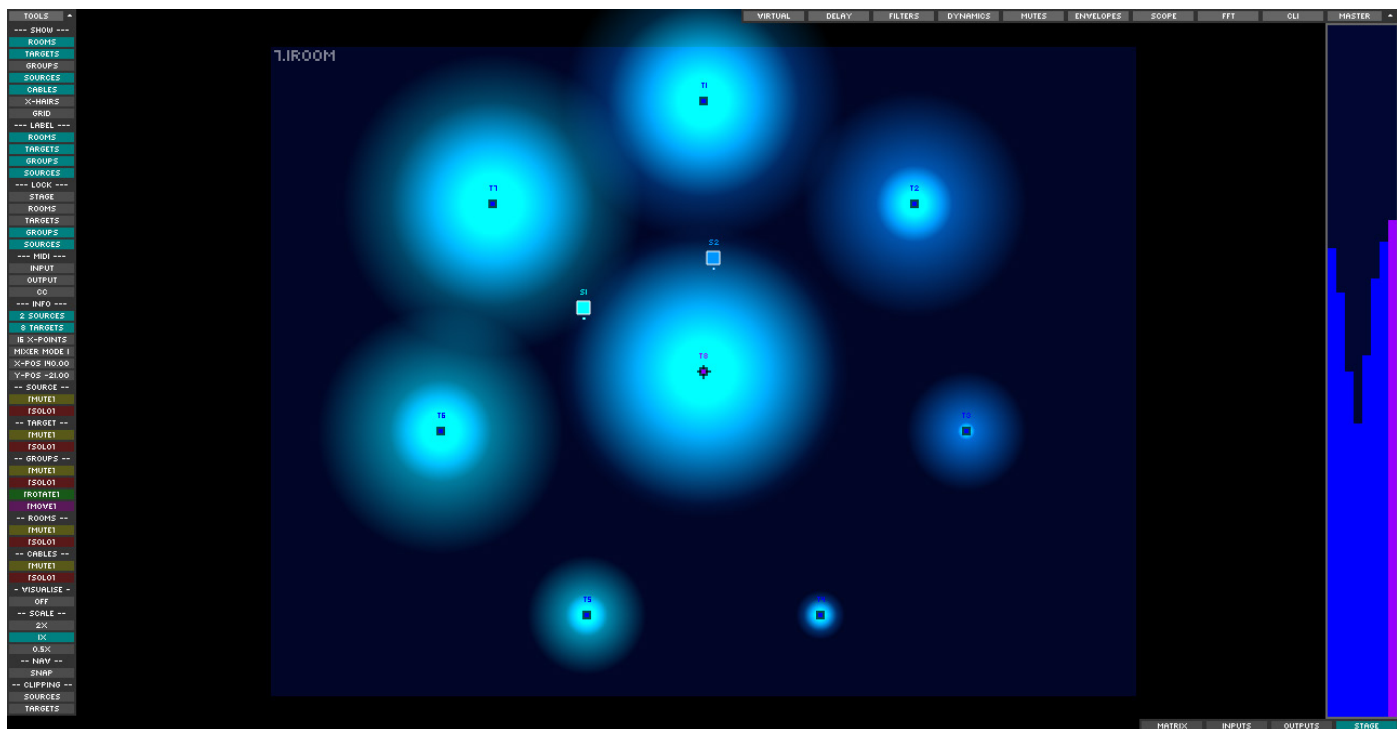


SOUNDSQUARES

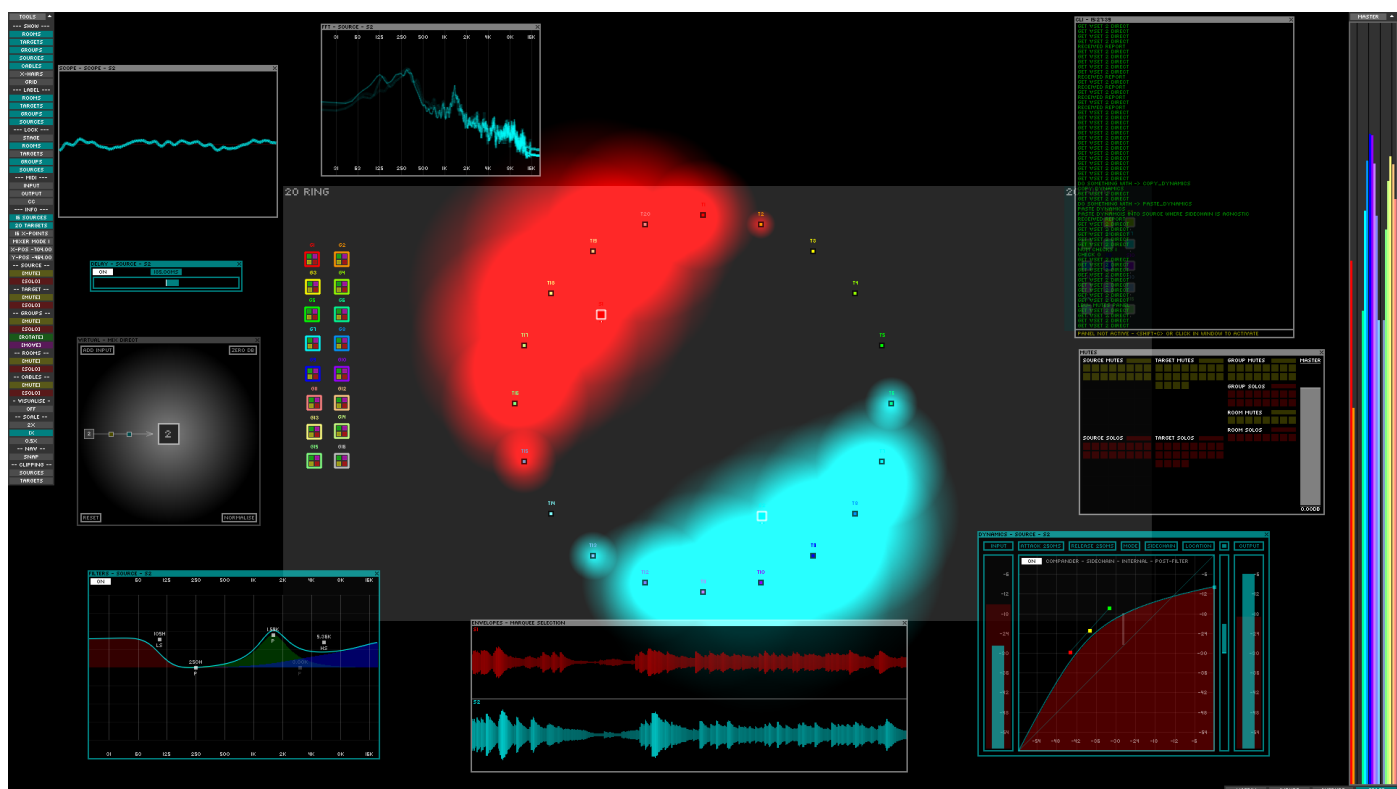
v1.F



basic 7.1 system

SOUNDSQUARES

v1.F



circular 20 speaker system

INDEX

INTRODUCTION :	1
DESCRIBING THE PROBLEM	1
BUT WHY SOUNDSQUARES?	2
TOPOLOGY :	4
INTERFACE :	5
STAGE PAGE	5
SOURCES	6
TARGETS	8
GROUPS	10
ROOMS	12
CABLES & CHAINS	14
MARQUEE	16
SUB PANELS & HOT KEYS	17
TOOLBOX	18
VIRTUAL	19
DELAY	20
FILTERS	21
DYNAMICS	22
MUTES	23
ENVELOPES	24
SCOPE & FFT	25
CLI	26
MASTER	27
DSP PIPELINE DEBUG OVERLAY	28
TARGET HEATMAP	29
INPUT / OUTPUT MIXER PAGES	30
MATRIX MIXER AND ROUTING PAGE	32
SYSTEM REQUIREMENTS	36
SETTING UP THE HOST DAW	37
SETTING UP SOUNDSQUARES	40
USAGE EXAMPLES	
MAKING A CIRCULAR SYSTEM	40
ADDING SUBWOOFERS	42
PROTECTION LIMITERS	44
MID / SIDE PROCESSING	46
MULTIBAND PROCESSING	48
ON ENVELOPE FOLLOWING	49

DESCRIBING THE PROBLEM

Imagine a scenario involving multiple loudspeakers deployed in a space. It might be a museum or art gallery, performance space, or other public environment, but unlike a generic public-address system in a supermarket where one sound source is heard through all speakers simultaneously, the usage demands that different sounds come from different speakers, and that their distribution be fluidly and reliably orchestrated.

This is well understood in the context of cinema where a number of speakers are used to create the experience, usually organised into a 5.1 topology where the 5 main speakers are used for mid and high frequency content, and the .1 speaker is used for low frequencies. This is sometimes extended to systems using the naming convention main-dot-sub, such as 7.1 as used in computer gaming setups, and higher-end cinema systems using arrangements of up to 22.2.

Broadly speaking, these dot-format speaker arrangements are usually arranged in an approximation of a circle with the audience expected to be located in static positions within the circle. In the case of larger cinematic systems the circle of speakers may break-out into 3 dimensions, but still generally follows the same notion of an audience seated inside of a uniform array.

But what if instead of addressing an already codified arrangement of speakers, the environment features an arbitrary set of locations, with no fixed position for the audience?

This presents several challenges when creating an audio installation or performance work. Let's put aside creative, aesthetic, theoretical, or critical reasoning behind the design of the physical layout itself, and instead look for a minute at the simple problem how to address a system using readily accessible audio production platforms and tools.

Looking at the accessibility and availability of tools based on the output format might help illuminate the scenario :

Stereo : cheap and powerful software mixing tools, widely available for almost 30 years now, and a set of well codified conventions as to their use - main narrative (or singing) voice placed in the centre of the stereo image, atmos and other spatial elements distributed between the two speakers to create a virtual sound-stage or stereo image, whether recorded or constructed.

5.1 : whilst also powerful, generally not so cheap or readily available (or widely used outside of cinema), but does, broadly speaking have a set of conventions as to what belongs where in the mix - again voices front-and-centre, environment sounds placed accordingly.

Ambisonics and pseudo holophonic arrays approaching wavefront synthesis : whilst the tools are certainly powerful, and the expectations of the 'virtual reality' these systems enable results in quite strict conventions, this is a somewhat more costly niche, which whilst some tools are freely available, their use is rather rarified, and somewhat less accessible than mixing to a more widely available commercial format.

Beyond and inbetween these formats : whilst in academic settings there are many powerful tools freely available, their use is even more niche, and whilst the creators challenge and construct the conventions of how the systems are used, there is yet to emerge any singular standard, and whilst 8 speakers in a ring with a centrally seated audience is somewhat of a common situation, these setups are not widely available outside of academia.

SoundSquares, then, is intended to facilitate the use of a large number of loudspeakers positioned in an arbitrary fashion, where for example, a lack of audience centrality challenges notions of the ideal listening spot, and intentions for output experience are not couched in terms of virtual sound stage or image. It is a mixing and audio distribution tool far more than it is a tool for the creation of virtual environments in the strictest of senses.

BUT WHY SOUNDSQUARES?

The sub-heading to this question might be - “aren’t there already a range of commercially available hardware mixers available that can do this?” ... well ... beyond the generic topology of a “regular” mixing desk, where multiple inputs are essentially funnelled to a small number of physical outputs, the way to handle an “any sound anywhere” situation would be to use a matrix mixer, basically a multiplexed patchbay where signals from any input can be arbitrarily routed to any output, and any output can receive any combination of input signals.

In commercial venue settings, it is not uncommon for hardware matrix mixers to be deployed “behind the scenes” and controlled via automation, wall-plates, and remote computer interfaces. Whilst on the DSP and routing side of things, these systems are incredibly flexible and obviously very powerful, they are generally not geared-up to the needs of an artist who might need to quickly and intuitively control the mix in realtime as part of a live performance. As powerful and available as these matrix based approaches are, they are just not designed with performance in mind, with their non-installer interfaces deliberately being reduced to a very cut-down set of features the installer implements before hand-off to the client - and the client’s eventual users - often resulting in an oversimplified ux-focus of “hit this button to bring the music into the bar, and this one to control the volume in the toilets”.

SoundSquares then, exists in a niche where the deployment of large speaker arrays crosses-over with the needs of artists, in both installation and performance, where existing conventions of both speaker position, and audience location need not apply.

Whilst at the level of DSP, SoundSquares also deploys a mixing matrix, it does so via an interface which presents a two dimensional visual mapping of the loudspeakers. Rather than rely entirely on a “cabled” or “point-to-point” topology, SoundSquares uses an intuitive visual approach to position sound source analogs in proximity to loudspeaker analogs as found in many 5.1 software mixer setups where a visual node, or perhaps a trackball, enables swift and accurate changes to be made. SoundSquares extends this regime far beyond mainstream main-dot-sub formats.

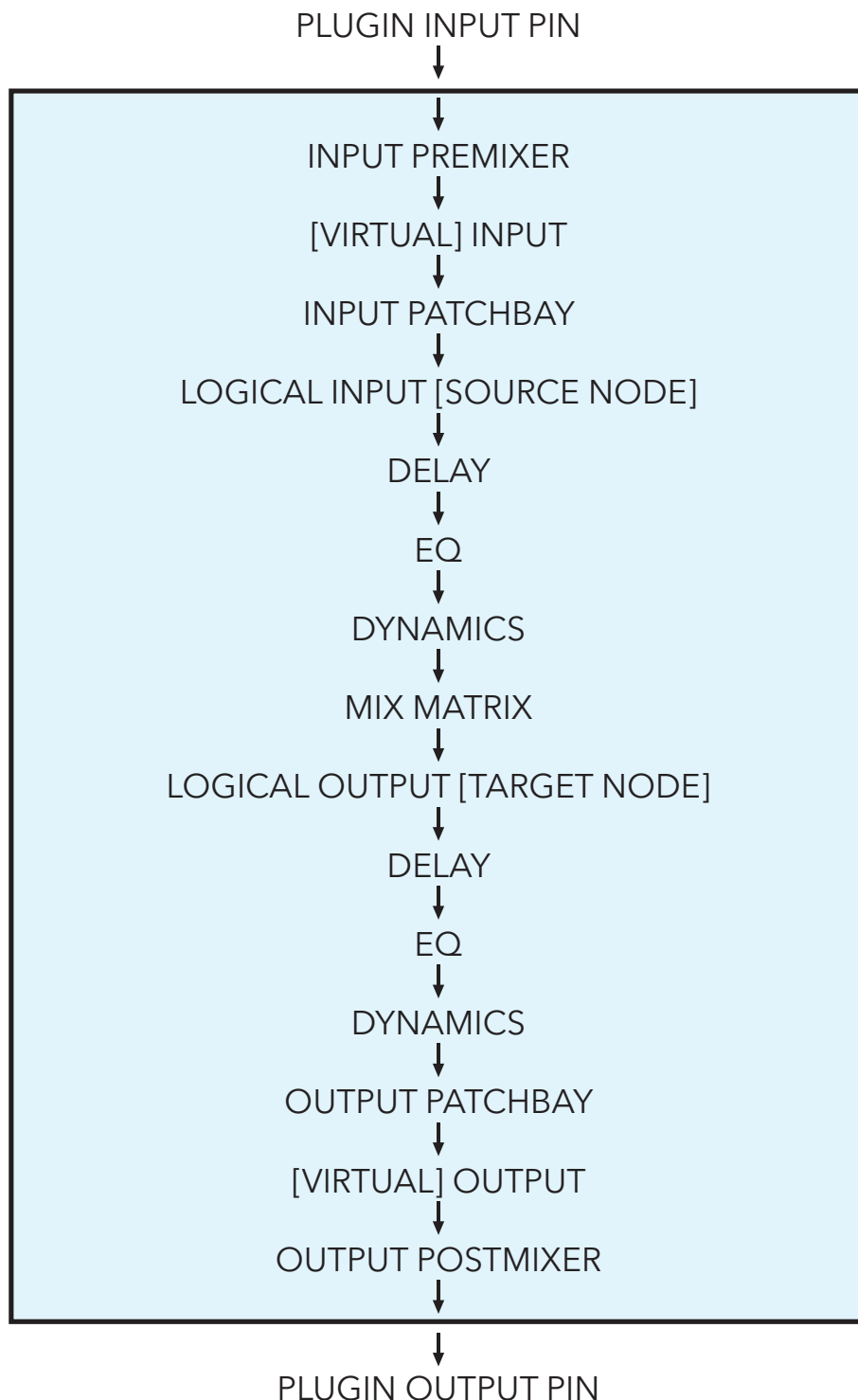
Imagine for a minute, you have 40 speakers, and an 8 channel recording to be positioned onto the speakers. Certainly this *is possible* using a point-to-point approach to routing a matrix, but will require quite some fine-tuning for the individual levels of the source’s sub-channels with respect to the speakers each is heard through. It may require as many as 320 cross-points within the matrix for every sound *everywhere*, however in practice it is much more likely to require somewhere between 8 and 40-or-so matrix cross-points ... and that’s still potentially a *lot* of time spent clicking-around with the mouse and a *significant* amount of brainshare required in understanding and remembering what you’re trying to achieve given the number of steps it might require.

Now imagine that this 8 channel recording is only one of many needing to be placed, and as you imagine this, bear in-mind that the public opening is only a few hours away and you’ve still got a bazillion parameters to have to refine just to get the positioning under control, and what about the crossovers, the filters, and the dynamics ... and the final rehearsal for the performance?

THAT’S WHY SOUNDSQUARES.

SoundSquares is implemented as a 32bit VST2 plugin for Windows. It is best used with the DAW Reaper - further setup details below - and as such it is assumed that all timeline and other creative filtering effects are managed using Reaper. SoundSquares is used on the master channel, or a group channel as a way to distribute the individual incoming channels to their relevant output locations. In SoundSquares there natively exists no such notion as Stereo or 5.1, and instead each input is considered its own singular, independent channel.

With that in mind, every plugin input routed from the DAW undergoes the following series of transformations before becoming a plugin output and going back into the DAW for further processing or routing directly to the output hardware.



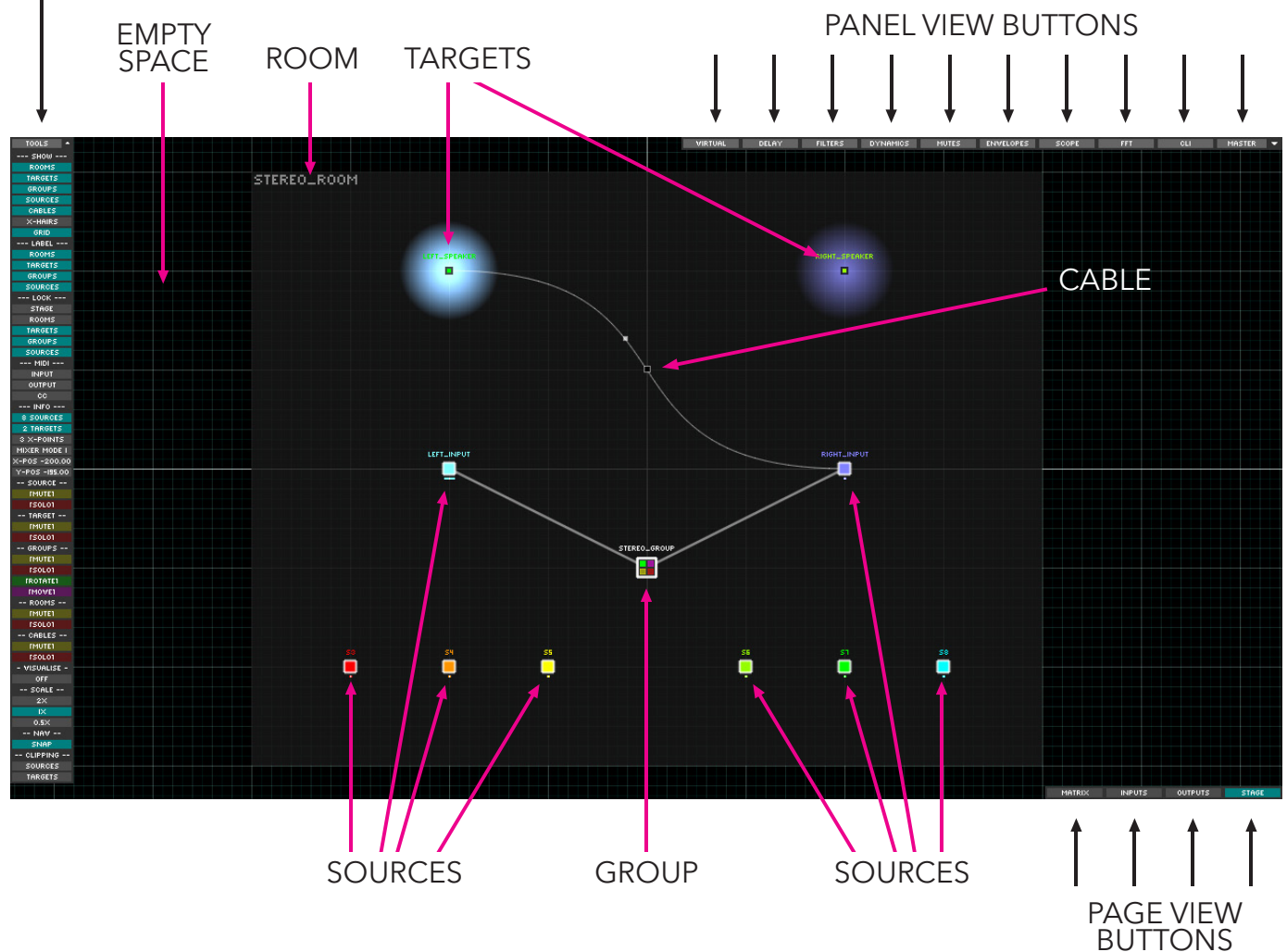
[*SHIFT + P*]

The SoundSquares interface is divided into 4 main pages, with each offering a different mode of address. It is best used 2-handed, with one hand on the mouse, and the other on the keyboard.

STAGE PAGE

This is where the majority of the action happens in SoundSquares, and the relationships between input and output are established and manipulated. Each element will be addressed individually below. For starters though, here an overview of the STAGE PAGE:

TOOLBOX



The background of the STAGE is considered as empty space ... everyone knows that “in space nobody can hear you scream” ... next-up the heirarchy is the ROOM, within which the SOURCES, GROUPS, and TARGETS can be found, along with CABLES.

In total, the STAGE may contain a maximum of 8 ROOMS, 64 SOURCES, 64 TARGETS, 16 GROUPS, and 1024 CABLES, which when combined might result in upto 4096 internal audio mixes in the central matrix mix stage of the pipeline.

When interacting with the STAGE PAGE, multiple *click+* modifiers are employed to perform different functions, in the same way one might *shift+click* or *ctrl+click* objects in other software. In SoundSquares, **many** additional **key+click** actions are used to enact a variety of functions.

SOURCE NODES

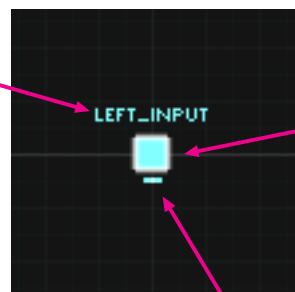
In SoundSquares, whilst a SOURCE is analogous to an input it does not necessarily have a one-to-one relationship with the plugin's actual input as seen from the host DAW. There is both the premix and the virtualisation stage to go through before audio arrives as the concept of a SOURCE. If both input premix and virtualisation steps are *transparent* then the SOURCE represents the plugin input, although there are good reasons these premixing steps exists, which will be revealed in the usage examples section.

