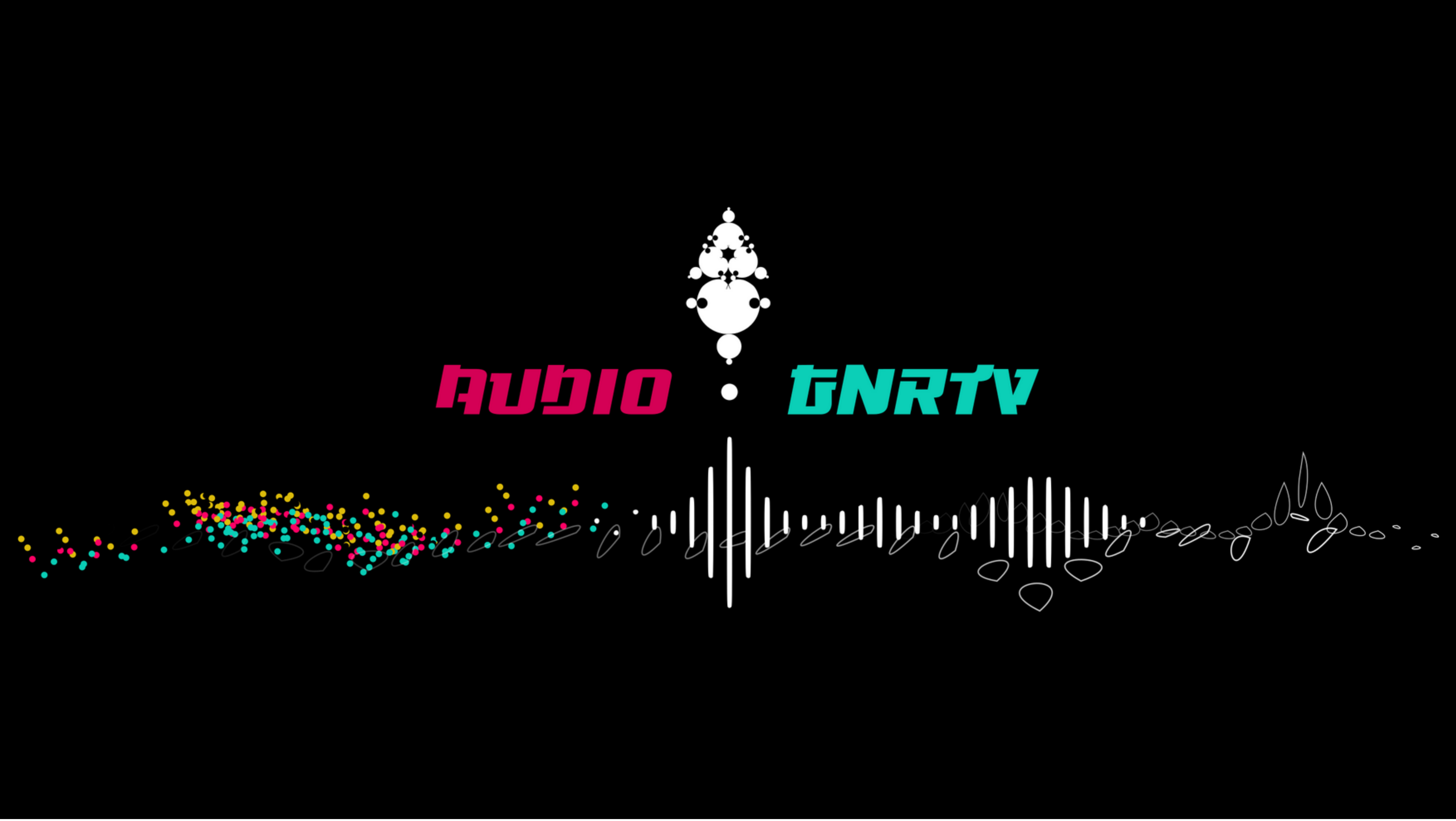


Overview *gnrtv.cells* Pd Toolkit

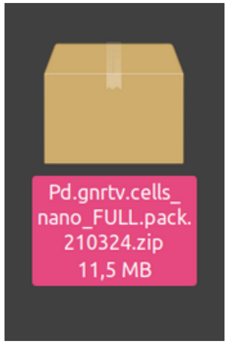
Quick view gnrtv.cells Pd Toolkit

***1**___Install purr-data in your OS. Here there are the repositories :*

purr-data
https://puredata.info/downloads/purr-data
https://github.com/agraef/purr-data/releases

