

---

# **PatchXR**

***Release 26/12/2020***

**Nicholas Borkowski**

**Aug 31, 2022**



# DOCUMENTATION CONTENTS:

<b>1</b>	<b>How to Install</b>	<b>1</b>
<b>2</b>	<b>Blocks</b>	<b>3</b>
2.1	Audio	3
2.1.1	allpass2	3
2.1.2	decay	3
2.1.3	envelope_ad	4
2.1.4	highpass	4
2.1.5	integrator	4
2.1.6	ladder	5
2.1.7	limiter	5
2.1.8	lowpass	5
2.1.9	mic	6
2.1.10	microphone	6
2.1.11	noise	6
2.1.12	onepole	7
2.1.13	oscillator	7
2.1.14	pink_noise	7
2.1.15	pulse	7
2.1.16	reverb	8
2.1.17	reverb_tank	8
2.1.18	sample_and_hold	8
2.1.19	statevariable	9
2.1.20	vel	9
2.1.21	wavefolder	9
2.2	Bubble samplers	9
2.2.1	bubble_popper	9
2.3	Connections	10
2.3.1	elbow	10
2.3.2	extension	10
2.3.3	fork	10
2.3.4	get	10
2.3.5	msg_in	11
2.3.6	msg_out	11
2.3.7	output	11
2.3.8	pass_in	11
2.3.9	pass_out	12
2.3.10	spacer	12
2.3.11	split	12
2.3.12	stoe	12

2.3.13	wireless_in	13
2.3.14	wireless_in_jolt	13
2.3.15	wireless_out	13
2.3.16	wireless_out_jolt	13
2.4	Deprecated	14
2.4.1	alu	14
2.4.2	digitalwaveguidestring	14
2.4.3	freeverb	14
2.4.4	ntor	15
2.4.5	ntou	15
2.4.6	outstate	15
2.4.7	rms	15
2.4.8	rton	16
2.4.9	rtou	16
2.4.10	speaker_2D	16
2.4.11	speaker	16
2.4.12	uton	17
2.4.13	utor	17
2.5	Donotinclude	17
2.5.1	cmd	17
2.5.2	ghost	18
2.5.3	linearladder	18
2.5.4	lowpass_old	18
2.5.5	m_add	18
2.5.6	pass	19
2.5.7	sample_player_cut	19
2.5.8	sample_player	19
2.5.9	sample_recorder	19
2.5.10	string_with_output	20
2.5.11	toggleboardcmd	20
2.6	Experimental	20
2.6.1	ableton_link	20
2.6.2	bar	20
2.6.3	chain	21
2.6.4	frequency_analyzer	21
2.6.5	global_reverb	21
2.6.6	hammer	21
2.6.7	karplus_strong_extended	21
2.6.8	karplus_strong	22
2.6.9	list	22
2.6.10	pitchshifter	22
2.6.11	plate	22
2.6.12	pluck	23
2.6.13	s_reader	23
2.6.14	sample_info	23
2.6.15	saturator	23
2.6.16	violin	24
2.7	Interface	24
2.7.1	back_button	24
2.7.2	button	24
2.7.3	controloutput	25
2.7.4	custom_slider	25
2.7.5	execute	25
2.7.6	ghost_looper	26

2.7.7	hapticcontrol	26
2.7.8	help_button	26
2.7.9	interaction_box	26
2.7.10	keyboard	27
2.7.11	knob	27
2.7.12	knobboard	28
2.7.13	laser_canvas	28
2.7.14	laserboard	28
2.7.15	laserkey	29
2.7.16	next_button	29
2.7.17	nknob	29
2.7.18	pad	30
2.7.19	peppermill	30
2.7.20	play_button	30
2.7.21	presetmanager	30
2.7.22	readout	31
2.7.23	reset_button	31
2.7.24	skip_button	31
2.7.25	slider_3d	31
2.7.26	slider_db	32
2.7.27	slider	32
2.7.28	sliderboard	32
2.7.29	stepped_dial	33
2.7.30	toggle	33
2.7.31	toggleboard	33
2.7.32	uknob	33
2.7.33	value	34
2.8	Kinetic	34
2.8.1	anchor	34
2.8.2	ball	34
2.8.3	block_distance	34
2.8.4	block_get_look	35
2.8.5	block_lookat	35
2.8.6	block_relative_position	35
2.8.7	block_set_position	35
2.8.8	box	36
2.8.9	funnel	36
2.8.10	handle	36
2.8.11	m_booster	36
2.8.12	m_compare	36
2.8.13	m_path	37
2.8.14	m_reader	37
2.8.15	m_spawner	37
2.8.16	m_switch	37
2.8.17	marble	37
2.8.18	octopus	38
2.8.19	octopus_inv	38
2.8.20	rail	38
2.8.21	rotor	38
2.8.22	string	38
2.8.23	wall	39
2.9	Lights	39
2.9.1	hsvtorgb	39
2.9.2	lightbulb	39

2.9.3	sunlight	40
2.10	Logic	40
2.10.1	bernoulli	40
2.10.2	compare	40
2.10.3	counter	41
2.10.4	cyclecounter	41
2.10.5	data_array	41
2.10.6	gate	42
2.10.7	if_else	42
2.10.8	nth	42
2.10.9	rangedfor	43
2.10.10	sequence	43
2.10.11	sequence_loop	43
2.10.12	sequence_random	44
2.10.13	watcher	44
2.11	Marble rain	44
2.11.1	marbleslide	44
2.11.2	marbletube	45
2.12	Math	45
2.12.1	abs	45
2.12.2	add	45
2.12.3	add_jolt	45
2.12.4	clamp	46
2.12.5	constant	46
2.12.6	cos	46
2.12.7	divide	47
2.12.8	divide_jolt	47
2.12.9	floor	47
2.12.10	floor_jolt	48
2.12.11	from_db	48
2.12.12	log	48
2.12.13	log_jolt	48
2.12.14	map	49
2.12.15	max	49
2.12.16	midi_to_freq	49
2.12.17	midi_to_freq_jolt	50
2.12.18	min	50
2.12.19	modulo	50
2.12.20	modulo_jolt	50
2.12.21	multiply	51
2.12.22	multiply_jolt	51
2.12.23	power	51
2.12.24	power_jolt	52
2.12.25	previous	52
2.12.26	random	52
2.12.27	round	52
2.12.28	sign	53
2.12.29	sin	53
2.12.30	subtract	53
2.12.31	subtract_jolt	54
2.12.32	tanh	54
2.12.33	to_db	54
2.13	Other	55
2.13.1	palette	55

2.14	Others	57
2.14.1	block_watcher	57
2.14.2	trigger_box	57
2.15	Props	57
2.15.1	particles_sprite	57
2.15.2	prop_firstmask	58
2.15.3	prop_neuron	58
2.15.4	prop_particles	58
2.15.5	prop_underwaterparticles	58
2.15.6	prop_wireframe	59
2.16	Samples	59
2.16.1	sample	59
2.17	Sky	59
2.17.1	venue_lightsaber	59
2.17.2	venue_mountains	59
2.17.3	venue_ozelot	60
2.17.4	venue_spacestation	60
2.17.5	venue_underwater	60
2.18	Stages	60
2.18.1	prop_highplatform	60
2.18.2	prop_platform	61
2.18.3	prop_swimmingpool	61
2.19	Terrains	61
2.19.1	venue_wireframe	61
2.20	Time	61
2.20.1	beat_time	61
2.20.2	clock	61
2.20.3	delay	62
2.20.4	delay_jolt	62
2.20.5	eggtimer	62
2.20.6	metronome	62
2.20.7	rhythm	63
2.20.8	timecontrol	63
2.21	Unclassified	63
2.21.1	prop_neurons	63
2.21.2	venue_brain	64
2.22	Venue	64
2.22.1	bg_nebula	64
2.23	Visual	64
2.23.1	camera	64
2.23.2	cameramanager	65
2.23.3	coloroven	65
2.23.4	device_box	65
2.23.5	image	65
2.23.6	pencil	65
2.23.7	plot	66
2.23.8	scope	66
2.23.9	spotlight	66
2.23.10	trail_pencil	66
2.23.11	txt	67

**3 Examples 69**

**4 Videos 71**

<b>5</b>	<b>Misc</b>	<b>73</b>
<b>6</b>	<b>Get Started</b>	<b>75</b>
<b>7</b>	<b>Other Resources</b>	<b>77</b>



## HOW TO INSTALL

Currently PatchWorld is under development for PC and only available to our developers, testers and close friends. If you would like to know more about Patch and our app on Meta Quest 2, [Patchworld](#). Please visit our website and follow us on [facebook](#) and [twitter](#)



## 2.1 Audio

### 2.1.1 allpass2

#### Description

Second order allpass filter for implementing phasors and such.

#### Inputs, output and other parts

*Signal input* (stream input) The input signal.

*Frequency* (stream input) The 'break' frequency of the filter.

*Radius* (stream input) The radius/steepness of the filter.

#### See also:

*lowpass, highpass, onepole, ladder, statevariable*

### 2.1.2 decay

#### Description

A decay envelope - decays the input value to target value over a duration of time.

#### Inputs, output and other parts

*Target value* (interactive) Sets the target value the output will decay towards.

*Half time* (stream input) Sets how long it will take for the output to go halfway towards the target value (in seconds).

*ButtonPivot* push

*Knob\_Pivot* value

*Knob* (interactive) Sets value from which the decay will begin.

*Button* (interactive) Triggers the decay envelope.