A SOURCE then, is a carrier for an audio signal, and with the mouse can be freely moved around the room it is in. The spatial relationship between SOURCE and TARGET, for the most part, determines the volume of the SOURCE *through* the TARGET.

SOURCES can also be CABLED directly to a TARGET, in which case the proximity relationship is not used to determine the SOURCE'S volume in *that* TARGET, but instead a volume control *within* the CABLE (see below) is used to determine the SOURCE'S volume.

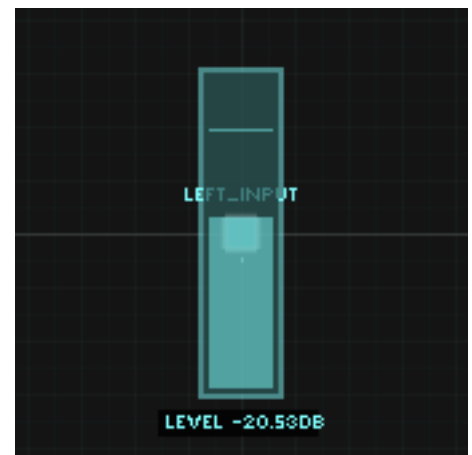
SOURCES can also be GROUPED, with GROUPS acting both as Voltage Controlled Amplifier (VCA) and also *grab-handles* enabling multiple SOURCES to be moved as a GROUP. A GROUP'S *mute*, *solo*, and *phase status* also applies to its SOURCE members - see section on GROUPS.

NAME LABEL

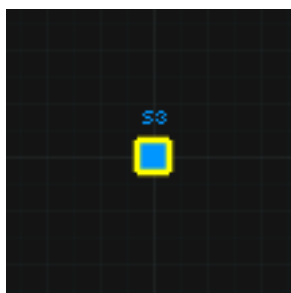


COLOUR

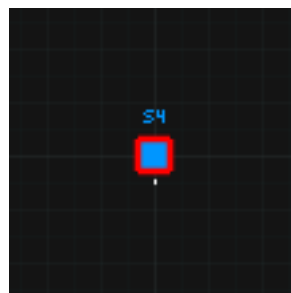
VU METER



SOURCE VOLUME
CONTROL - *ctrl+click*



MUTED
[YELLOW OUTLINE]



SOLO'D
[RED OUTLINE]



PHASE REVERSED
[CYAN INDICATOR]

SOURCE RELATIONSHIP WITH ROOMS

When moving a SOURCE, it is limited to being moved *within* the room it currently occupies. To move a source outside of a room, use *shift+drag*

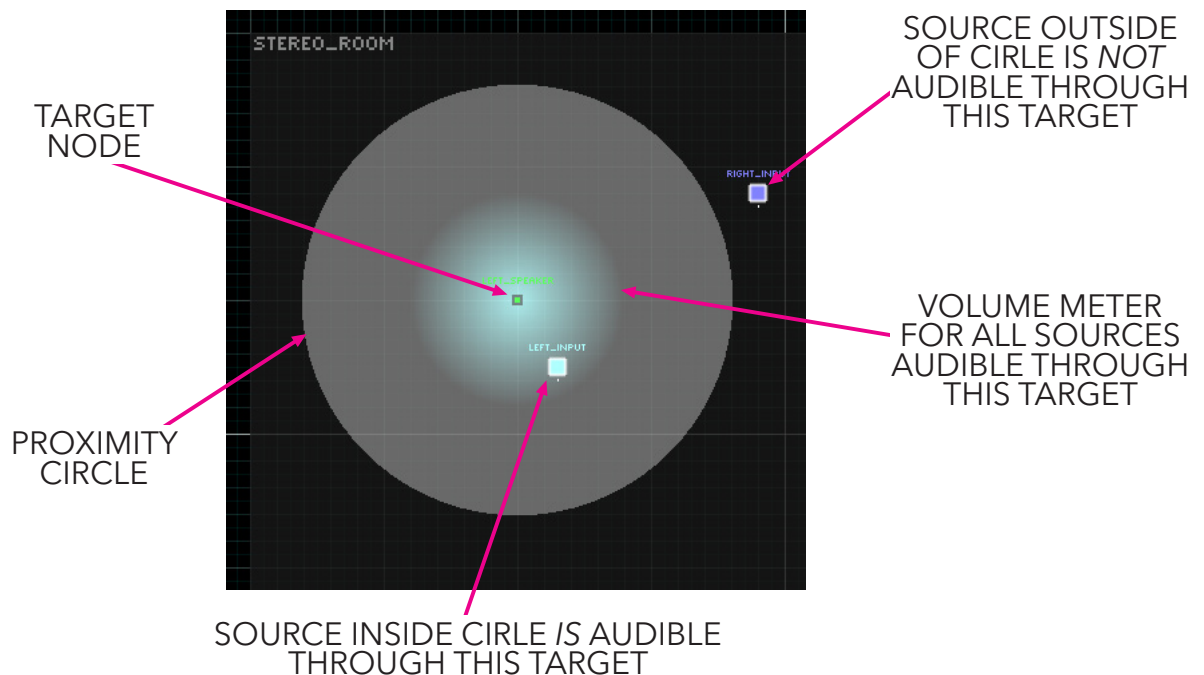
SOURCE MOUSE-CLICK ACTIONS

A	MUTE
S	SOLO
D	HIDE (VISUAL ONLY : KEEPS MIXING)
F	FREE / RELEASE FROM GROUP(S)
B 1,2,3...0 SHIFT + 1,2,3 ...0	SELECT COLOUR FROM DIALOG PRESET COLOUR 1 TO 10 PRESET COLOUR 11 TO 20
C	START / END CABLE IF CABLE COMES FROM A SOURCE IT BECOMES THIS SOURCE'S SIDECHAIN
X	REMOVE ALL CABLES
I	ROUTE TO SCOPE AND FFT VIEWER
E	ROUTE TO ENVELOPE VIEWER
P	SWITCH PHASE
Z	SET VOLUME TO ZERO dB
CTRL + DRAG	VOLUME SLIDER
SHIFT + DRAG	FREE DRAG (NOT BOUND TO ROOM)
CTRL + SHIFT	SPREAD SLIDER
SHIFT + Z	RESET SPREAD
SHIFT + X	REMOVE SIDECHAIN (RESET TO INTERNAL)
RIGHT CLICK	OPEN GROUP MEMBER & VIRTUAL CHANNEL DIALOG
DOUBLE CLICK	SET NAME
R	RESET

TARGET NODES

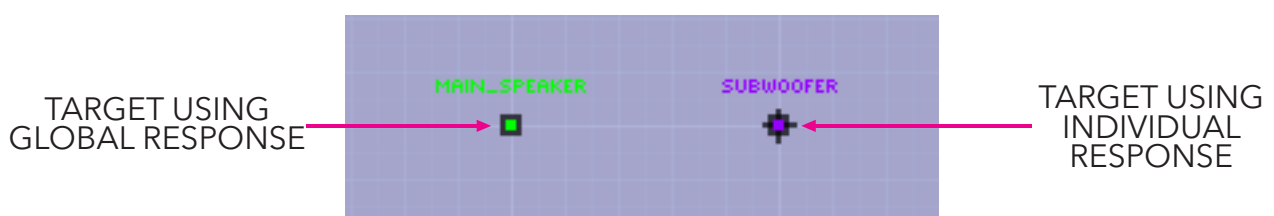
In SoundSquares, whilst a TARGET is analogous to an output it does not necessarily have a one-to-one relationship with the plugin's actual output as seen from the host DAW. There is both the virtualisation and postmix stage to go through before audio from a TARGET arrives at the plugin output pins. If both output virtualisation and postmix steps *are transparent* then the TARGET represents the plugin output, although there are good reasons these postmixing steps exists, which will be revealed in the usage examples section.

A TARGET then, is also carrier for an audio signal, and can be freely moved around the ROOM it occupies. The spatial relationship between SOURCE and TARGET, is determined by the TARGET'S *proximity circle* which defines the distance between SOURCE and TARGET where the SOURCE becomes *inaudible*. SOURCES have a control to individually *multiply* this distance using the *spread slider* via *ctrl+shift* drag.



TARGETS have the same *mute*, *solo*, and *phase* controls that SOURCES have with equivalent red, yellow, and cyan indicators. TARGET *response circles* are edited by *ctrl+click+dragging*, and individual TARGET *output level faders* can be controlled using *shift+click+dragging*.

In setting-up a system it is common for evenly spaced TARGETS to all use the same *response circle*, but there are circumstances where each might need its own independent *response*, such as the TARGET representing a *sub-woofer*. By default, all TARGETS used a *global response circle* as indicated by their square appearance. To switch a TARGET over to *individual response* use *backslash-click*, and it will be drawn as a square with protrusions indicating its singular response:



TARGET MOUSE-CLICK ACTIONS

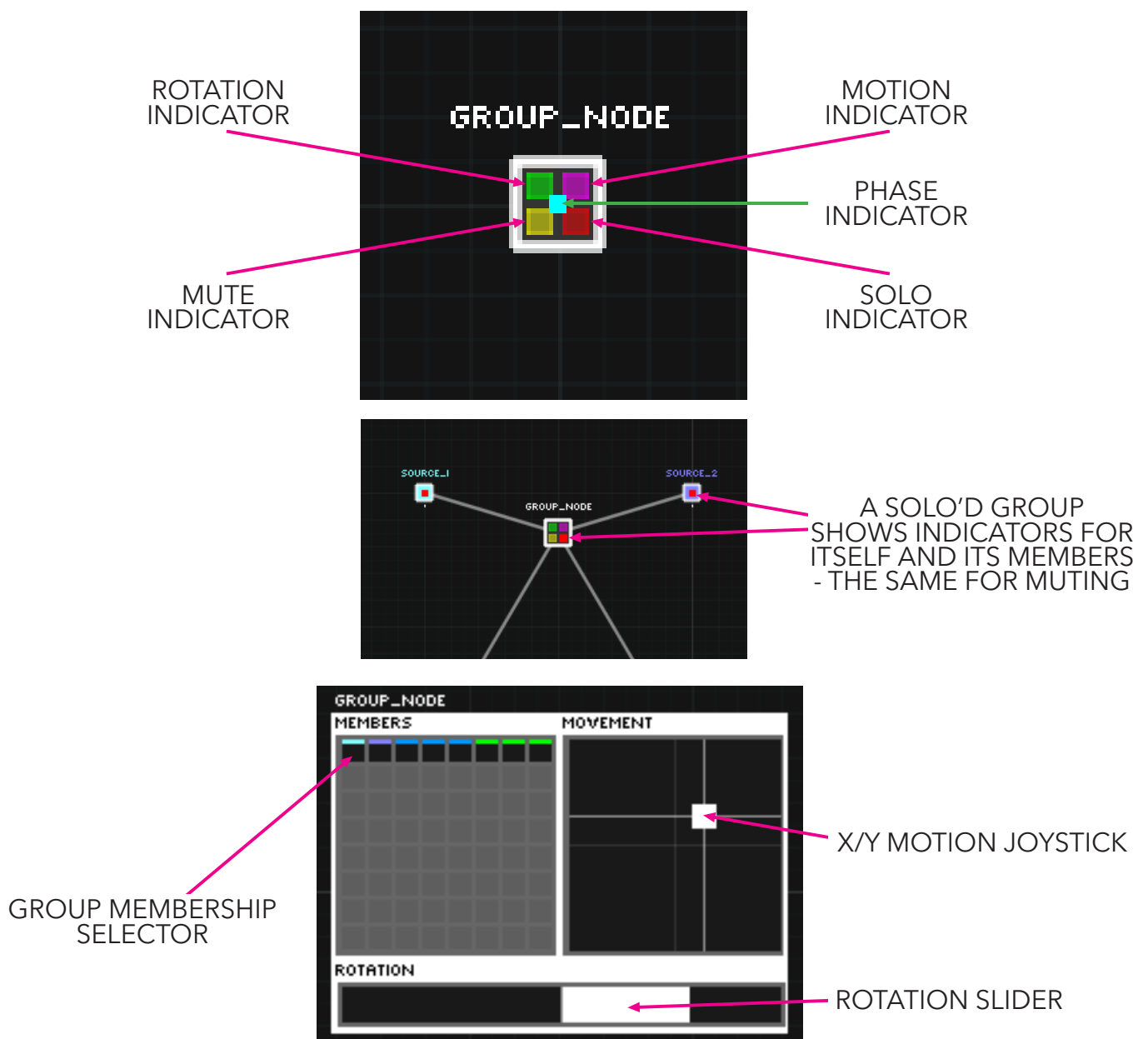
A	MUTE
S	SOLO IF CABLE STARTED, IT BECOMES THIS TARGET'S SIDECHAIN
D	HIDE (VISUAL ONLY : KEEPS MIXING)
B 1,2,3...0 SHIFT + 1,2,3 ...0	SELECT COLOUR FROM DIALOG PRESET COLOUR 1 TO 10 PRESET COLOUR 11 TO 20
C	START / END CABLE
X	REMOVE ALL CABLES
I	ROUTE TO SCOPE AND FFT VIEWER
E	ROUTE TO ENVELOPE VIEWER
P	SWITCH PHASE
Z	SET VOLUME TO ZERO dB
BACKSLASH	TOGGLE INDIVIDUAL / GLOBAL RESPONSE
CTRL + DRAG	CHANGE RESPONSE
SHIFT + DRAG	VOLUME SLIDER
CTRL + SHIFT	FREE DRAG (NOT BOUND TO ROOM)
SHIFT + X	REMOVE SIDECHAIN (RESET TO INTERNAL)
RIGHT CLICK	OPEN VIRTUAL CHANNEL DIALOG
DOUBLE CLICK	SET NAME
R	RESET

GROUP NODES

Acting as VCA's for their *members*, GROUPS can also be *muted* and *solo'd*, and have their *phase* toggled. Any changes to the GROUP are applied to the *members* **in addition** to whatever settings each individual *member* may have : a SOURCE which has *reversed phase* when placed into a GROUP which also has *reversed phase* will therefore have *double-reversed phase* - i.e. normal phase as a result of **both** phase switches being employed.

In addition to their role as VCA, GROUPS are also used as part of the positioning and movement of SOURCES. Right-clicking a GROUP opens the *group selector dialog* where the members of the GROUP can be determined along with both *rotation* **and** *linear motion* controls. When this dialog is closed, the status of *mute*, *solo*, *phase*, *rotation*, and *motion* is displayed on the GROUP'S node handle.

The *volume slider* for a GROUP, like SOURCES, is accessed via *ctrl+click*. SOURCES may be members of multiple GROUPS, in which case their VCA functions cascade.



GROUP MOUSE-CLICK ACTIONS

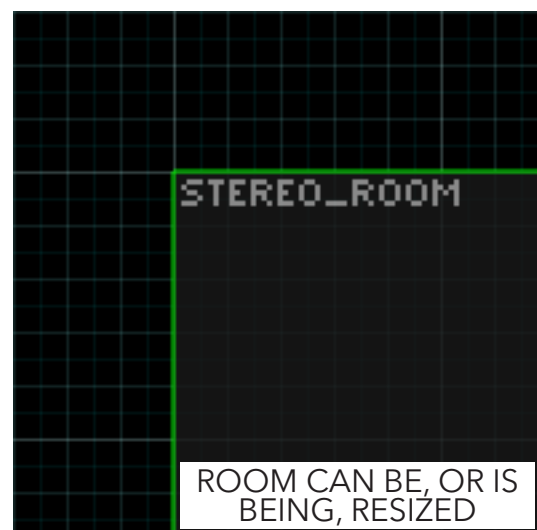
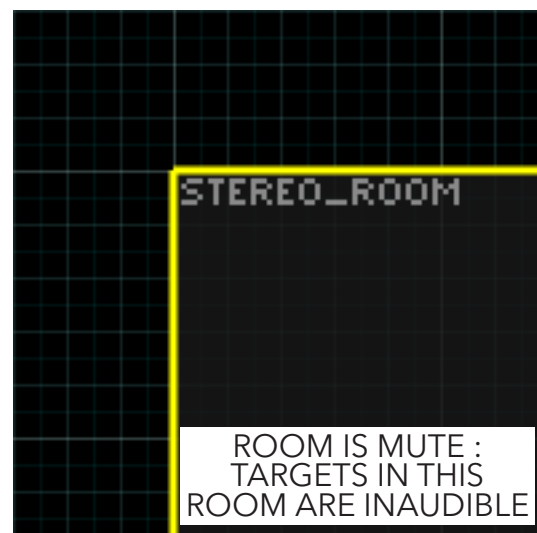
A	MUTE
S	SOLO
D	HIDE (VISUAL ONLY : KEEPS MIXING)
F	FREE ALL SOURCES FROM GROUP
B	SELECT COLOUR FROM DIALOG
1,2,3...0	PRESET COLOUR 1 TO 10
SHIFT + 1,2,3 ...0	PRESET COLOUR 11 TO 20
X	REMOVE CABLES FROM MEMBER SOURCES
I	ROUTE MEMBERS TO ENVELOPE VIEWER
Q	TOGGLE ROTATION
E + DRAG	ADJUST ROTATION SPEED
V + DRAG LEFT/RIGHT	ROTATE MEMBERS MANUALLY
CTRL + CLICK ROTATION SLIDER TO RESET ROTATION	
W	TOGGLE MOTION
T + DRAG	ADJUST MOTION SPEED & DIRECTION
CTRL + CLICK MOTION JOYSTICK TO RESET X AXIS	
SHIFT + CLICK MOTION JOYSTICK TO RESET Y AXIS	
CTRL + SHIFT + CLICK MOTION JOYSTICK TO RESET X & Y AXIS	
C + DRAG	EXPAND / CONTRACT GROUP
+ SHIFT	HORIZONTAL ONLY
+ CTRL	VERTICAL ONLY
P	SWITCH PHASE
Z	SET VOLUME TO ZERO dB
BACKSLASH	TOGGLE INDIVIDUAL / GLOBAL RESPONSE
CTRL + DRAG	VOLUME SLIDER
SHIFT + DRAG	MOVE WITHOUT MOVING MEMBERS
CTRL + SHIFT	MOVE MEMBERS NOT BOUND TO ROOM
RIGHT CLICK	OPEN MEMBERSHIP SELECTION DIALOG
DOUBLE CLICK	SET NAME
R	RESET

ROOMS

As previously hinted at in the note about the STAGE, ROOMS are a way to discretise the SOURCE-TARGET relationship. Only when a SOURCE is in the same ROOM as a TARGET is it audible through that TARGET. ROOMS in SoundSquares are also analogous to *output groups* on a regular mixing desk, and in addition to being containers of SOURCES and TARGETS operate as VCA's for TARGETS via a *ctrl+click+drag volume slider*.

ROOMS therefore also have *mute*, *solo*, and *phase* toggles, and as with the SOURCE-GROUP relationship, any *mute*, *solo* or *phase toggle* of a ROOM cascades to all TARGETS in that ROOM **in addition** to any *mute*, *solo*, or *phase toggle* of the TARGET. In other words, a *reverse phase* TARGET in a *reverse phase* ROOM has its *original phase*.

The *perimeter* of the ROOM along with its name indicate its status, and with mouse-over the perimeter the ROOM can be resized when indicated in green:

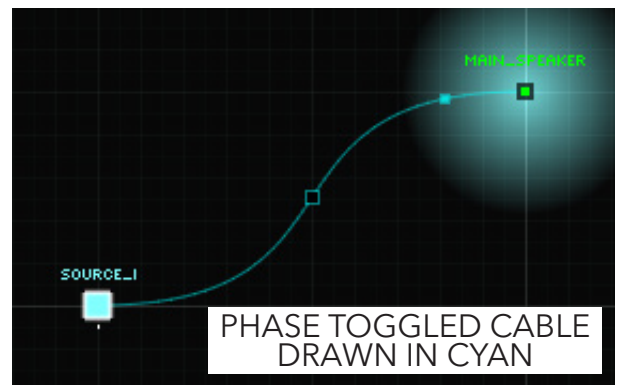
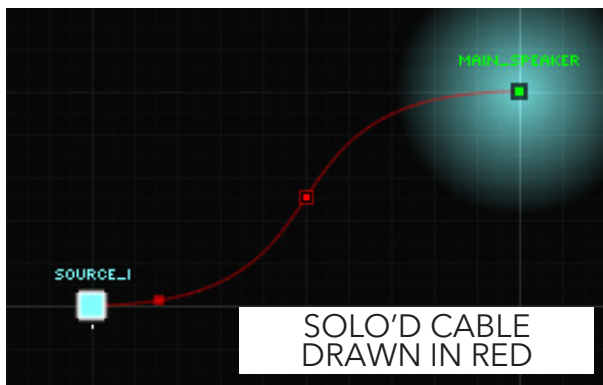
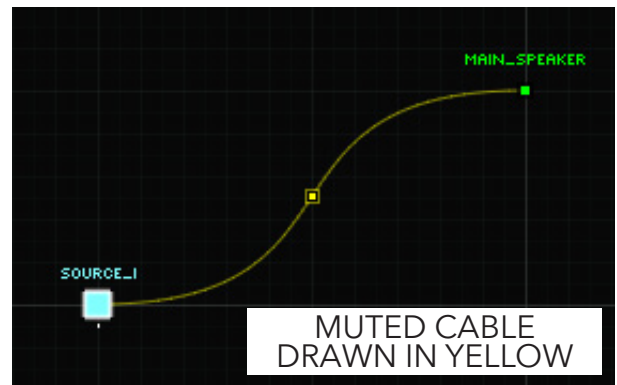
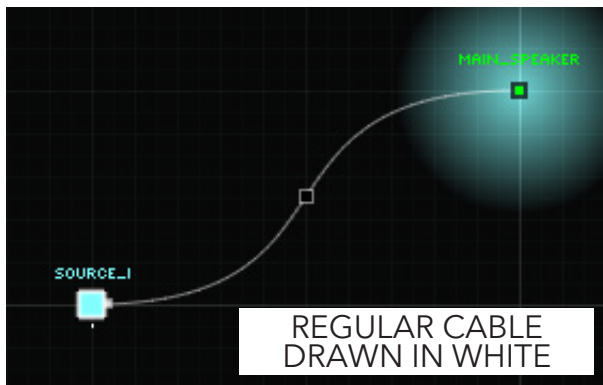


ROOM MOUSE-CLICK ACTIONS

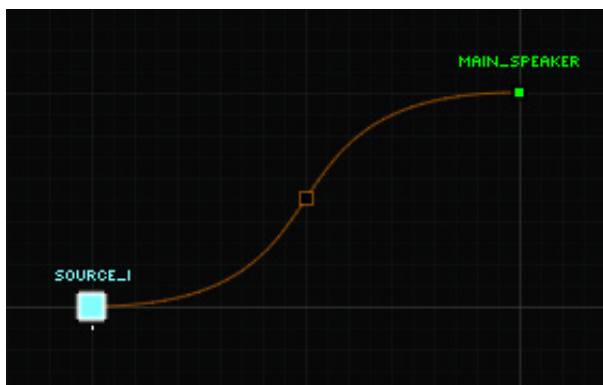
A	MUTE
S	SOLO
F	FREE ALL SOURCES FROM GROUP
B	SELECT COLOUR FROM DIALOG
P	SWITCH PHASE
Z	SET VOLUME TO ZERO dB
BACKSLASH	TOGGLE INDIVIDUAL / GLOBAL RESPONSE
CLICK + DRAG	DRAW MARQUEE SELECTION
CTRL + DRAG	VOLUME SLIDER
SHIFT + DRAG	MOVE ROOM WITH CONTENTS
CTRL + SHIFT + DRAG	MOVE ROOM WITHOUT CONTENTS
RIGHT CLICK	OPEN MAIN MENU
DOUBLE CLICK	SET NAME
R	RESET

CABLES AND CHAINS

CABLES in SoundSquares come in a variety of flavours, and are primarily used to link SOURCES to TARGETS, overriding their spatial relationship. CABLES have their own internal *volume slider* and act as VCA's on SOURCE nodes. Note that GROUP VCA functions cascade through CABLES, so a regular cable does not override a GROUP'S contribution to the eventual mix. CABLES can also be *muted* and *solo'd*, as well as offering *phase toggling*. Interacting with a CABLE is done by clicking on its *central handle*.

