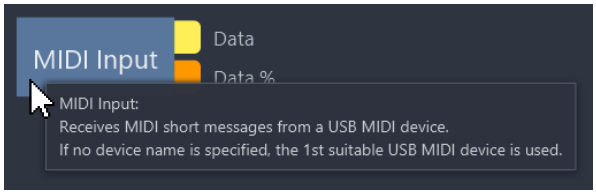
buster .lighting

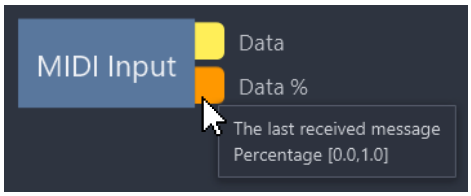
User Manual v1.3.1

Table of Contents

1 Introduction

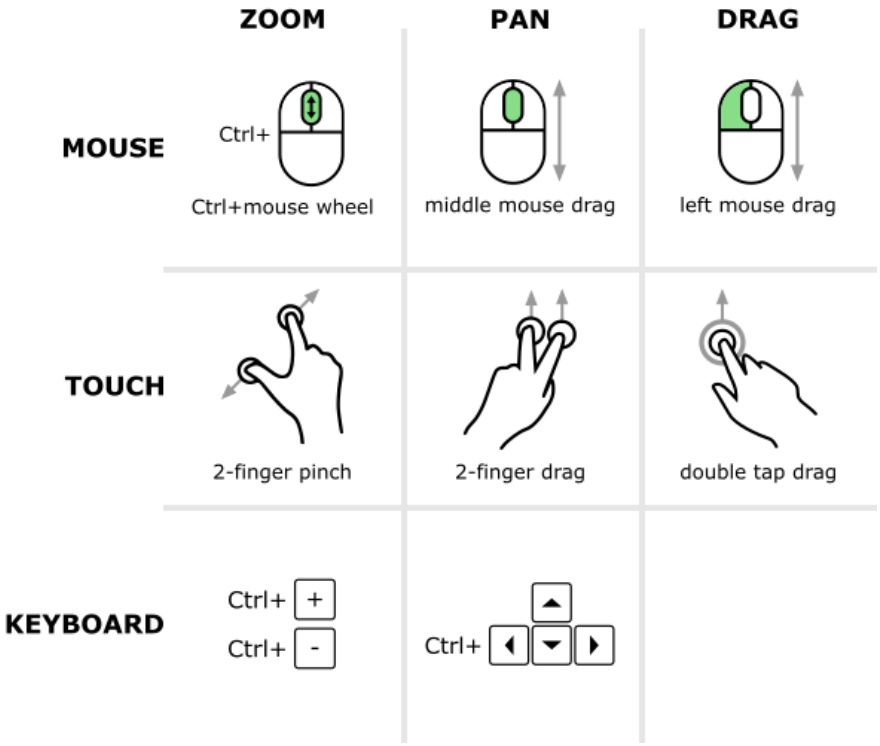
This manual is intended as an introduction to the Buster products. It explains the software's main principles and describes some tips and pitfalls, so you can get experimenting quickly and efficiently. Instead of describing every functionality in this document, a lot of documentation was added to the software itself: for example, if you hover the mouse cursor over blocks or their ports, messages pop up explaining their purpose and usage.





If you find documentation lacking, if you experience a problem, if you encounter a software bug, or if you have an idea for an improvement or a new feature, please let us know. You can register your issues at the request and bug tracker, reachable via https://buster.lighting.

Buster software supports zooming, panning, and dragging with mouse, touchpad, touchscreen, and keyboard. Here is an overview of the different gestures:



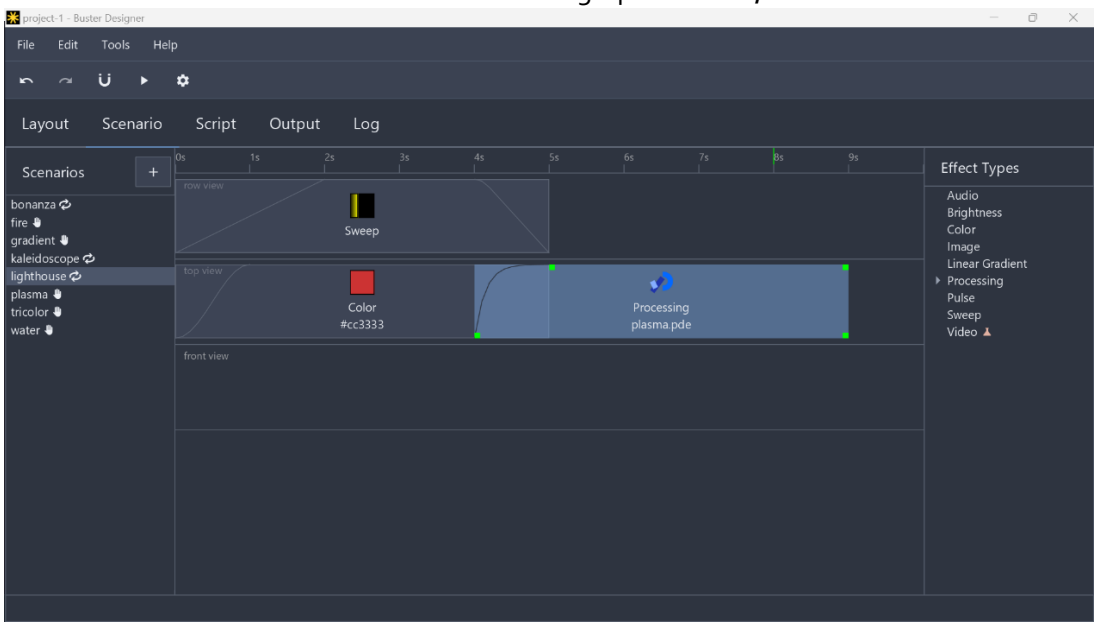
2 Overview

Buster Lighting is software for dynamic lighting:

