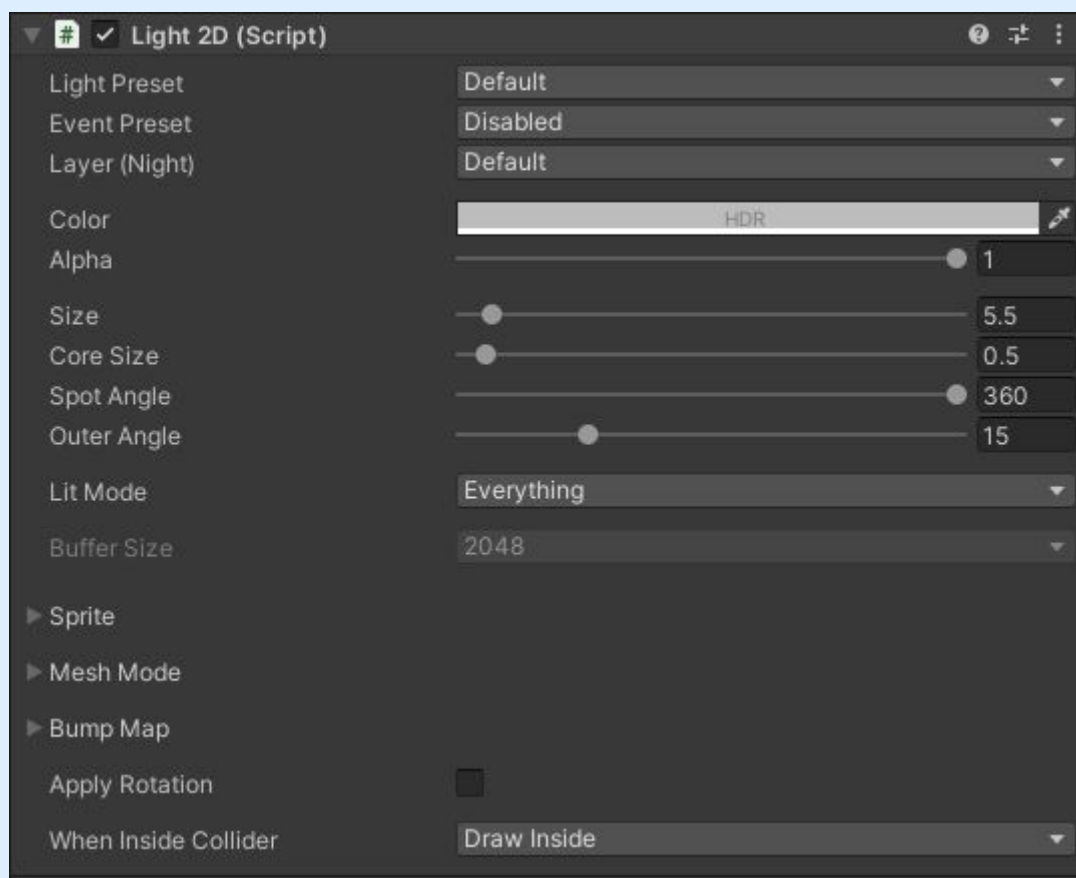
Creates a mesh from Collider components and attach it to the mesh renderer. Mostly used to display basic demo scenes without any images.

Component Reference

LightingManager2D			
appearance			
description	<p>Only one Lighting Manager 2D is allowed per scene. This component is automatically generated on the first 2D lighting API call.</p>		
information	Profile		
	Cameras		

Light2D

appearance



description

This component emits light. Can use custom texture to set up a unique light look.