*Set state* (jolt input) Sets the value from which the decay will begin and triggers the decay envelope.

#### See also:

*envelope\_ad*

### 2.1.3 envelope\_ad

#### Description

A two-stage, Attack-Decay envelope with linear response. Attack begins from 0 and rises to 1. Decay begins from 1 and falls to 0. Connect the 'End of attack' output to the 'Decay trigger' input to trigger the decay stage right after attack finishes.

#### Inputs, output and other parts

*Attack time* (stream input) Attack stage time (in seconds).

*Decay time* (stream input) Decay stage time (in seconds).

*End of attack* (jolt output) Emits jolt once the attack stage finishes. Connect to the 'Decay trigger' input to trigger the decay stage right after attack finishes.

*End of decay* (jolt output) Emits jolt once the decay stage finishes.

*Attack trigger* (jolt input) Triggers the attack stage of the envelope.

*Decay trigger* (jolt input) Triggers the decay stage of the envelope.

#### See also:

*decay*

### 2.1.4 highpass

#### Description

12 dB/oct (two-pole) biquad high-pass filter.

#### Inputs, output and other parts

*Signal input* (stream input) The input signal.

*Cutoff frequency* (stream input) Cutoff frequency of the filter (in Hz). Must not be less than one.

*Resonance* (stream input) Resonance of the filter. Must not be less than zero.

#### See also:

*lowpass, onepole, ladder, statevariable, allpass2*

### 2.1.5 integrator

#### Description

A leaky integrator with separate constants for the rise and fall time. This block can be used to smooth out an incoming signal so that the change in the signal level cannot exceed a certain value per second. Combine with the 'abs' block to create an envelope follower.

#### Inputs, output and other parts

*Signal input* (stream input) The signal to integrate.

*Rise control* (stream input) The integration constant to use when the signal is rising.

*Fall control* (stream input) The integration constant to use when the signal is falling.

## 2.1.6 ladder

### Description

A 24 dB/oct (four-pole) bandpass-like filter, with center frequency and resonance settings.

### Inputs, output and other parts

*Signal input* (stream input) The input signal.

*Center frequency* (stream input) Center frequency of the filter (in Hz). Must not be less than one.

*Resonance* (stream input) Resonance of the filter. Must not be less than zero.

### See also:

*lowpass, highpass, onepole, statevariable, allpass2*

## 2.1.7 limiter

### Description

Applies brick wall limiting to the incoming signal, to control the amplitude.

### Inputs, output and other parts

*Signal input* (stream input) The signal to which the limiting will be applied.

*Release time* (stream input) How fast the limiter will stop suppressing the input signal once the threshold is no longer crossed.

*Look-ahead time* (stream input) Time 'ahead' at which the limiter will start suppressing the signal once the threshold has been crossed.

*Attack time* (stream input) How fast the limiter will start suppressing the input signal once the threshold level has been crossed.

*Threshold level* (stream input) The signal level above which the limiting will be applied. Set in linear units. (0-1 range)

## 2.1.8 lowpass

### Description

12 dB/oct (two-pole) biquad low-pass filter.

### Inputs, output and other parts

*Signal input* (stream input) The input signal.

*Cutoff frequency* (stream input) Cutoff frequency of the filter (in Hz). Must not be less than one.

*Resonance* (stream input) Resonance of the filter. Must not be less than zero.

### See also:

*highpass, onepole, ladder, statevariable, allpass2*

## 2.1.9 mic

### Description

Outputs microphone pitch (MIDI note number) and amplitude (0-1).

### Inputs, output and other parts

*Pitch* (jolt output) Emits the pitch of the audio coming into the microphone in the form of a MIDI note number.

*Amplitude* (jolt output) Emits the amplitude of the audio coming into the microphone in the form of the normalized level value (0-1).

*note*

*DB*

*Grab sphere VR*

*Grab sphere VR (1)*

### See also:

*microphone*

## 2.1.10 microphone

### Description

Outputs microphone audio signal as a stream.

### Inputs, output and other parts

*Grab sphere VR*

### See also:

*mic*

## 2.1.11 noise

### Description

Generates white noise - a signal with uniform energy across the frequency spectrum.

### Inputs, output and other parts

*glass*

### See also:

*pink\_noise*

## 2.1.12 onepole

### Description

6 dB/oct (one-pole) low-pass filter.

### Inputs, output and other parts

*Signal input* (stream input) The input signal.

*Cutoff frequency* (stream input) Cutoff frequency of the filter (in Hz). Must not be less than one.

### See also:

*lowpass, highpass, ladder, statevariable, allpass2*

## 2.1.13 oscillator

### Description

An oscillator with four waveform modes: sinusoid, triangle, sawtooth & square. Frequency is set in Hz (cycles per second).

### Inputs, output and other parts

*Reset phase* (jolt input) Sets the phase of the oscillator.

*Waveform* (interactive) The waveform of the oscillator. [sine/triangle/sawtooth/square]

*Wave type* (jolt input) Sets the waveform of the oscillator. [sine/triangle/sawtooth/square]

*Frequency (Hz)* (stream input) Sets the frequency of the oscillator (in Hertz).

*Signal output* (stream output) Oscillator output.

*Phase offset* (stream input) Sets the phase offset of the oscillator.

## 2.1.14 pink\_noise

### Description

Pink noise generator, outputting a signal with  $1/f^a$  energy across the frequency spectrum.

### Inputs, output and other parts

*glass*

### See also:

*noise*

## 2.1.15 pulse

### Description

Outputs a pulse with a duration set by the duration input. A pulse is a stream with a value of one for a specified period and zero elsewhere.

### Inputs, output and other parts

*Duration* (stream input) The duration of the pulse. The default value 0.00003 corresponds to one sample.

*Trigger* (interactive) Triggers the pulse.

### 2.1.16 reverb

#### Description

Adds reverberation to a stream using a feedback delay network algorithm.

#### Inputs, output and other parts

*Signal input* (stream input) The stream to apply reverb to.

*Decay time* (stream input) Controls how long it takes for the sound to die out (in seconds)

*Absorption* (stream input) Controls the decay time of the reverbs high frequencies. (0-1)

*Mix* (stream input) Controls the dry/wet mix of the output.

#### See also:

*reverb\_tank, delay*

### 2.1.17 reverb\_tank

#### Description

Applies reverb to the incoming signal.

#### Inputs, output and other parts

*Signal input* (stream input) The stream to apply reverb to.

*decay\_time* (stream input) Controls how long it takes for the sound to die out (in seconds).

*Absorption* (stream input) Controls the decay time of the reverbs high frequencies. (0-1)

*Mix* (stream input) Controls the dry/wet mix of the output.

#### See also:

*reverb, delay*

### 2.1.18 sample\_and\_hold

#### Description

Stores the value of a signal when a clock or event is received.

#### Inputs, output and other parts

*Clock input* (stream input) Each time when this values goes high the input signal value will be held.

*Signal input* (stream input) Signal to be sampled.

*sample* Sample



### 2.1.19 statevariable

#### Description

A state variable filter, with center frequency, resonance and filter type settings.

#### Inputs, output and other parts

*Signal input* (stream input) The input signal.

*Center frequency* (stream input) Center frequency of the filter (in Hz). Must not be less than one.

*Resonance* (stream input) Resonance of the filter. Must not be less than zero.

*Filter type* (jolt input) Sets the type of the filter.

#### See also:

*lowpass, highpass, onepole, ladder, allpass2*

### 2.1.20 vel

#### Description

Outputs the difference (delta) between the current and the previous sample value.

#### Inputs, output and other parts

*Signal input* (stream input) Signal to be analyzed.

*Grab sphere VR*

### 2.1.21 wavefolder

#### Description

A wave shaper that looks for values out of the [-1 and 1] range and mirrors the excess back into the [-1 and 1] range.

#### Inputs, output and other parts

*Signal input* (stream input) Signal to be folded.

*body*

## 2.2 Bubble samplers

### 2.2.1 bubble\_popper

#### Description

An underwater creature you can use to pop bubbles to remove them. Grab and move bubbles close to the spikes to make them pop.

#### Inputs, output and other parts

## 2.3 Connections

### 2.3.1 elbow

#### Description

A 90 degree bend that simply passes the stream from input to output.

#### Inputs, output and other parts

*input* (stream input)

### 2.3.2 extension

#### Description

Like an extension cable, connects two blocks to pass a stream event through a longer distance.

#### Inputs, output and other parts

*multibreak\_ICO*

*Input* (stream input) Extension cable input.

*Output* (stream output) Grabbable extension cable output.

#### See also:

*split, fork*

### 2.3.3 fork

#### Description

Extension cable and/or splitter, transmits one input stream as one or more stream outputs over a distance.

#### Inputs, output and other parts

*Input* (stream input) Stream input to be passed to output(s) unchanged.

*Make new cable* (interactive) Pull while holding trigger to create a new connection.

#### See also:

*split, extension*

### 2.3.4 get

#### Description

Emits jolt event when triggered. Value is determined by the stream input.

#### Inputs, output and other parts

*Value* (stream input) Sets the value that will be emitted (without triggering the output).

*Output* (jolt output) Emits jolt output when triggered.

*Button* (interactive) Triggers the output to emit the set value.

#### See also:

*stoe*

### 2.3.5 msg\_in

#### **Description**

Receive an OSC message.

#### **Inputs, output and other parts**

*e\_out*

*text\_IN*

### 2.3.6 msg\_out

#### **Description**

Send an OSC message.