Download packages depending on OS >
macos > https://github.com/agraef/purr-data/releases/download/2.19.3/purr-data-2.19.3-macos-x86_64.zip
windows > https://github.com/agraef/purr-data/releases/download/2.19.3/purr-data-2.19.3-mingw-x86_64.zip
linux ubuntu / deb > https://github.com/agraef/purr-data/releases/download/2.19.3/purr-data-2.19.3-ubuntu-x86_64.zip
https://software.opensuse.org//download.html?project=home%3Aaggraef%3Apurr-data-jgu&package=purr-data

***2**___Download gnrtv.cells package and unzip in any location in your lapotp*



***3**___gnrtv.cells is a toolkit to **easily design sonic generative algoryhtms with pd**, which will require all folders of the toolkit arranged in the same level. Those directories/folders are the following :*

***CODE** open **gnrtv.cells.nano__CORE.template.pd** which is the main template. Save it with other name in the same folder or create another one for ex.'projects' in the same level of the rest of folders, otherwise several functions will not work.*
Also if you want to increase interaction in your instruments there are those 3 folders which contains examples and methods to interact with line in / microphone / capture signal / Midi devices and OSC tunnelling messages (OSC listen and sending)

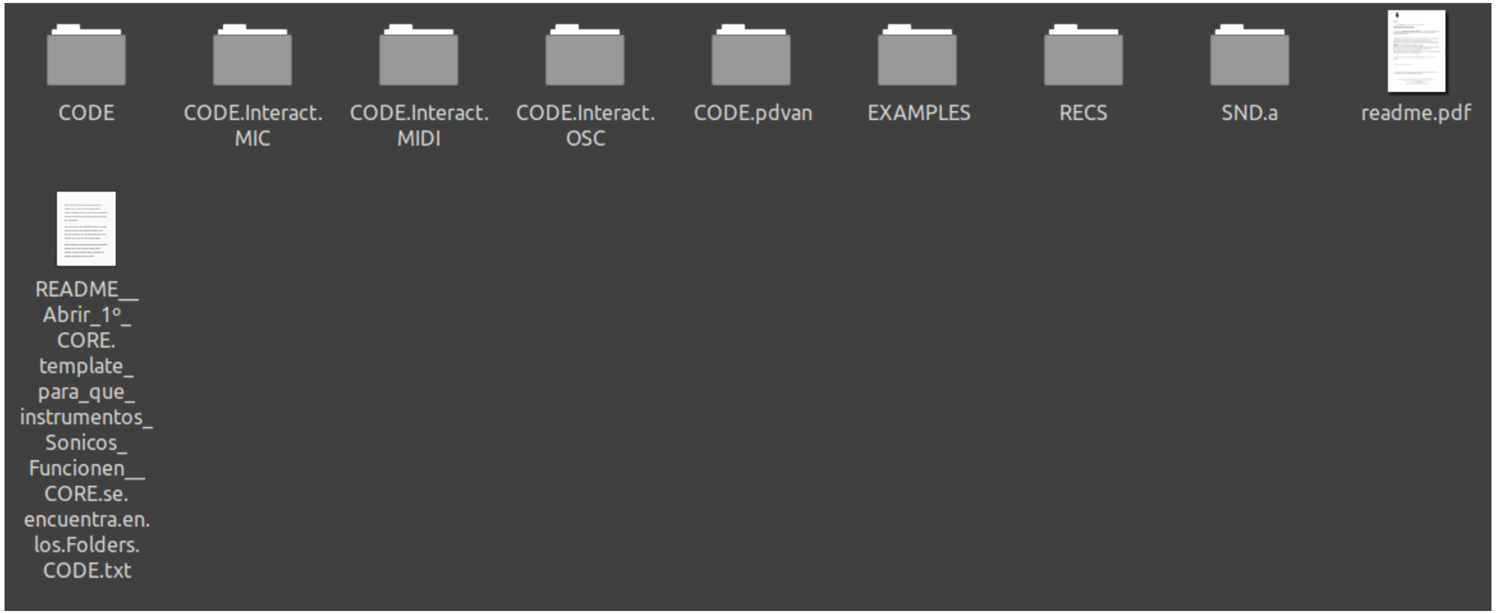
CODE.Interact.MIC
CODE.Interact.MIDI
CODE.Interact.OSC

