Documentation

Thank you for choosing the **Light Switch System!**
I hope this asset enhances the creative possibilities of your game and provides a smooth experience. If you encounter any issues or have questions, please refer to this documentation. In case you need further assistance, don't hesitate to underline reach out to me.

Your feedback matters! If you find the Light Switch System valuable and enjoy using it, I kindly ask for your support by leaving a positive review on the Unity Asset Store. Your review helps me improve the product and inspires me to create even more assets for the community.
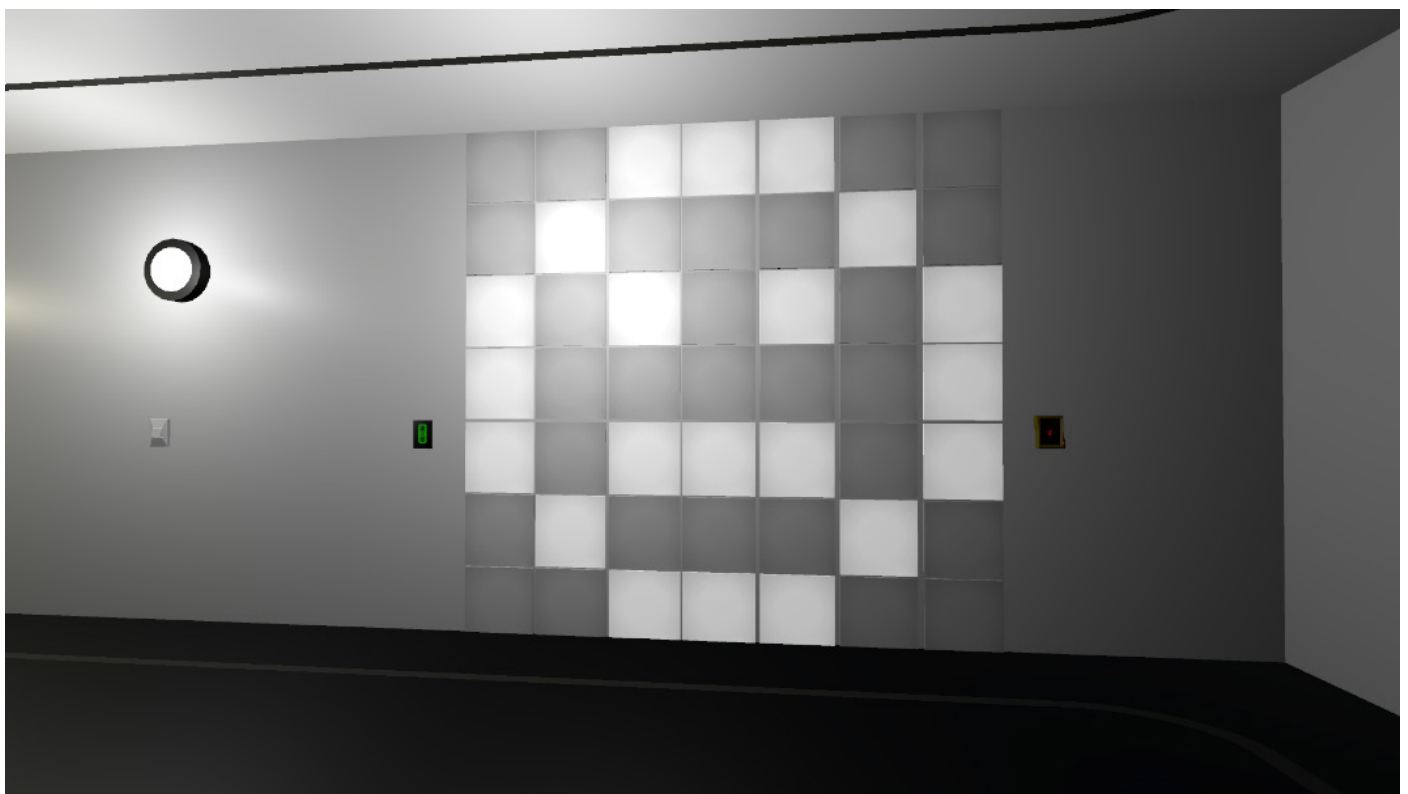Thank you for your time and for being a part of our user community!

The **Light Switch System** is a complete solution that makes it easy to turn lights on and off in your game in a flexible way.