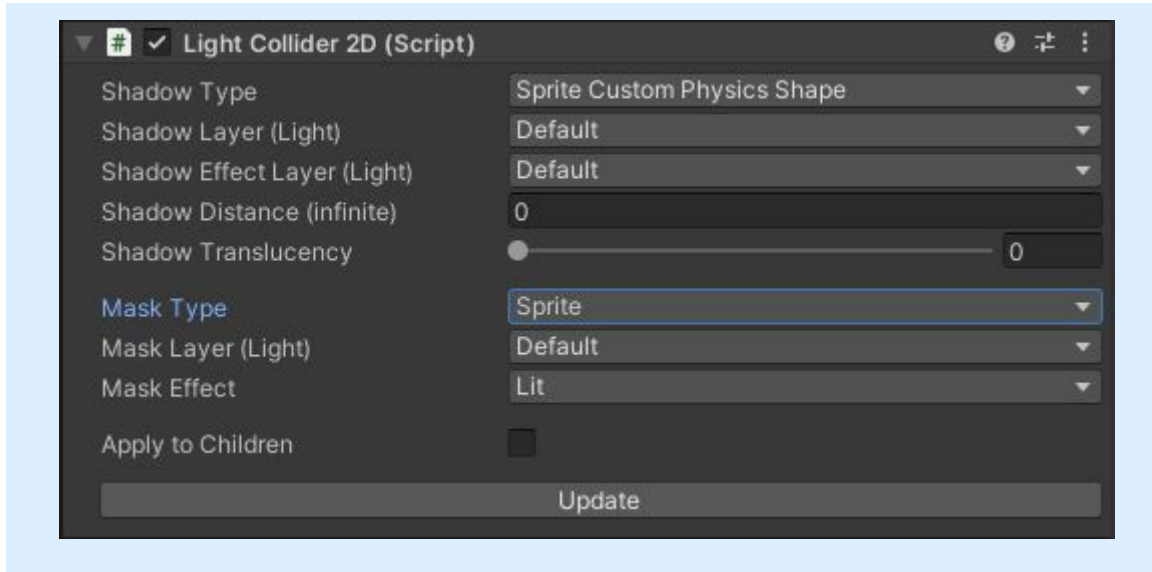
information

Layer Count	Int [0 - 31]	Layer count that will be included in light calculations. The more layers, the less optimized light calculation will be. 3 layers should be enough for making quite complicated or tricky lighting scenes.
Layers	Layer Object List	The list of layer objects. For each layer select it's Id that is used in every light collider. Type gives an option to draw colliders or masks only. Order allows you to sort the shadows and masks according to specific statement (Distance To Light, Y Axis)
Color	24 Bit Color	The color of light. The darker the color, the less visible it will appear. Black color is not visible at all.
Alpha	Float [0 - 1]	Transparency of light. The higher alpha value, the more visible light appears to be.
Size	Int [0 - Unidentified]	The size of light, keep in mind that increasing the size of light does not automatically increase it's "buffer size", very large lights require larger buffer size, otherwise pixelated artifacts appear.
Buffer Size	Enumerator	The resolution of the light buffer. Larger buffer leads to more crispy shadow details, however it costs more performance.
Light Sprite	Default	Default texture which is being applied for the light.
	Custom	Enables custom sprite texture to use for the light.
Sprite	Sprite	After enabling LightSprite/Custom, you are able to

			<i>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

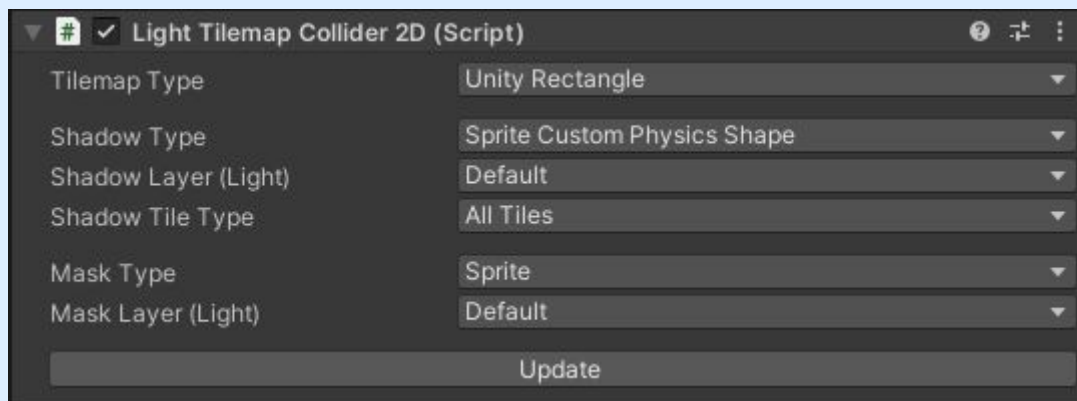
Versatile light collider component. Can be used to set up shadow casting for sprites & solo collider components.

