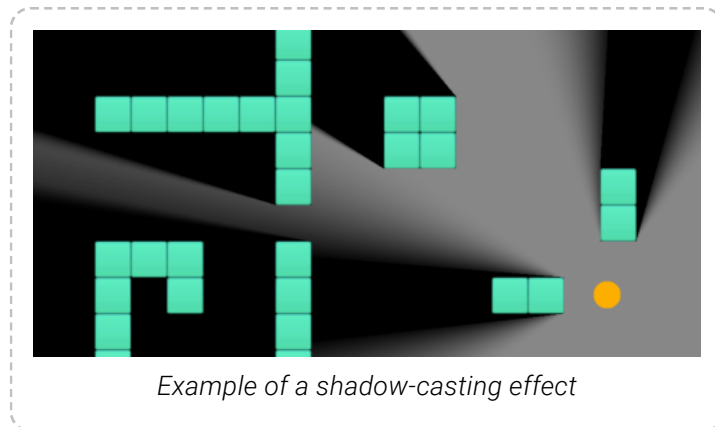


SHADOW LIGHT

View online: <https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/shadow-light>

The **Shadow light** object can render real-time shadows from other objects with the **Shadow caster** behavior.

Construct comes with a number of examples of shadow-casting effects. Search for *Shadows* in the **Example Browser** to locate them.



Scripting

When using JavaScript or TypeScript coding, the features of this object can be accessed via the **IShadowLightInstance** script interface.

Shadow rendering

The Shadow light object renders shadows adjacent to objects with the *Shadow caster* behavior, using the object's collision polygon and the relative location of the light. Shadows are filled in over the background, as opposed to rendering sections of light. The Z order of the Shadow light object determines whether the shadows appear above or below other objects.

The light can be set to have a radius. If the radius is 0, it acts like a point source, and shadows have hard edges. If the radius is larger, it accurately renders penumbras (the transition from lightness to darkness) at the edges of shadows. However in this mode the light height is ignored and all shadows extend off the screen.

Shadow limitations

Due to the shadow rendering algorithm, there are a couple of limitations:

- The shadow casters can only use **convex collision polygons**. Shadows will not render correctly if they use concave polygons. If you need a concave shape, this can always be

achieved by placing multiple shadow caster objects next to each other to compose a concave shape out of convex parts.

- Antumbra (beyond where the umbra converges to a point) are not rendered. To avoid the umbra converging to a point on-screen, avoid using shadow casters smaller than the light radius. Stick to large shadow casters and a small light radius.
- When using a light radius, avoid placing the light very close to or directly over a shadow caster. Shadows can fail to render correctly in these circumstances.

Tags

By default, all shadow lights cast shadows off all shadow casters. In some cases it is desirable to only have certain shadow lights cast shadows off certain shadow caster objects. Each shadow light object can be assigned a tag, and the *Cast from* property set to only cast shadows from that object off shadow caster behaviors with the same (or different) tags.

Shadow light properties

Light height

The height of the light, used with the shadow caster object heights to calculate the length of shadow to cast. This property only has an effect if the *Light radius* is 0, otherwise shadows always extend offscreen.

Light radius

The radius of the light. If the radius is 0, the light acts like a point source and shadows are hard-edged. If the radius is larger the object will render penumbras at the edges of shadows. The larger the radius, the wider the penumbras will be. For correct rendering avoid using a large radius, and especially avoid making the radius larger than any of the shadow caster objects. If the radius is not 0, the light height is ignored and shadows always extend offscreen.

Cast from

Which shadow caster objects to render shadows for from this object. The options are:

- **All:** every shadow caster object will get a shadow rendered for this light.
- **Same tag:** shadows will only be rendered for shadow casters with the same Tag property.
- **Different tag:** shadows will only be rendered for shadow casters with a different Tag property.

Tag

Used to determine which shadow casters to render shadows for, depending on the *Cast from* mode. If *Cast from* is set to *All*, the tag is ignored.

Preview Paid plans only

Enable to run a preview of the shadow casting effect directly in the Layout View.

Shadow light conditions

The Shadow light object does not have any of its own conditions.

Shadow light actions

Set cast from

Set light height

Set tag

Set the corresponding object properties. For more information see *Shadow light properties*.

Set light position

Sets the position in the layout from which shadows are cast from.

Note that using the normal Set position action will also update the light position. However the Shadow Light object automatically positions itself in the middle of the viewport in order to draw over the whole screen. Using Set position to set the light position in the middle of the viewport may conflict with its automatic positioning, so this action can be used as a more reliable way to guarantee the light position is placed at the given location.

Set shadow color

Set the color of the shadows that are rendered by the light. The default is black. Use an expression of the form `rgb(red, green, blue)`. To set the opacity of the shadows, change the opacity of the *Shadow light* object.

Shadow light expressions

LightX

LightY

The X and Y co-ordinates of the light source in the layout. Note a quirk: the light source is moved using the ordinary *Set position* actions, but due to the way the object rendering works

the ordinary *X* and *Y* expressions always return a position relative to the viewport instead. The *LightX* and *LightY* expressions return the actual position of the light source.

Tag

Return the current tag of the object.