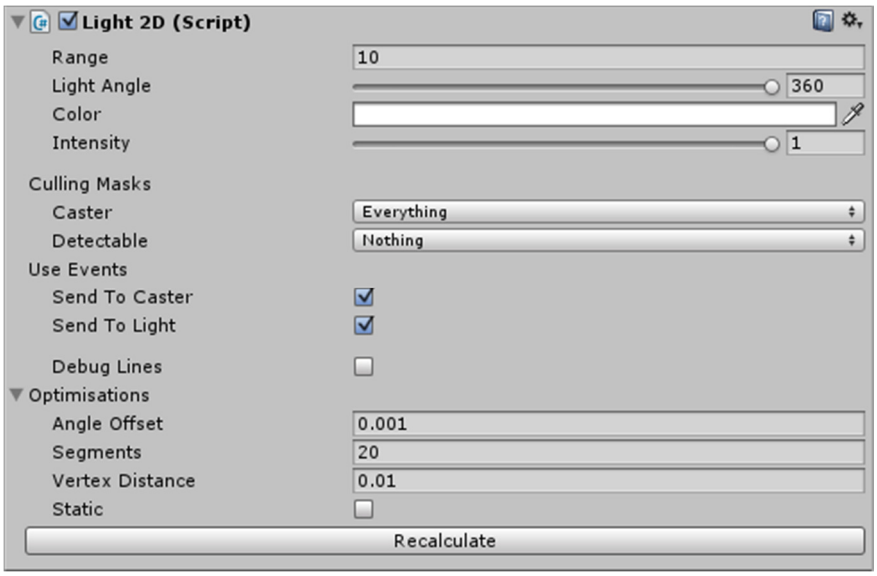
Light 2D allow for lighting in 2D scenes, allowing sprites and 2D objects to cast shadows on their surroundings. Light 2D uses collider information to achieve this, thus all objects intended to cast shadows must have one of the 4 Collider2D types attached.