Collision Type	None	Disables all shadow casting for this object.
	Collider	Use 2D collider for geometry of shadow casting. (Box2D, Circle2D, CapsuleCollider2D, Polygon2D, Edge2D)
	Sprite Custom Physics Shape	Use Sprite's custom physics shape which can be accessed with Unity Sprite Editor.
	Mesh	Uses Mesh Filter Mesh to cast shadows.
Collision Layer	Int [0 - 31]	Lighting layer of the object, this layer should be included in the lighting source layer list.
Mask Type	None	Disables all masking for this object.
	Sprite	Uses sprite from sprite renderer of this object for the mask.
	Collider	Uses 2D Collider geometry for the mask.
	Sprite Custom Physics Shape	Use Sprite Custom Physics Shape geometry to mask the object.
Mask Layer	Int [0 - 31]	Lighting layer of the object, this layer should be included in the lighting source layer list.

A diagram showing two light blue rectangular buttons. The left button is labeled 'Update' and the right button is labeled 'Editor Button'. Both buttons have a thin black border and a subtle gradient.

Press this object to re-initialize geometry of the collider. This is 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.;

appearance



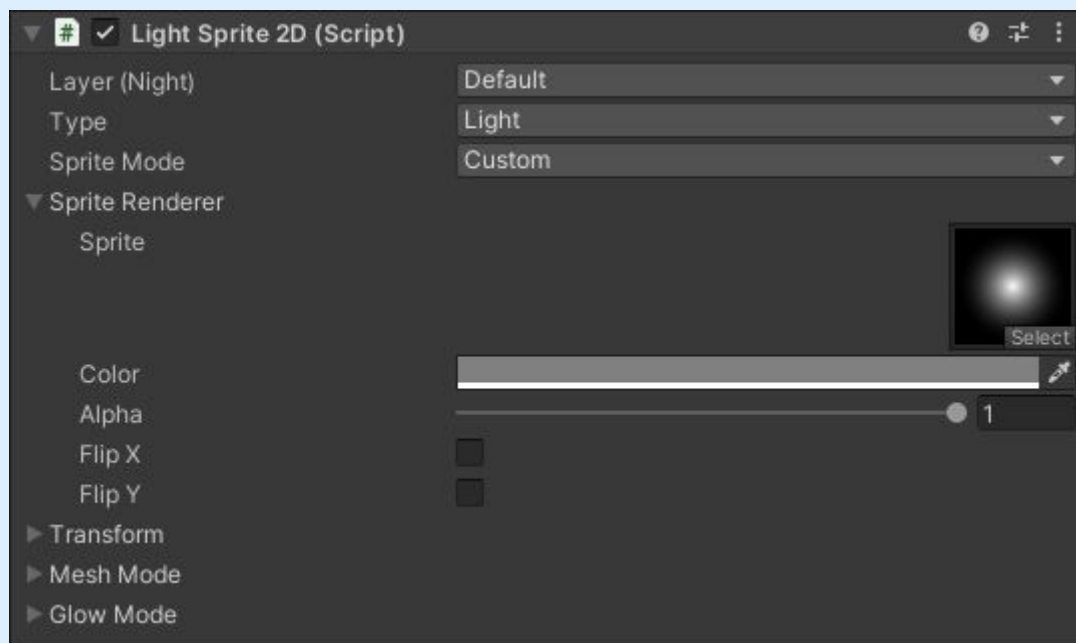
description

Tilemap Light Collider component can be used to setup shadow casting for standard unity Tilemap component and [Super Tilemap Editor](#) system.

information

Tilemap Type	Unity Engine Tilemap	Use standard tilemap for shadow casting. No additional collider components are needed for this. Lighting system will take sprites used in the tileset and apply their selected properties for collisions.
	Super Tilemap Editor	Use Super Tilemap Editor for the shadow casting.
Collision Type	None	Disable shadow casting for this tilemap object.
	Rectangle	All tiles are treated like rectangles
	Sprite Custom Physics Shape	Try using sprite's custom physics shape for shadow casting.
Collision Layer	Layer Enumerator [0 - 31]	Lighting layer of the object, this layer should be included in the lighting source layer list.
Mask Type	None	Disable masking for this object.
	Sprite	Uses tile sprites for masking.
	Rectangle	Use rectangle shape for masking for mask this tilemap.
	Sprite Custom Physics Shape	Use tile sprite custom physics shape.
Mask Layer	Layer Enumerator [0 - 31]	Lighting layer of the object, this layer should be included in the lighting source layer list.
Day Height & Size	Enumerator Float [0 - Undenified]	Enable sun's shadow casting in the daylighting system.
Batch Sprite Masking	Boolean	This is optimization. Enable this option when whole tile palette consist of same texture file. This should improve batch calls from lighting system.
Update Collisions	Editor Button	Press this object to re-initialize geometry of the collider. This is 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.;