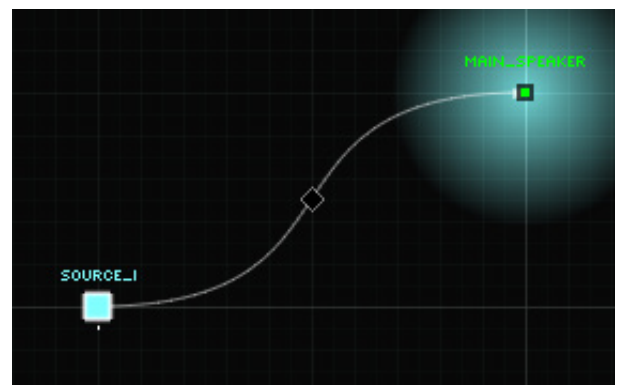


phase toggled CABLES that are also solo'd are drawn in pink



EXCLUSION CABLES

Drawn in orange, *exclusion cables* block the SOURCE from the TARGET in the same way as *muting* or setting the CABLE to $-\infty$ dB, but they are not impacted by the *mute / solo* status of other CABLES.

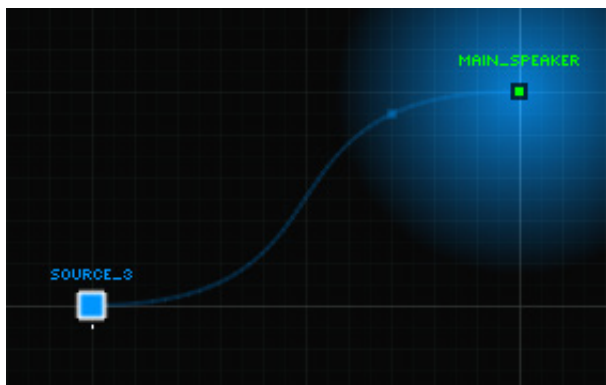


ANTI-GROUP CABLES

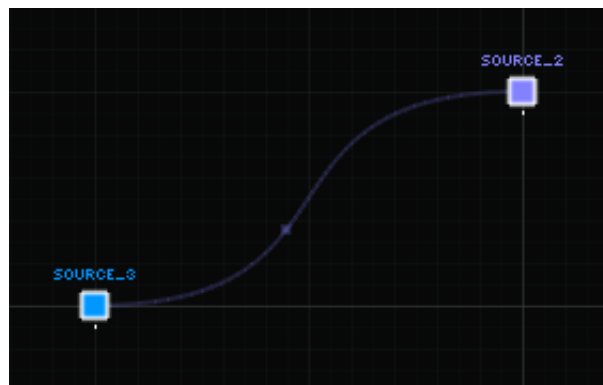
Drawn with a diamond-shaped centre instead of a square, *anti-group cables* allow the SOURCE to *ignore* the GROUP VCAs whilst not removing the SOURCE from the GROUP for *rotation* or *motion*.

SIDECHAINS

All dynamics processors in SoundSquares can be controlled using SIDECHAINS from SOURCES. A SOURCE'S dynamics processors can only be sidechained to another SOURCE. TARGETS can be *SIDECHAINED* from both SOURCE and TARGETS. SIDECHAINS are drawn in the same way as CABLES, except that they *do not* feature a central control node, and are always drawn in the same colour as the SOURCE the SIDECHAIN originates *from*:



"MAIN_SPEAKER" is being SIDECHAINED by "SOURCE_3"



"SOURCE_3" is being SIDECHAINED by "SOURCE_2"

CABLE MOUSE-CLICK ACTIONS

Hold C, then click-drag on SOURCE or TARGET to start a CABLE.

Hold C, then click-drag on SOURCE or TARGET to terminate a CABLE.

Hold C, then click in background (or room) to cancel CABLE.

Hold E when terminating a cable to turn it into an exclusion CABLE.

Hold S when terminating a cable at a TARGET turn it into a SIDECHAIN

Hold C when terminating a CABLE from a SOURCE to a SOURCE to turn it into a SIDECHAIN

Click on the cable centre to :

A	MUTE
S	SOLO
P	SWITCH PHASE
Z	SET VOLUME TO ZERO dB
E	SET CABLE AS EXCLUSION CABLE (TO REVERT, USE Z)
G	TOGGLE CABLE AS ANTI GROUP CABLE
X	REMOVE CABLE
CLICK + DRAG	VOLUME SLIDER

MARQUEE

SoundSquares' MARQUEE (lasso tool) offers a simple yet powerful mechanism for controlling multiple stage nodes simultaneously. It works in combination with the TOOLBOX LOCKING functions, and the NODE HIDE function, using the following logic:

*If a node is under the marquee **and** the node is not hidden **and** the node's type is not locked ... then it is susceptible to a variety of group-based actions and effects syndication functions*

MARQUEE-BASED KEY+CLICK COMMANDS :

A	MUTE SOURCES/GROUPS
S	SOLO SOURCES/GROUPS
D	HIDE SOURCES/GROUPS
F	FREE SOURCES/GROUPS ALL SOURCES/GROUPS
P	SWITCH PHASES OF SOURCES/GROUPS
X	REMOVE CABLES FROM SOURCES
R	RESET NODE
E	TOGGLE SELECTION TO ENVELOPE VIEWER
SHIFT + A	MUTE TARGETS IN MARQUEE
SHIFT + S	SOLO TARGETS IN MARQUEE
SHIFT + D	HIDE TARGETS IN MARQUEE
SHIFT + X	REMOVE CABLES FROM TARGETS IN MARQUEE
SHIFT + P	SWITCH PHASES OF TARGETS

If CAPSLOCK is *ON* there is no need to use shift to address TARGETS, for example, just use A + click to mute, instead of SHIFT + A.

SHIFT + [1...8]	ADD/REMOVE SOURCES FROM GROUP [1...8]
SHIFT + CTRL + [1...8]	ADD/REMOVE SOURCES FROM GROUP [9...16]
CTRL + [1...8]	MAKE ROOM [1...8] FROM MARQUEE
CLICK + DRAG	MOVE EVERYTHING INSIDE MARQUEE
CLICK + SHIFT + DRAG	MOVE JUST THE MARQUEE

MORE MARQUEE-BASED TIME SAVERS :

Setting up a circle of TARGETS can be achieved by drawing the MARQUEE around a number of TARGETS and using the *position* functions found in the menu system.

Syndicating the same DYNAMICS or FILTER settings to a selection of nodes can be achieved by drawing the MARQUEE around them, and then using the relevant SUB PANEL as a common interface.

Pasting FILTER or DYNAMICS settings to multiple nodes can be achieved by using the MARQUEE and the *paste* functions found in the menu system, making for fast configuration of systems.

NOTE : FOR MARQUEE FUNCTIONALITY TO WORK, NODE NAMES MUST NOT BEGIN WITH A NUMBER

SUB PANELS & HOT KEYS

SoundSquares uses a variety of SUB PANELS in order to cram-in all of its controls. They are organised as a series of floating windows with their own minimal window manager. Some of them work directly with the audio nodes, some of them more globally, and some are purely for visual feedback.

VIEW	HOT KEY	DESCRIPTION
TOOLBOX	<i>SHIFT + T</i>	GLOBAL STAGE COMMANDS AND GUI PARAMETERS
VIRTUAL MIXER	<i>F1</i>	INPUT / OUTPUT PRE/POST MIXERS
DELAY	<i>F2</i>	SOURCE / TARGET DSP DELAY
FILTERS	<i>F3</i>	SOURCE / TARGET DSP FILTERS
DYNAMICS	<i>F4</i>	SOURCE / TARGET DSP DYNAMICS
MUTES	<i>F5</i>	ALL NODES MUTE / SOLO OVERVIEW
ENVELOPES	<i>F6</i>	MULTIPLE SOURCE / TARGET ENVELOPE VIEWER
SCOPE	<i>F7</i>	SOURCE / TARGET WAVEFORM SCOPE
FFT	<i>F8</i>	SOURCE / TARGET SPECTRUM ANALYSER
CLI	<i>SHIFT + C</i>	COMMAND LINE INTERFACE USED BY MENU SYSTEM
MASTER	<i>SHIFT + Y</i>	METER FOR ALL TARGETS
<i>F9 + MOUSE WHEEL / SCROLL</i>		ZOOM STAGE IN / OUT
	<i>F10</i>	TOGGLE FULLSCREEN VIEW MODE
DSP PIPELINE DEBUG	<i>F11</i>	DEVELOPMENT TOOL TO EXAMINE DSP INTERNALS
TARGET HEATMAP	<i>SHIFT + F</i>	VISUALISE THE SYSTEM'S PANNING LAW(S)
	<i>F12</i>	HIDE ALL SUB PANELS

In addition to the controls listed on the coming pages, the following key commands work in the STAGE PAGE :

ITEM	HOT KEY	DESCRIPTION
STAGE ROOMS	<i>]</i>	TOGGLE LOCKING* OF STAGE
	<i>H</i>	TOGGLE ROOM VISIBILITY ON STAGE
	<i>SHIFT + H</i>	TOGGLE DISPLAY OF ROOM NAMES
	<i>CTRL + H</i>	TOGGLE LOCKING* OF ROOMS
TARGETS	<i>J</i>	TOGGLE TARGETS VISIBILITY ON STAGE
	<i>SHIFT + J</i>	TOGGLE DISPLAY OF TARGET NAMES
	<i>CTRL + J</i>	TOGGLE LOCKING* OF TARGETS
	<i>SHIFT + @ (')</i>	CHANGE NUMBER OF TARGETS
GROUPS	<i>K</i>	TOGGLE GROUPS VISIBILITY ON STAGE
	<i>SHIFT + K</i>	TOGGLE DISPLAY OF GROUP NAMES
	<i>CTRL + K</i>	TOGGLE LOCKING* OF GROUPS
	<i>L</i>	TOGGLE SOURCES VISIBILITY ON STAGE
SOURCES	<i>SHIFT + L</i>	TOGGLE DISPLAY OF SOURCE NAMES
	<i>CTRL + L</i>	TOGGLE LOCKING* OF SOURCES
	<i>SHIFT + ; (:)</i>	CHANGE NUMBER OF SOURCES
	<i>N</i>	IF SOURCES AND TARGETS VISIBLE, VIEW CABLES TOO
CABLES	<i>M</i>	TOGGLE DRAWING OF CROSSHAIRS
X-HAIRS	<i>[</i>	TOGGLE DRAWING OF GRID
GRID	<i>SHIFT + Q</i>	TOGGLE GROUP ROTATIONS
ROTATION	<i>SHIFT + W</i>	TOGGLE GROUP MOTIONS
MOTION	<i>SHIFT + V</i>	TOGGLE MIX VISUALISATIONS
VISUALISATION	<i>SHIFT + S</i>	TOGGLE SNAP TO GRID
SNAP		

* Locking only applies to positions on STAGE, and does *not* impact setting of levels, CABLES, SIDECAINS but it **does** impact MENU *and* MARQUEE action such as *syndication* of FILTERS and DYNAMICS.

[*SHIFT + T*]

The **TOOLBOX** puts a selection of useful functions into a menu on the left side of the screen.



Click the **TOOLBOX** *title* to show/hide the **TOOLBOX**, which is visible by default.

SHOW / HIDE the various parts of the interface by using these toggles.
GREY means *HIDDEN*, *BLUE* means *SHOWING*.
 This selection is saved with the current preset.

SHOW / HIDE the various labels on the interface by using these toggles.
GREY means *HIDDEN*, *BLUE* means *SHOWING*.
 This selection is saved with the current preset.

LOCK the position of the various parts of the interface by using these toggles.
GREY means *LOCKED*, *BLUE* means *UNLOCKED*.
LOCKS only impact positions - faders and dialogs remain usable.
 This selection is saved with the current preset.

MIDI functionality is *not yet running* in this version of the plugin.
 Ignore this part of the toolbox for the moment...

DISPLAY current state of the system - number of mixing cross-points, number of modulating cross-points, and mouse position.
 To change number of **SOURCES** or **TARGETS**, click their field.
 New **SOURCES / TARGETS** arrive in the last clicked-in **ROOM**

DEFEAT BUTTONS

When **SOURCES**, **TARGETS**, **GROUPS**, **ROOMS** or **CABLES** are either *solo'd* or *muted*, these indicators will illuminate.
 To override (defeat) *solos* or *mutes*, click the relevant button

If any **GROUPS** have their *rotation* or *movement* enabled, the *rotate* and *move* buttons become coloured.
 To activate *rotation* or *movement* click the toggle buttons.
 These settings are saved with the current preset.

To see a live visualisation of the entire mix, click *VU meters*. **TARGETS** illuminate with relevant **SOURCE** levels.

Zoom the interface to 2.0x, 1.0x, or 0.5x visual scale

Toggle *SNAP-TO-GRID* (works best with *GRID visible*)

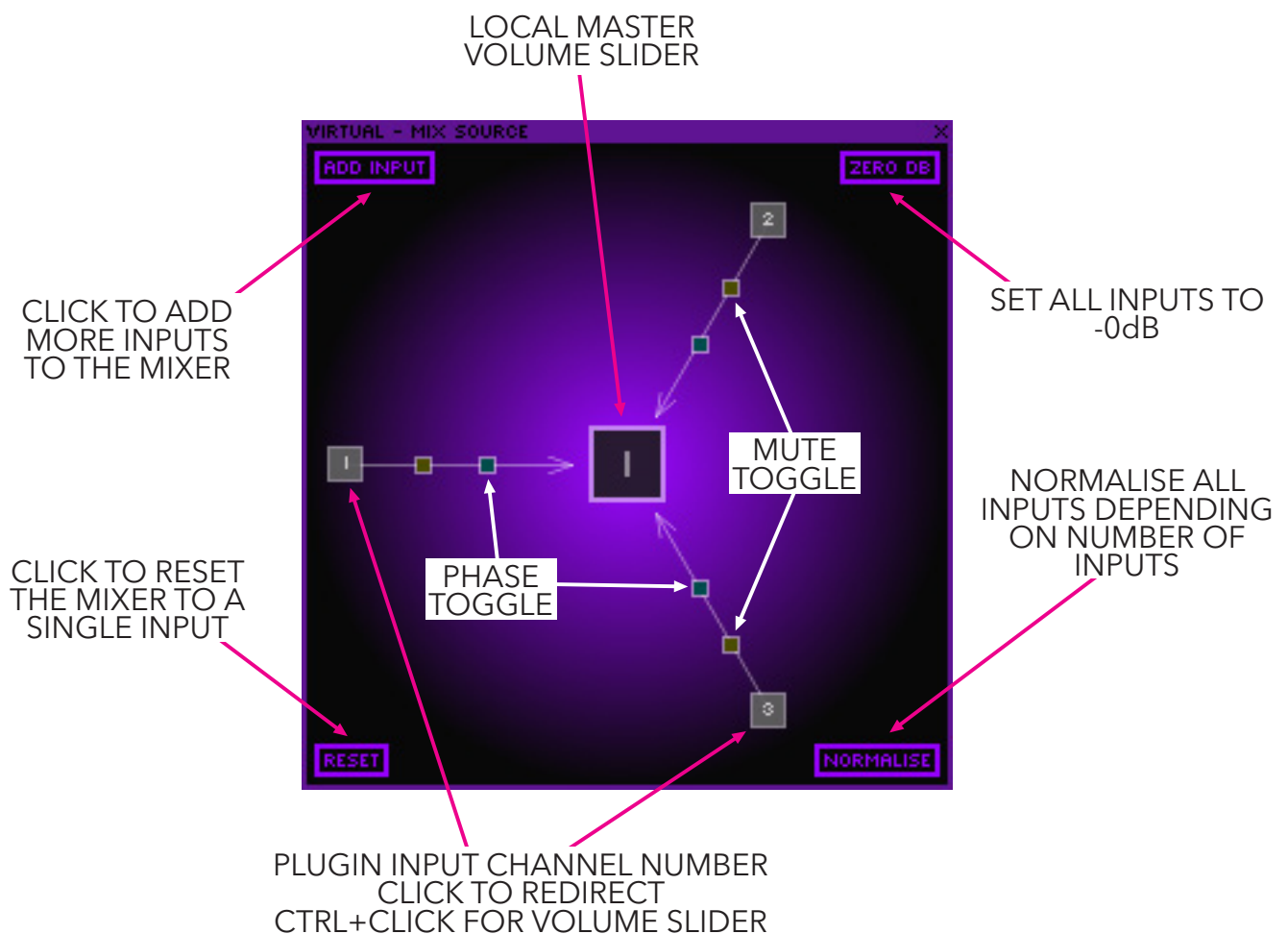
When a **SOURCE'S** input or a **TARGET'S** output level clips, these illuminators will flash red (*as will the offending item*) - click to dismiss.

VIRTUAL PREMIXERS AND POSTMIXERS

Before audio arriving at the plugin input becomes a SOURCE node, and before audio mixed to a TARGET node becomes a plugin output, there are two additional stages acting to virtualise and route the signals. The first input and last output stages take the form of simple N:1 pre/post mixers, offering control over *individual levels, mutes and phases* in addition to a *local-master volume control*.

INPUT PREMIXERS : mix a VIRTUAL INPUT from plugin inputs

OUTPUT POSTMIXERS : mix a plugin output from VIRTUAL OUTPUTS



For most uses, these mixers can most likely be ignored, especially if a simplified on-to-one routing relationship between plugin inputs/outputs and SOURCES/TARGETS is required.

However if you need to work with mono channels sub-mixed from multiple plugin inputs, or side-channels derived from anti-phasing one side of a stereo pair, an INPUT PREMIXER is what you need - one example might be creating a multichannel dynamics effect with a common sidechain mixed from all the constituent sub-channels.

On the output side of things, POSTMIXERS can be most effectively used in the scenario whereby multiple channels of a playback mix need bussing to another location for monitoring, much in the same way that multiple headphone mixes can be post-mixed alongside a front-of-house mix in live performance settings.

[F2]



DELAY SUB PANEL

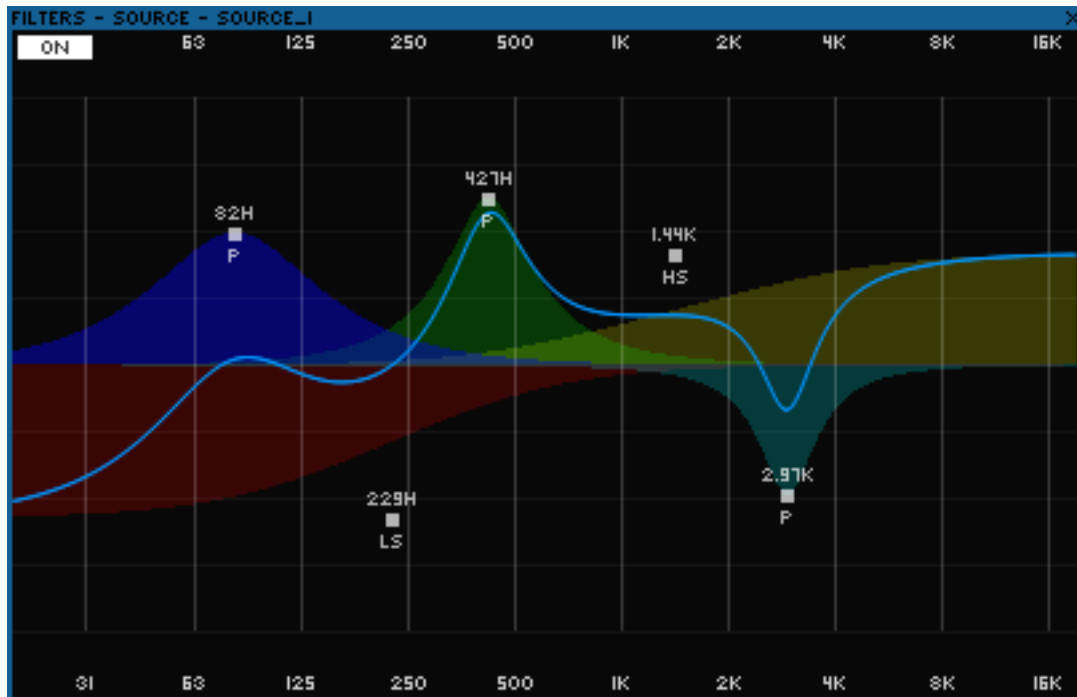
All SOURCES and TARGETS feature a simple *non-feedback* DELAY. It is not a creative effect per se, but can come in handy when dealing with large numbers of speakers in a system. Further to the normal mode of applying a positive-going delay time, the use of *negative delays* allows a SOURCE or a TARGET to be *pulled ahead* of their peers.

Obviously this doesn't mean the audio is actually being *pulled back from the past* but instead when entering a negative delay time, all the peers' delay times are increased to take into account the negative delay of the selected node.

To change the DELAY TIME, drag the fader left or right. For incrementing the DELAY in quantised units of 100ms, hold down *shift* whilst dragging. For incrementing the DELAY in more fine-grain increments than normal, hold down *ctrl*. To manually enter the desired time, click in the display to open a keyboard input dialog.

To open or close the DELAY SUB PANEL press F2.

[F3]



FILTERS SUB PANEL

All SOURCES and TARGETS feature a 5 band fully parametric equaliser section enabling up to +/-12dB cut/boost per band, and the following filter shapes :

LOW PASS
HIGH PASS
LOW SHELF
HIGH SHELF
EQ PEAK
BAND PASS

To change a band's type

To switch a band on / off

To move a band's frequency or gain

To change a filter band's bandwidth

To move all bands frequencies together

To move all bands gains together

To amplify / diminish the magnitude of all band gains

ctrl+click it

ctrl+shift+click it

simply drag it about.

shift+drag up/down.

ctrl+click then left/right.