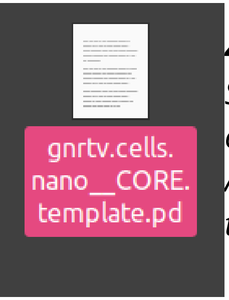
***EXAMPLES** a bunch of tiny examples of generative sonic design.*

***RECS** all live sets can be recorded with the start.rec / stop.rec buttons that appears in the bottom section of the Core's Rack (block which contains FX mixer, signal sends and main clock).*
Those recordings if you have Core in tyour patch will be stored in this folder.

***SND.a** a couple of samples in .wav in case you want to try samplers (cells is mainly a generative synth tool).*
Remember to use samples in .wav with 16bits

*In addition there is a **CODE.pdvan** in case you have to use cells with PdVanilla version which is the classical version of pd which works in many plkatforms and devices, very useful also in luthierism applications.*

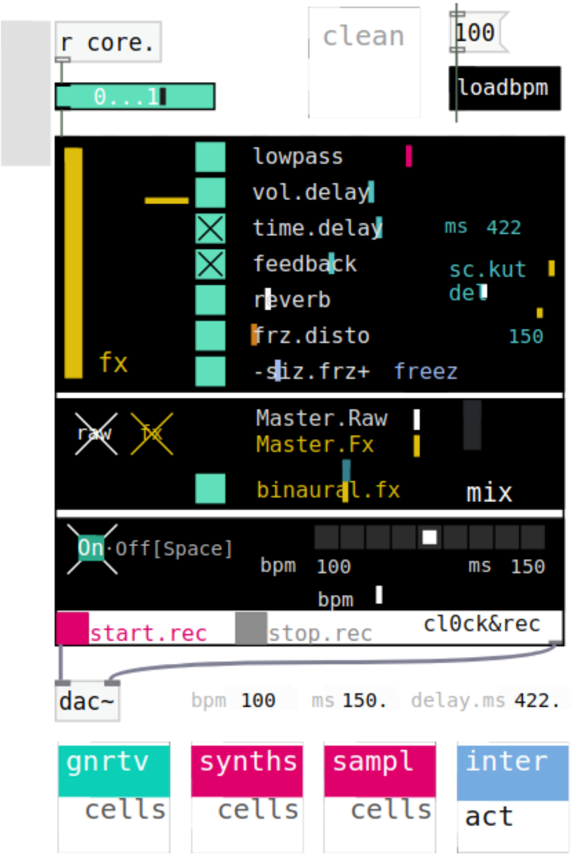

CODE
CODE.Interact. MIC
CODE.Interact. MIDI
CODE.Interact. OSC
CODE.pdvan
EXAMPLES
RECS
SND.a
readme.pdf
README__Abrir_1º_CORE.template_para_que_instrumentos_Sonicos_Funcionen__CORE.se.encuentra.en.los.Folders.CODE.txt