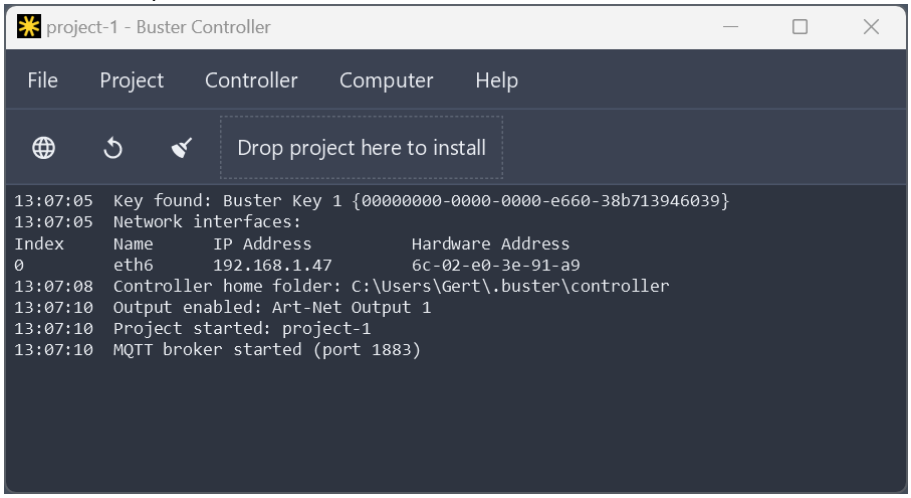
- Create 2-dimensional layouts for your fixtures, pixels and LED strips.
- Add dynamic lighting scenarios that can run simultaneously and independently.
- Automate scenarios with alarms, calendars, astronomical events …
- Control scenarios with buttons and sensors, network protocols, a web interface …
- Design, simulate and run your lighting project with a single app.
- Use industry standard communication protocols: Art-Net, sACN, MQTT, MIDI …

Buster Lighting comprises following products:

- Buster Designer: desktop software to design, simulate and run a new lighting project, or control an occasional installation. It has various graphical tools, like a built-in simulator.



- Buster Controller Desktop: desktop software to simply run a lighting project. The Controller has no design or simulation capabilities, but is less demanding on the computer hardware than the Designer. Perfect for stand-alone operation, when the computer may not have a screen or keyboard attached.

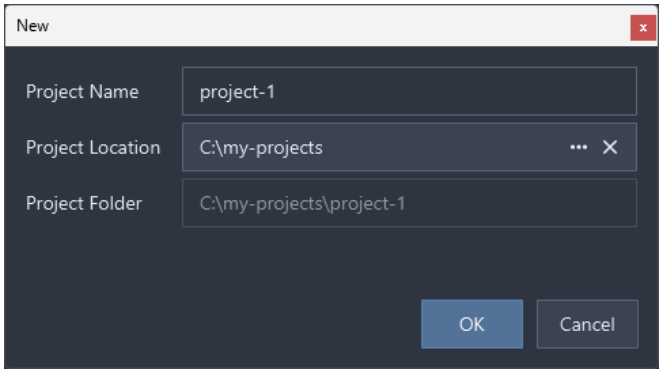


- Buster Controller Service: a non-graphical alternative to Buster Controller Desktop, operating as a background program. It does not have graphics hardware support though, due to restrictions in the Windows operating system.

- Buster Key: a small USB device that enables Buster software to communicate with your lights. To download and test the software, or design and simulate a project, a Key is not required. However, to actually drive your lights, you'll need to purchase a Key and attach it to the computer that the Buster software is running on.

3 Buster Designer

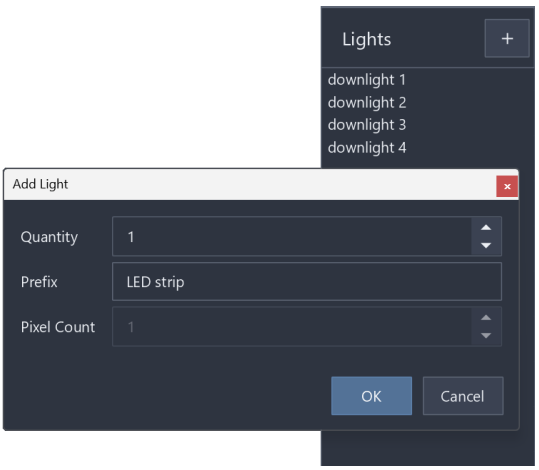
When creating a new lighting project, the Designer will immediately ask you where to save it on your computer. A project is not saved as a single file, but as a folder, containing an essential project.xml file. As your project grows, more files (like images or configuration files) may be added to this project folder.