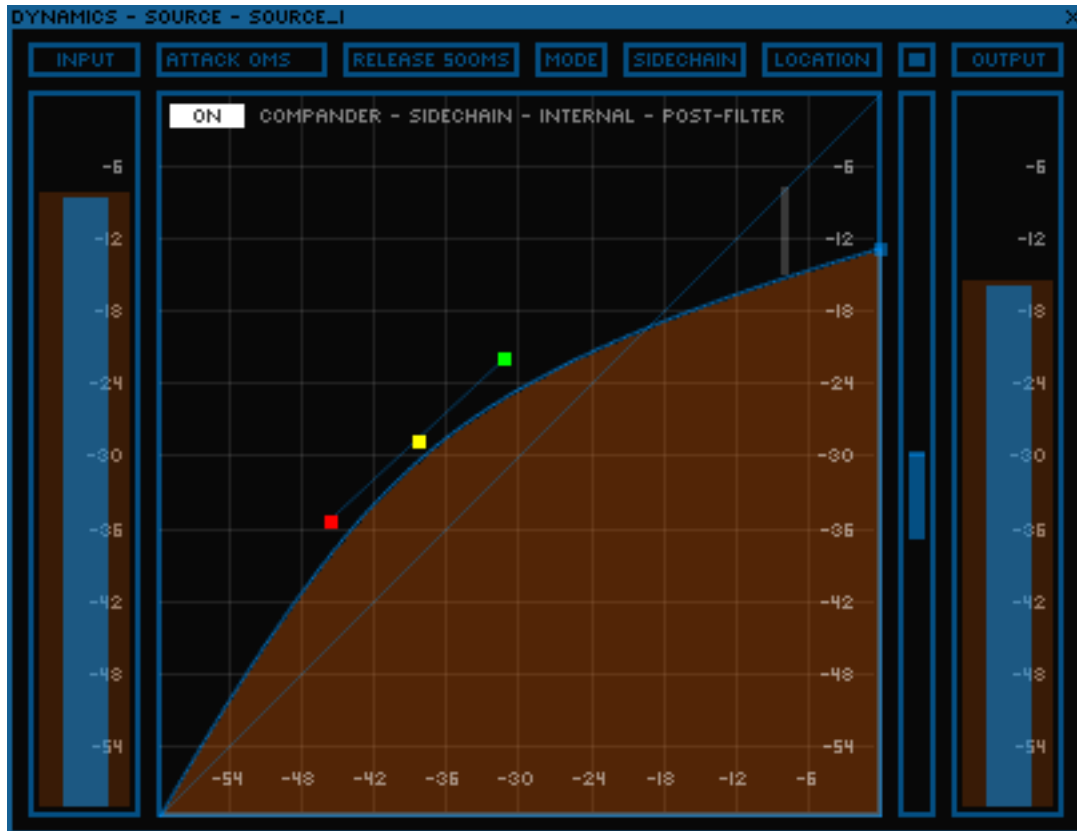
shift+click then drag up/down

ctrl+shift+click background then drag left/right

To open the menu, *right-click* in the background, where you'll find options to copy/paste/reset and to load a selection of presets, including crossover settings at a range of frequencies of poles configured in Butterworth, Linkwitz-Riley, or Bessel modes.

NOTE : All settings made in the FILTERS SUB PANEL are *syndicated* to **ALL** SOURCE and TARGET nodes under the MARQUEE, *unless the node is either hidden (d+click) or the node-type is locked from the TOOLBOX.*

[F4]



DYNAMICS SUB PANEL

All SOURCES and TARGETS feature a powerful Bezier-curve based, sidechainable DYNAMICS PROCESSOR offering the following models :

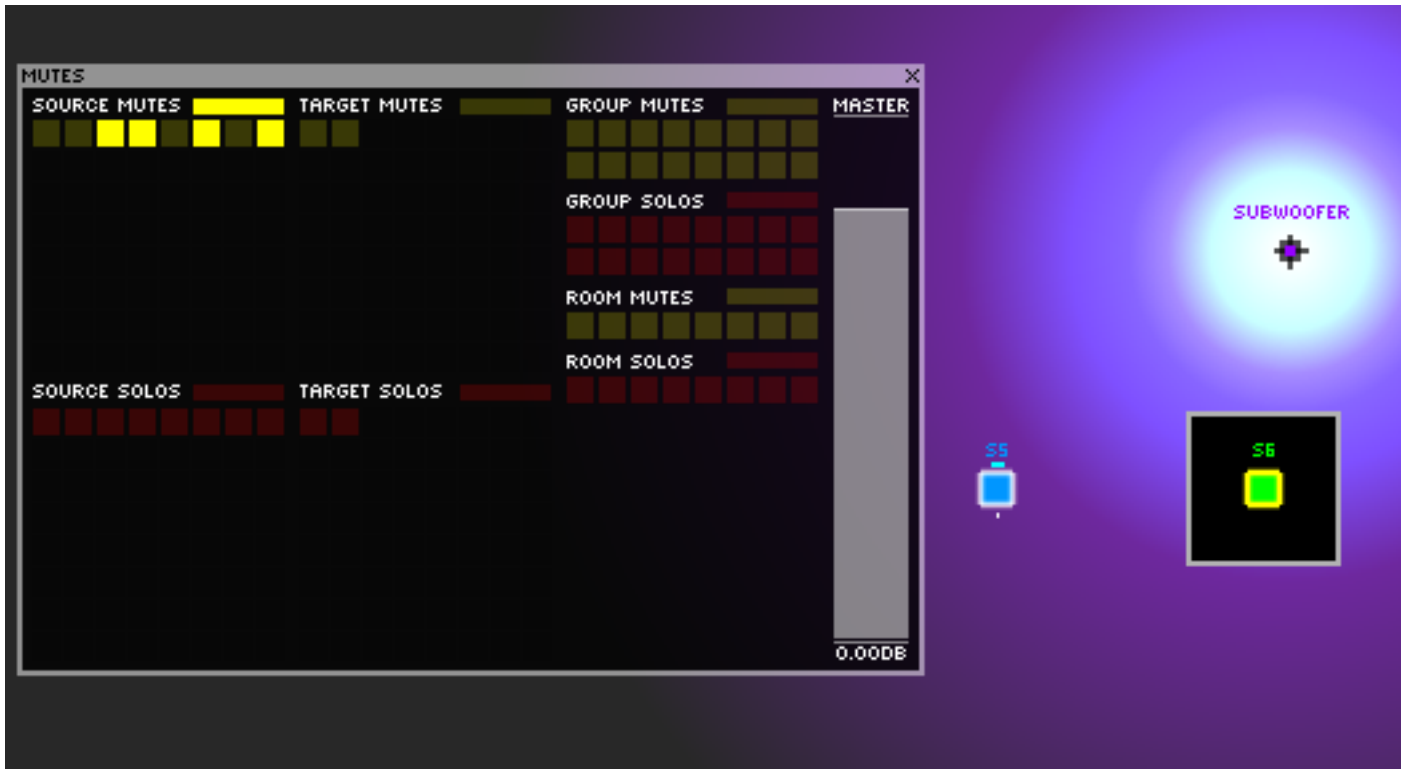
- COMPRESSOR
- GATE
- LIMITER
- COMPANDER
- FREEFORM
- ENVELOPE FOLLOWER
- ENVELOPE SHAPER
- DYNAMIC EQ

Attack and release are entered manually by clicking in their text fields, and *mode*, *sidechain*, and *location* are selected via drop-down menus from their fields, or using the pop-up menu by right-clicking in the response graph. The *location* selector allows for both internal and external sidechains to be drawn from *pre-delay*, *pre-filter*, or *post-filter* positions in the process pipeline, and for TARGETS, this location can also be drawn from *post-dynamics* when the sidechain used is a SOURCE. In *dynamic EQ* mode, the sidechain is used to mix the node's *pre-filter* signal with its *post-filter* signal, and will have no effect *unless the filters are also operational*

TARGETS and SOURCES can be sidechained from SOURCES, but (at preset) only TARGETS can be sidechain from TARGETS.

NOTE : All settings (except *sidechain*) made in the DYNAMICS SUB PANEL are syndicated to **ALL** SOURCE and TARGET nodes under the MARQUEE, *unless the node is either hidden (d+click) or the node-type is locked from the TOOLBOX.*

[F5]



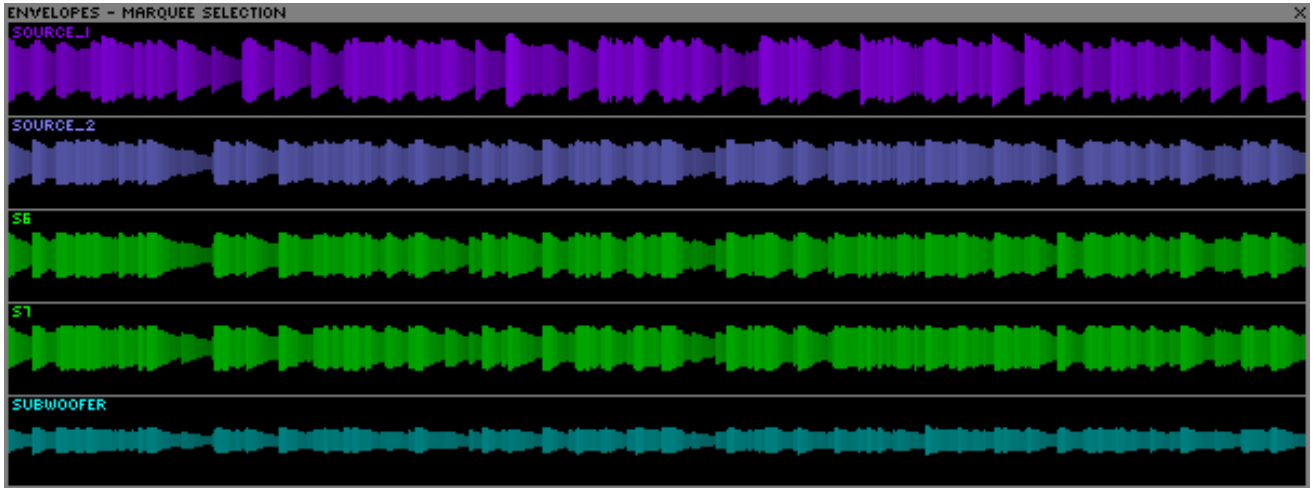
MUTES SUB PANEL

This sub panel offers an additional method of *muting and solo'ing* any SOURCE, TARGET, GROUP, or ROOM, in addition to offering access to the *system-wide MASTER VOLUME*.

For each category of item, if it is *solo'd* or *muted* the *defeat buttons* next to each category can be used to *unsolo* and *unmute* each category.

In addition to presenting an alternative location for *mutes and solos*, when you mouse-over any of the buttons in this panel, the node it relates to is highlighted wherever it is on the STAGE ... which can come in quite handy when the setup is looking busy

[F6]



ENVELOPES SUB PANEL

The ENVELOPES PANEL offers a quick and simple way to see what is going-on with any selection of SOURCES *and/or* TARGETS.

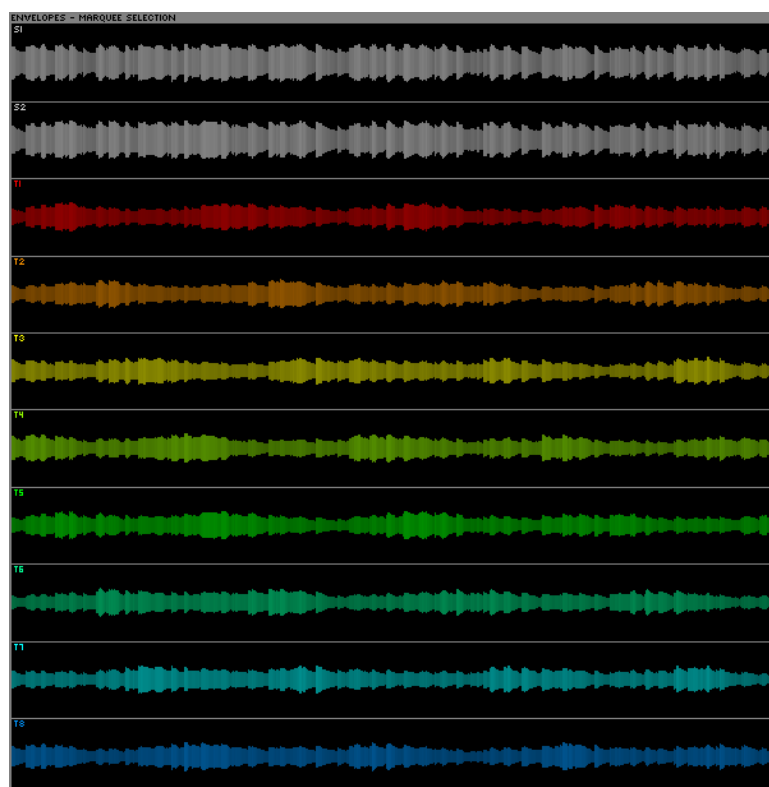
To route a single SOURCE or TARGET to the ENVELOPES PANEL, *hold E* and click on the node.

Multiple nodes can be routed to the ENVELOPES PANEL in two ways :

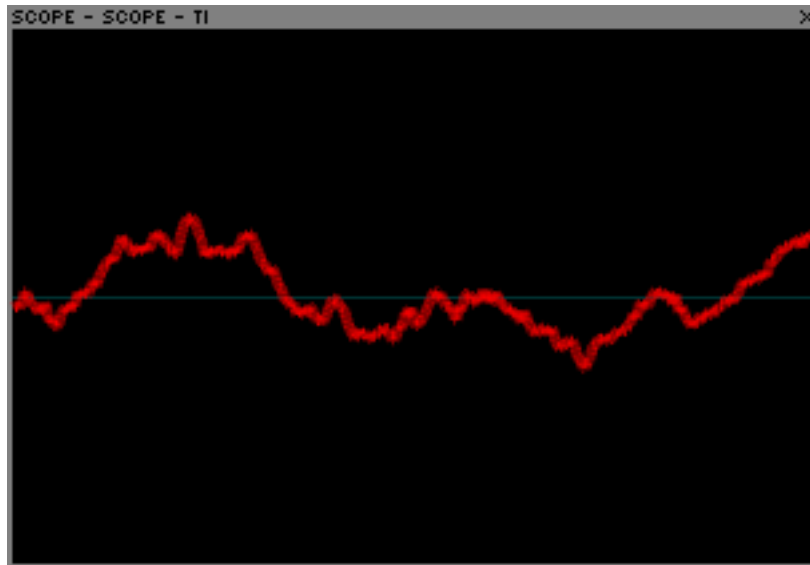
To show envelopes for all members of a GROUP, *hold I*, and click on the GROUP node.

To show envelopes for an arbitrary selection of SOURCE and TARGETS, draw a MARQUEE around them, then *hold E* and click on the MARQUEE.

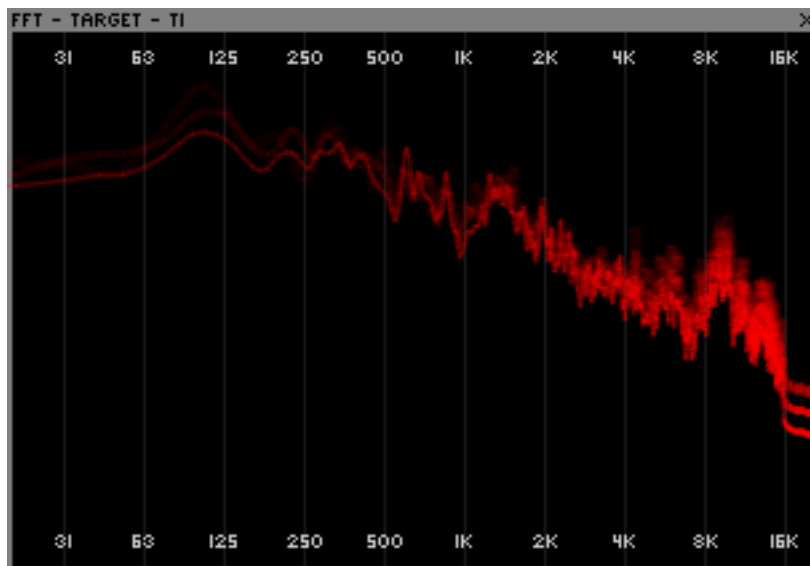
The ENVELOPES PANEL can be resized by dragging from its *bottom border*



[F7]



[F8]



SCOPE & FFT SUB PANELS

The SCOPE and FFT PANELS offer additional views into what is going-on with any *single* SOURCE or TARGET.

To route a SOURCE or TARGET to *both* PANELS, *hold I* and click on the node.

Only a single node at a time can be routed to these PANEL *at present*.

[SHIFT + C]

```

CLI - 19:09:09
RECEIVED REPORT
BUILD INPUT TRIM FADER
TRIM FADER BUILDER
SOURCE VIRTUALS UPDATED
TARGET VIRTUALS UPDATED
RECEIVED REPORT
RECEIVED REPORT
RIGHT CLICK MAIN
DO SOMETHING WITH -> GROUPIFY SOURCES GROUPS

CLI OBJECT >
> COMMAND STRING > GROUPIFY SOURCE S1 S2 GROUP ONE
> ACTION > GROUPIFY
> ARGUMENTS > SOURCE S1 S2 GROUP ONE
> MODE > 1
> START > 0
> NUM STARTS > 0
> STARTS > 0
> END > 0
> NUM ENDS > 0
> ENDS > 0
> VALUE > 0.000000
> STRING >
> VARS BY REF > 3
-> TYPE I1 INDEX I01 RANKI1 STRING IS11
-> TYPE I1 INDEX I11 RANKI21 STRING IS21
-> TYPE I31 INDEX I01 RANKI51 STRING IONE1

>>>GROUPIFY<<<
GROUP TO ADD 0
RECEIVED REPORT
RIGHT CLICK MAIN
DO SOMETHING WITH -> CIRCLE TARGETS

CLI OBJECT >
> COMMAND STRING > CIRCLE TARGET T1 T2 T3 T4 T5 T6 T7 T8
> ACTION > CIRCLE
> ARGUMENTS > TARGET T1 T2 T3 T4 T5 T6 T7 T8
> MODE > 2
> START > 0
> NUM STARTS > 0
> STARTS > 0
> END > 0
> NUM ENDS > 0
> ENDS > 0
> VALUE > 0.000000
> STRING > T8
> VARS BY REF > 8
-> TYPE I21 INDEX I01 RANKI11 STRING IT11
-> TYPE I21 INDEX I11 RANKI21 STRING IT21
-> TYPE I21 INDEX I21 RANKI31 STRING IT31
-> TYPE I21 INDEX I31 RANKI41 STRING IT41
-> TYPE I21 INDEX I41 RANKI51 STRING IT51
-> TYPE I21 INDEX I51 RANKI61 STRING IT61
-> TYPE I21 INDEX I61 RANKI71 STRING IT71
-> TYPE I21 INDEX I71 RANKI81 STRING IT81

GENERATE METRICS ...
8 ITEMS
MAKE CIRCLE
>CIRCLE TARGETS

```

CLI SUB PANEL

A majority of SoundSquares menu-based functions and a selection of internal functions are text-based commands executed by a *virtual command line*. If you pay close attention to it you might be able to figure-out some useful things ... eventually I might document things properly ...

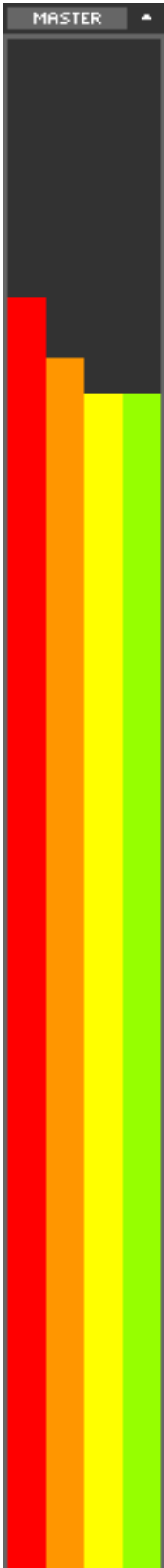
To capture nodes to execute commands it uses the following syntax styles

- [VERB] [NODE_TYPE] [NODE ID RANGES] [OPTIONS]
- [VERB] [PLURAL_NODE_TYPE] [OPTIONS]
- [VERB] [NODE_NAME_LIST]

Some examples :

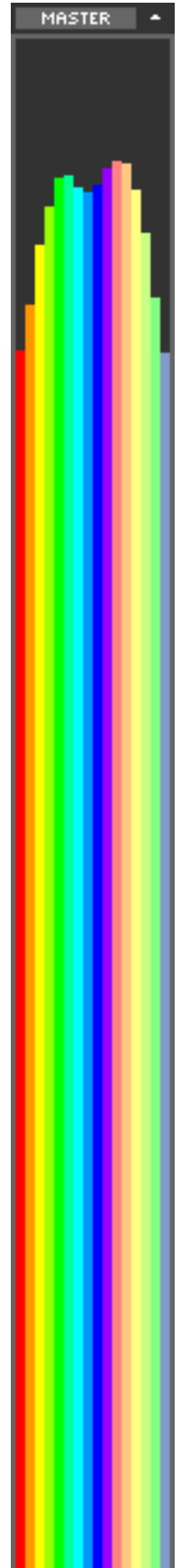
- NAME SOURCE 1-6 9-12 SRC // RENAME SOURCES 1-6 & 9-12 TO "SRC_" + [n]
- ALIGN TARGETS LEFT // ALIGN ALL TARGETS UNDER MARQUEE
- GLOBALZ TARGETS // TARGETS UNDER MARQUEE : GLOBAL RESPONSE
- CABLE S1 T3 // ADD CABLE BETWEEN "S1" & "T3"
- MUTE S1 S2 S3 // MUTE NODES "S1", "S2", & "S3"

I find it useful. You'll maybe not ... yet ...

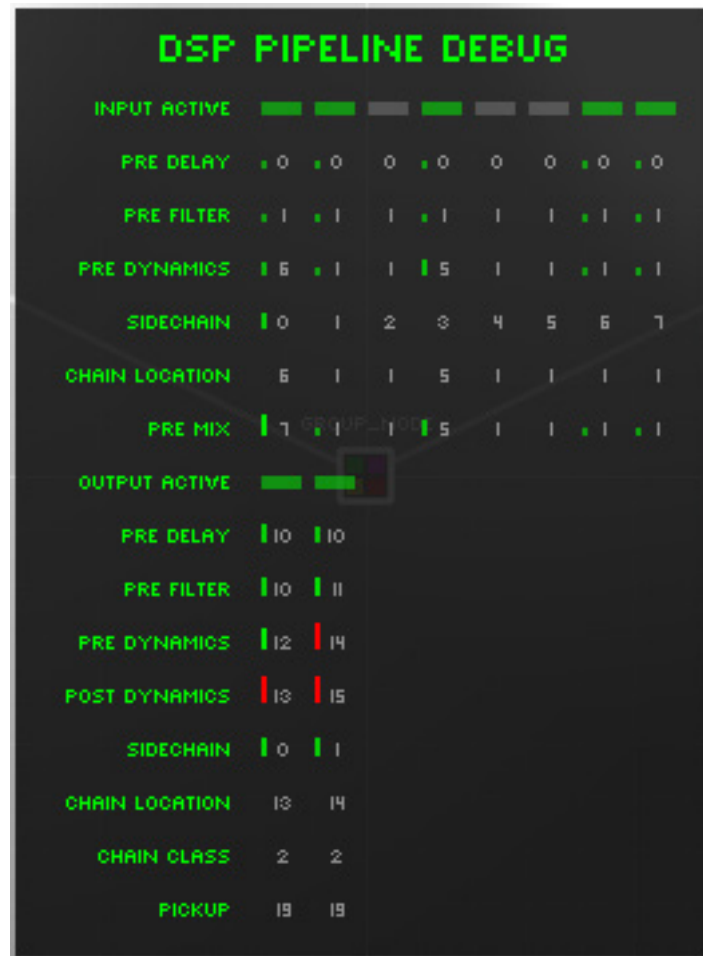
[*SHIFT* + Y]**MASTER SUB PANEL**

Whether 2 TARGET outputs, or 64, or indeed any number inbetween, the MASTER SUB PANEL simply shows the outgoing levels of all TARGETS.