#### **Inputs, output and other parts**

*e\_in\_hot*

*s\_IN*

*text\_IN*

### 2.3.7 output

#### **Description**

Sends sound signal from Patch to the currently active audio output.

#### **Inputs, output and other parts**

*input* Audio stream input.

*pan* Left/right pan.

*spatialness* Controls how much the output's position influences the output sound.

### 2.3.8 pass\_in

#### **Description**

#### **Inputs, output and other parts**

*eR\_hot*

*Emitter*

### 2.3.9 pass\_out

#### Description

#### Inputs, output and other parts

*eR\_hot*

*Emitter*

### 2.3.10 spacer

#### Description

Connect to streams away from each other.

#### Inputs, output and other parts

*arm\_handle*

*s\_IN* Stream input

### 2.3.11 split

#### Description

Splitter, transmits one input stream as two identical stream outputs (A and B) - like a thru box or splitter box.

#### Inputs, output and other parts

*BODY*

*Input* (stream input) Stream input to be passed to both (A and B) outputs unchanged.

*Output A* (stream output) First stream output.

*Output B* (stream output) Second stream output.

#### See also:

*extension, fork*

### 2.3.12 stoe

#### Description

Send a jolt value each time the input stream changes. (max 90Hz by default)

#### Inputs, output and other parts

*input* (stream input) The stream to convert.

*output* (event output) Emits the value of the incoming stream as a jolt.

### 2.3.13 wireless\_in

#### Description

Sends stream from one place in your patch to the corresponding ‘wireless\_out’ blocks. Each ‘wireless\_in’ and ‘wireless\_out’ set of blocks within one connection need to have the same name. Each wireless connection needs to have a unique name.

#### Inputs, output and other parts

*Connection name* (text) Name of the connection - the same as the name of corresponding ‘wireless\_out’ block.

*Input* (stream input) Stream to be sent.

#### See also:

*wireless\_out*

### 2.3.14 wireless\_in\_jolt

#### Description

Receives stream sent from the corresponding ‘wireless\_out\_jolt’ blocks. Each ‘wireless\_in\_jolt’ and ‘wireless\_out\_jolt’ set of blocks within one connection need to have the same name. Each wireless connection needs to have a unique name.

#### Inputs, output and other parts

*Connection name* (text) Name of the connection - the same as the name of the corresponding ‘wireless\_out\_jolt’ block.

*Output* (jolt output) Emits received jolt.

### 2.3.15 wireless\_out

#### Description

Receives stream sent from the corresponding ‘wireless\_in’ blocks. Each ‘wireless\_in’ and ‘wireless\_out’ set of blocks within one connection need to have the same name. Each wireless connection needs to have a unique name.

#### Inputs, output and other parts

*Connection name* (text) Name of the connection - the same as the name of the corresponding ‘wireless\_in’ block.

#### See also:

*wireless\_in*

### 2.3.16 wireless\_out\_jolt

#### Description

Sends jolt from one place in your patch to the corresponding ‘wireless\_in\_jolt’ blocks. Each ‘wireless\_in\_jolt’ and ‘wireless\_out\_jolt’ set of blocks within one connection need to have the same name. Each wireless connection needs to have a unique name.

#### Inputs, output and other parts

*Input* (jolt input) Jolt to be sent.

*Connection name* (text) Name of the connection - the same as the name of corresponding ‘wireless\_in\_jolt’ block.

## 2.4 Deprecated

### 2.4.1 alu

#### Description

All logical operations, A block that allows users to do all arithmetic operations, with the opportunity to select the operations with a dial

#### Inputs, output and other parts

*s\_IN*

*eR\_hot*

*Emmitter*

*Knob\_Type*

*SetTypeReciever*

### 2.4.2 digitalwaveguidestring

#### Description

A string you can play by plucking it or hitting it with a marble.

#### Inputs, output and other parts

*k\_sustain*

*sustainEventReceiver*

*k\_brightness*

*brightnessEventReceiver*

*Grab sphere VR*

*string*

*ChangeStringRadius*

*ChangeLength*

### 2.4.3 freeverb

#### Description

#### Inputs, output and other parts

*s\_IN\_0\_in*

*s\_IN\_1\_roomsize*

*s\_IN\_2\_feedback*

*s\_IN\_3\_frezze*

### 2.4.4 ntor

#### Description

Does  $(x + 1) * 0.5 * (B-A) + A$  so you can map  $\{-1 \text{ to } 1\}$  values to  $\{A \text{ to } B\}$  values.

#### Inputs, output and other parts

*s\_IN\_value*

*s\_IN\_1\_min*

*s\_IN\_2\_max*

### 2.4.5 ntou

#### Description

Does  $(x + 1) * 0.5$  so you can map  $\{-1 \text{ to } 1\}$  values to  $\{0 \text{ to } 1\}$ .

#### Inputs, output and other parts

*s\_IN*

### 2.4.6 outstate

#### Description

Split optimized for feedback loops.

#### Inputs, output and other parts

*s\_IN*

*~OUT\_feedback*

*OUT\_Set*

### 2.4.7 rms

#### Description

Calculates the root mean square of the incoming signal over a period of time. For example can be used to make an envelope follower.

#### Inputs, output and other parts

*Signal input* (stream input) Signal for which the root mean square will be calculated.

*Window size* (stream input) Size of the analysis window (in seconds).

## 2.4.8 rton

### Description

Maps our input from {a to b} to {-1 to 1}

### Inputs, output and other parts

*s\_IN\_0\_value*

*s\_IN\_1\_min*

*s\_IN\_1\_max*

## 2.4.9 rtou

### Description

Maps an input from range {A to B} to unit range {0 to 1}

### Inputs, output and other parts

*Input* (stream input) Value to be mapped.

*A* (stream input) Input range lower value.

*B* (stream input) Input range upper value.

### See also:

*utor, map, clamp*

## 2.4.10 speaker\_2D

### Description

A non-spatialised sound source outputting directly to your physical speakers or headphones. Speakers translate the values of what you are building into audio. Use the second input to control panning (-1 to 1).

### Inputs, output and other parts

*s\_IN* Audio input

*s\_IN\_pan* Left/Right pan (-1 to 1)

## 2.4.11 speaker

### Description

This is your sound source. Speakers translate the values of what you are building into audio.

### Inputs, output and other parts

*s\_IN*



## 2.4.12 uton

### Description

Converts a unit to normal

u unit: 0 to 1

n: normal -1 to 1

r: range : any range

### Inputs, output and other parts

*s\_IN*

## 2.4.13 utor

### Description

Converts a unit range to any set {A to B} range. Example: use a slider that goes from 0 to 1, connect it to the top part of the utor and then use the other two inputs to set output range.

u (unit): 0 to 1

n (normal): -1 to 1

r (range): any range

### Inputs, output and other parts

*Top*: get a signal to drive the conversion

*left*: top

*Input* (stream input) Value to be mapped.

*A* (stream input) Output range lower value.

*B* (stream input) Output range upper value.

### See also:

*rtou, map, clamp*

## 2.5 Donotinclude

### 2.5.1 cmd

#### Description

#### Inputs, output and other parts

*esReciver*

*Button*

## 2.5.2 ghost

### Description

Place a recorded controller and change the timing.

### Inputs, output and other parts

*k\_offset* Offset the animation in time.

*b\_offset* Restart when pressed.

*k\_tempo* Set the base tempo.

*k\_beat* Set how many beats it lasts before looping.

*Grab sphere VR*

*text\_IN*

*k\_startTrim*

## 2.5.3 linearladder

### Description

Frequency bandpass filter.

### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

*s\_IN\_2*

## 2.5.4 lowpass\_old

### Description

Low pass filter.

### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1\_cutoffFreq*

## 2.5.5 m\_add

### Description

### Inputs, output and other parts

*s\_IN*

## 2.5.6 pass

### Description

Passes an event from a jolt input to jolt output without modification.

### Inputs, output and other parts

*Input* (jolt input) Jolt input to be emitted from output.

*Output* (jolt output) Emits the received jolt.

## 2.5.7 sample\_player\_cut

### Description

### Inputs, output and other parts

*speed*

*buffer\_index*

*GRAB (1)*

*play*

*end\_of\_sample*

*start\_position*

*duration*

## 2.5.8 sample\_player

### Description

### Inputs, output and other parts

*cursor*

*buffer\_index*

*GRAB (1)*

*end\_of\_sample*

*num\_beats*

## 2.5.9 sample\_recorder

### Description

### Inputs, output and other parts

*input*

*cursor*

*Record\_command*

## 2.5.10 string\_with\_output

### Description

A physical model of a string that can interact with marbles and your controller. Has a stream output to connect it to the rest of your patch.

### Inputs, output and other parts

*sustain* (knob) Control how long the string will ring out after it's been excited.

*brightness* (knob) Controls how bright the string sound is

*string* (interactive) The string you can hit with

*inharmonicicity* (interactive) Pull to change the radius of the string, making the sound more bell-like.

*length\_pitch* (interactive) Pull/push to change the length of the string and thereby its pitch.

## 2.5.11 toggleboardcmd

### Description

Editable array of toggle buttons.