3.1 Lights

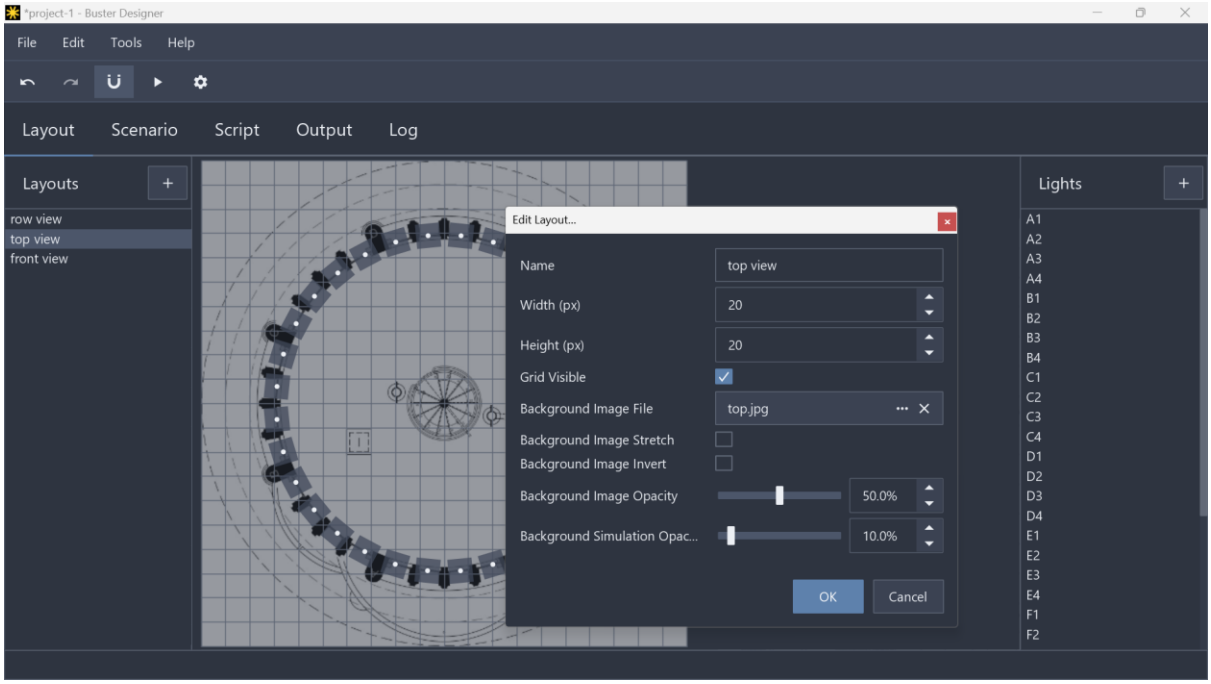
For every individually addressable light source (or pixel) in your lighting installation, add a light to your project. Note that:

- Lights are technology-agnostic: at this stage there's no need to specify whether a light is an incandescent bulb, fluorescent lamp, LED strip … or whether it uses DMX, Art-Net, sACN … for communication.
- Lights are full-color-capable by design: if your actual, physical light sources cannot show all colors, don't worry about that right now. You'll be able to configure it when adding outputs to the project.
- Lights are currently single-pixel devices. In a future software release, lights will be able to have multiple pixels, making it easier to model pixelized LED strips.



3.2 Layouts

A layout is a rectangular, 2-dimensional view of the lighting installation. You can have many layouts in your project, for example a front view, a top view, or a view per room in a building. A layout is basically a grid of specific width and height, on which you can arrange your lights.

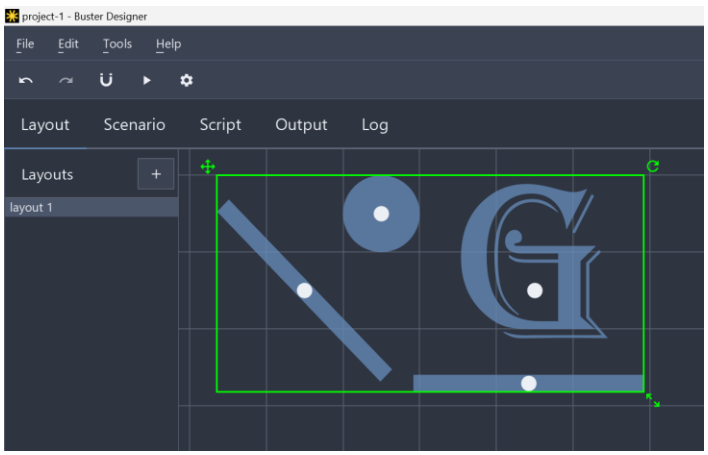


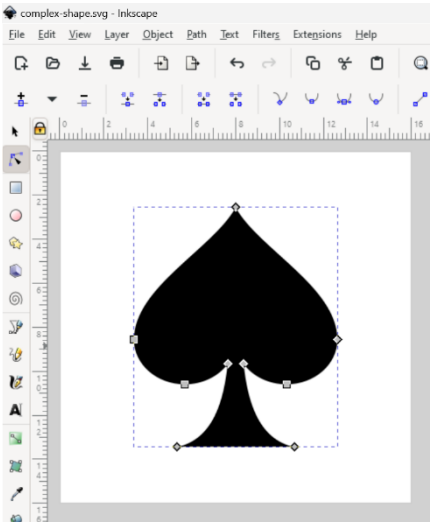
You should choose the size of a layout based on the number of lights it will contain. On the layout shown above for example, 32 lights are arranged in a circle.