If the output routing is 1:1 then this shows exactly what the plugin is sending back to the host. If however, the routing is more complicated, this panel will only illuminate what is going-on with *LOGICAL* TARGETS.



[F11]



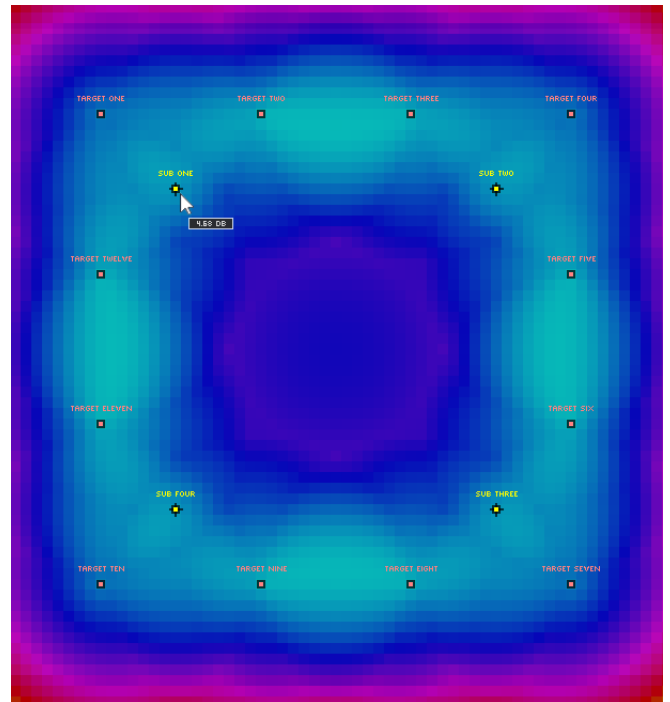
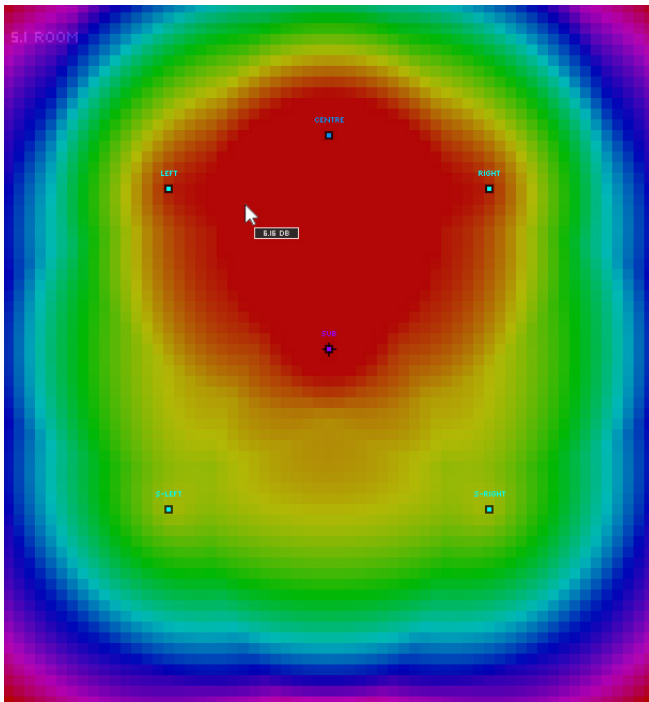
DSP PIPELINE DEBUG OVERLAY

As with the command line, this overlay is really only there for ongoing development purposes. It shows information about the various contents of the different stages of the processing pipeline.

SoundSquares' mix engine is based on a mixed precision part-interleaved parallel processing pipeline using AVX, SSE, and SCALAR code to be as fast and efficient as I can possibly make it. As such, it is quite an ... interesting ... setup internally, and uses a series of staged buffers whose locations change depending on the filters and dynamics processes in-use at the time. Hence, hunting down the correct location of sidechain channels within this "walking" setup has been a little challenging, so this overlay was thrown-together to make writing and debugging the code a bit easier.

It can be fun to observe all the blinken-lights generated by a mix though ;)

[*SHIFT + F*]



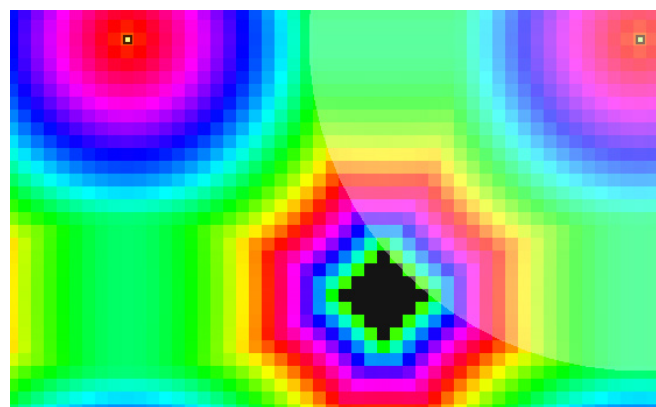
TARGET HEAT MAP

When putting together and mapping a system, it's sometimes useful to consider how uniform or 'flat' the mapping is with respect to whatever the multi-speaker equivalent of *panning law* is. The HEAT MAP offers an easy way of visualising the flatness of the system, and gives a read-out for the decibel level of a SOURCE when placed at the position of the mouse.

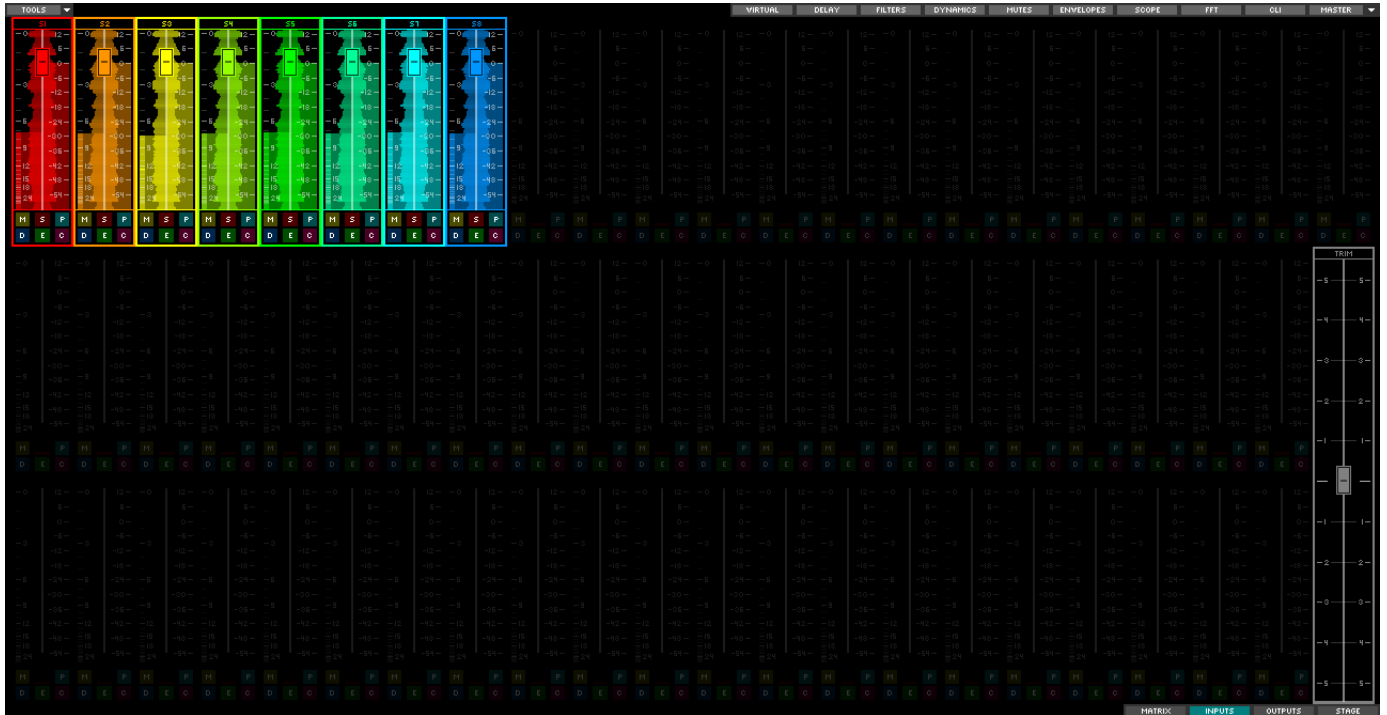
The HEAT MAP is made by scanning all positions of the STAGE to determine the *normalised average* level a SOURCE will create through (*visible, not-muted*) TARGETS when place at that position. Colour is mapped ot relative leve, so the less the localised colour variation between TARGETS, the flatter the mapping's response will be.

In terms of stereo, a -6dB (by value) panning law is established between any adjacent pair of TARGETS when their response circles cross terminate at their adjacent *partner*. At their mutual centre each TARGET contributes half of the level at that position. It becomes an interesting question when using multiple speakers as to what panning law *might even mean*.

This map is by no means a scientific tool, but it does at least give an indication of how variable the locations in the ROOM might be - if there are any black patches, and a SOURCE is put there, it will be *inaudible* unless CABLED to any TARGETS, or it has it's *spread slider* increased.

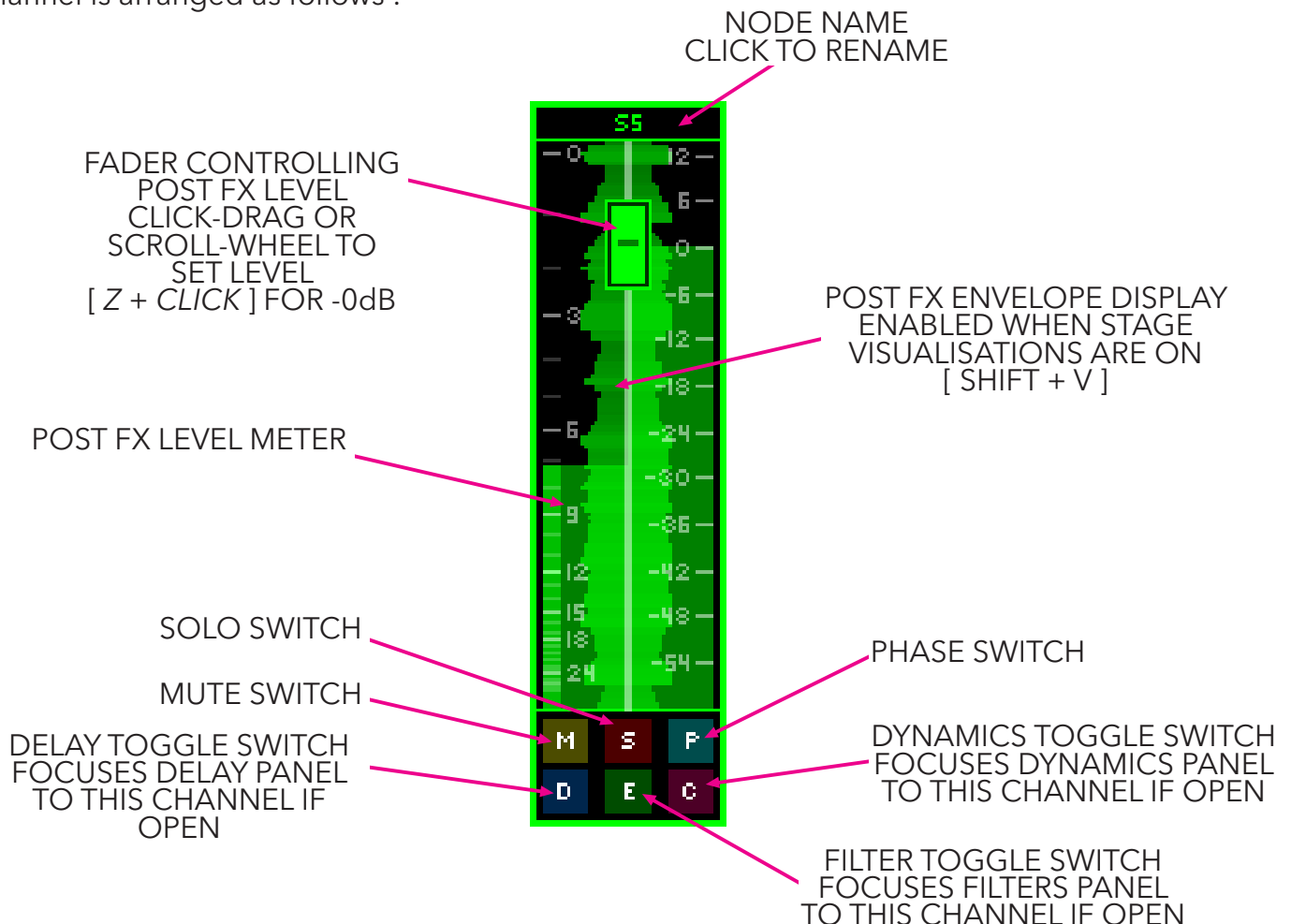


[SHIFT + I & SHIFT + O]

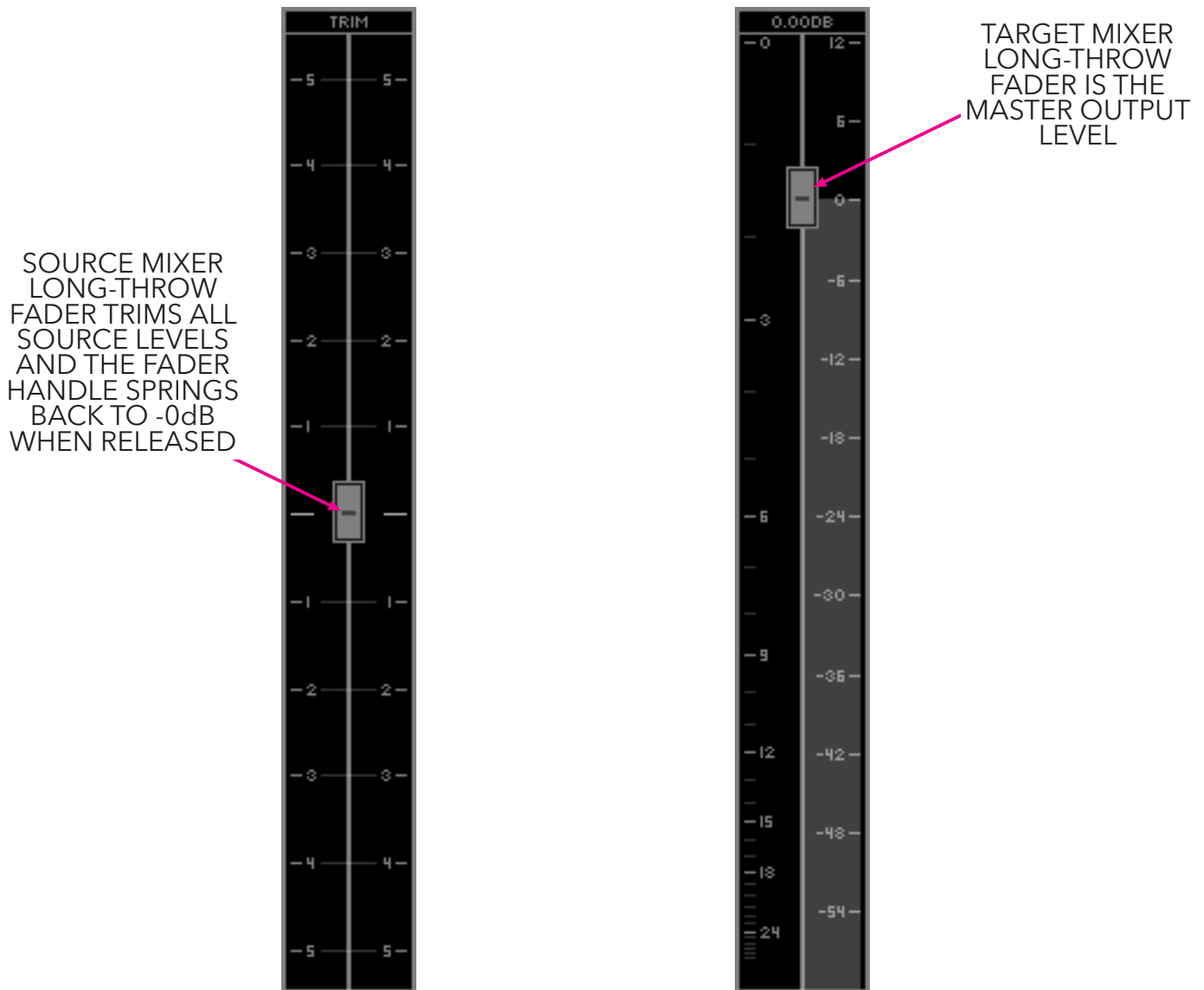


INPUT AND OUTPUT MIXER PAGES

The SoundSquares MIXER PAGES arrange all SOURCES and all TARGETS into 2 traditional mixing desk layouts showing only active nodes, and should be trivially familiar to use. Each individual channel is arranged as follows :

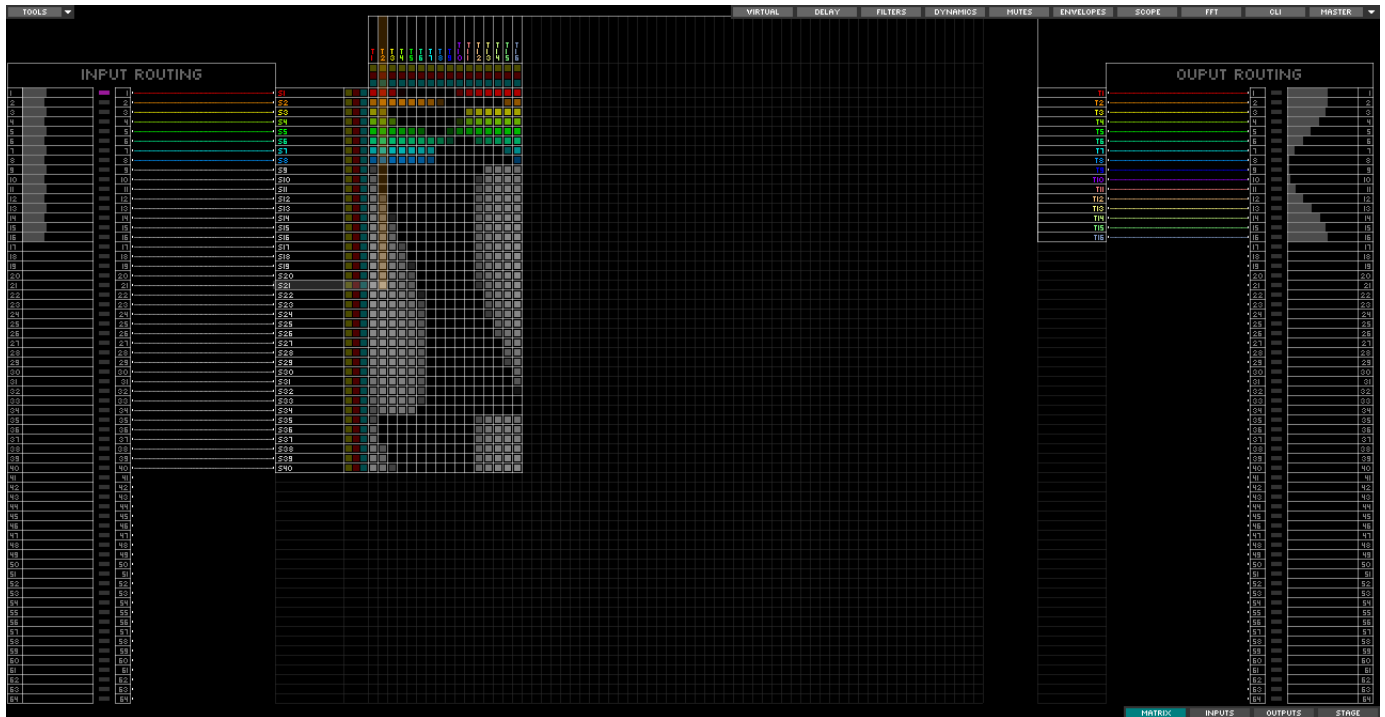


The main difference between INPUT MIXER PAGE and OUTPUT MIXER PAGE, other than their different foci being on SOURCE and TARGETS respectively, is the function of their long-throw faders. The SOURCE MIXER's long-throw fader acts as a **trim control for all SOURCE** nodes, and the TARGET MIXER's long-throw fader acts as the MASTER OUTPUT LEVEL CONTROL FADER.



Overall, these generic MIXER PAGES are far less exciting and interesting than the STAGE PAGE and the MATRIX PAGE, but they have their uses

[SHIFT + U]



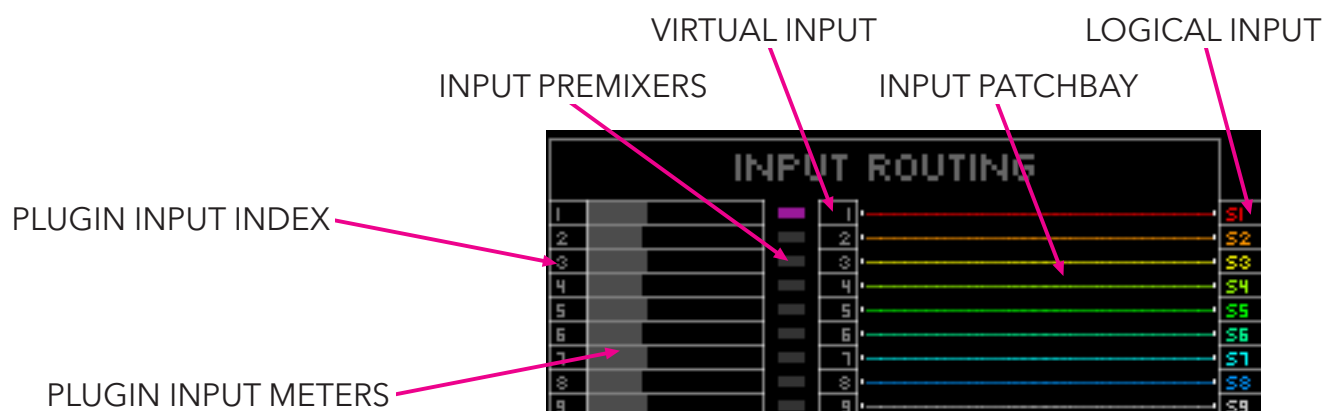
MATRIX MIXER PAGE

The SoundSquares MATRIX MIXER PAGE shows the MATRIX representation of the mix generated by the STAGE PAGE. It can also be used to interact with the mix, with any new cross-points generated being reflected back to the STAGE PAGE as new CABLES.