### Inputs, output and other parts

## 2.6 Experimental

### 2.6.1 ableton\_link

#### Description

#### Inputs, output and other parts

*Sphere*

*knob*

*btn\_debug*

*e\_OUT*

### 2.6.2 bar

#### Description

#### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

*s\_IN\_2*

*s\_IN\_3*

### 2.6.3 chain

#### Description

#### Inputs, output and other parts

*emitter*

*eR\_main*

### 2.6.4 frequency\_analyzer

#### Description

#### Inputs, output and other parts

*signal\_input*

*trigger\_analysis*

*max\_freq*

*max\_freq\_amplitude*

### 2.6.5 global\_reverb

#### Description

#### Inputs, output and other parts

*s\_IN\_1\_decayTime*

*s\_IN\_2\_absorb*

*s\_IN\_3\_freeze*

### 2.6.6 hammer

#### Description

#### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

*s\_IN\_2*

### 2.6.7 karplus\_strong\_extended

#### Description

#### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

*s\_IN\_2*

*s\_IN\_3*

*s\_IN\_4*

*s\_IN\_5*

*s\_IN\_6*

## 2.6.8 karplus\_strong

### Description

#### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

*s\_IN\_2*

*s\_IN\_3*

## 2.6.9 list

### Description

Select between multiple stream inputs.

#### Inputs, output and other parts

*Emitter\_000*

*Emitter\_001*

*es\_hot*

*s\_IN(Clone)*

## 2.6.10 pitchshifter

### Description

Transpose sound in real time.

#### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

## 2.6.11 plate

### Description

plate filter : input, freq, hammerX, hammerY, freqDedLoss, freqInDepLoss

#### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

*s\_IN\_2*

*s\_IN\_3*

*s\_IN\_4*

*s\_IN\_5*

## 2.6.12 pluck

### Description

#### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

*s\_IN\_2*

## 2.6.13 s\_reader

### Description

When 'IN gets an event, 'OUT fires the value of ~IN

#### Inputs, output and other parts

*e\_OUT* 'OUT

*s\_IN* ~IN

*e\_IN\_HOT* 'IN

*e\_IN\_Hot*

## 2.6.14 sample\_info

### Description

#### Inputs, output and other parts

*trigger*

*GRAB* (1)

*sample\_duration*

*num\_beats*

## 2.6.15 saturator

### Description

Applies a simple tanh distortion/saturation to the incoming signal.

#### Inputs, output and other parts

*s\_IN*

*body*

## 2.6.16 violin

### Description

Violin physical model generator

### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1*

*s\_IN\_2*

*s\_IN\_3*

## 2.7 Interface

### 2.7.1 back\_button

#### Description

Pressing the button will emit a number value of 1 out of the output.

#### Inputs, output and other parts

*Button* (interactive) Triggers output to emit 1.

*Output* (jolt output) Emits the value 1 when button is pressed.

*Trigger* (jolt input) Triggers output to emit 1.

*State* (jolt input) Sets the visual state of the button, 0 - button is dark, >0 - button is lit, <0 - button looks inactive.

#### See also:

*help\_button, next\_button, play\_button, reset\_button, skip\_button*

### 2.7.2 button

#### Description

Pressing the button will emit a number out of the output, number can be set by the stream input.

Buttons are one of the basic blocks used to trigger stuff in your world. You can e.g. wire a the output of a button to the 'play' input of a sample to trigger playback of that sample.

#### Inputs, output and other parts

*Value* (stream input) Sets the number value that the ouput will emit.

*Button* (interactive) Triggers the output to emit the set number value.

*Output* (jolt output) Emits the number value.

*Trigger* (jolt input) Triggers the output to emit the set number value.

#### See also:

*value, slider, knob*



### 2.7.3 controloutput

#### Description

#### Inputs, output and other parts

*Trigger Right*

*Grip Right*

*Button A Right*

*Button B Right*

*Head Distance*

*Trigger Left*

*Grip Left*

*Button A Left*

*Button B Left*

*Distance Between Controllers*

### 2.7.4 custom\_slider

#### Description

A slider that be used to control various things via jolts.

Sends out a jolt with a value between 0 and 1 that indicates the position of the slider thumb. The jolt input can be used to set the slider's value from elsewhere.

#### Inputs, output and other parts

*Thumb* (interactive) Press trigger button and move to change the slider's value.

*Output* (jolt output) Sends out the slider's value whenever it changes.

*Input* (jolt input) Sets the value of the slider.

### 2.7.5 execute

#### Description

Execute console command when triggered. Using up to 4 inputs :0 :1 :2... as parameter. ex : bgcolor :1 :2 :3

#### Inputs, output and other parts

*text\_IN*

*esReciver*

*Button*

*s\_IN*

## 2.7.6 ghost\_looper

### Description

Record ghosts.

### Inputs, output and other parts

## 2.7.7 hapticcontrol

### Description

### Inputs, output and other parts

*s\_IN\_Right*

*s\_IN\_Left*

## 2.7.8 help\_button

### Description

Press the button - triggers a jolt input when activated.

### Inputs, output and other parts

*Button* Button

*Emitter* Emits jolt of active value

*e\_I* Jolt input

## 2.7.9 interaction\_box

### Description

A highly flexible interface for building different kinds of interactions.

A box-shaped area that can be divided into segments along each axis. Provides a variety of different jolt outputs for each controller inside the interaction area, and lets you control the controller's haptics.

### Inputs, output and other parts

*Resize handle* (interactive) Drag to resize the interaction area.

*Right X* (jolt output) Emits the X position of the right controller.

*Right Y* (jolt output) Emits the Y position of the right controller.

*Right Z* (jolt output) Emits the Z position of the right controller.

*Right present* (jolt output) Emits a 1 when the right controller enters the area and 0 when it leaves.

*Right trigger* (jolt output) Emits a value between 0 and 1 depending on how much the right controller trigger button is pressed. If 'Continuous trigger' is turned off via the inspector it sends out 1/0 when the trigger is pressed/released.

*Left X* (jolt output) Emits the X position of the left controller.

*Left Y* (jolt output) Emits the Y position of the left controller.

*Left Z* (jolt output) Emits the Z position of the left controller.

*Left present* (jolt output) Emits a 1 when the left controller enters the area and 0 when it leaves.

*Left trigger* (jolt output) Emits a value between 0 and 1 depending on how much the left controller trigger button is pressed. If ‘Continous trigger’ is turned off via the inspector it sends out 1/0 when the trigger is pressed/released.

*Segments X* (knob) Controls the number of segments in the X direction.

*Segments X* (jolt input) Controls the number of segments in the X direction.

*Segments Y* (knob) Controls the number of segments in the Y direction.

*Segments Y* (jolt input) Controls the number of segments in the Y direction.

*Segments Z* (knob) Controls the number of segments in the Z direction.

*Segments Z* (jolt input) Controls the number of segments in the Z direction.

*Haptics left* (stream input) Controls the vibration strength of the left controller.

*Haptics right* (stream input) Controls the vibration strength of the right controller:

## 2.7.10 keyboard

### Description

A keyboard interface that can be used to play notes by pressing the keys or to play music from a specific scale using the jolt input.

### Inputs, output and other parts

*MIDI Out* (jolt output) Outputs the MIDI-number of the key you press, or the corresponding note if you send a jolt to INPUT.

*Input* (jolt input) Send a jolt to make the keyboard output the MIDI-number of the corresponding note in the scale, starting form zero. E.g. sending 0 will send out the root note, or if you have a major or minor scale selected sending a 2 will output the third.

*Root note* (knob) Sets the root note of the keyboard and scale. Maps “C” key to choosen note.

*Root note* (jolt input) Send a jolt to set the root note of the keyboard and scale. Maps “C” key to choosen note.

*Play/Scale Toggle* Switch between Play/Scale edit mode.

## 2.7.11 knob

### Description

A knob.

### Inputs, output and other parts

*knobBody*

*ValveEventReciver*

## 2.7.12 knobboard

### Description

Editable array of knobs.

### Inputs, output and other parts

*btn\_debug*

*resize*

*OUT*

*WRITE*

*rows*

*columns*

*KnobB(Clone)*

## 2.7.13 laser\_canvas

### Description

A canvas that lets you control stuff with the laser pointer.

When you point your controller at the canvas it will activate the laser pointer (like e.g. in the hub). Jolt outputs lets you get the controller state.

### Inputs, output and other parts

*Resize* (interactive) Move with trigger to resize the canvas.

*Canvas* (interactive) Point your controller at the canvas to interact with it.

*Active* (jolt input) Send a 1 to activate the laser, send a 0 to deactivate it.

*x* (jolt output) Emits the normalized (0 to 1) x coordinate of where you're pointing at the canvas.

*y* (jolt output) Emits the normalized (0 to 1) y coordinate of where you're pointing at the canvas.

*hover* (jolt output) Emits a 1 when you start pointing at the canvas and 0 when you stop.

*trigger* Emits a 1 when you press the trigger while pointing at the canvas, and 0 when you stop or exit the canvas.

## 2.7.14 laserboard

### Description

### Inputs, output and other parts

*emiiter\_ON*

*emitter\_OFF*

*s\_IN\_base*

*GRAB\_ico*

*s\_IN0*

*s\_IN1*

*s\_IN2*

*s\_IN3*

*s\_IN4*

*s\_IN5*

*s\_IN6*

*s\_IN7*

*s\_IN8*

*s\_IN9*

*s\_IN10*

*s\_IN11*

### 2.7.15 laserkey

#### Description

#### Inputs, output and other parts

*EmitterNoteON*

*EmitterNoteOFF*

*s\_IN*

### 2.7.16 next\_button

#### Description

Press the button - triggers a jolt of any input stream value when activated.

#### Inputs, output and other parts

*Button* Button

*Emitter* Emits jolt of active value

*e\_I* Jolt input

### 2.7.17 nknob

#### Description

A stream constant value.