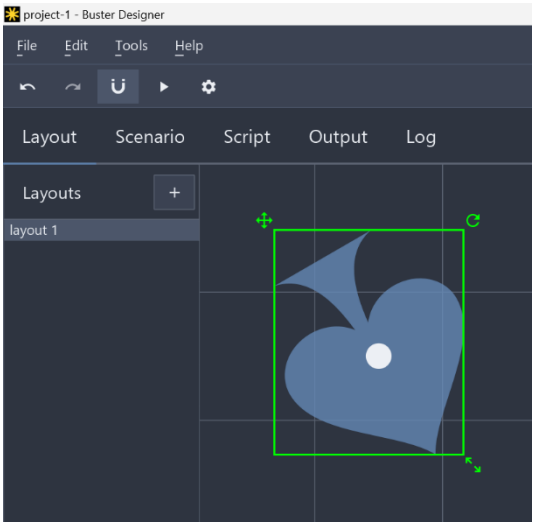
- With a layout size of 20 x 20, all lights occupy a different 'square' in the grid, which is probably what you want.
- A size of 10 x 10 (lower resolution) would result in bigger squares. Lights with anchors (the bright, central dot) in the same square, will always have the same colors. So, in most cases you'll want to avoid lights sharing the same square.
- A size of 1000 x 1000 (higher resolution) would take up more computer memory, without any real advantages.

You can use an image (an architectural drawing, for example) as a layout background. A background image makes it easier to arrange your lights, and makes simulations more realistic and more convincing.

When a light is dragged and dropped onto a layout, it will get the default simple, rectangular 1 x 1 shape. That shape is customizable, though. For example, you could resize to make it look like a thin, rectangular LED strip, a circle (ellipse), or a character (text). You could even give it an arbitrarily complex shape with an SVG path (saved to an SVG file).



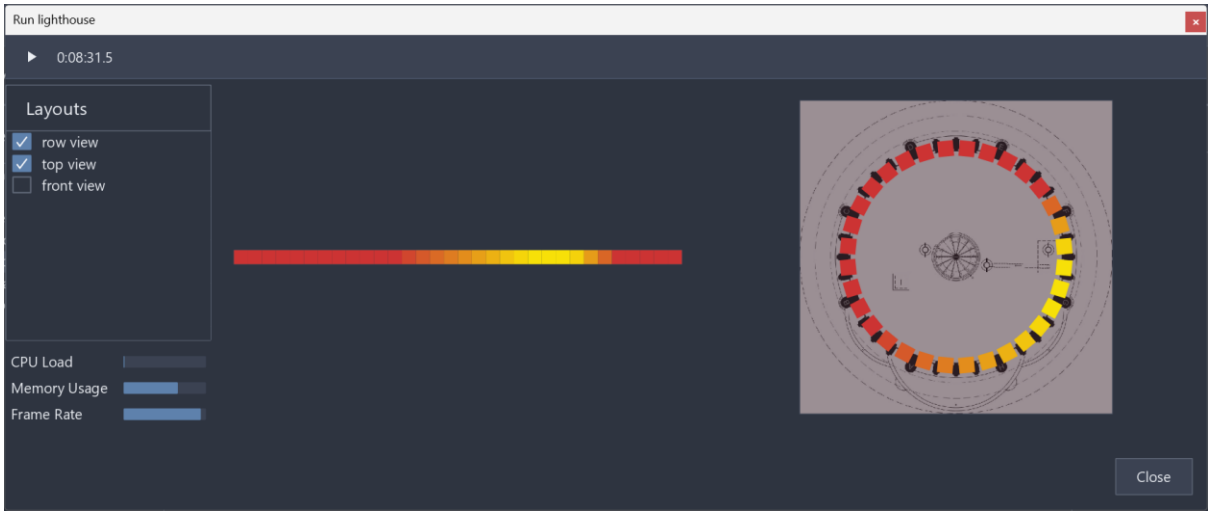




A simulation can of course benefit from custom shapes. However, a light's color is not determined by its shape, but solely by the position of its anchor.

A light can be placed on many layouts, for example on a front view and on a top view. On each layout, the light can be given a different shape.

Layouts do not necessarily need to correspond to the physical reality. In the following image, notice how in reality the lights are set up in a circle. However, adding a layout that arranges them differently (in a single row, for example) can create interesting possibilities, as you'll learn in the next chapter.