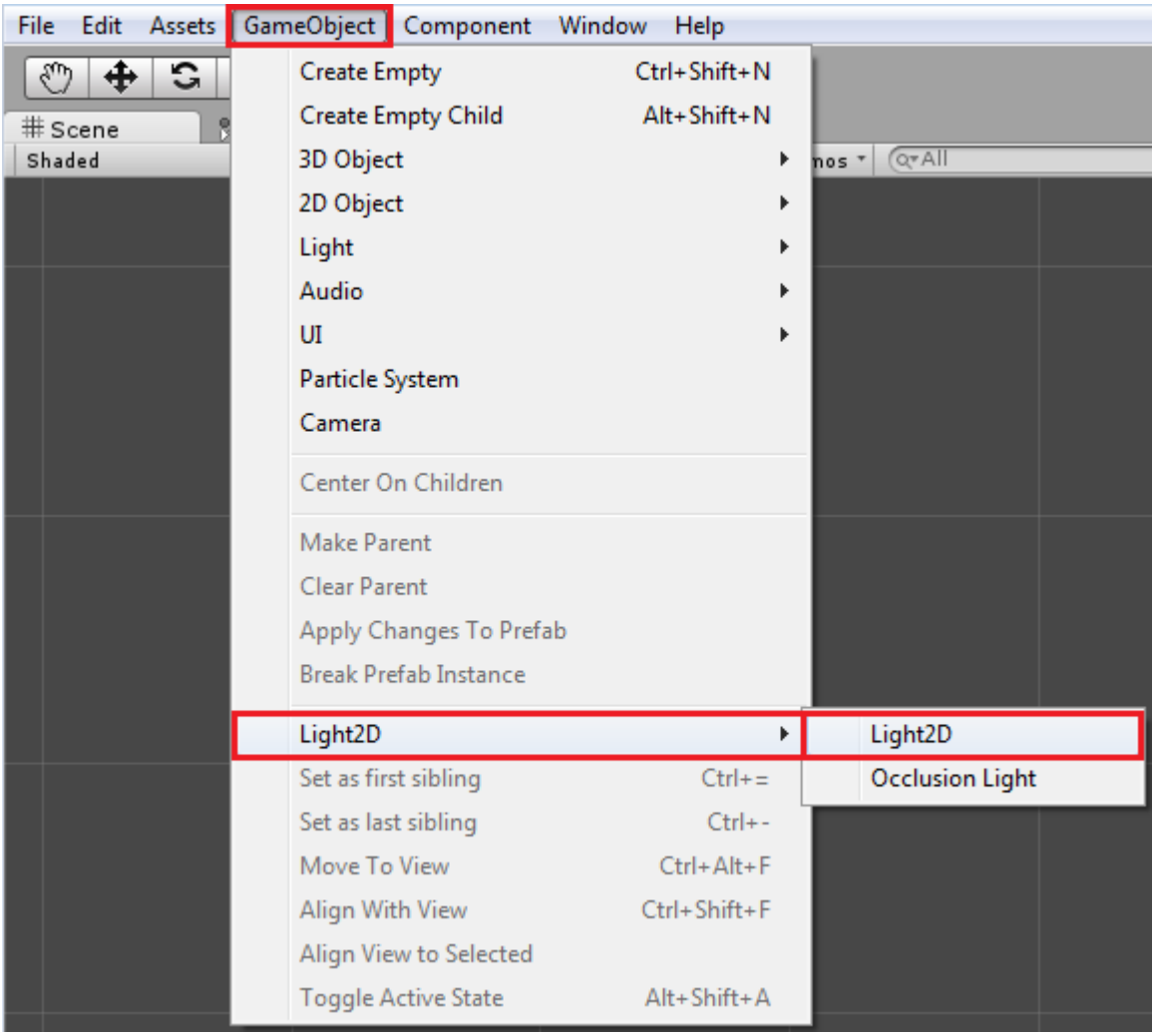


Property	Function
Range	How far the light is emitted from the center of the light. The light radius.
Light Angle	Determines the angle of the spot light in degrees.
Color	Color of light emitted
Intensity	Brightness of the light.
Caster Mask	Use to selectively exclude layers casting shadows and interacting with the light. See Layers.
Detectable Mask	Use to selectively exclude layers interacting with the light. See Layers. Note objects on this mask will trigger events but not cast shadows.
Send To Caster	Send triggered event to object.
Send To Light	Send triggered event to game object attached with the Light2D.
Debug Lines	Show debug lines
Angle Offset	Offset between penumbra raycasts. Decrease if: • light overlaps objects, • light interacting with very small game objects or the range is very large
Segments	Number of initial fan raycasts used, effectively the angular resolution of the light mesh.
Vertex Distance	Minimum allowed distance for light vertices. If artifacting is present then decrease.
Static	Stops light from recalculating each frame. Used for background lighting only.
Recalculate	Recalculate light – Used for static lighting when objects/light is moved.

Contact:
support@discoursegames.co.za

Adding 2D lighting to your scene.

Light2D can easily be added using GameObject menu. See figure below.



Note: For the menu to work, the Light2D and OcculsionLight2D must be in the filepath "Assets/Light2D/Prefabs". If you change this, change the filepath variable in Light2D_inspector to maintain functionality.

Alternatively Light2D can be added by dragging the prefab from Light2D.prefab into the scene view from "Assets/Light2D" **or** the Light2D.cs script can be added to any GameObject. If script is added directly, the light material must be added to the object.

Limitations and Notes

- CircleCollider2D must maintain circularity. Scaling must be equal in x and y.
- CircleCollider2D must be above (closer to the camera) the light mesh to ensure no overlap.
- Overlapping colliders will cause visual artifacting
- If shadow casting object has an attached Rigidbody2D, ensure Interpolate is set to None. Setting this to Interpolate or Extrapolate will cause extremely noticeable visual artifacting.