The system comes with everything you need to put controllable lights in your game, including 10 switch models, 4 lamp models, and scripts that make it easy to use the system without coding.
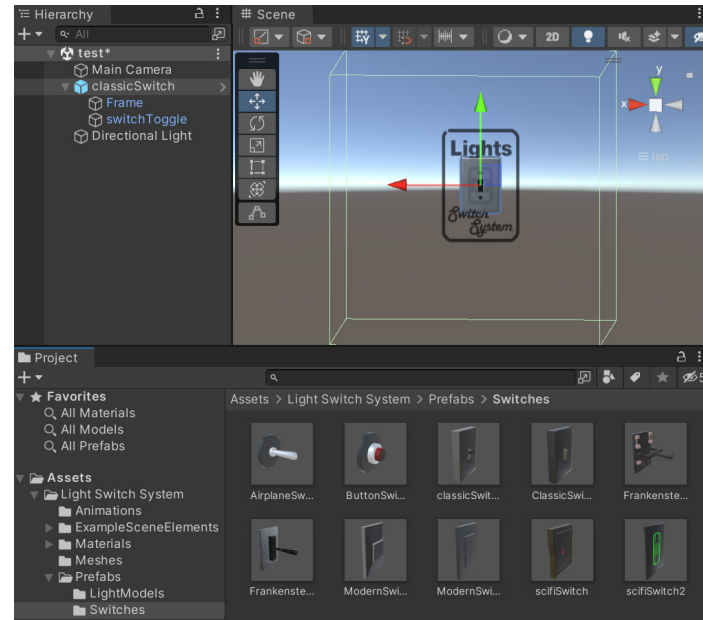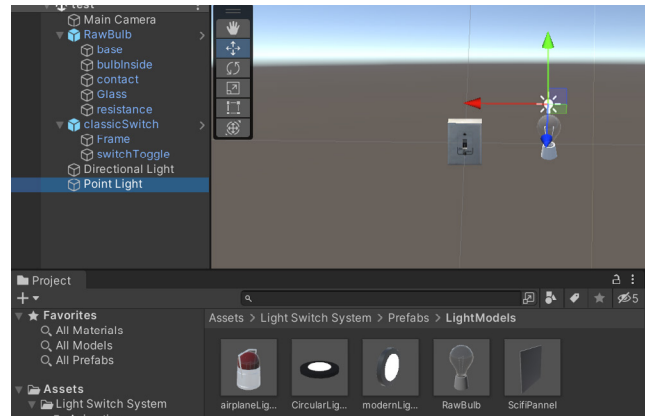
There are two modes available: Lite mode and Pro mode. Lite mode is a fast and simple option that lets you use one switch to control one light. Pro mode is more advanced, and allows you to control multiple lights in more complex ways. For example you can have a master switch that controls many lights, as well as individual switches for each light.
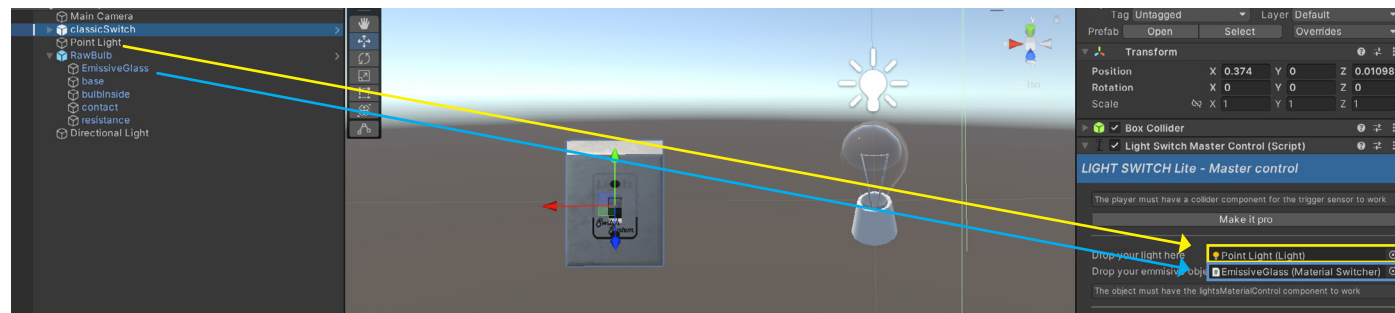
► Quick Start

1- Choose your preferred switch and lamp prefabs, and drop them on the scene

2- Add 3D light close to the lamp.

3- In the "Light Switch Master Control" component of the switch, add the 3D light and the object that contains the emissive material (the one with the script" MaterialSwitcher") on its respective places, so they be associated to the switch.