#### Inputs, output and other parts

*knobBody*

#### See also:

[knob](#), [custom\\_knob](#), [custom\\_knob2](#)

### 2.7.18 pad

#### Description

A pad you can hit; sends a trigger and the hit position on the pad.

#### Inputs, output and other parts

*pad* Pad

*eReciver\_toggleColor* Jolt toggles color

*emiter\_vel* Emits velocity value

*emiter\_x* Emits x position of hit

*emiter\_y* Emits y position of hit

*Hit\_Resize* Hit resize

### 2.7.19 peppermill

#### Description

#### Inputs, output and other parts

*ValveRoot*

*ValveEventReciver*

*valveHandle*

*Grab sphere VR*

### 2.7.20 play\_button

#### Description

Press the button - triggers a jolt input when activated.

#### Inputs, output and other parts

*Button* Button

*Emmitter* Emits jolt of active value

*e\_I* Jolt input

### 2.7.21 presetmanager

#### Description

Create presets for your instruments.

#### Inputs, output and other parts

## 2.7.22 readout

### Description

Display a number. Used for building interfaces, debugging and learning.

### Inputs, output and other parts

*Display* (visual) Shows the number set via the stream input

*Input* (stream input) Send a stream or jolt to display it's value.

## 2.7.23 reset\_button

### Description

Press the button - triggers a jolt input when activated.

### Inputs, output and other parts

*Button* Button

*Emitter* Emits jolt of active value

*e\_I* Jolt input

## 2.7.24 skip\_button

### Description

Press the button - triggers a jolt input when activated.

### Inputs, output and other parts

*Button* Button

*Emitter* Emits jolt of active value

*e\_I* Jolt input

## 2.7.25 slider\_3d

### Description

A three-dimensional slider with x, y, and z outputs.

### Inputs, output and other parts

*hide\_show* (jolt input) Send a value of one to show the ball or zero to hide it.

*extend\_x* (interactive) Pull to extend the x-axis.

*extend\_y* (interactive) Pull to extend the x-axis.

*extend\_z* (interactive) Pull to extend the z-axis.

*ball* (interactive) The ball whose position determines the output values.

*output\_x* (stream output) Outputs the x position of the ball.

*output\_y* (stream output) Outputs the y position of the ball.

*output\_z* (stream output) Outputs the z position of the ball.

### 2.7.26 slider\_db

#### Description

#### Inputs, output and other parts

*Grab sphere VR*

*eR\_set*

*PULL\_extend*

*PULL\_fader*

### 2.7.27 slider

#### Description

You can change the value with your controller. Goes from 0 to 1.

#### Inputs, output and other parts

*Grab sphere VR*

*eR\_set*

*PULL\_extend*

*PULL\_fader*

### 2.7.28 sliderboard

#### Description

Editable array of silders.

#### Inputs, output and other parts

*btn\_debug*

*ResizeHandle*

*OUT*

*WRITE*

*s\_IN\_0\_rows*

*s\_IN\_1\_columns*

*Grab sphere VR*

*eR\_set*

*PULL\_extend*

*PULL\_fader*



### 2.7.29 stepped\_dial

#### Description

A dial with a configurable number of steps.

Used for switching between two or more discrete options.

#### Inputs, output and other parts

*input* (jolt input) Sets the currently selected step.

*output* (jolt output) Emits the currently selected step when it changes.

*Dial* (Interactive) Press trigger and turn to move the dial.

### 2.7.30 toggle

#### Description

Set and send state as on or off, with indicator.

#### Inputs, output and other parts

*TOGGLE\_state\_indicator*

*EventReceiver*

### 2.7.31 toggleboard

#### Description

Editable, resizable array of toggles.

#### Inputs, output and other parts

*s\_IN\_rows\_0* Stream input - number of rows

*s\_IN\_columns\_1* Stream input - number of columns

*resize* Resize

*OUT* Out

*WRITE* Write

*btn\_debug*

*toggleboard\_piece(Clone)*

### 2.7.32 uknob

#### Description

a unit knob t

u unit: 0 to 1

#### Inputs, output and other parts

*knobBody*

## 2.7.33 value

### Description

A button that emits a number value. Hit the sphere, or send a Jolt to the input to emit a value. Set the value that will be emitted by adjusting the number with the trigger button.

### Inputs, output and other parts

\*\*

*Trigger* (jolt input) Triggers the output to emit the set number value.

*Output* (jolt output) Emits the set value when triggered.

*Value* (interactive) Sets the value that will be emitted (without triggering the output).

### See also:

*button, get, slider, knob*

## 2.8 Kinetic

### 2.8.1 anchor

#### Description

Sets player position and rotation to the position and rotation of the anchor block when triggered. Sending a zero will transport the player instantly, while sending any positive number will transport the player gradually over time - time (in ms) is set by received value.

#### Inputs, output and other parts

*Direction* (pointer) Sets the direction of the player's view according to the direction of the elongated tube. Once the action is completed, the player's head will end up where the block is located.

*Teleport* (jolt input) Sets and triggers the time it takes for the player to reach the new position and rotation in milliseconds.

### 2.8.2 ball

#### Description

#### Inputs, output and other parts

*ball*

### 2.8.3 block\_distance

#### Description

#### Inputs, output and other parts

*InteractiveSelector*

*distance output*

## 2.8.4 block\_get\_look

### Description

Output angle between front/right or up direction of the origin, and the target position.

### Inputs, output and other parts

*origins*

*targets*

*Angle Z* Is the front of the origin facing the target ?

*Angle Y*. Is the right side of the origin facing the target ?

*Angle X* Is the up direction of the origin facing the target ?

## 2.8.5 block\_lookat

### Description

### Inputs, output and other parts

*InteractiveSelector*

*speed*

## 2.8.6 block\_relative\_position

### Description

### Inputs, output and other parts

*InteractiveSelector*

*e\_OUT\_x*

*e\_OUT\_y*

*e\_OUT\_z*

## 2.8.7 block\_set\_position

### Description

### Inputs, output and other parts

*InteractiveSelector*

*knob*

### 2.8.8 box

#### Description

#### Inputs, output and other parts

*bottom\_intr*

*px\_intr*

*nx\_intr*

*pz\_intr*

### 2.8.9 funnel

#### Description

#### Inputs, output and other parts

*Grab*

### 2.8.10 handle

#### Description

Lets you attach object together. Click and drag from handle to the object.

#### Inputs, output and other parts

*handle*

### 2.8.11 m\_booster

#### Description

#### Inputs, output and other parts

*s\_IN*

### 2.8.12 m\_compare

#### Description

#### Inputs, output and other parts

*s\_IN*

### 2.8.13 m\_path

**Description**

**Inputs, output and other parts**

*EmitterV*

*emitter*

*emitterLenght*

*knobOrig*

*Collider*

*PathBodyExtendHandleHandelExtend\_001*

*Grab*

### 2.8.14 m\_reader

**Description**

**Inputs, output and other parts**

*emitter*

### 2.8.15 m\_spawner

**Description**

**Inputs, output and other parts**

*ButonPivot*

*eR\_triggerController*

*knob*

### 2.8.16 m\_switch

**Description**

**Inputs, output and other parts**

*s\_IN*

### 2.8.17 marble

**Description**

**Inputs, output and other parts**

### 2.8.18 octopus

#### Description

Lets you attach object together. Click and drag from octopus to the object.

#### Inputs, output and other parts

*octopus*

### 2.8.19 octopus\_inv

#### Description

#### Inputs, output and other parts

*octopus\_inv*

*btn\_dbug*

### 2.8.20 rail

#### Description

Attach other blocks to it use click and drag. Then changing its input value will translate them.

#### Inputs, output and other parts

*length*

*position*

*octobody (1)*

### 2.8.21 rotor

#### Description

Attach other blocks to it use click and drag. Then changing its input value will rotate them.

#### Inputs, output and other parts

*rotation*

*grabber*

### 2.8.22 string

#### Description

A string (like a guitar string) you can strike with your controller or hit with marbles to make sound.

A simulation or e.g. a guitar string with a natural sound. Can interact with other kinetic object such as your controller or marbles. Changes its sound subtly depending on where you hit it.

#### Inputs, output and other parts

*Sustain* (knob) Controls for how long the string will ring out when struck (in seconds).

*Sustain* (jolt input) Send a jolt to set the sustain of the string (in seconds).

*String* (interactive) Strike or hit with a marble to make sound

*Length* (interactive) Use the trigger button and push/pull to change the length/pitch of the string.

*Pitch* (jolt input) Send a MIDI note (e.g. from a keyboard) to set the pitch of the string.

*Output* (stream output) Sends out the sound of the string. Will mute the string when plugged in.

## 2.8.23 wall

### Description

A wall to bounce collision objects like marbles

### Inputs, output and other parts

*Grab*

*extensionHandle*

## 2.9 Lights

### 2.9.1 hsvtorgb

#### Description

#### Inputs, output and other parts

*sO\_R*

*sO\_G*

*sO\_B*

*Hue*

*Saturation*

*Value*

*Grab sphere VR*

### 2.9.2 lightbulb

#### Description

Adds an HSV-colored light source to a scene.

#### Inputs, output and other parts

*s\_IN\_h* Hue

*s\_IN\_s* Saturation

*s\_IN\_v* Intensity

*hue*

*saturation*

*value*

## 2.9.3 sunlight

### Description

Let you change color and sun orientation.

### Inputs, output and other parts

*s\_IN\_r* Red amount

*s\_IN\_g* Green amount

*s\_IN\_b* Blue amount

*s\_IN\_f* Global intensity

*hue*

*saturation*

*value*

*Grab sphere VR*

## 2.10 Logic

### 2.10.1 bernoulli

#### Description

Routes an incoming jolt to either of two outputs based on probability. With probability value set closer to 0 the jolt will more like go though output A, while if value will be set closer to 1 - though output B.