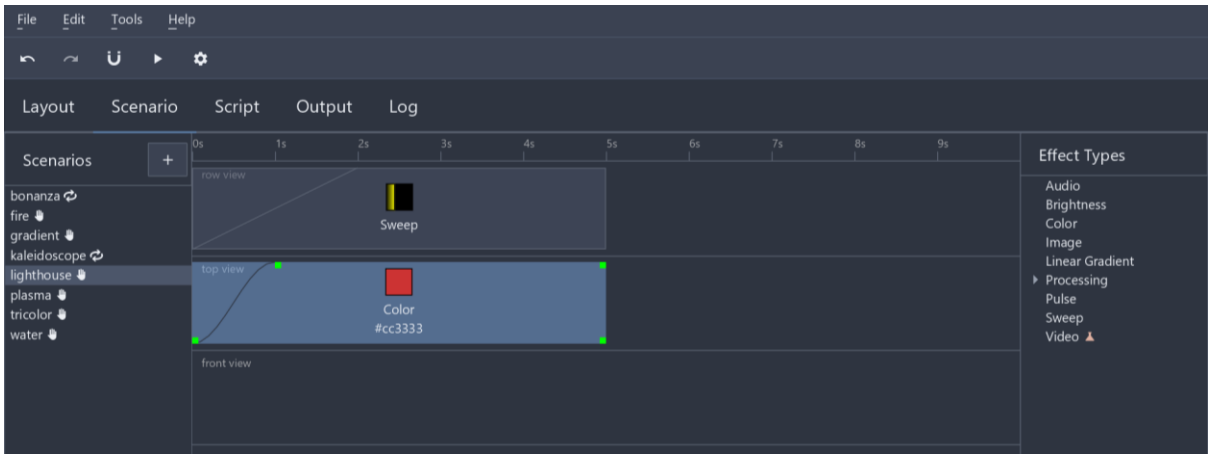


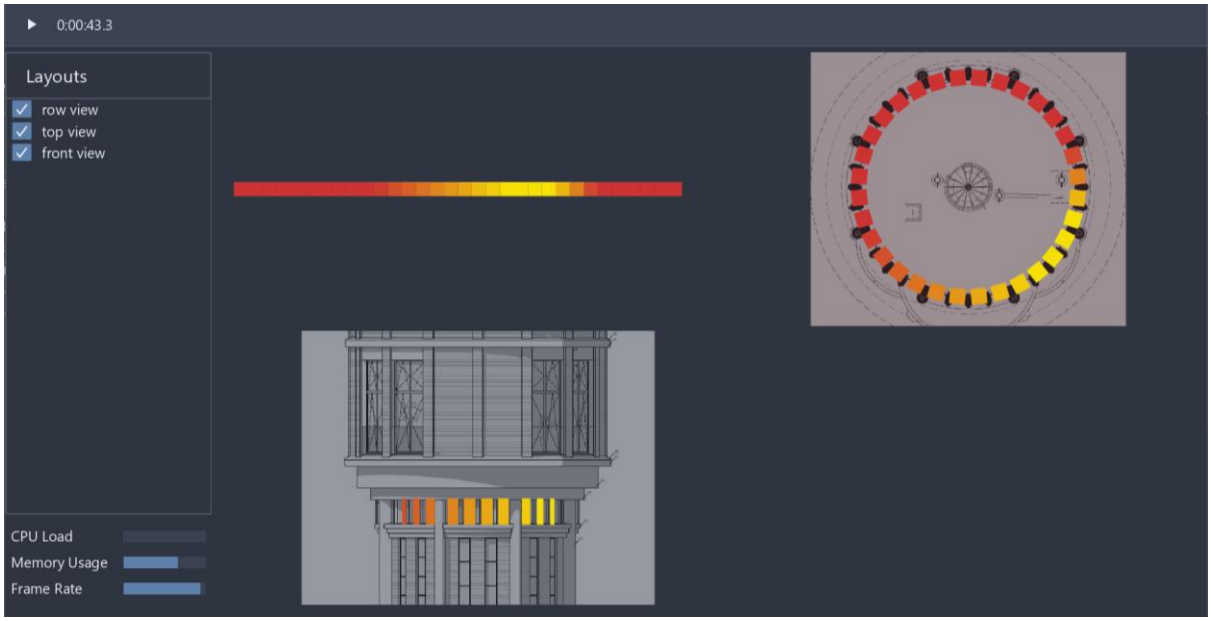
3.3 Scenarios

A scenario is a sequence of lighting effects. When such an effect is running, it draws stuff on a specific layout and generally animates it. The lights arranged on that layout, will simply pick up its changing colors.

A scenario provides tracks, on which effects can be placed. You get a track for each layout in the project. For example, the following image shows a scenario named 'lighthouse'. When it is started, it will run a color effect (red) on the 'top view' layout and a sweep effect (yellow) on the 'row view'

layout. Note that there are no effects on the 'front view' track. When you right-click 'lighthouse' and choose 'Run', a simulation is started.





This scenario paints the row view partly yellow and the top view completely red. All lights are on both layouts, so which color will they adopt? The software actually blends effects by these rules:

1. Effects on different tracks are blended according to the layout list order: the row view is above the top view, so the yellow sweep effect appears on top of the red color effect, partly covering it up.
2. Effects on the same track are blended according to their start time: effects starting later will appear on top of those started earlier.

So, if you would move the top view above the row view, then the red color effect would appear above the yellow sweep effect, covering it up.

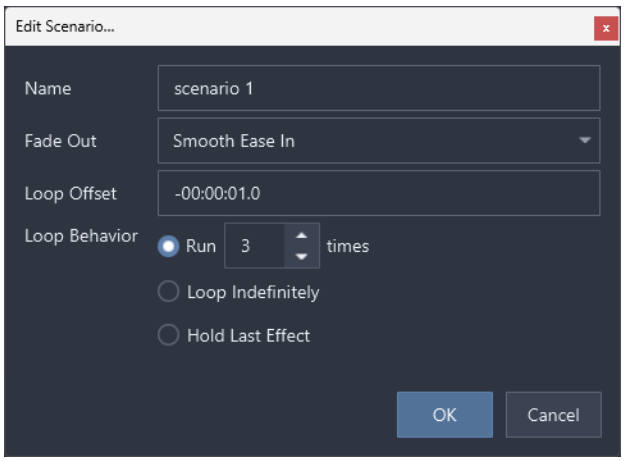
Also note that the sweep effect would affect the lights quite differently if it were active on another layout. This sweep effect animates a vertical line across the row view: viewed on the top view however, the line appears to move around in circles. This shows that layouts and effects can be combined creatively to achieve surprising results.

3.3.1 Life cycle

Effects can be dropped on the track of a scenario, dragged around and resized. The effect's timing is determined by four points in time, visualized by green handles:

A scenario can have many effects on many layouts. By default, a scenario runs all its effects once and then ends, but you can actually change that end behavior:

- Run a specified number of times.
- Loop indefinitely.
- Hold the last effect.



When a scenario is set to loop, it will start again at its (end point + loop offset). For example, in the images above, the scenario ends after 15 seconds. Taking into account the loop offset of -1 sec, it will start its next iteration after 14 seconds. This is visualized by the green line in the time bar.