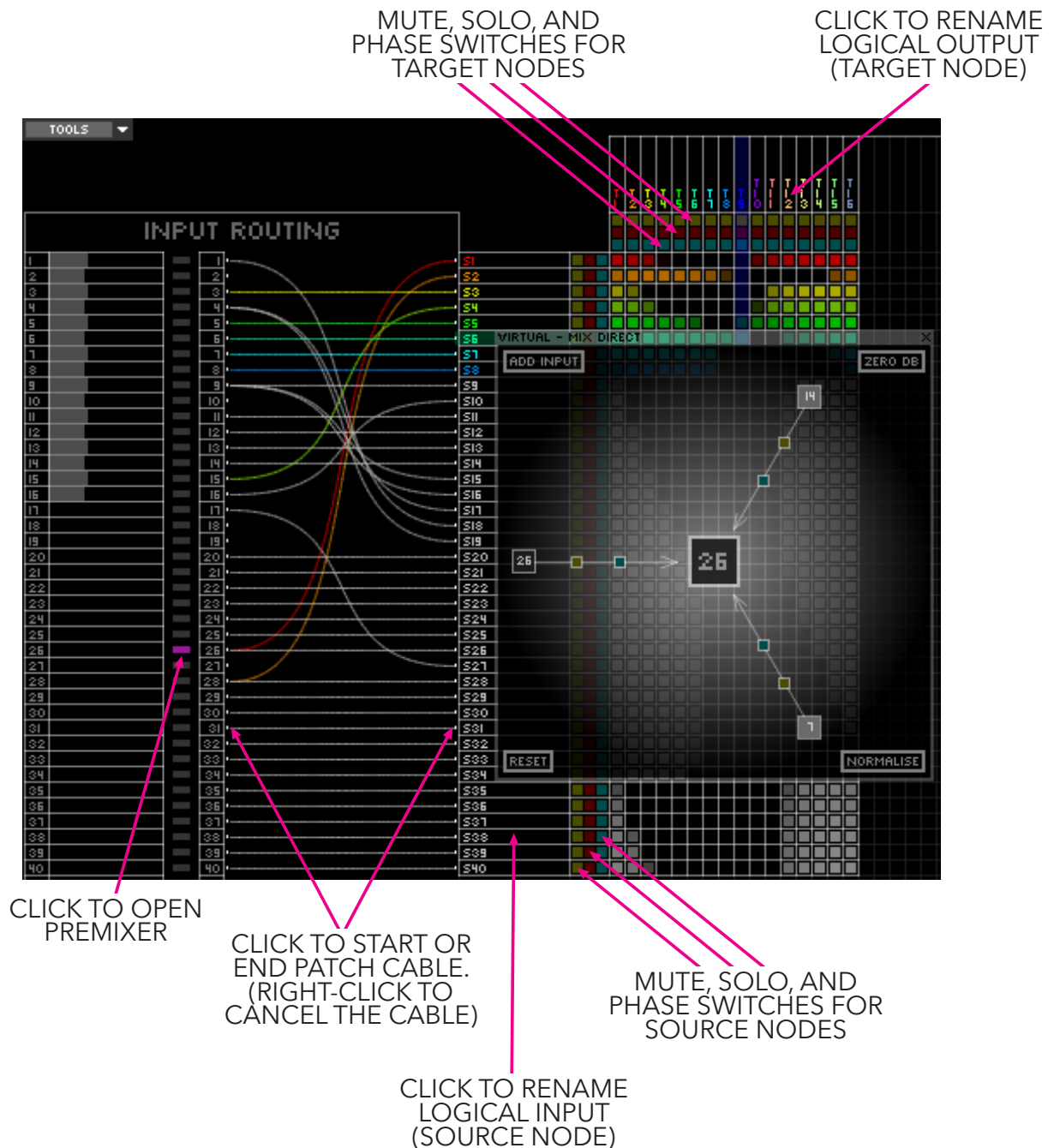
The page, left-to-right, is composed of 5 main sections :

- PLUGIN INPUT METERS AND PRE MIXERS
- INPUT PATCHBAY
- MATRIX
- OUTPUT PATCHBAY
- OUTPUT POST MIXERS AND PLUGIN OUTPUT METERS

INPUT PRE MIXERS and OUTPUT POST MIXERS use the VIRTUAL PANEL as described in the PANELS SECTION. The job of the INPUT PRE MIXER is to mix *plugin inputs* together to become *virtual inputs*, and the job of the INPUT PATCHBAY is to route the *virtual inputs* to become *logical inputs* in the form of SOURCE nodes. Mirroring this, the job of the OUTPUT PATCHBAY is to turn *logical outputs* in the form of TARGET nodes into *virtual outputs*, and the job of the OUTPUT POSTMIXER is to mix *virtual outputs* together to become *plugin outputs*.



In the following example, INPUT PREMIXER 26 is MIXING audio from PLUGIN INPUTS 7, 14 & 26 to become VIRTUAL INPUT 26, and the INPUT PATCHBAY is taking VIRTUAL INPUT 26 and routing it to LOGICAL CHANNEL 1, which becomes SOURCE 1 in the STAGE PAGE :



The output side functions the same as the input side, except the order of operations is reversed. On the input side the *plugin inputs* are mixed to become *virtual inputs* which are routed to *logical inputs*, and on the outputs side, *logical outputs* are routed to become *virtual outputs*, which are then mixed to become *plugin outputs*.

If this all seems like overkill, consider the case of a multichannel compressor, or a mid/side parallel processor : this I/O virtualisation and pre/post-mix arrangement allows for side-channel delivery into the stage, and also reuse of sidechains which can be premixed from multiple sources. Taken in combination, this functionality is what enables SoundSquares to be used to construct a wide array of multichannel multiband dynamics processors.

Using the MATRIX is straight-forward. When the mouse is over a cross-point, both SOURCE and TARGET are highlighted, and when you *click-drag* on a cross-point a fader appears allowing the level of the cross-point to be set. *ALL* new cross-points inserted in this way become CABLES on the STAGE PAGE. When *stage visualisations* are on, the x-points are coloured PER-SOURCE, and animated via its corresponding SOURCE volume.

To set a cross-point to -0dB, *hold Z* and click the cross-point.

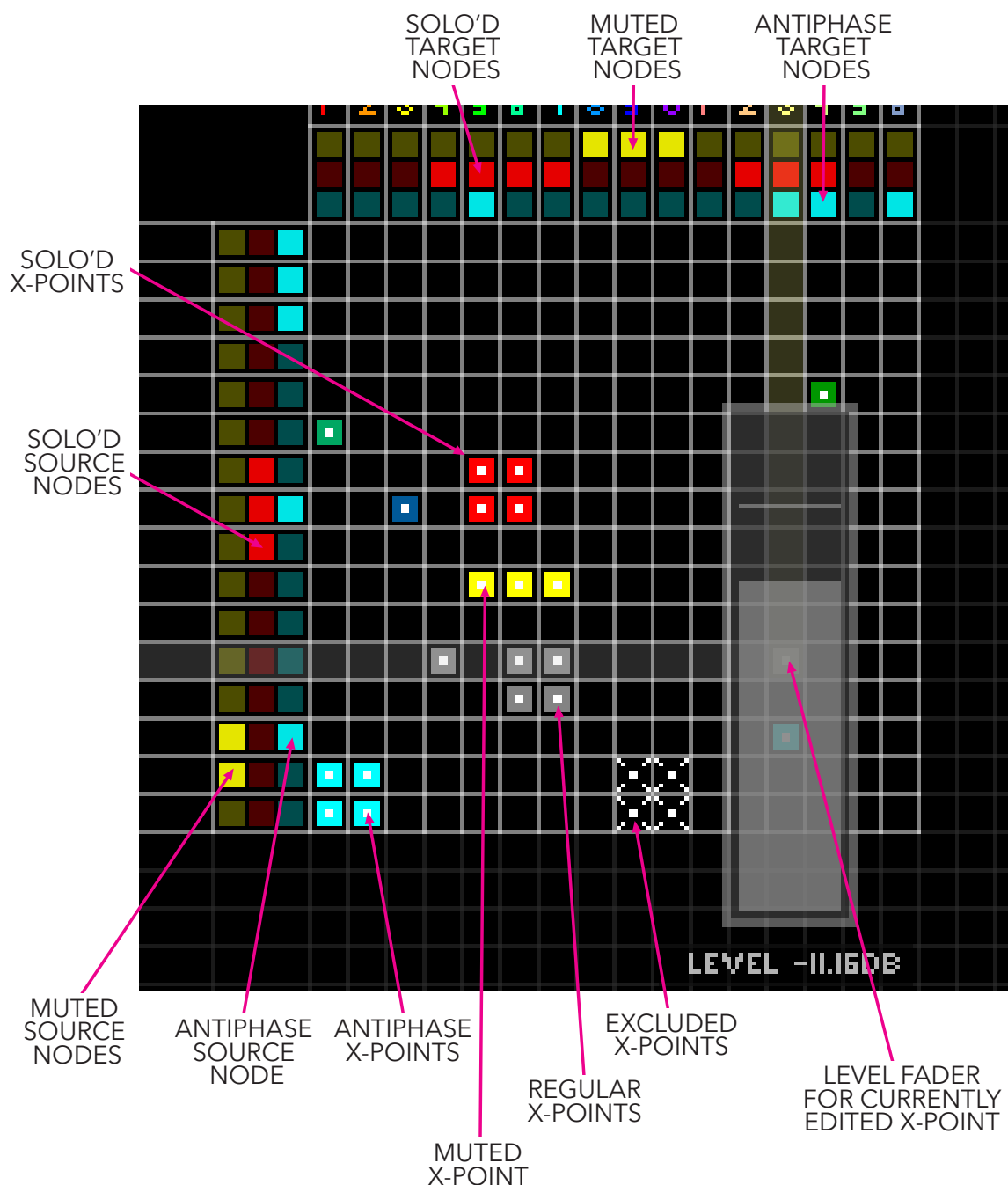
To mute a cross-point, *hold A* and click the cross-point.

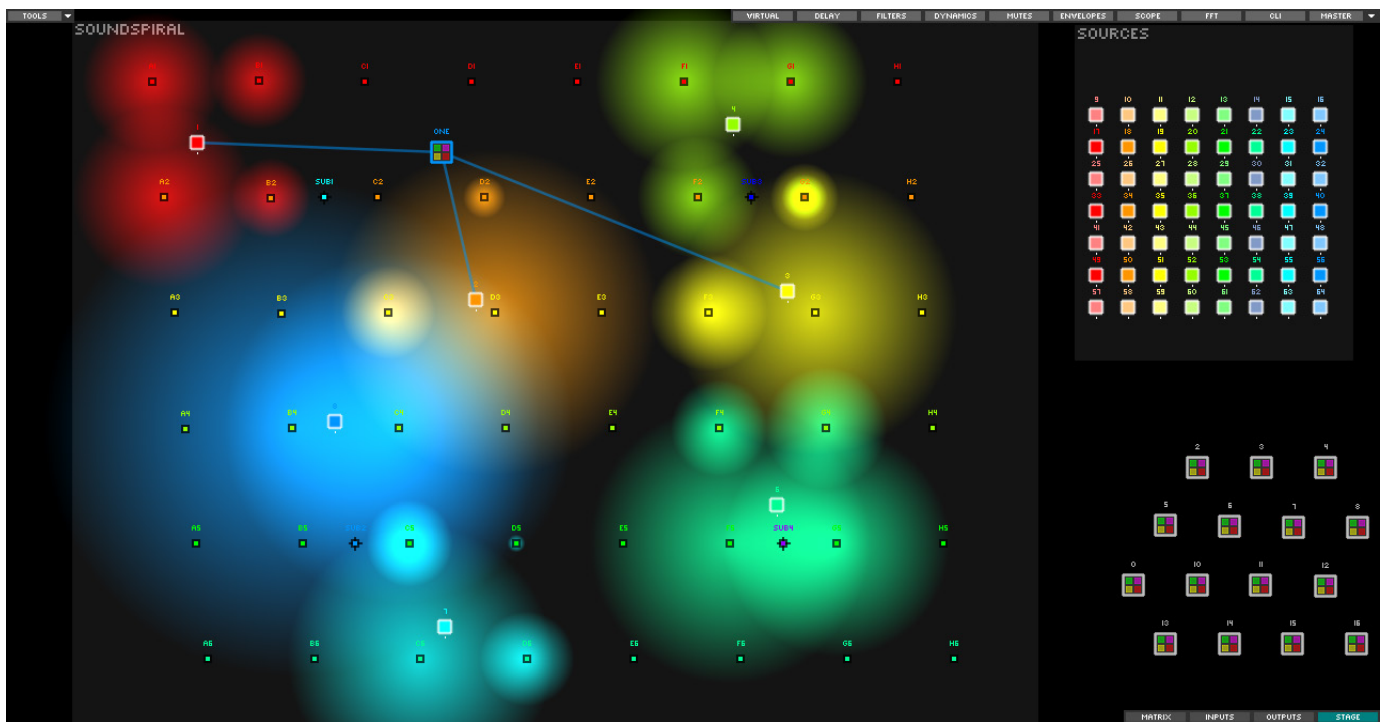
To solo a cross-point, *hold S* and click the cross-point.

To reverse the phase of a cross-point, *hold P* and click the cross-point.

To exclude a cross-point (exclusion cable), *hold E* and click the cross-point.

To remove a cross-point, *hold X* and click the cross-point.

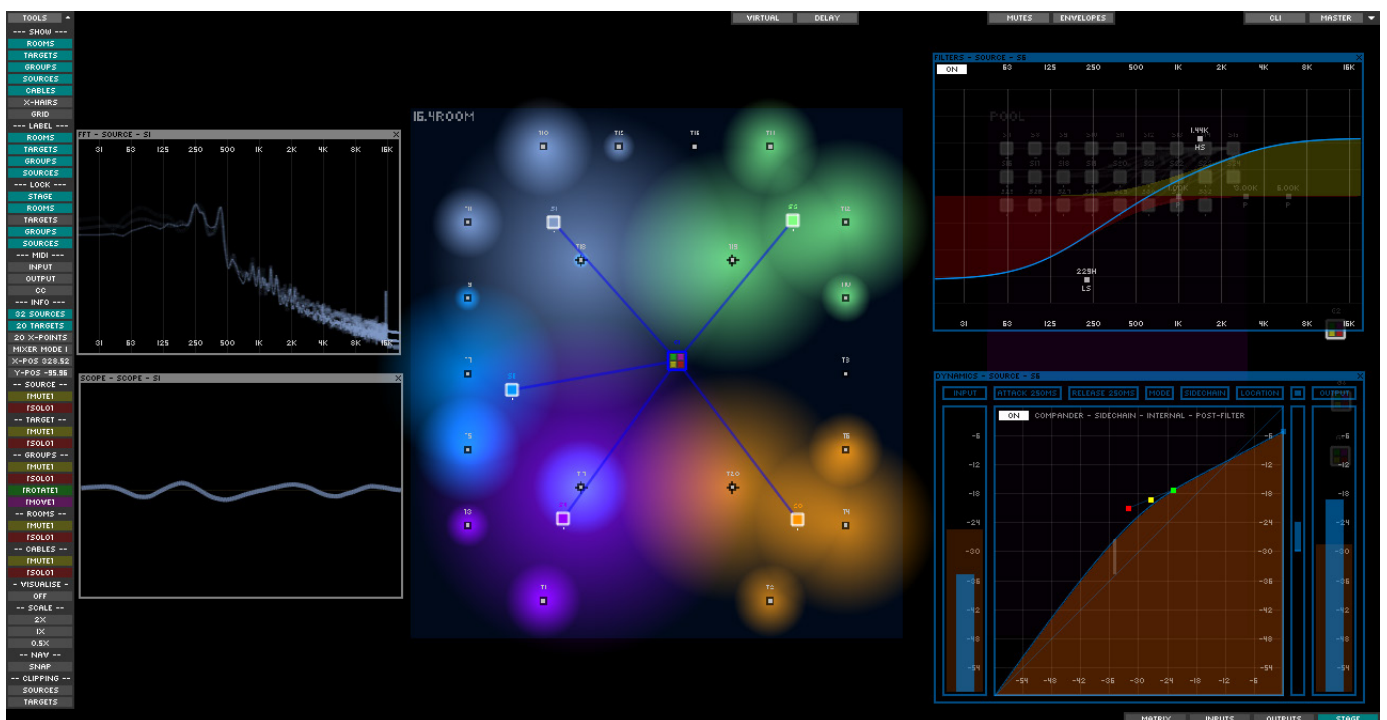




SoundSpiral 48.4 portable venue, London 2012 Cultural Olympiad

SOUNDSQUARES

v1.F



OtherRanks 20 speaker installation, Royal Armories Museum, Leeds, UK

SoundSquares is a 32bit (x86) VST plugin, and as such requires an x86 capable plugin host (DAW) running on Windows 7 or later.

The recommended host DAW is Reaper (4.7 or later) as it is both lightweight and fast, and also incredibly flexible when it comes to routing audio. SoundSquares has successfully been tested using Reaper's built-in bridging when using the x64 version.

Will it run on Linux or Mac using emulation or a virtual machine?

I've honestly no idea, although if you're already successfully running a DAW under such circumstances, then give it a go, as it'll probably run as long as it's on an x86 or x64 ISA.

As for the physical machine to run things on, the likelihood is that if you're running a modern audio setup already then you'll not need to worry about physical system specifications. For the avoidance of doubt however, SoundSquares *requires* an AVX2 compatible processor, which can calculate 256-bit wide floating point vectors. Thankfully compatible processors have been in production since 2013 so if your CPU is Intel Haswell, AMD Excavator or later, then you're good to go.

If in doubt, before running SoundSquares, check that your machine has a processor generation listed on this page :

https://en.wikipedia.org/wiki/Advanced_Vector_Extensions#CPUs_with_AVX2

In terms of system memory, SoundSquares doesn't require anywhere near the amount of RAM you might need to run modern sampler plugins, and whilst not fully optimised for memory usage, its footprint by comparison is relatively modest.

One final point to bear in-mind when it comes to system requirements is that whilst it is perfectly possible to make good use of SoundSquares as a stereo tool, where it really shines is in its intended use-case of addressing multichannel playback environments. To do so, you'll therefore also need a multichannel audio interface running ASIO drivers, with enough channels for your intended setup. An all-in-one interface is recommended, although if you're on a tight budget, ASIO4ALL can be used to bridge multiple smaller-channel-count interfaces into a single driver address space.

And of course you'll also need enough amplifiers, cables, mounting-points and speakers to physically realise your system ... not to mention the small matter of a physical venue to host it in.

In these example setup steps, we'll be using Reaper as the reference DAW, simply due to its large sub-channel per bus count. Broadly speaking the same approach will apply to any multichannel capable DAW you might wish to try, however it must be mentioned that SoundSquares was developed solely for use on top of Reaper as the hosting DAW, and as such may not be fully-featured when using a different platform, especially around the use of keyboard-based commands.

In other words, I'd recommend using Reaper for this stuff, and if not, please remember that your mileage may vary, and I'm not personally in a position to investigate or provide fixes or updates based on anything other than my own usage patterns and intentions, which will remain rooted on Reaper.

STEP ONE : INSTALL THE PLUGIN

This is, by now, pretty vanilla stuff. Simply copy the contents of the .zip file to your plugins folder of preference, and then run Reaper and rescan for plugins using the preferences dialog under PLUGINS > VST. Once this step is complete, SoundSquares will appear in the "Add FX to Track" dialog box when inserting new plugins.

STEP TWO : CONFIGURE YOUR SYSTEM TO REACH YOUR OUTPUTS

As mentioned in system requirements, you'll need a multichannel audio interface, and this will mean pointing Reaper at all of your relevant output channels. In Reaper preferences under AUDIO > DEVICE, make sure that "Output Range" reaches to the highest audio output for the system you intend to target.

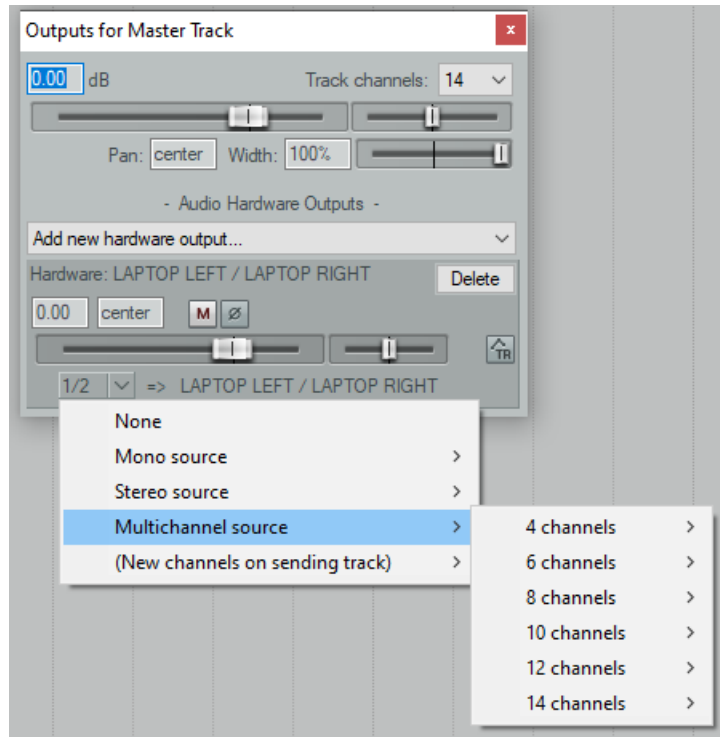
Once you're certain Reaper can connect to your hardware, you can proceed to routing the master track of the mixer to your individual hardware output channels, so start an empty project and open the mixer panel.

Add an instance of SoundSquares to the Master Channel using your normal method of inserting a plugin. Right-click in the left-hand menu (if plugin is docked), or on the title bar if plugin is floating and select "Send all keyboard input to plugin" - this is definitely required to gain full keyboard-based functionality!

Next, in the Master channel, open the I/O routing panel "Outputs for Master Track" and set track channels to the number of physical outputs you intend to use - don't worry if this is more than the number of speakers in your system as Reaper works in stereo pairs, so if your system has 13 speakers you want to address, you'll need to have Reaper setup with 14 Master Track channels.

Next-up you'll need to tell Reaper where to output your Master Track to - which hardware outputs you want to use. If it doesn't already have an existing hardware output, create one using the "Add new hardware output" drop down, and then using the channel select menu (which by default is usually set to "1/2"). Select "Multichannel source", the next menu is where you chose how many channels you want to output to, and the remaining menu after that is where you chose the output range of channels on the device.