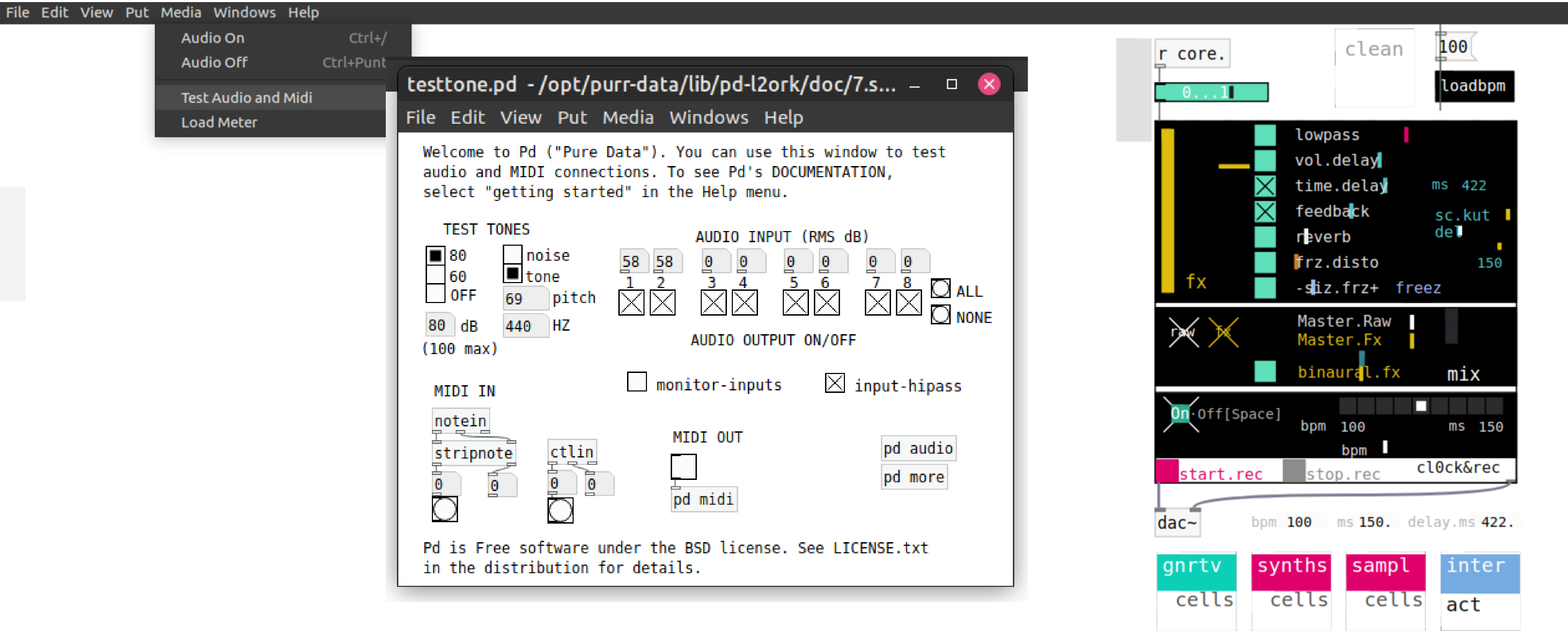
*4__Open **gnrtv.cells.nano__CORE.template.pd** which is the main template. (In any CODE folder there is a CORE.template)*
Save it with other name in the same folder or create another one for ex.'projects' in the same level of the rest of folders, otherwise several functions will not work.
Also if you want to increase interaction in your instruments there are those 3 folders which contains examples and methods to interact with line in / microphone / capture signal / Midi devices and OSC tunnelling messages (OSC listen and sending)

CODE.Interact.MIC
CODE.Interact.MIDI
CODE.Interact.OSC

important! Remember that you can only have 1 CORE file opened, otherwise if you have several CORE files opened, internal messaging will be dubbed and application will not work.