When the scenario is set to 'hold its last effect', it will determine its last effect release point and prevent all effects with that release point from fading out. In other words, the scenario will keep its last effects running. This is useful to statically illuminate a room, for example.

3.3.2 Notable effects

3.3.2.1 Image & video effect

An image effect displays a (scaled) image on a layout. If you have a drawing of your lighting installation, you could colorize it with a drawing program[1] and load it into Buster Designer. The image effect also accepts animated GIFs, that you can easily find online or create yourself[2].

[1] For example, with Gimp or Adobe Photoshop
[2] For example, with lottiefiles.com or Adobe Illustrator.

A video effect displays a (scaled) video on a layout. Take care to use videos that are small enough, because long, high-resolution videos will make your project take longer to save, upload, and start. If your original video is big, you can use an editor to scale it down and trim it[3].

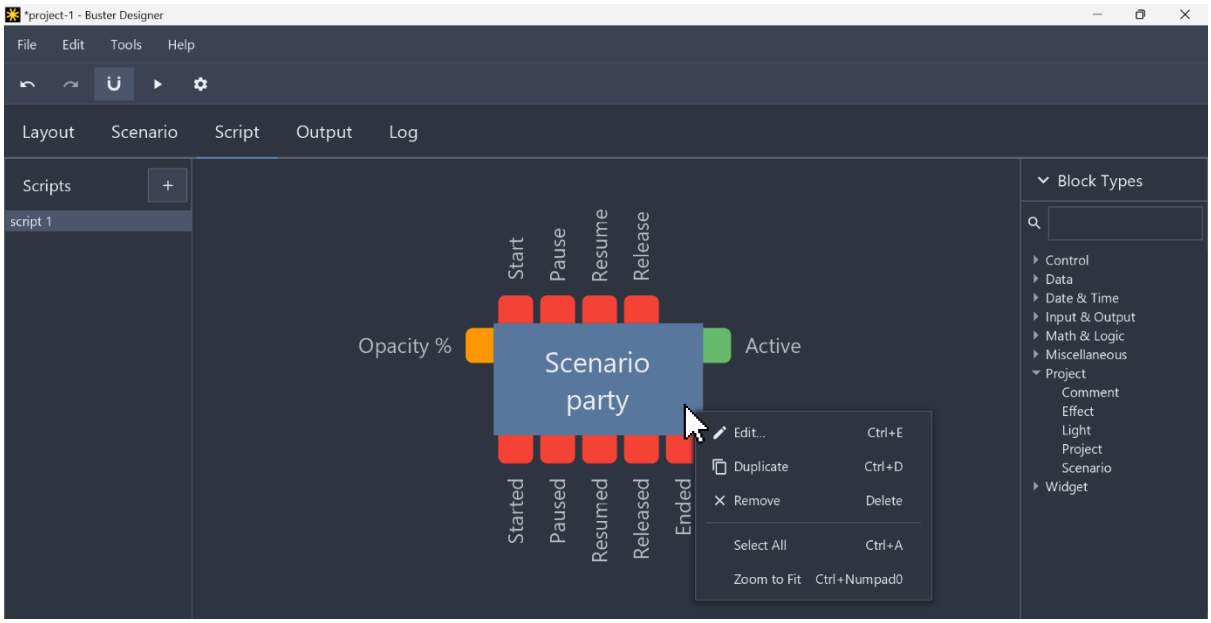
3.3.2.2 Processing effect

Processing is a coding language to create visual art. Although there is a learning curve to use this effect, its possibilities are endless. Buster Designer includes a few templates for the Processing effect. You can find more inspiration and many examples online and in books.

3.4 Scripts

Buster software lets you add control and automation to your project with scripts. Scripts are graphical programs that determine your lighting installation's behavior. For example, you could use them to:

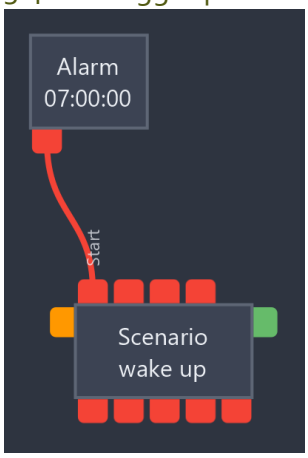
- Start, pause, or stop a scenario. Multiple scenarios can run simultaneously and independently.
- Set up an alarm or react to a sunset.
- Receive input signals, for example from a physical button, a temperature sensor, or a network message.
- Transform signals with math and control logic.



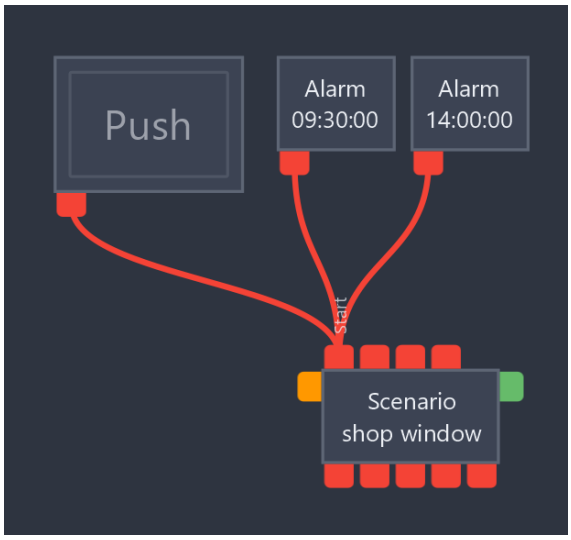
A script offers you an area to add, arrange and interconnect blocks. In the image above, a scenario block was added to 'script 1'. With this block, the 'party' scenario can be controlled and monitored. You can edit any block to change its properties, in this case to change the scenario it controls, for example.