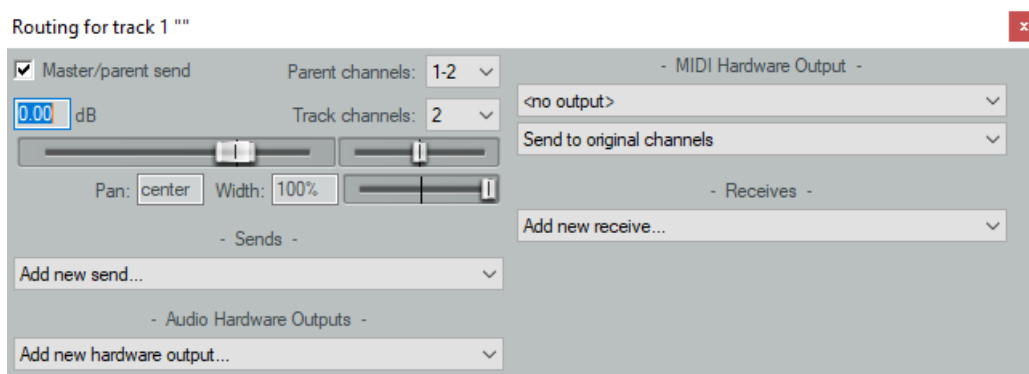
If you're fortunate enough to have more physical hardware outputs than the number of speakers you intend to use, there's nothing to prevent you from starting your system using output 4 or 5 or whatever. For convention's sake, I usually build my rigs from channel 1 upwards, so if I have any hardware outputs left-over, they're at the higher-end of the indexing. Either way, you should now be setup on the master-side of things and ready to start routing audio into SoundSquares



STEP THREE : ROUTING AUDIO INTO SOUNDSQUARES

This is where things can start to get a little complicated, as whilst Reaper is certainly flexible, what we're doing is far from a "normal" use case, and so the step of getting audio from a track channel and into the plugin, whilst straight-forward, can be a little tedious, especially if using an alternative setup to the one presented here.

Let's assume you have 2 stereo track channels in Reaper, and you want to send them to SoundSquares plugin inputs 1&2 and 3&4. To do so, open each channel's I/O routing dialog "Routing for Track ..." and you'll see something like the this :



What this dialog shows is that the 2 subchannels of this track are routed to channels 1&2 of the Master channel. To route them to different input channels of SoundSquares, simply use the "Parent channels" drop-down to select your destination. It helps to enable "Show non-standard stereo channel pairs" under Preferences>Audio, so you can route stereo to not just 1&2, 3&4, etc but also to 2&3, 4&5 and so-on. To route audio to a single plugin input channel, you'll need a combination of the routing pan fader, and width fader so as to "pre-mix" to the destination input as Reaper doesn't have a concept of strictly mono channels, since it expects by default that you'll be mixing to stereo.

To route more channel to SoundSquares, simply rinse and repeat.

It is worth noting, that multiple instances of SoundSquares can be run in a mix. This would entail inserting the plugin in a group channel and then routing the subchannels to the 'parent' group. Furthermore, the instances that you use can be cascaded in series, such that the first instance might be used for creative means, and a second instance might be used as a post-production processor. The general topology of routing this is broadly the same except for the key notable point that by default, Reaper's track channels are pre-configured to be routed directly to the master channel, so you'll need to switch that off per channel if using SoundSquares in a group channel instead of on the master.

The most simple setup for SoundSquares to run is in a very basic 2-in, 2-out configuration for use as a simplistic stereo mixer, so we'll use that as the base example before moving-on to something more interesting.

As well as 2 outputs for stereo, let's assume you have 8 inputs to SoundSquares, and have setup your channel routing in Reaper as described above. If so, then all you will need to do to start mixing is to tell SoundSquares you want to use 8 inputs, and 2 outputs.

There's a pre-existing patch for that, and actually it is the initial default patch using 8 SOURCES and 2 TARGETS, along with straight-through routing on both input and output sides.

So actually, you'll need to do very little other than press play to find that your 8 inputs are now available as SOURCES with the SoundSquares mix environment.

Great, but what if you wanna add more inputs?

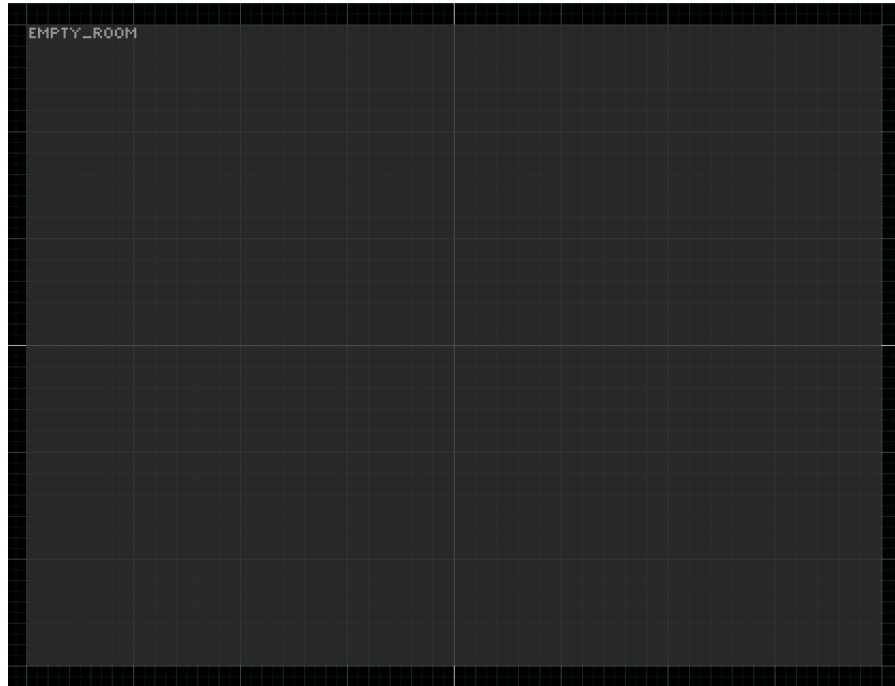
Simply click in the toolbox and enter the number of SOURCES you want to use, and you'll find your new sources arrive in the last ROOM you clicked-in. The same can be said of TARGETS - just tell SoundSquares the total number you want to use, and any new TARGETS will arrive in the last ROOM you clicked-in.

If for whatever reason the routing gets messed-up and your TARGETS don't reach the Track Channels you intend them to in Reaper, you might try selecting "RESET ROUTING" from the main SoundSquares menu available by right-clicking anywhere on the STAGE PAGE which isn't a node or PANEL.

MAKING A CIRCULAR SYSTEM

STEP ONE : TIDY THE STAGE

Remove all pre-existing SOURCES and TARGETS by setting both counts to 0 using the input fields found in the TOOLBOX, or using the keyboard [shift + ;] and [shift + @]. You should see now an empty ROOM : click inside the ROOM.

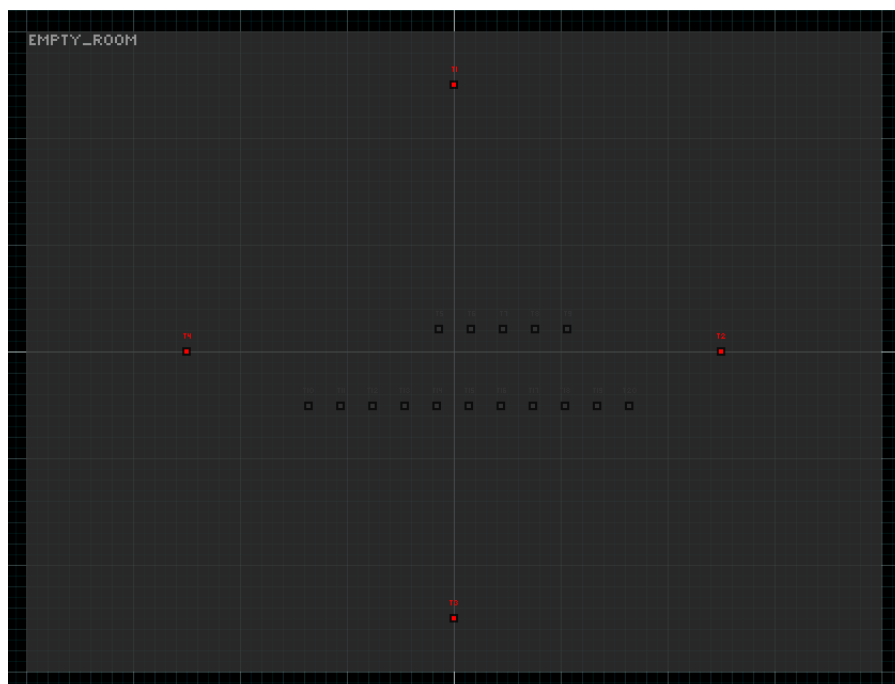


STEP TWO : CREATE NEW TARGETS

Enter the desired number of TARGETS in the same way as the previous step - the new TARGETS will arrive in a line at the top of the last click-in ROOM.

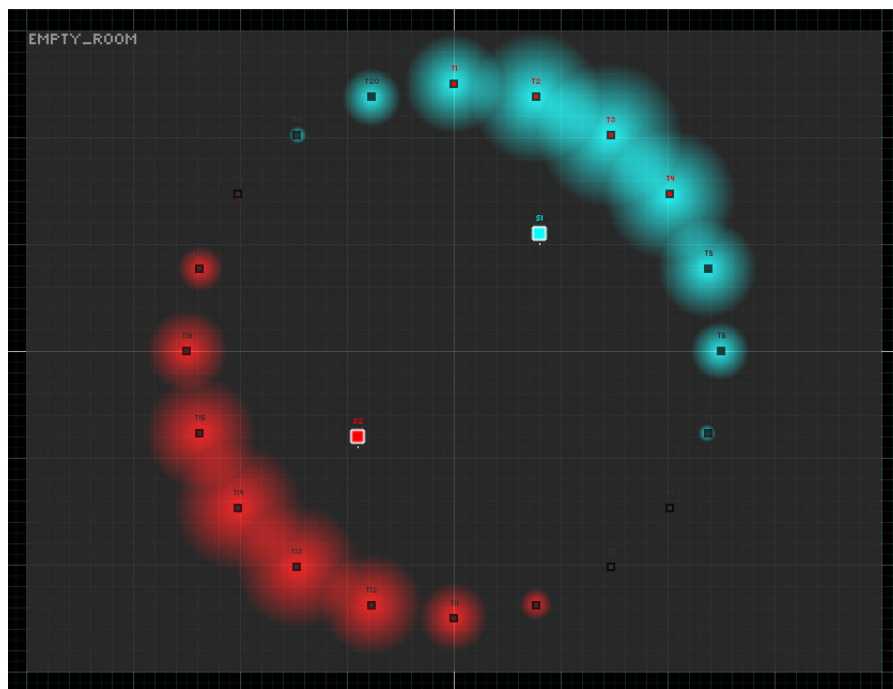
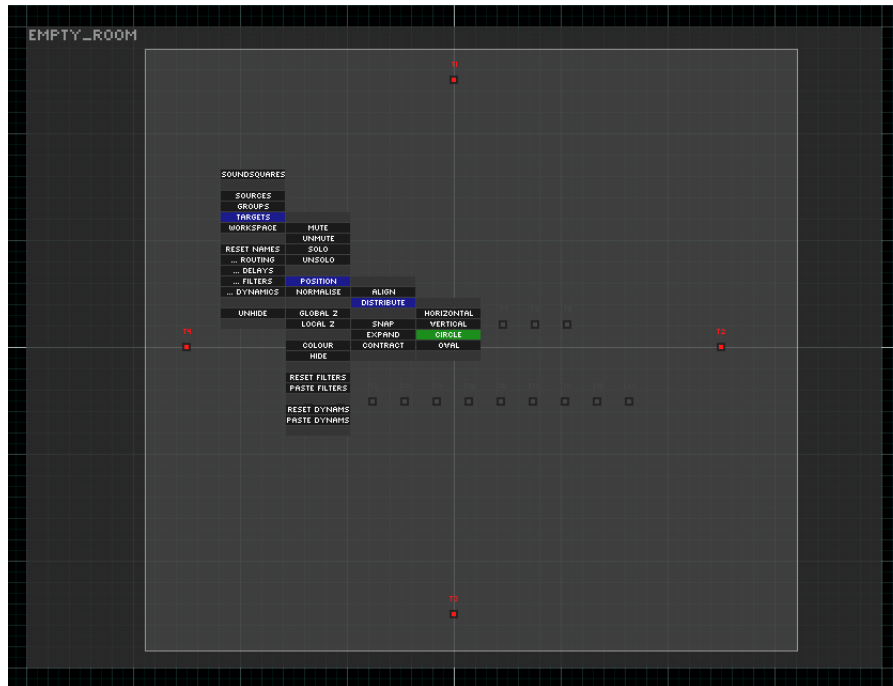
STEP THREE : DEFINE THE PERIMETER

Move the TARGETS to the middle of the ROOM, and then place ONE TARGET EACH at the LEFT, RIGHT, TOP, and BOTTOM edges marking the perimeter of the new *system model*.



STEP FOUR : AUTO-POSITIONING

Draw the MARQUEE around all of the TARGETS, including those which are marking the perimeter, and from the RIGHT-CLICK MENU, SELECT : **TARGETS : POSITION : DISTRIBUTE : CIRCLE**



STEP FIVE (OPTIONAL) : NORMALISE

If you wish to set all the TARGET response circles such that the response of the system when a SOURCE is positioned dead-centre is set to -0dB, go back in the MENU and select : **TARGETS : NORMALISE**. You can inspect the system response using the HEAT MAP overlay via **SHIFT + F**.

YOU NOW HAVE A SET OF TARGETS DISTRIBUTED IN A CIRCLE. YOU CAN NOW START TO INTRODUCE SOME SOURCES TO THE STAGE AND START MAKING SOME NOISE.

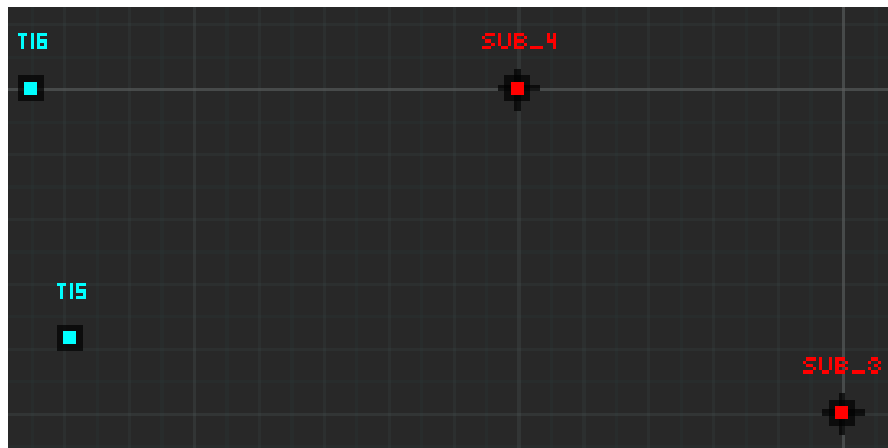
WE'VE GOT SPEAKERS/TARGETS IN SPACE NOW, SO LET'S ADD A COUPLE MORE AND USE THEM AS SUBWOOFERS, AND ADD SOME CROSSOVERS TO SOUNDSQUARES.

STEP ONE : ADD MORE TARGETS

Do this like you did before, remembering to first click in the ROOM where you want the new TARGETS to arrive. Then put TARGETS where you want them. Optionally, they can be coloured by using [1...0] + click or SHIFT + [1...0] + click on each TARGET. If you like, you can name your TARGETS by double-clicking them and typing a name.

STEP TWO : SET THE RESPONSE OF THE NEW TARGETS TO LOCALISED INSTEAD OF GLOBAL

Backslash+click each of the TARGETS - they should now have a different shape indicating local response.

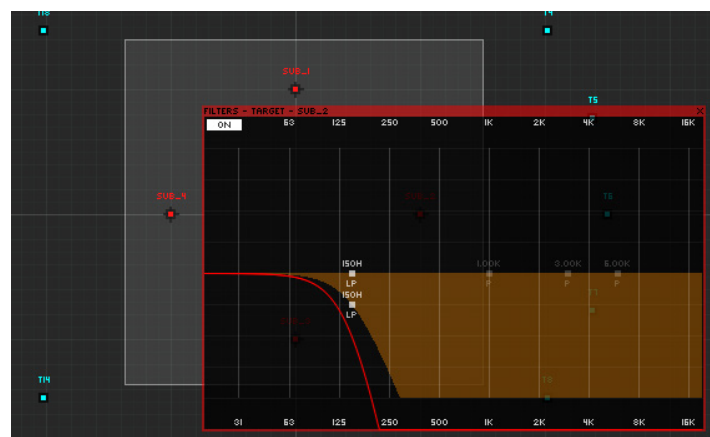


STEP THREE : ADJUST THEIR RESPONSE

CTRL + CLICK + DRAG each new subwoofer to individually set their response circles. Fewer subs will require a larger response circle, but we don't perceive low frequency as particularly direction, so the subs having a different panning law from the rest of the system isn't generally a problem as long as you're reasonably "sensible" about it.

STEP FOUR : LOW PASS CROSSOVERS FOR SUBS

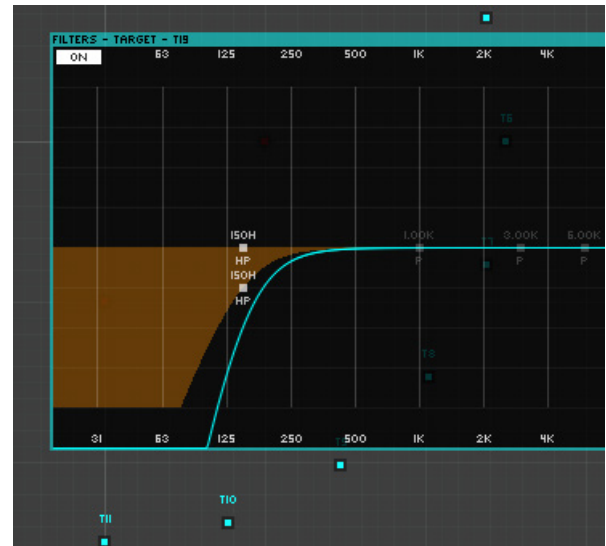
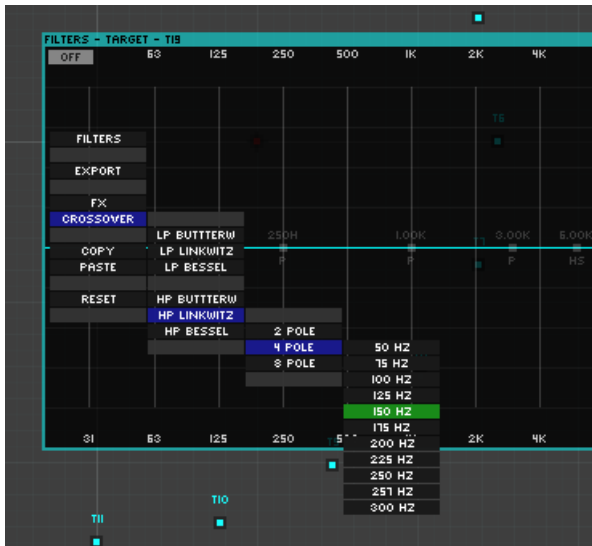
Click on one of the new TARGETS and MARQUEE around the subwoofer TARGETS, and then open the FILTERS PANEL. From the FILTER PANEL right-click menu select **CROSSOVER : LOW PASS** : [AND THEN YOUR CHOICE OF MODEL AND FREQUENCY]



STEP FIVE : HIGH PASS FOR ALL OTHER TARGETS

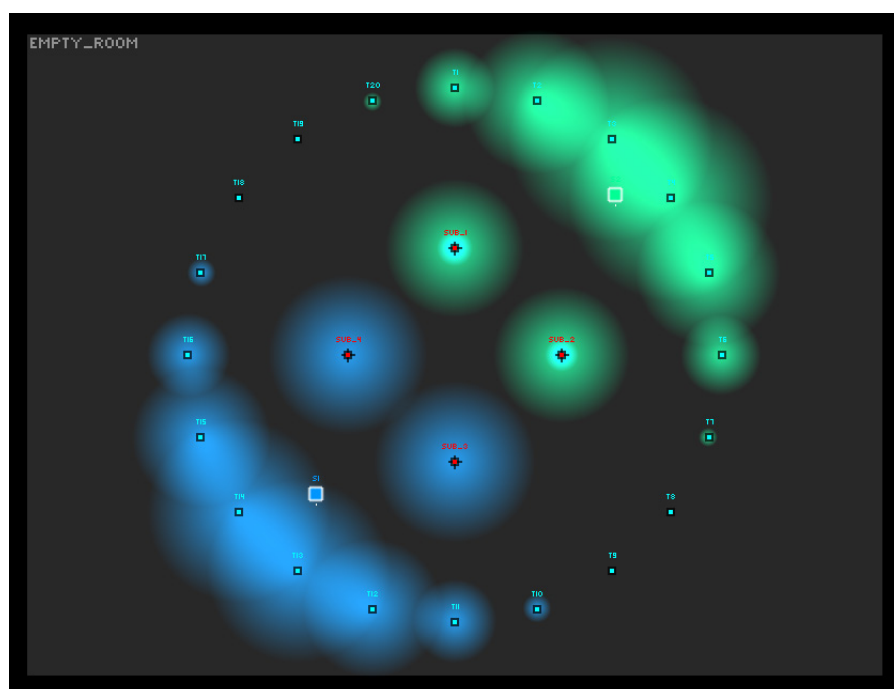
To apply high-pass to all the other TARGETS repeat the above step with one crucial difference. The new SUB TARGETS will need to be hidden to prevent them being syndicated by the next step. D+CLICK with hide them.

Once the SUB TARGETS are hidden, draw the MARQUEE around all the other TARGETS, and click one of them to refocus the FILTERS panel, and from the right-click menu select **CROSSOVER : HIGH PASS : [AND THEN YOUR CHOICE OF MODEL AND FREQUENCY]**



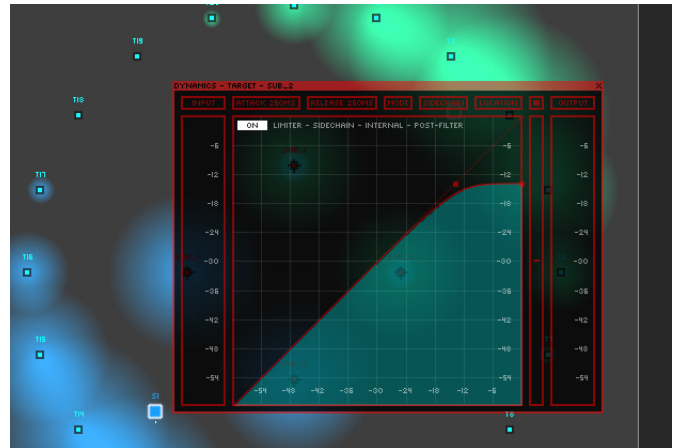
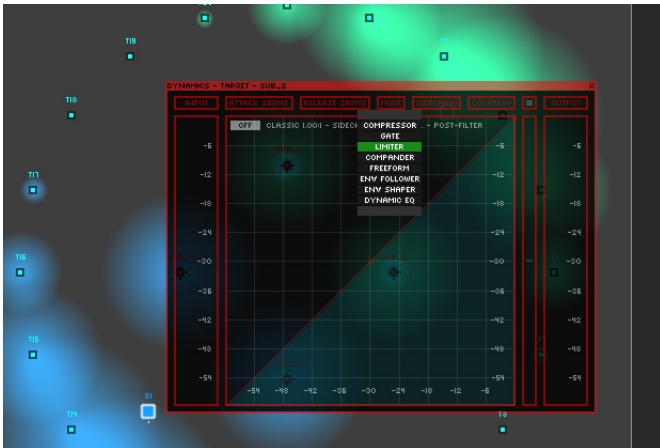
You have just created a [main.sub] sound system with including crossovers. The SUB TARGETS can now be unhidden, and mixing can commence.