Great! You're ready to go. Now every time the player press the button "Fire1" on the trigger area the light will toggle on and off.
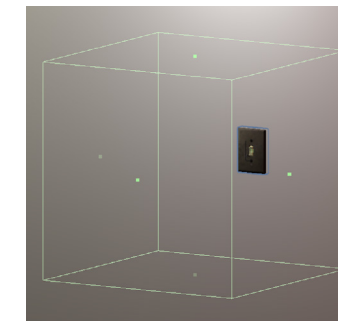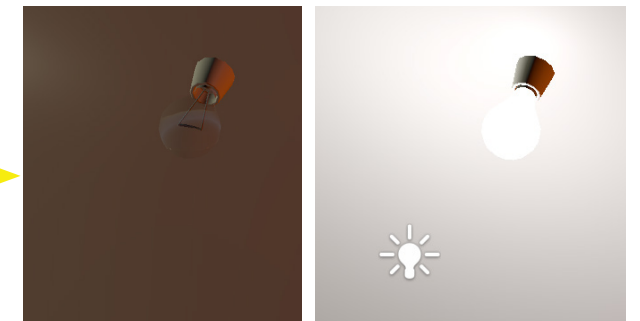

Try the demo scene here

► How it works

If you plan to use your own switch models this are the concepts you should know to integrate them with the scripts. To make the complete system work you'll need 3 objects: the Switch, a 3D light and a lamp with an emissive material,* and also 3 scripts.

- **The light system workflow**

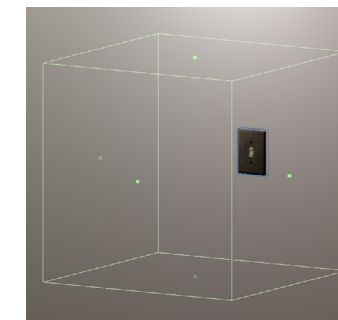The player inside the sensor presses the "fire 1 button"

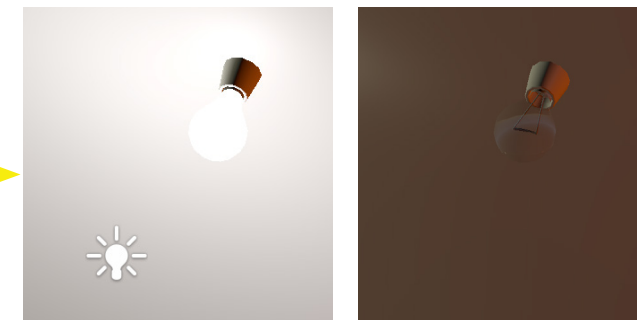The linked light component and emissive material will turn to on mode

At the start of the game all switches will associate with the lights they are connected and give them a script that allows the light to know how many associated switches it has and on/off requests.

When the player gets in the sensor area and presses the input button on a turned off switch, the toggle will move to the on position and it will send a on request. If its associated light has **all its linked switches on,** it will turn on*. Then the light component will be enabled and the emissive material will be set to the On state material. When is off, the opposite will happen.

The Switch should contain a collider in trigger mode that will act as sensor and let the system know if the player is trying to interact with the switch, if the player is outside it wont react to the player input.
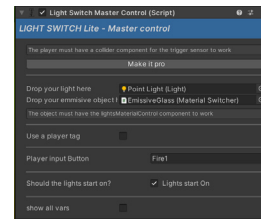

If player inside the sensor presses the "fire 1 button" again

The linked light component and emissive material will turn to off mode

The light and emissive materials should have all it's switches on to be able to turn on, for example if one light has two switches, it will turn on only if both switches are on. If at least one of them turns off. The light will turn off too.

► Light Switch Master Control

This is the most important component of the light switch, here you can control how the system is going to work.

With the "make it pro/lite" button you can choose between 2 Lite and pro modes. The lite mode controls one light and one emissive material. Ideal to make a fast and simple setup.