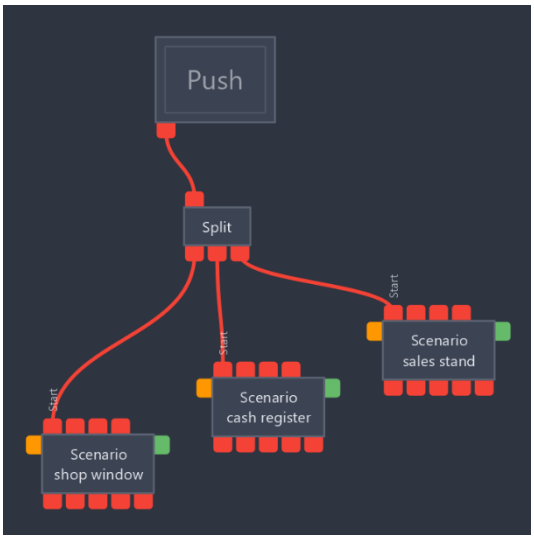
[3] For example, with ClipChamp, OpenShot, or DaVinci Resolve

3.4.1 Trigger ports



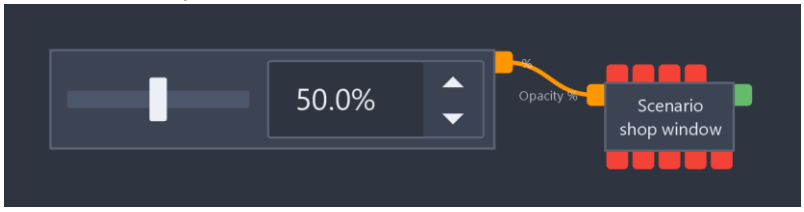
A scenario block has red ports at the top and at the bottom. The top ports are inputs and can accept triggers. The bottom ports are outputs and can generate triggers. Consequently, a red bottom port can be connected to a red top port. Triggers are signals that can travel through a red wire, so generally vertically, from the top to the bottom.





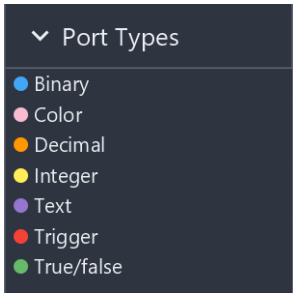
A top port can accepts many wires. A bottom port however, can only have a single wire attached. If you want to send a trigger to multiple top ports, you'll need a split block to distribute it (left to right).

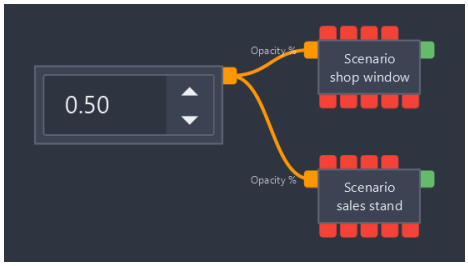
3.4.2 Value ports

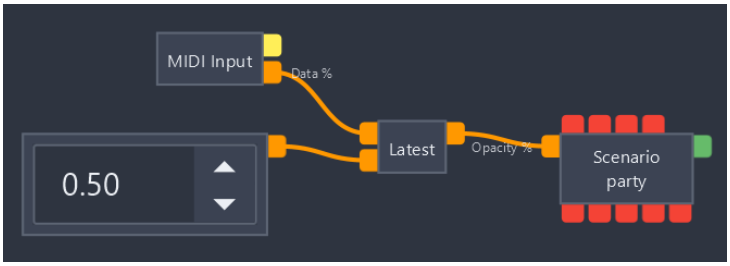


A scenario block also has ports on the left and right side: the left ports are inputs and the right ports are outputs. Note that the left and right ports come in different colors. A left port and right port can only be connected if they have matching colors. In the example above, an orange right port is connected to an orange left port to transport values (more specifically decimal values).

Binary values use blue wires, color values use pink, decimal values use orange, integer values use yellow, text values use purple, and true/false values use green wires. All values travel generally horizontally, from left to right.



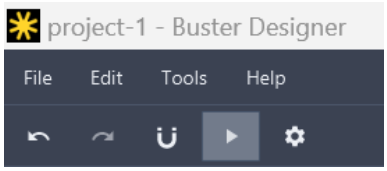




A right port can have many wires attached. A left port however, can only accept a single wire. In the above example, a latest block will select the incoming value that changed last, either from a incoming MIDI message or the decimal spinner. But you could also select the highest value with a maximum block, or compute a sum with an add block, or …

3.4.3 Simulation

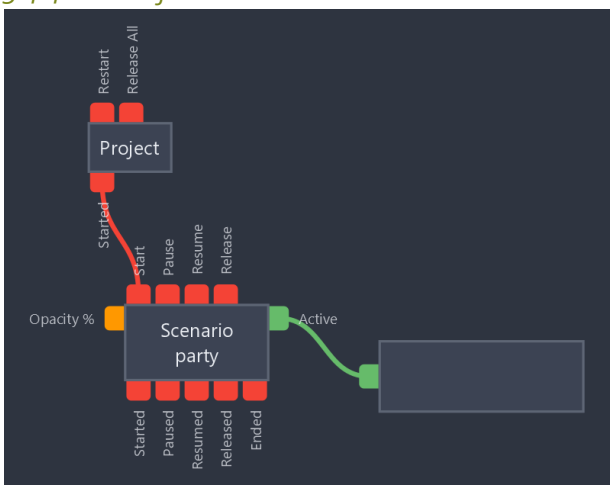
As explained previously, you can start scenario simulation from the scenarios list on the 'Scenario' tab. However, if you want to simulate a script, you need to run the complete project (in contrast to a single scenario). You can start project simulation with the play button in the Designer's upper toolbar. Under the hood, this will fire up a full-blown Controller.