#### Inputs, output and other parts

*Probability* (stream input) Probability distribution value.

*Output A* (jolt output) Emits jolt value based on set probability.

*Output B* (jolt output) Emits jolt value based on set probability.

*Input* (jolt input) Receives jolt value which will be passed to either of the outputs.

#### See also:

*random, sequence\_random*

### 2.10.2 compare

#### Description

Compares a jolt value with a stream value and outputs result of the comparison in the form of a boolean value (false - 0, true - 1).

#### Inputs, output and other parts

*Output* (jolt output) Emits boolean result (false - 0, true - 1) of jolt and stream values comparison on each received jolt.

*Compare with* (stream input) Sets the value that will be compared with the incoming value from the jolt input (without triggering the output).

*Input* (jolt input) Receives value which will be compared with value set in the stream input, triggers the comparison.



*Comparison operation* (interactive) Sets the type of the comparison operation.

*SetTypeREciever*

**See also:**

*if\_else*

### 2.10.3 counter

#### Description

Counts the sum of incoming jolts values.

#### Inputs, output and other parts

*Set* (jolt input) Sets current count value and triggers the output.

*Output* (jolt output) Emits the current count value as a jolt.

*Increment* (jolt input) Counter increment value. Each received jolt value is added to the count and triggers the output.

*Knob* (interactive) Sets the value.

*Button* (interactive) Triggers the value set by the knob.

**See also:**

*cyclecounter, nth*

### 2.10.4 cyclecounter

#### Description

Counts the number of received jolts in a looping manner.

#### Inputs, output and other parts

*cyclecounter*

*Cycle size* (stream input) Sets the value at which the counter will reset and start counting from zero again.

*Output* (jolt output) Emits the current count value as a jolt.

*Increment* (jolt input) Counter increment trigger. Each received jolt triggers the count to increase by one.

*Set* (jolt input) Sets current count value and triggers the output.

**See also:**

*counter*

### 2.10.5 data\_array

#### Description

Store and retrieve values from an array using indices to build presets, sequencers, etc..

The `data_array` block can be used to store values to be recalled later. Sending an index

#### Inputs, output and other parts

*Read index* (jolt input) Send a jolt to read the value at a given index.

*Write index* (jolt input) Send a jolt to select which index will be written to next time a value is received.

*value* (jolt input) Send a jolt to write a value to the index selected via the 'Write index' input

*value* (jolt output) Emit the stored value when a jolt is received through the 'Read index' jolt input.

## 2.10.6 gate

### Description

Allows or disallows jolts to pass based on the set stream value.

### Inputs, output and other parts

*Output* (jolt output) Emits the received jolt value as a jolt if gate value is equal or greater than 0.5

*Gate* (stream input) Sets the gate state. Values equal or greater than 0.5 will set gate state to 'open'.

*Input* (jolt input) Receives jolt value which will be passed to the output if gate value is equal or greater than 0.5

### See also:

*sequence*

## 2.10.7 if\_else

### Description

Compares a jolt value with a stream value and outputs a jolt value through one of the two outputs, depending on whether the condition is met or not

### Inputs, output and other parts

*SetTypeReceiver*

*Input* (jolt input) Receives value which will be compared with value set in the stream input, triggers the comparison.

*Comparison operation* (interactive) Sets the type of the comparison operation.

*Compare with* (stream input) Sets the value that will be compared with the incoming value from the jolt input (without triggering the output).

- *Output if\** (jolt output) Emits received jolt value if the condition is met.
- *Output else\** (jolt output) Emits received jolt value if the condition is not met.

### See also:

*compare*

## 2.10.8 nth

### Description

Sends only one event every N events it receives.

### Inputs, output and other parts

*N value* (stream input) Sets the value at which the counter will reset, start counting from zero again and trigger the output.

*Input* (jolt input) Counter increment trigger. Each received jolt triggers the count to increase by one.

*Set* (jolt input) Sets current count value.

*Output* (jolt output) Emits a jolt every time the count has reached the N value.

**See also:**

*rangedfor*, *counter*

## 2.10.9 rangedfor

### Description

Instantly sends N events, from zero to N-1, when triggered.

### Inputs, output and other parts

*Set* (jolt input) Triggers the process of sending jolts from 0 to N-1 (where N is set by the received jolt value).

*Output* (jolt output) Emits jolts from zero to N-1.

*Knob* (interactive) Sets N, the number of jolt to output when triggered.

*Button* (interactive) Triggers the process of sending jolts from 0 to N-1 (where N is set by the knob).

**See also:**

*nth*

## 2.10.10 sequence

### Description

Routes an incoming jolt to one of many outputs.

### Inputs, output and other parts

*Select output* (stream input) Selects to which output the incoming jolt will be routed to.

*Pull* (interactive) Pull to set the number of output gates.

*Input* (jolt input) Receives the jolt which will be routed to the selected output.

*Output* (jolt output) Emits the routed jolt value. One of the possible outputs the input may be routed to.

**See also:**

*sequence\_loop*, *sequence\_random*

## 2.10.11 sequence\_loop

### Description

Routes an incoming jolt to one of many outputs in a looping manner.

### Inputs, output and other parts

*Input* (jolt input) Receives the jolt which will be routed to the currently selected output and proceed to the next output.

*Pull* (interactive) Pull to set the number of output gates.

*Select output* (jolt input) Select which output should be used next time an input is received.

*Output* (jolt output) Emits the routed jolt value. One of the possible outputs the input may be routed to.

**See also:**

*sequence, sequence\_random*

## 2.10.12 sequence\_random

### Description

Routes an incoming jolt to a random output.

### Inputs, output and other parts

*Input* (jolt input) Receives the jolt which will be routed to one of the outputs at random.

*Pull* (interactive) Pull to set the number of output gates.

*Output* (jolt output) Emits the routed jolt value. One of the possible outputs the input may be routed to.

**See also:**

*sequence, bernoulli*

## 2.10.13 watcher

### Description

Compares two stream values. Every time the condition is met, signal output will send the value of 1.

### Inputs, output and other parts

*Signal input* (stream input) Stream input value to compare with value from second inlet.

*Compare with* (stream input) Sets the value that will be compared with the incoming value from the signal input.

*Output* (jolt output) Sends jolt value every time the condition begins to be true

*SetTypeReceiver*

*Comparison operation* (interactive) Sets the type of the comparison operation.

**See also:**

*compare*

## 2.11 Marble rain

### 2.11.1 marbleslide

#### Description

A slide that will guide marbles with realistic physics.

#### Inputs, output and other parts

*marbleSlide*

*Handle*

## 2.11.2 marbletube

### Description

A tube with a funnel that will catch and guide marbles.

### Inputs, output and other parts

*marbleTube*

*Handle*

## 2.12 Math

### 2.12.1 abs

#### Description

Calculates and outputs absolute value of an incoming signal. Any negative number will become positive. Examples: -8 will output 8. -12.3 will output a 12.3

#### Inputs, output and other parts

*Number* (stream input) Outputs absolute (non-negative) value of given number

#### See also:

*round*

### 2.12.2 add

#### Description

Adds two numbers together and outputs the result using a simple  $A + B$  math operation, can be used to sum two signals together. Example: Connect an oscillator to each input, set each to different frequency, connect the output to a speaker block to hear both together.

#### Inputs, output and other parts

*A* (stream input) The first value, added to B.

*B* (stream input) The second value, added to A.

#### See also:

*subtract, multiply, divide, add\_jolt*

### 2.12.3 add\_jolt

#### Description

Adds number from stream input to the jolt input and outputs the sum as jolt.

#### Inputs, output and other parts

*Add* (stream input) Sets the value that will be added to the incoming value from the jolt input (without triggering the output).

*Result* (jolt output) Emits sum of jolt and stream values on each received jolt.

*Calculate* (jolt input) Receives value to which the value set in the stream input will be added, triggers the calculation.

**See also:**

*subtract\_jolt, multiply\_jolt, divide\_jolt, add*

## 2.12.4 clamp

### Description

Clamps a value inside a specified range - meaning, input is truncated within the range from set minimum to set maximum.

### Inputs, output and other parts

*Input* (stream input) Value to be clamped within the range.

*Range minimum* (stream input) Minimum of the clamp range

*Range maximum* (stream input) Maximum of the clamp range.

**See also:**

*map, rtou, utor*

## 2.12.5 constant

### Description

Outputs various useful math constants: pi: the trigonometric constant sr: sample rate sp: sample period e: Euler's constant bpm: tempo in bpm bp: beat period (sec. pr. beat).

### Inputs, output and other parts

*Output* (jolt output) Emits the value of the selected constant whenever it changes or a jolt is received.

*Constant* (knob) Selects which constant to output.

*Trigger* (jolt input) Triggers the output to emit the selected constant.

## 2.12.6 cos

### Description

Calculates cosine value of A

### Inputs, output and other parts

*s\_IN\_0* a

*A* (stream input) Value for which the cosine will be calculated.

**See also:**

*sin*

## 2.12.7 divide

### Description

Divides one number by the other number and outputs the result using a simple  $A / B$  math operation. Remember to avoid dividing by zero.

### Inputs, output and other parts

*A* (stream input) The first value, which will be divided by *B*.

*B* (stream input) The second value, by which the *A* will be divided. Must not be zero.