If you choose pro mode you can control many lights with one switch. You can group them in different light kits to keep your project organized. Every light kit can have many lights and many emissive materials.
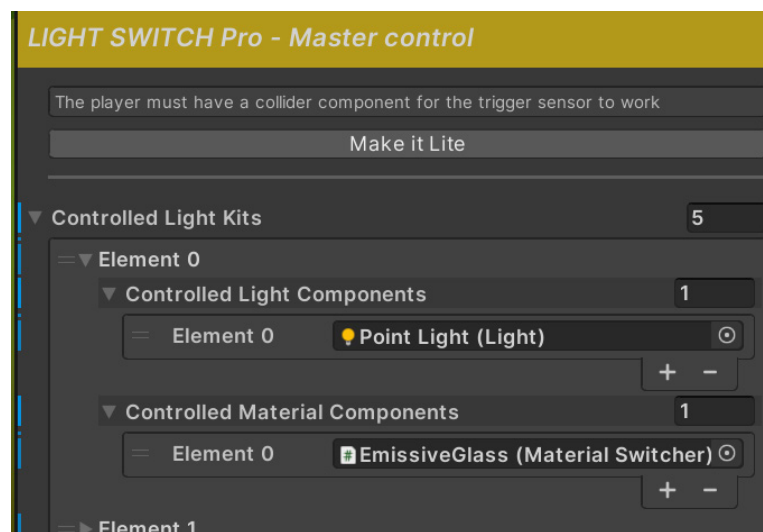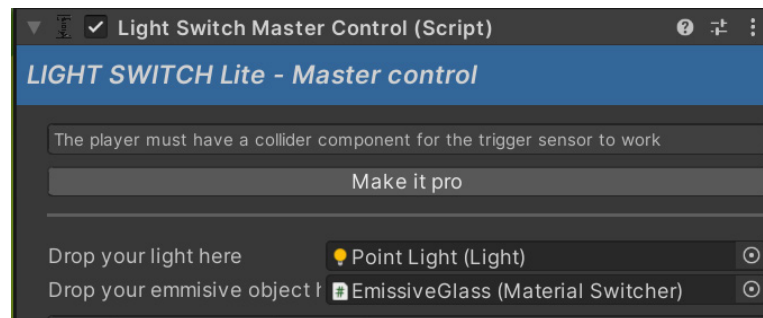
You can also link each lights to different switches. In that way you can have a master switch that controls all lights and each light with its own individual switch.

In order to turn on all the associated switches must be turned on. To turn off at least one of the associated switches must be turned off

Check this box to make the sensor react just when the tagged player is there and prevent other objects to activate it.

Here you can choose the player input button used to interact with the lights. This will work on legacy or mixed input systems.

If you are using the new input system, you could call the function *"SwitchLightsCheck (bool skipSensor)"* from the LightSwitchMasterControl of the switch you want to toggle. The bool value lets you skip the sensor if it's true.
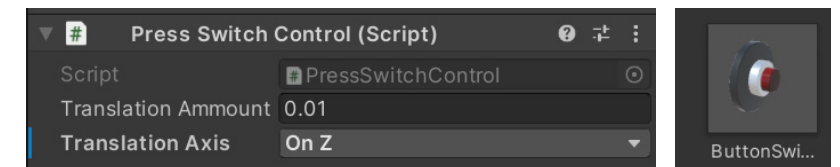
Here you can select the lights initial state.

The lamp and the 3D light are optional and you can choose between one of this elements or both.
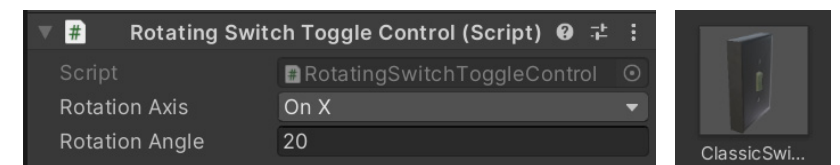
► Secondary scripts

• Light Switch Toggle Control

This script should be located in a child Game Object of the Game Object that contains the Master Control. It function is to control the moving part of the switch, the parent class handles the on/off signal and the child classes controls the reactions of the switch. There are 4 different types:
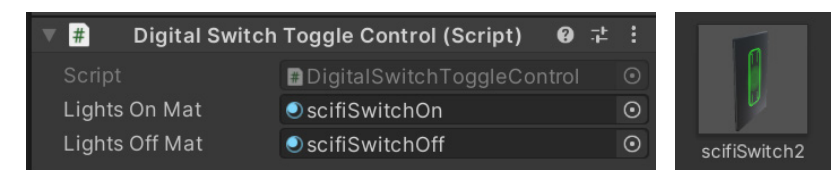
Press switch control, for buttons or sliders. The translation amount controls the distance it's going to move between the on and off position, the translation axis let's you sellect the direction.
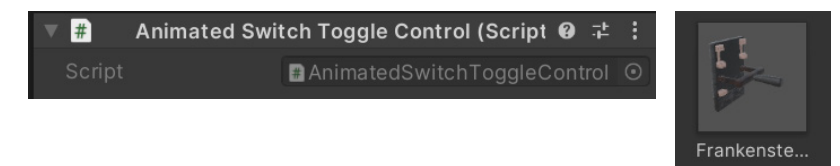