When simulating the project, all your scenarios and script are available. Scripts are constantly evaluated (approximately 60 times per second). Scenarios are started only when requested from a script, or from the Controller web interface. On the left side of the simulator window, you can show or hide the layouts and scripts that are of interest to you, by (un)checking them.

3.4.4 Notable blocks

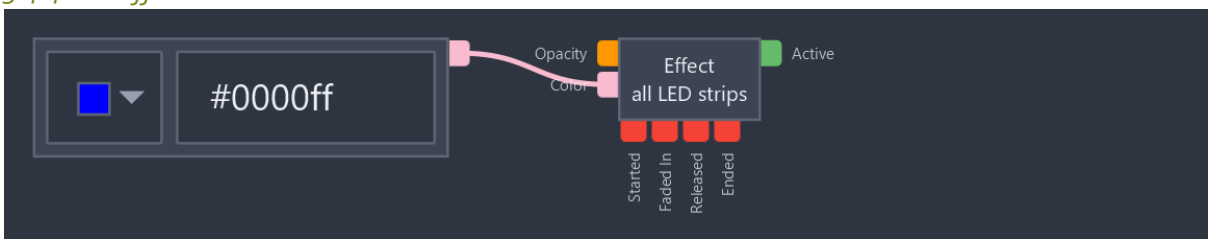
3.4.4.1 Project and scenario block



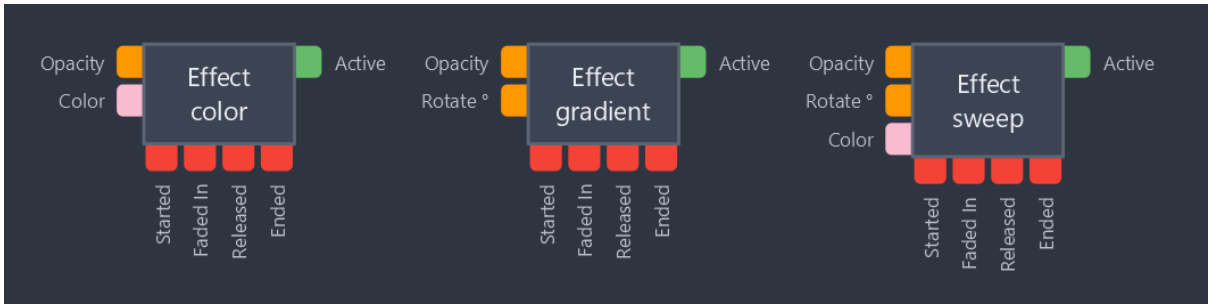
The scenario block has ports to control and monitor a specific scenario: you can feed it triggers to start, pause, resume, and release it (start fading it out). Because a scenario can be controlled from many places in many scripts, but also from the web interface, this block also provides bottom ports that output triggers when the scenario's state changes. Moreover, the right port will output a 'true' value while the scenario is actually running.

If you want to start a scenario immediately when the projects starts, you can connect a scenario block to a project block. The project block also provides a handy top port to release all running scenarios. This may be handy when all lighting needs to be turned off.

3.4.4.2 Effect block

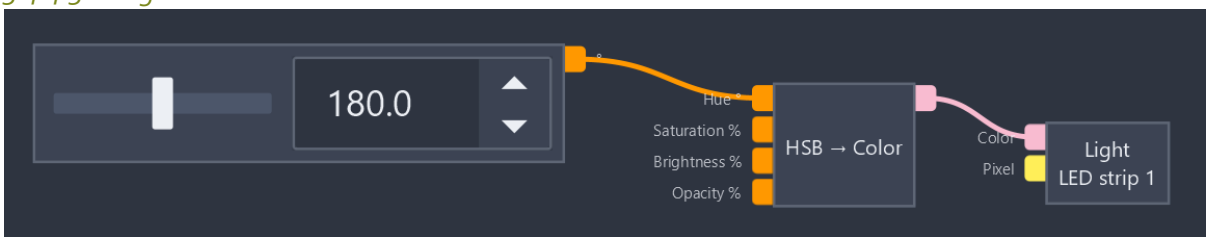


The effect block has ports to control and monitor a specific effect (on a specific track and scenario). In the example above, 'all LED strips' is a color effect that probably colorized a layout that holds all the projects LED strip lights. On the scenario tab, you are limited to setting a single, fixed color for a color effect. But in a script, you can actually dynamically change the effect's color, for example with a color (chooser) block.



Depending on the referenced effect type (color effect, linear gradient effect, image effect ...), an effect block may show effect-specific ports. With these you can make your project even more dynamic and interactive.

3.4.4.3 Light block



Scenarios and effects offer a powerful way to create complex animation for your lighting installation. But sometimes you just want to control a specific light, for example when you are setting up a new lighting installation.

In the above example, a slider (degrees block) outputs decimal values, that are transformed into color values and fed into a light block. The light block specifically controls 'LED strip 1'. With this configuration, you can easily traverse all possible color hues, to determine the best one for your LED strip.

If a light block and a scenario should both try to control the same light, the scenario will win. In Buster software, scenarios/effects have a higher priority than scripts/blocks.

This is a work in progress: more to come soon!