### See also:

*multiply, add, subtract, divide\_jolt*

## 2.12.8 divide\_jolt

### Description

Divides incoming jolt number by the set stream value and outputs the result as jolt. Remember to avoid dividing by zero.

### Inputs, output and other parts

*Calculate* (jolt input) Receives value which will be divided by the stream value, triggers the calculation.

*toggle* Toggle

*Result* (jolt output) Emits calculated value as jolt.

*Divider* (stream input) Sets the value by which the jolt input value will be divided (without triggering the output). Must not be zero.

### See also:

*multiply\_jolt, add\_jolt, subtract\_jolt, divide*

## 2.12.9 floor

### Description

Rounds input float numbers down to their nearest integer values (i.e. 5.9 will output 5).

### Inputs, output and other parts

*Input* (stream input) The number that will be rounded down to the nearest whole number.

### See also:

*round, modulo, floor\_jolt*

### 2.12.10 floor\_jolt

#### Description

Rounds jolt input float numbers down to their nearest integer values (i.e. 5.9 will output 5).

#### Inputs, output and other parts

*Input* (jolt input) Jolt that will be rounded down to the nearest whole number.

*Output* (jolt output) Emits result of rounding down jolt input to nearest whole number.

#### See also:

*modulo\_jolt, floor*

### 2.12.11 from\_db

#### Description

Converts value from dB to amplitude

#### Inputs, output and other parts

*s\_IN\_0* a

*Input* (stream input) Value which will be converted from dB.

#### See also:

*to\_db*

### 2.12.12 log

#### Description

Returns the logarithm of value A to the base of B value.

#### Inputs, output and other parts

*A* (stream input) Logarithm value A.

*B* (stream input) Logarithm base value B.

#### See also:

*power, log\_jolt*

### 2.12.13 log\_jolt

#### Description

Returns the logarithm of jolt input to the base of the stream value.

#### Inputs, output and other parts

*Base* (stream input) Sets the logarithm base value (without triggering the output).

*Calculate* (jolt input) Receives value of which the logarithm value will be calculated, triggers the calculation.

*Result* (jolt output) Emits calculated value as jolt.

*toggle* Toggle



**See also:**

*power\_jolt, log*

### 2.12.14 map

**Description**

Scales values from a defined (min & max) input range to defined (min & max) output range. For example can be used to scale slider 0-1 range to 1-1000 range.

**Inputs, output and other parts**

*Input* (stream input) Value to be mapped.

*Input range min* (stream input) Input range lower value.

*Input range max* (stream input) Input range upper value.

*Output range min* (stream input) Output range lower value.

*Output range max* (stream input) Output range upper value.

**See also:**

*clamp*

### 2.12.15 max

**Description**

Returns the highest of the two (A and B) incoming stream values.

**Inputs, output and other parts**

*A* (stream input) Value A

*B* (stream input) Value B

**See also:**

*min*

### 2.12.16 midi\_to\_freq

**Description**

Converts MIDI note number (0-127) to corresponding frequency (Hz). E.g. 69 is A5 (440Hz) and 60 is C4 (261.63Hz).

**Inputs, output and other parts**

*MIDI note* (stream input) MIDI note value (0-127) to convert.

**See also:**

*midi\_to\_freq\_jolt*

### 2.12.17 midi\_to\_freq\_jolt

#### Description

Converts MIDI note number (0-127) jolt to corresponding frequency (Hz) jolt. E.g. 69 is A5 (440Hz) and 60 is C4 (261.63Hz).

#### Inputs, output and other parts

*MIDI note input* (jolt input) Receives a jolt with value of MIDI note to convert.

*Output* (jolt output) Emits jolt with calculated note frequency.

#### See also:

*midi\_to\_freq*

### 2.12.18 min

#### Description

Returns the lowest of the two (A and B) incoming values.

#### Inputs, output and other parts

*A* (stream input) Value A

*B* (stream input) Value B

#### See also:

*max*

### 2.12.19 modulo

#### Description

Returns the remainder after dividing stream A by stream B.

#### Inputs, output and other parts

*A* (stream input) Dividen value.

*B* (stream input) Divisor value.

#### See also:

*round, floor, divide, modulo\_jolt*

### 2.12.20 modulo\_jolt

#### Description

Returns the remainder after dividing jolt by stream input value.

#### Inputs, output and other parts

*Divisor* (stream input) Sets the value by which the jolt will be divided (without triggering the output).

*Calculate* (jolt input) Receives value which will be divided by stream input value, triggers the calculation.

*Result* (jolt output) Emits calculated remainder value as jolt.

*toggle*

**See also:**

*floor\_jolt, divide\_jolt, modulo*

### 2.12.21 multiply

#### Description

Multiplies two numbers and outputs the result using a simple  $A * B$  math operation.

#### Inputs, output and other parts

*A* (stream input) The first value, multiplied with *B*.

*B* (stream input) The second value, multiplied with *A*.

**See also:**

*divide, add, subtract, multiply\_jolt*

### 2.12.22 multiply\_jolt

#### Description

Calculates multiplication of the jolt value with the stream value and outputs the result as jolt.

#### Inputs, output and other parts

*Multiplier* (stream input) Sets the value with which the jolt input value will be multiplied (without triggering the output).

*Result* (jolt output) Emits calculated value as jolt.

*Calculate* (jolt input) Receives value which will be multiplied with the stream value, triggers the calculation.

**See also:**

*divide\_jolt, add\_jolt, subtract\_jolt, multiply*

### 2.12.23 power

#### Description

Calculates stream value *A* to the power of stream value *B*

#### Inputs, output and other parts

*A* (stream input) Base value *A*

*B* (stream input) Exponent value *B*

**See also:**

*log, power\_jolt*

### 2.12.24 power\_jolt

#### Description

Calculates jolt input to the power of stream input.

#### Inputs, output and other parts

*Exponent* (stream input) Sets the exponent value (without triggering the output).

*Calculate* (jolt input) Receives value which will be raised to the power of stream input, triggers the calculation.

*toggle* Toggle

*Result* (jolt output) Emits calculated value as jolt.

#### See also:

logarithm\_jolt, *power*

### 2.12.25 previous

#### Description

Delays the incoming stream by a single sample.

#### Inputs, output and other parts

*Input* (stream input) The stream to delay by a single sample.

### 2.12.26 random

#### Description

Outputs a random decimal number between 0 and  $\langle b \rangle \text{Max} \langle /b \rangle$ .

#### Inputs, output and other parts

*Button* (interactive) Triggers the output to emit a new random value.

*Max random value* (interactive) Sets random max value.

*Calculate* (jolt input) Send a jolt value to this input to trigger this block to emit a new random value. The value of the jolt will determine the maximum value of the random number.

*Output* (jolt output) Emits random value.

#### See also:

*noise, pink\_noise, bernoulli*

### 2.12.27 round

#### Description

Rounds incoming number with decimal value to the nearest whole number.

#### Inputs, output and other parts

*s\_IN\_0* a

*Input* (stream input) The number that will be rounded to the nearest whole number.

**See also:**

*floor, modulo*

### 2.12.28 sign

**Description**

Sign of incoming value: 1 if positive, -1 if negative.

**Inputs, output and other parts**

*s\_IN\_0* a

*Input* (stream input) Value for which the sign will be determined.

**See also:**

*compare*

### 2.12.29 sin

**Description**

Calculates sine value of A

**Inputs, output and other parts**

*s\_IN\_0* a

*A* (stream input) Value for which the sine will be calculated.

**See also:**

*cos*

### 2.12.30 subtract

**Description**

Subtract one number from another and outputs the result using a simple  $A - B$  math operation, can be used to calculate difference between two signals.

**Inputs, output and other parts**

*A* (stream input) Value from which the B will be subtracted

*B* (stream input) Value to be subtracted from A

**See also:**

*add, multiply, divide, subtract\_jolt*

### 2.12.31 subtract\_jolt

#### Description

Subtracts stream number from jolt input and outputs the result as jolt.

#### Inputs, output and other parts

*Subtract* (stream input) Sets the value to be subtracted from the jolt input value (without triggering the output).

*Calculate* (jolt input) Receives value from which the stream value will be subtracted, triggers the calculation.

*Result* (jolt output) Emits results of subtracting stream value from jolt value on each received jolt.

*toggle* Toggle

#### See also:

*add\_jolt, multiply\_jolt, divide\_jolt, subtract*

### 2.12.32 tanh

#### Description

Hyperbolic tangent of the stream input. Can be used to create for example soft clipping at 1 and -1.

#### Inputs, output and other parts

*s\_IN\_0* a

*Input* (stream input) Value for which the hyperbolic tangent will be calculated.

#### See also:

*wavefolder*

### 2.12.33 to\_db

#### Description

Converts from amplitude to dB

#### Inputs, output and other parts

*s\_IN\_0* a

*Input* (stream input) Amplitude value which will be converted to dB.

#### See also:

*from\_db*

## 2.13 Other

### 2.13.1 palette

#### Description

Used in VR to spawn block and access console.