Contact:
support@discoursegames.co.za

Code Reference

Variable	Description	Range
AngleOffset	Offset between penumbra raycasts. Decrease if: • light overlaps objects, • light interacting with very small game objects, • light range is very large.	Value must be greater than 0.0.
color	The color of the light, applied as vertex color of light mesh.	
detectableObject_Mask	LayerMask for selectively culling which layers interact with Light2D events, but do not cast shadow	
DetectableObjects	List of gameobjects in the lit area not casting shadow (Read Only)	
Intensity	The alpha value of the light.	Value must be between 0 and 1.
isStatic	Stops light from recalculating each frame. Used for background lighting only.	
NumberOfSegments	Number of initial fan raycasts used, effectively the angular resolution of the light mesh.	Value must be greater than 3.
Range	Range, or radius of the light. How far the light is emitted from the center.	Value must be greater than 0.0
shadowCaster_Mask	LayerMask for selectively culling which layers cast shadows	
ShadowCasters	List of gameobjects casting a shadow (Read Only)	
showDebugLines	Show debug lines: • White: Initial raycast fan lines • Cyan: Light Bounds • Green: Shadow Edges	
TotalCoverageAngle	Angle in degrees the light covers (360 being a point light)	Value must be between 0 and 360, inclusive.
useEvents_SendToCaster	Send triggered event to object.	
useEvents_SendToLight	Send triggered event to game object attached with the Light2D.	
VertexDistanceTolerance	Minimum allowed distance for light vertices. If artifacting is present then decrease.	Value must be greater than 0.0.

Contact:
support@discoursegames.co.za

Public Functions

Function	Description
RecalculateLight()	Recalculates light mesh. Use to reset for static lighting if environment or light moves.
setAngle(float newAngle)	Set rotation Z component of transform in degrees.
setColorWithoutAlpha(Color newColor)	Set color of the light without affecting the intensity

Event System

Events can be sent when objects enter or exit the lit area.

Events can be called on any scripts on the GameObject that entered/exited the lit area and the GameObject with Light2D attached, based on settings editable via inspector.

Notes:

- Events are not called on disabled MonoBehaviours.
- Events are called during LateUpdate()

OnLight2DEnter

This is called when a GameObject with an attached Collider2D enters the bounds of Light2D mesh.

```
using UnityEngine;
using System.Collections;

public void ExampleClass: MonoBehaviour {
    void OnLight2DExit(GameObject other)
    {
        ExampleFunction(other);
    }
}
```

OnLight2DExit

This is called when a GameObject with an attached Collider2D exits the bounds of Light2D mesh.

```
using UnityEngine;
using System.Collections;

public void ExampleClass: MonoBehaviour {
    void OnLight2DEnter (GameObject other)
    {
        ExampleFunction (other);
    }
}
```


Contact:
support@discoursegames.co.za

How to set up Occlusion Light

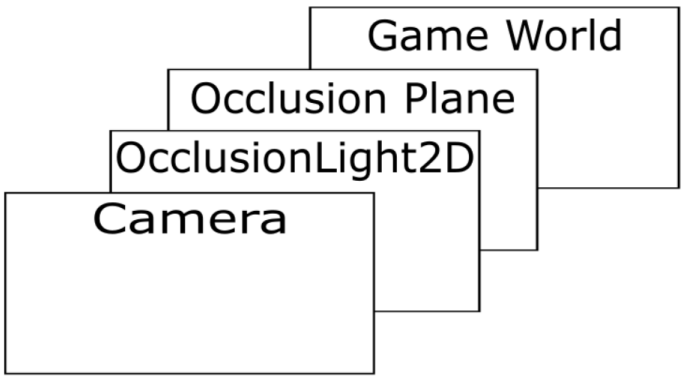
The occlusion system needs three elements to work; An Occlusion Plane used to obstruct the game world, a second occlusion camera and an OcclusionLight2D. Further a new “Occlusion” Layer must be created.

Occlusion plane; which can be a coloured copy of the background, a simple plane or anything to achieve the desired graphical effect. This occlusion plane must fully obstruct the view from the camera (i.e be between the camera and the gameworld). This plane must be set to the Occlusion Layer, and placed between the GameWorld and the camera.

Occlusion Camera; must be a child of the main camera, Clear Flags set to Depth Only,Culling Mask to Occlusion and the Depth to be in front of the main (default -1).

OcculusionLight2D; can be added via the GameObject dropdown, or constructed by adding the Light2D.cs script with the Light2D Occlusion material (found Light2D/Materials+Shaders). Must be placed between the Occlusion plane and the game world.

The arrangement should be as follows: Camera, OcclusionLight2D, OcclusionPlane, Gameworld. See figure below.



See Occlusion Example

Contact:
support@discoursegames.co.za