Rotating Switch Toggle Control for, classic switches and levers. The translation axis let's you select the rotation orientation. The rotation angle will set the orientation of the On and Off position positive and negative respectively, on the image would be +20 for On and – 20 for Off

Digital Switch Toggle Control for switches with emissive materials. It will change the material with an On and Off state

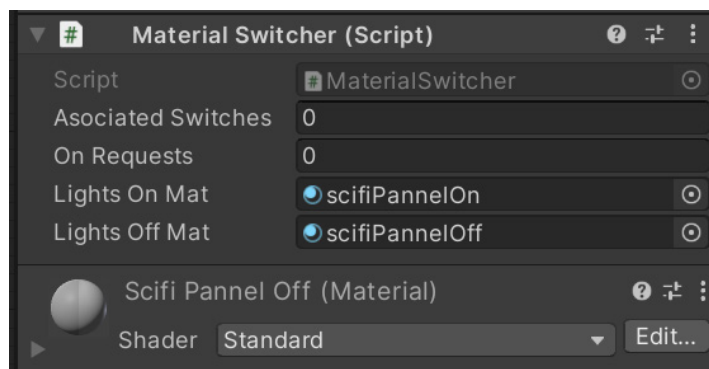**Animated Switch Toggle Control** for Switches that includes animations. The Animator component should contain in its parameters, a bool that trigger the on and off respective animation states

You can create you own Toggle class, to do it just have to inherit from the "SwitchToggleControl " class. Then implement the "changeState(bool toggle)" function, you'll become a true and false value for On and Off respectively.

• **Material Switcher**

This is the script you should add to the emissive part of the lamp if you have one. The "Associated switches" shows how many light switches are linked to this object.
The on requests shows how many linked switches are on.
The "Lights On Mat" is where you should place your On state material.
The "Lights Off Mat" is where you should place your Off state material.

Material Switcher (Script)		
Script	MaterialSwitcher	
Asociated Switches	0	
On Requests	0	
Lights On Mat	scifiPannelOn	
Lights Off Mat	scifiPannelOff	
Scifi Pannel Off (Material)		
Shader	Standard	Edit...

► **Recap**

• Add the "Light Switch Master Control" to your switch
• Add the "material switcher control" to the emissive part of your lamp with an On and Off material
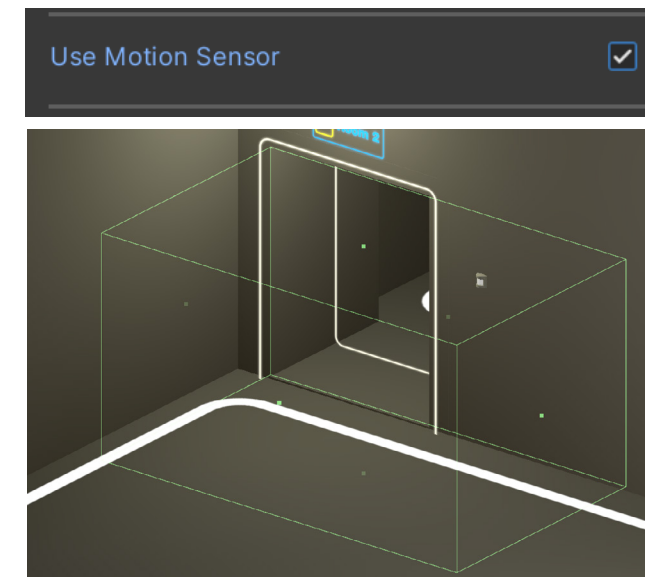• Link the light and the emissive part of the lamp to the switch.

► **Updates V 1.1.0**

• **New Function, Motion Sensor**

This option allows you to turn on or off lights when the player enters a trigger collider.
The light will switch their state when the game object with the chosen tag enters the area defined by the collider. If you start with the light off, the sensor will turn the lights on but if you start with the lights on the sensor will turn the lights off. Very usefull for horror games where you want to scare the player by turning the lights off when entering a certain area
There is also a new sensor 3D model included.

Use Motion Sensor ☑

★ ★ ★ ★ ★

Lights *Switch System*

★ ★ ★ ★ ★

If you like the product please remember to leave a positive review on the asset store, that would help me a lot to improve this asset.
Here you can take a look at my other assets :)

https://assetstore.unity.com/publishers/77287

Terms of use

• You may use this asset in your personal or commercial projects.
• You may not sell this asset or modify it in any way to resell it as similar asset.
• The customer support will be available as log as long as the source code is not modified. Anyways the code is documented so intermediate and advanced users will be able to modify it with ease if needed.
• You can find the full Terms here https://unity.com/legal/as-terms