#### Inputs, output and other parts

*toggle\_minimize*

*0*

*1*

*2*

*5*

*6*

*8*

*3*

*4*

*7*

*9*

*.*

*/*

*-*

*A*

*C*

*D*

*E*

*B*

*H*

*I*

*G*

*F*

*J*

*K*

*N*

*L*

*M*

*O*

*Q*

*P*

*R*

*T*

*U*

*V*

*S*

*W*

*Y*

*• -*

*CapsLock*

*X*

*Z*

*b*

*r*

*up*

*down*

*left*

*right*

*“*

*:*

*utor*

*ntor*

*rtor*

*s\_mul*

*s\_add*

*s\_div*

*s\_sub*

*clock*

*osc*

*noise*

*decay*

*freeverb*

*ladder*

*statevariable*

*connellyextender*

*split*



*spacer*  
*elbow*  
*keys\_tab*  
*s\_tab*  
*e\_tab*  
*io\_tab*  
*speaker*

## 2.14 Others

### 2.14.1 block\_watcher

#### Description

Attaches to other blocks and emits jolts when stuff happens.

#### Inputs, output and other parts

*Selector* (selector) Pull out sticks to attach blocks to be watched.

*Hover* (jolt output) Emits a jolt when you start or stop hovering your controller over an attached block.

*Grab* (jolt output) Emits a jolt when you grab or release an attached block.

*Hit* (jolt output) Emits the velocity of the controller when you hit and attached block (if it's hittable).

*Marble* (jolt output) Emits the velocity of the marble when an attached block gets hit by a marble (if it supports marble hits).

### 2.14.2 trigger\_box

#### Description

A resizable area that can detect if a player or a ghost is inside. Detection of ghosts can be disabled via the inspector.

#### Inputs, output and other parts

*Player inside* (jolt output) Emits a jolt with value 1 when the player (or a ghost) enters the area and a 0 when they leave.

*Resize handle* (interactive) Grab and move to resize the trigger area.

## 2.15 Props

### 2.15.1 particles\_sprite

#### Description

#### Inputs, output and other parts

*Emission Freq*

*Emission Radius*

*Emission Random Dir*

*Color H*

*Color S*

*Color V*

*Particle Speed*

*Particle Size*

*Particle Lifetime*

*Noise intensity*

*Noise frequency*

*Image block*

## **2.15.2 prop\_firstmask**

**Description**

**Inputs, output and other parts**

*VR Grab Handle Named*

## **2.15.3 prop\_neuron**

**Description**

**Inputs, output and other parts**

*VR Grab Handle Named*

## **2.15.4 prop\_particles**

**Description**

**Inputs, output and other parts**

*VR Grab Handle Named*

## **2.15.5 prop\_underwaterparticles**

**Description**

**Inputs, output and other parts**

*VR Grab Handle Named*

## 2.15.6 prop\_wireframe

### Description

### Inputs, output and other parts

*VR Grab Handle Named*

## 2.16 Samples

### 2.16.1 sample

#### Description

#### Inputs, output and other parts

*Grab\_interactiveBase*

*Hit\_interactiveBase*

## 2.17 Sky

### 2.17.1 venue\_lightsaber

#### Description

Background for the environment

write in the console:

spawn venue\_lightsaber

#### Inputs, output and other parts

*VR Grab Handle*

### 2.17.2 venue\_mountains

#### Description

Background for the environment

write in the console:

spawn bg\_mountains

#### Inputs, output and other parts

*VR Grab Handle*

### 2.17.3 venue\_ozelot

#### Description

Background for the environment

write in the console:

spawn venue\_ozelot

#### Inputs, output and other parts

*VR Grab Handle*

### 2.17.4 venue\_spacestation

#### Description

Background for the environment

write in the console:

spawn bg\_spacestation

#### Inputs, output and other parts

*VR Grab Handle*

### 2.17.5 venue\_underwater

#### Description

Background for the environment

write in the console:

spawn venue\_underwater

#### Inputs, output and other parts

*VR Grab Handle*

## 2.18 Stages

### 2.18.1 prop\_highplatform

#### Description

#### Inputs, output and other parts

*VR Grab Handle Named*

## 2.18.2 prop\_platform

### Description

### Inputs, output and other parts

*VR Grab Handle Named*

## 2.18.3 prop\_swimmingpool

### Description

### Inputs, output and other parts

*VR Grab Handle Named*

## 2.19 Terrains

### 2.19.1 venue\_wireframe

#### Description

#### Inputs, output and other parts

*VR Grab Handle*

## 2.20 Time

### 2.20.1 beat\_time

#### Description

The progress of your beat. You can control tempo / stop / restart with timecontrol block.

#### Inputs, output and other parts

*Sphere*

*e\_OUT*

### 2.20.2 clock

#### Description

The clock allows you to set an increase to your given value every second.

#### Inputs, output and other parts

*s\_IN* (INa)

*eR\_seState* set clock state

*eR\_setState*

*Sphere*

### 2.20.3 delay

#### Description

Creates an audio-rate delay line with streams.

#### Inputs, output and other parts

*input* (stream input) Stream input

*delay* Delay time in seconds

### 2.20.4 delay\_jolt

#### Description

Delay a jolt by some amount of time.

#### Inputs, output and other parts

*input* (jolt input) The jolt to delay.

*speed\_of\_time* (stream input) A multiplier for the delay constant.

*delay\_time* (stream input) How much to delay the jolt.

*output* (jolt output) Sends out the same jolt as received through 'input', but delayed by the time set via the 'delay\_time' stream input.

### 2.20.5 eggtimer

#### Description

The egg timer delays an event depending on the value of INa

#### Inputs, output and other parts

*s\_IN* (INa) delay time (seconds)

*eR\_hot* input

*e\_OUT*

### 2.20.6 metronome

#### Description

Ticks off regular events at a specified interval.

#### Inputs, output and other parts

*s\_IN* (INa) freq

*eR\_set* Set phase

*Emitter* Emits events, with each tick having a value of 1

## 2.20.7 rhythm

### Description

Outputs events at musical intervals. The master beat\_time is used as the default clock, unless something is attached to the clock stream input. Also outputs a normalized stream value indicating how far along we are in the period.

### Inputs, output and other parts

*offset* (knob) How many beats the output event is offset in time compared to the input clock.

*period* (knob) How many subdivisions there are pr. period. E.g. a value of 3 will cause an event to fire every third subdivision.

*subdivision* (knob) How many times to divide the beat when counting a period. E.g. a value of 2 will count two times every beat.

*clock* (stream input) Usually an increasing counter, counting the beats that determine when the period\_begin should fire.

*period\_begin* (event output) Emits an event whenever a new period starts.

*Sphere*

## 2.20.8 timecontrol

### Description

Pause / Resume / speed up tempo affecting phasors.

### Inputs, output and other parts

*Grab sphere VR*

*Back\_Button\_*

*Pause\_Button\_*

*Skip\_Button\_Left*

*Skip\_Button\_Right*

*Forward\_Button\_End*

*tempo\_knob*

## 2.21 Unclassified

### 2.21.1 prop\_neurons

#### Description

#### Inputs, output and other parts

*VR Grab Handle Named*

## 2.21.2 venue\_brain

### Description

Background for the environment

write in the console:

spawn venue\_brain

### Inputs, output and other parts

*VR Grab Handle*

## 2.22 Venue

### 2.22.1 bg\_nebula

#### Description

Background for the environment

write in the console:

spawn bg\_nebula

#### Inputs, output and other parts

*VR Grab Handle*

## 2.23 Visual

### 2.23.1 camera

#### Description

Records the gameplay as a video file. Active cameras appear as the gameview. You can record or set cameras active either through the 2D desktop GUI or by pressing the buttons physically on the component in VR. Press the record/stop icon for recording, press the camera icon to set cameras active. Press the folder icon to open the saving destination, it is set to `\\C:\Users\XXX\Documents\PatchXR\`

#### Inputs, output and other parts

\*\*\* connect a button and press it to set cameras active

*camera*



### 2.23.2 cameramanager

#### Description

Gives you a preview screen and manages cameras

#### Inputs, output and other parts

\*\*\* \*

*cameramanager*

### 2.23.3 coloroven

#### Description

#### Inputs, output and other parts

*red*

*green*

*blue*

*body*

### 2.23.4 device\_box

#### Description

A resizable and stylizable box to make interfaces for your devices.

#### Inputs, output and other parts

*Resize* (interactive) Grab and move to resize the box.

### 2.23.5 image

#### Description

#### Inputs, output and other parts

*MuX\_ImgCorner\_Msh*

*grab*

*esReciver*

### 2.23.6 pencil

#### Description

Allow you to draw in space, using a color, width of the line and lifetime.

#### Inputs, output and other parts

*k\_width* Width of the line.

*k\_life* Life time of the line.

*k\_HSV\_H* Color : Hue

*k\_HSV\_S* Color : Saturation

*k\_HSV\_V* Color : Brightness

### 2.23.7 plot

#### Description

#### Inputs, output and other parts

*s\_IN*

*esReciver*

### 2.23.8 scope

#### Description

Displays signal data.

#### Inputs, output and other parts

*s\_IN\_0*

*s\_IN\_1\_inWindow*

*toggle*

### 2.23.9 spotlight

#### Description

HSV-colored spot light.

#### Inputs, output and other parts

*hue*

*saturation*

*value*

*Angle*

### 2.23.10 trail\_pencil

#### Description

#### Inputs, output and other parts

*Width*

*Trail Life Time*

*Hue*

*Saturation*

*Brightness*

### 2.23.11 txt

#### Description

Let you place floating text.

#### Inputs, output and other parts

*text\_IN*



**EXAMPLES**

example are...



---

**CHAPTER  
FOUR**

---

**VIDEOS**

Please Visit our [YouTube channel](#)





Misc blocks are...



## **GET STARTED**

For more information about Patch, please see our website. [www.patchxr.com](http://www.patchxr.com)



## OTHER RESOURCES

Find us on [Discord](#) Find us on the [PatchWorld Forum](#)