IDEA : To avoid any further syndication to the TARGETS you might like to lock all TARGETS using CTRL+J or by using the relevant button in the TOOLBOX, which will prevent accidental changes to their crossovers.

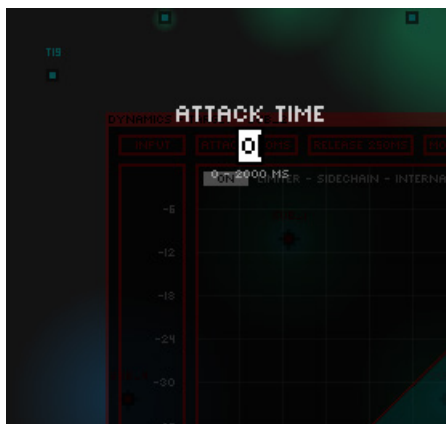


Depending on your usage and system requirements, you may wish to add limiters to the TARGETS. Whilst the SoundSquares DYNAMICS processors do not provide true *brickwall* limiting functionality they can certainly offer a level of protection from overload, distortion, or misuse.

With all TARGETS visible and unlocked, draw the MARQUEE around them all, and open the DYNAMICS PANEL, and select *LIMITER* from the *mode* dropdown.



For most limiting situations a **fast attack** and **relaxed release** is generally desirable, so once you've set the *output limit* by dragging the nodes in the DYNAMICS PANEL interface, click in the fields for *attack* and *release* and enter your required values in milliseconds via text entry.



These values are just a suggestion, and will probably need some fine-tuning depending on your program material. Another thing you might like to experiment with is the *limiter's knee* - how quickly the response turns from compression to limiting. This can be done by CTRL + CLICK + DRAG on the DYNAMICS PANEL NODE that sits on the diagonal line. The knee can be set anywhere from loose and angular.

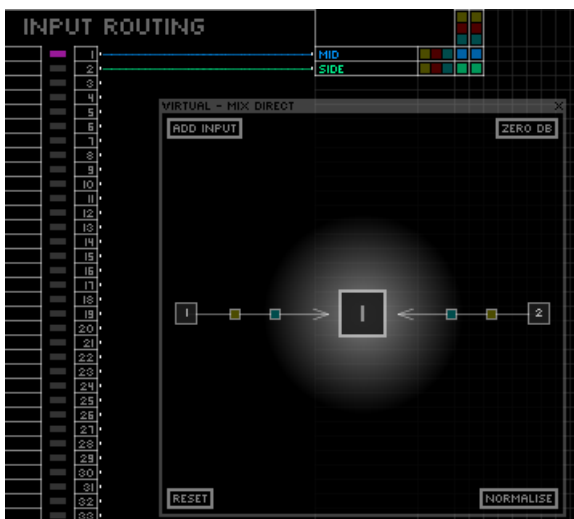


The following examples use stereo as a basic introduction to what might be considered more advanced techniques for creating multiband and multichannel processes. What follows, with a little effort, can easily be extrapolated to multichannel setups.

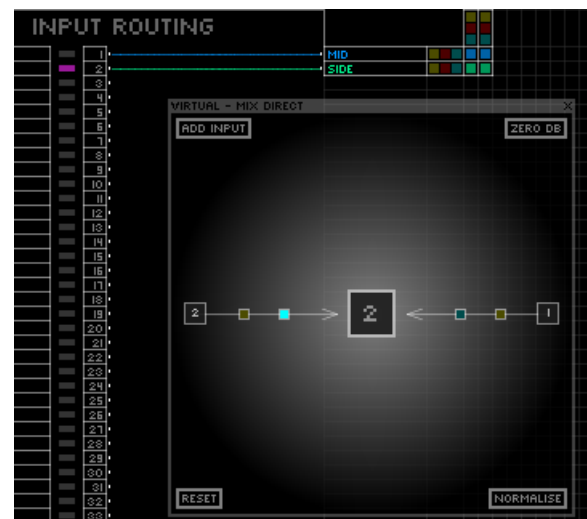
With MID/SIDE EQ, the idea is to ingest stereo audio, but configured as MID and SIDE channels, and instead of processing LEFT and RIGHT channels, to process the MID and SIDE independently before recombining them to output in STEREO.

To achieve this, we'll use the INPUT PREMIXERS to process incoming STEREO into MID and SIDE.

The MID channel is composed of both LEFT and RIGHT channels, normalised to -6dB, and the SIDE channel is composed of the LEFT channel *minus* the RIGHT channel. This will therefore mean using 2 instances of PREMIXER : one to make the MID channel, and the other to make the SIDE channel. MID will then appear as one SOURCE, and SIDE as the OTHER.



Using *plugin channel one's* PREMIXER, add both channel one and channel two to the mixer, and then hit the normalise button. This creates the MID *mono* signal using -6dB of each input channel

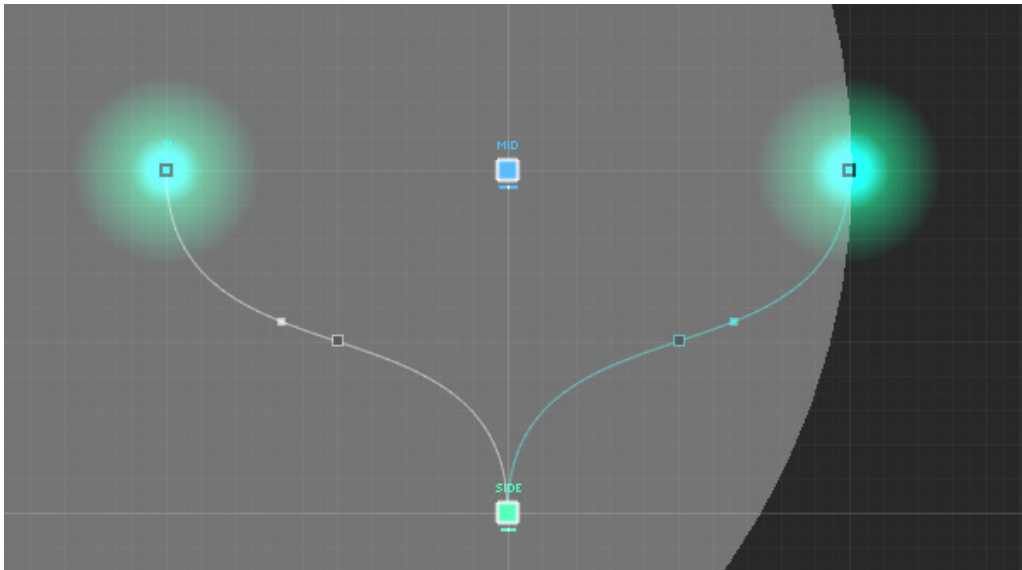


Using *plugin channel two's* PREMIXER, add both channel one and channel two to the mixer, and then *invert the phase of channel two*. This creates the SIDE channel using *LEFT MINUS RIGHT*

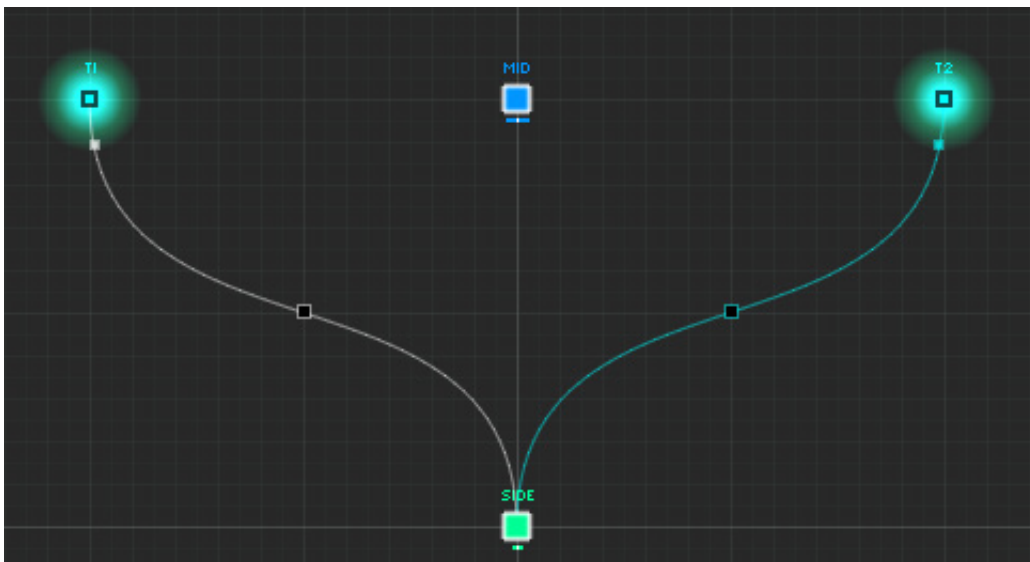
The INPUT PATCHBAY can be left in straight-through mode where *virtual input one* becomes SOURCE ONE and *virtual input two* becomes SOURCE TWO.

MID and SIDE channels can now be processed independently on the STAGE using whichever combination of DELAY, FILTERS, and DYNAMICS you find appropriate. If you're wondering how to use a common SIDECAIN for M/S processing, read-on to the next example ...

To reconstruct the STEREO signal from MID and SIDE channels means adding -6dB of MID to both LEFT and RIGHT, and undoing the *phase subtraction* introduced by PREMIXER TWO which created the SIDE CHANNEL.



Handily enough, -6dB is exactly what we get IF the TARGET responses are setup to meet each other, so use CTRL + CLICK + DRAG on one of the TARGETS in a *stereo pair* with them both set to use a *global response*, and then place the MID SOURCE at the centre-point between the TARGETS. Listening back to just the MID like this will result in *mono playback*.



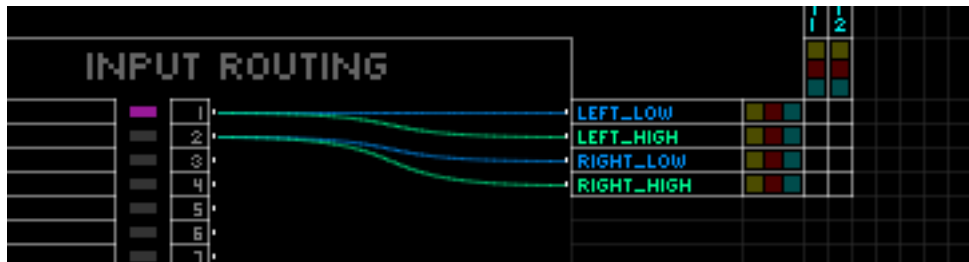
To add the SIDE channels back into the STEREO OUTPUT, use a CABLE to route the SIDE SOURCE to both TARGETS, then Z + CLICK the CABLE CENTRES to set both CABLES to -0dB. From here, all that is needed is for the channel that was phase-inverted by the INPUT PREMIXER to be phase-inverted here too. That was *input 2*, so P + CLICK on the CABLE linking SIDE to TARGET 2, and the result will be back in stereo with the correct left and right sides maintained.

Any changes introduced using DELAY, FILTERS, or DYNAMICS on the MID SOURCE will only affect the *centre* of the stereo image, and any changes introduced on the SIDE COURSE will only affect the stereo *portion* of the image.

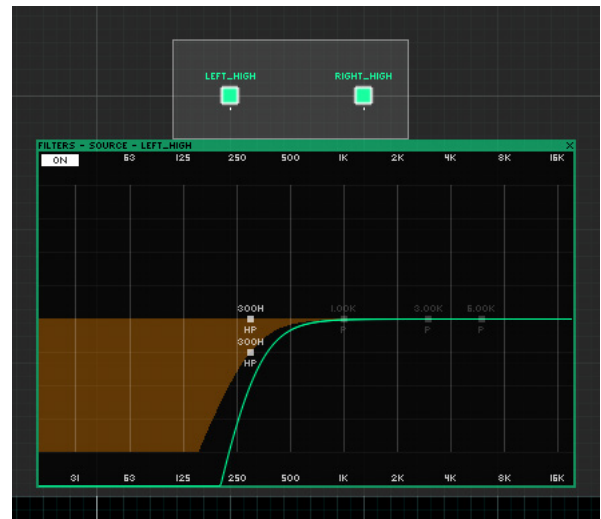
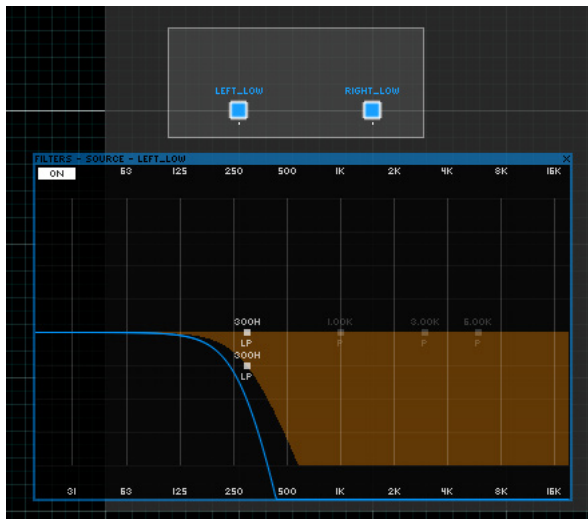
Play around, see what works ...

The second example using stereo as a model for further exploration is the idea of *multiband processing*. Here we will apply different compression settings to the treble and bass portions of an incoming stereo signal using *process syndication*.

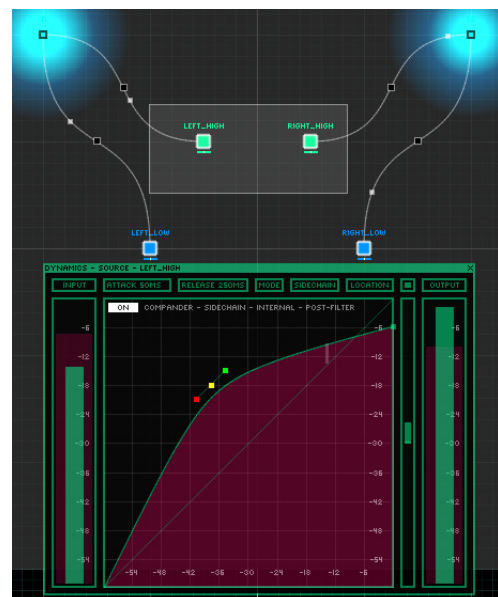
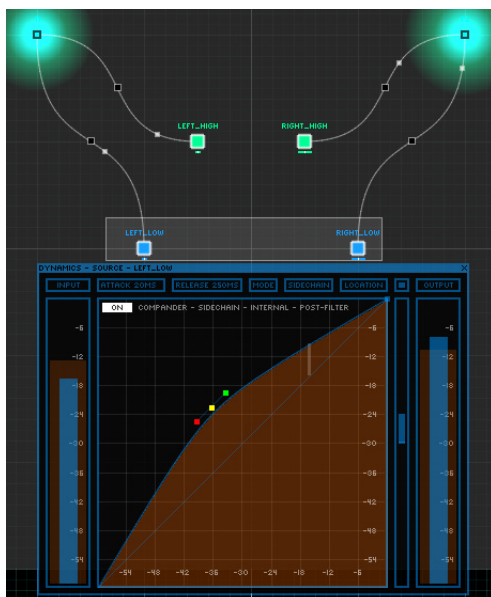
The first step is to create 2 SOURCES for *both left and right inputs*, and to use the INPUT PATCHBAY to route the unprocessed plugin inputs via their *virtual inputs* to the relevant nodes.



The next step is to setup some FILTERING for the LOW and HIGH channels.

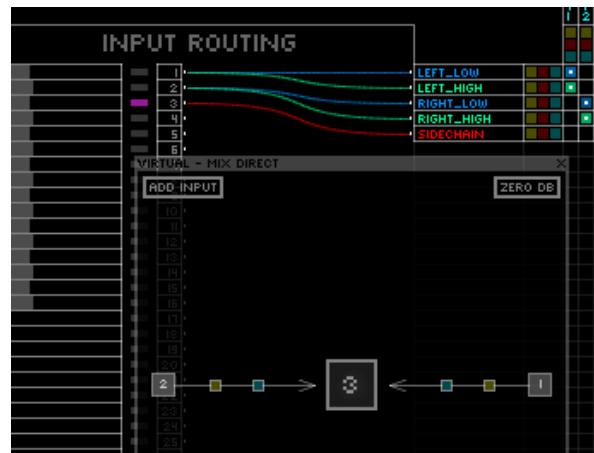


Now, also using *syndication*, different DYNAMICS settings can be applied to *both* LOW bands and *both* HIGH bands, with all outputs being routed at -0dB via cables to the LEFT and RIGHT TARGETS.

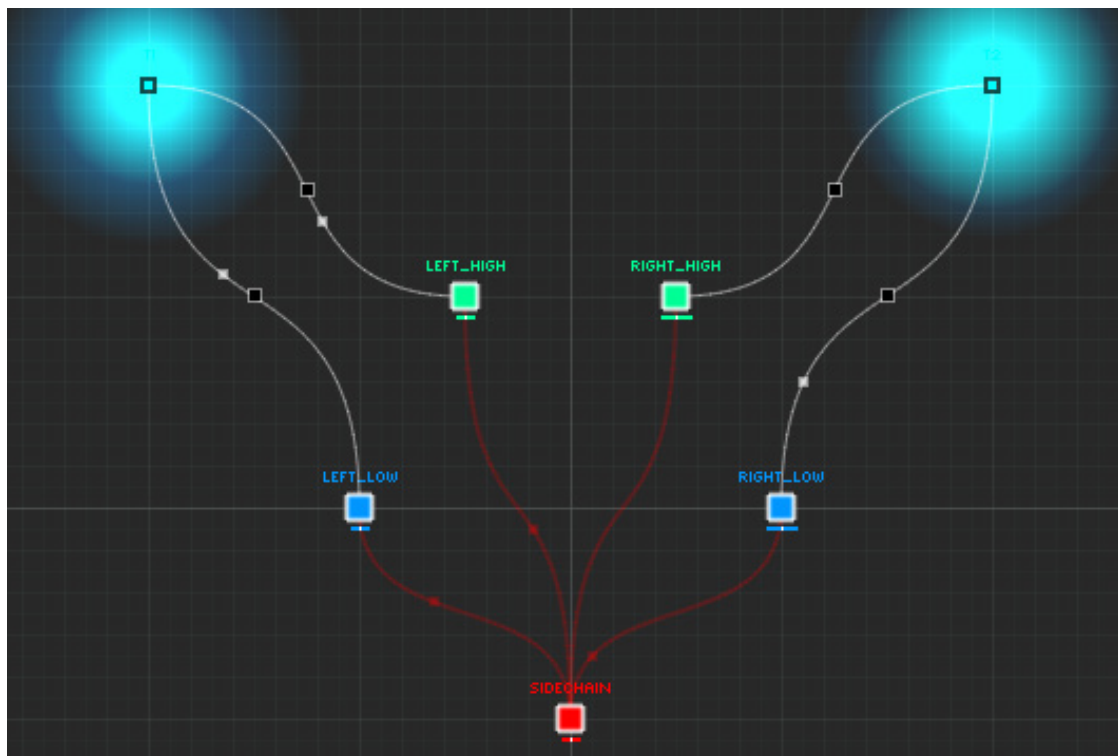


So far so good - we've managed to setup split-band processing and a total of 4 DYNAMICS processors - 2 each for LEFT and RIGHT - or put another way : 2 each for LOW and HIGH bands. This isn't yet really *true stereo yet though*, as all we've managed to do is to create two independent sets of processes. The key to gluing things together is to have all of the DYNAMICS processes use the same *sidechain signal summed* from both LEFT and RIGHT inputs.

To do this, we'll use an INPUT PREMIXER to create an addition SOURCE, using the same idea as creating a MID channel - mixing LEFT and RIGHT inputs, normalised to -6dB, and then use this as the combined SIDECHAIN signal for all DYNAMICS PROCESSES ...



... To do so, CABLE the SIDECHAIN SOURCE NODE to *all* 4 of the other SOURCES nodes. In their DYNAMICS PANELS they will now display the SIDECHAIN input with the colour *taken from the* the node uses as SIDECHAIN. From here, it is quite straight-forward to apply filtering to the SIDECHAIN SOURCE via the FILTER PANEL.



This MULTIBAND *technique* can quite easily be combined with the MID/SIDE *technique* to create some very powerful processes, taking different approaches to the different bands of LEFT, RIGHT, MID and SIDE channels ...

So far we've looked at triggering a DYNAMICS process using itself, or variations of itself as the SIDECHAIN. SoundSquares makes it very easy to use any input as a SIDECHAIN, and also optionally as triggers which can be used in a variety of ways, loosely coming under the banner of ENVELOPE FOLLOWING.

Musically, a SIDECHAIN might be used to GATE a signal, such as a bass drum triggering a synth sound.

In radio, a SIDECHAIN might be used to DUCK a signal, such as is used when a voice-over automatically turns-down the background music when a presenter talks. With some sneaky routing using DELAYS, the ducking effect can also be made to happen *before* the presenter talks, making a *lookahead ducker*.

A more fluid form of a SIDECHAIN GATE is a generic ENVELOPE FOLLOWER, which rather than take a binary approach to switching a signal on or off in response to the SIDECHAIN, instead maps the volume of the SIDECHAIN onto the volume of the modulated SOURCE, which can make for a more musically sympathetic GATING effect. Furthermore, it is the basis for VOCODING.

SoundSquares offers enough SOURCE channels and processors that it can be used to create a 30 band *mono* or a 21 band *stereo* VOCODER. I'll not show you directly how to do that, but instead layout the logic here incase you fancy exploring. Here's the stereo algorithm :

- 1) route the SIDECHAIN (**modulator**) signal to 21 SOURCES
- 2) apply a different *band pass* FILTER to each of the SIDECHAIN SOURCES
- 3) route the affector (**carrier**) signal to 21 SOURCES for LEFT and another 21 SOURCES for RIGHT
- 3) apply the same set of FILTERS to the CARRIER SOURCES
- 4) setup each FILTERED carrier SOURCE'S DYNAMICS processor to respond to its corresponding FILTERED modulator SOURCE, and have the processor run in *envelope follower* mode.
- 5) recombine the individual carrier SOURCE *bands* into an OUTPUT / TARGET either via CABLES or via a *proximity mix* using the STAGE.
- 6) play some ambient synths through the carrier channels, and an epic drumkit through the modulator channels : notice how cool this is, and then consider that rather than being limited to *stereo output*, each band of the vocoded output can potentially now be routed to a different speaker ...

And one final thought on ENVELOPE FOLLOWING is to point-out that the DYNAMICS processors also have a mode which can be used to create *dynamic EQ effects*. This is where the SIDECHAIN, rather than trigger volume changes, is used to control the *wet/dry* mix between a PRE-FILTERED signal and a POST-FILTERED signal. This enables volume to now also impact frequency. To achieve this, use a combination of the techniques shown above, setting-up the affected channel with some change to its FILTERS. When in *dynamics EQ* mode, the channel's PRE-FILTER and POST-FILTER signals will now be controlled by the SIDECHAIN, including the use of *attack and release*.

oh, and P.S. ... the *envelope shaper* using fast attack and release can also be used to make distortion effects and add harmonics to your audio ... worth checking-out using FFT and some sinewaves.