appearance




description

*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**)*

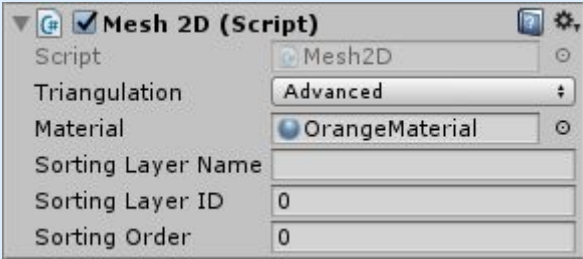
information

	Type	Particle	Additive shader effect for this component.
		White Mask	Applies 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	Applies black mask for the object, object and everything underneath is completely not visible.
	Sprite Mode	Custom	Select your own sprite for this component.
		Sprite Renderer	Synchronize sprite variable with Sprite Renderer component attached to the same game object.
	Color	24 Bit Color	Color of the effect. This is not taking any effect when using white mask or black mask.
	Alpha	Float [0 - 1]	Transparency of this effect. This is not taking 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 it's strength.
	Blur Iterations	Int [1 - 10]	The times blur algorithm is being applied.

	<i>Apply Blur</i>	Boolean	<i>For this option to be used, you need to enable sprite write/read setting.</i>
	<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 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	<i>Triangulation</i>	<i>Triangulation method to be used when generating 2D mesh.</i>
	<i>Material</i>	<i>The material of the object.</i>

Tools/Light2D			
appearance			
description	<p><i>Only one Lighting Manager 2D is allowed per scene.</i></p> <p><i>This component is automatically generated on first 2D lighting API call.</i></p>		
information	Rendering Mode	On Render	Additional Sorting Order option will appear for this setting. You can set specific sorting order for lighting buffer.
		Pre Render	Game objects with sorting order will appear above the lighting buffer. Game objects with lower sorting order ID will appear below the lighting buffer.
		Post Render	Not recommended to use. Lighting buffer is drawn on post process rendering loop. It seems to have many issues if you want to use post-processing effects.
	Darkness Color	24 Bit Color	The darker color, the darker scene will be drawn. For day lighting effects you should set up darkness color very bright. Can be used to get "tint" which could represent dusk or dawn.
	Shadow Darkness	Float [0 - 1]	The darkness of day lighting shadows. 0 - not visible, 1 - opaque.
	Sun Rotation	Radians	Sun rotation will affect all lighting colliders with day shadows. This variable can be manipulated in real time to achieve day & night cycle.
	Draw Additive Lights	Boolean	Enable additive lights drawing. When disabled, it will skip all checks for additive lights drawing. If you are not using this feature in any of the lights, it is suggested to disable this.
	Draw Rooms	Boolean	Enable rooms feature, mostly used for scenes with day lighting effects.
	Draw Occlusion	Boolean	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.
	Draw Day Shadows	Boolean	Enable day shadow casting for lighting colliders. It is recommended to disable this if you are not using day lighting effects.
	Draw Main Buffer	Boolean	When disabled, it hides lighting buffer from main camera. This setting is similar to "Disable Engine" except all lighting calculations will be still performed, but not drawn.
	Draw Scene Buffer	Boolean	When enabled, lighting sources can be seen in scene view.
	Lighting Resolution	Float [0.125 - 1]	The resolution of Lighting buffer. The higher resolution is, the more detail 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.
	Fixed Light Buffer & Size	Boolean Enumerator	This option enables better poll system for light source system. Improves performance and is recommended for mobile build. When enabled, custom light buffer size is not available, all lights will be having same texture size.
	Batch Collider Mask	Bool	When enabled, lighting will use 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 performance.

	Debug	Bool	<i>When enabled, additional lighting information will be displayed in game view. This helps to benchmark lighting performance. Mostly used by Smart Lighting 2D Developer.</i>
	Disable Engine	Bool	<i>This option disables all lighting features, no lighting calculations will be applied.</i>