5___*Check if your Sound Card is working : go to menu media > 'test audio and midi'and select the 80 value in the Test Tones area. This should emit a sinusoid of 440hz which corresponds to the note (La / A4).*

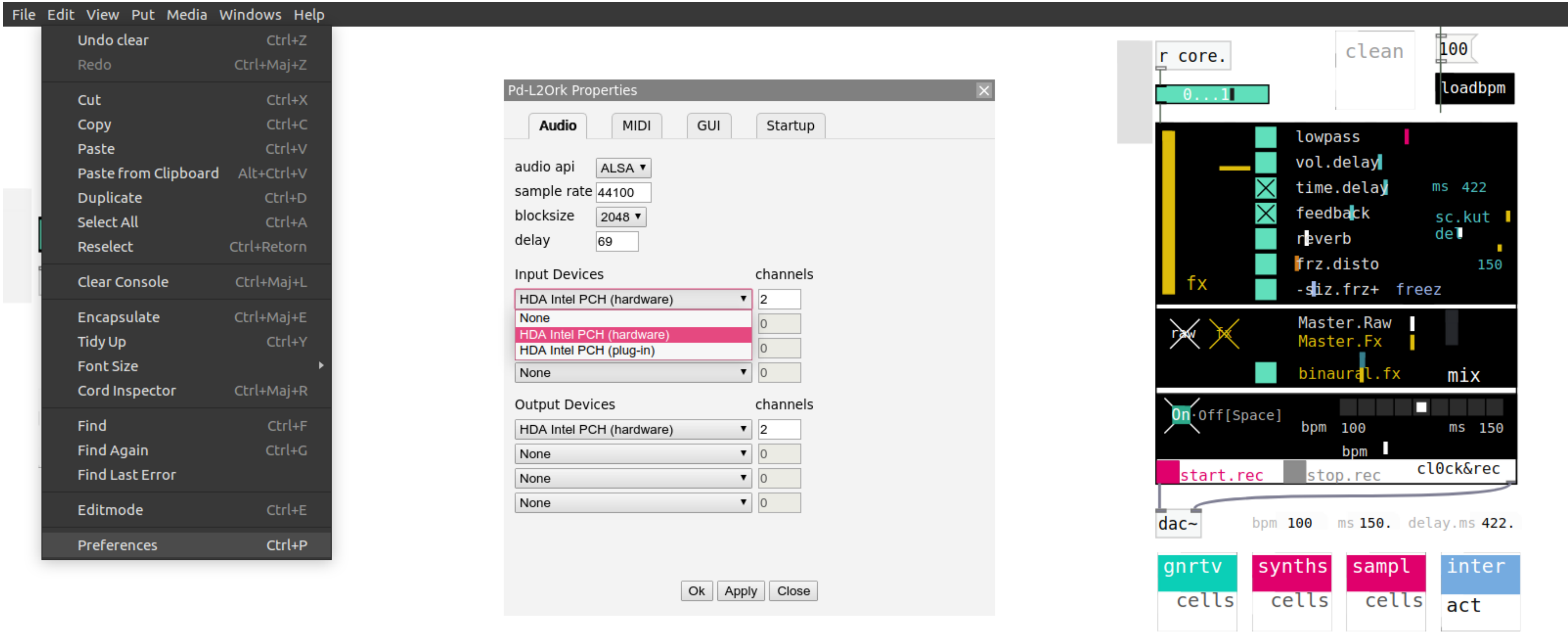


Welcome to Pd ("Pure Data"). You can use this window to test audio and MIDI connections. To see Pd's DOCUMENTATION, select "getting started" in the Help menu.

Pd is Free software under the BSD license. See LICENSE.txt in the distribution for details.

In case sinusoid tone does not appear, you have to check your audio equipment and also the setting up of the soundcard in menú > edit > preferences

A pop up is emerging with different parameters to tweak > sample rate usually 44100 / memory DSP block size / delay or latency (best results over 50ms/60ms) and also set up of input and output channels in your dac~/ soundcard



Pd-L2Ork Properties

Audio | MIDI | GUI | Startup

audio api ALSA

sample rate 44100

blocksize 2048

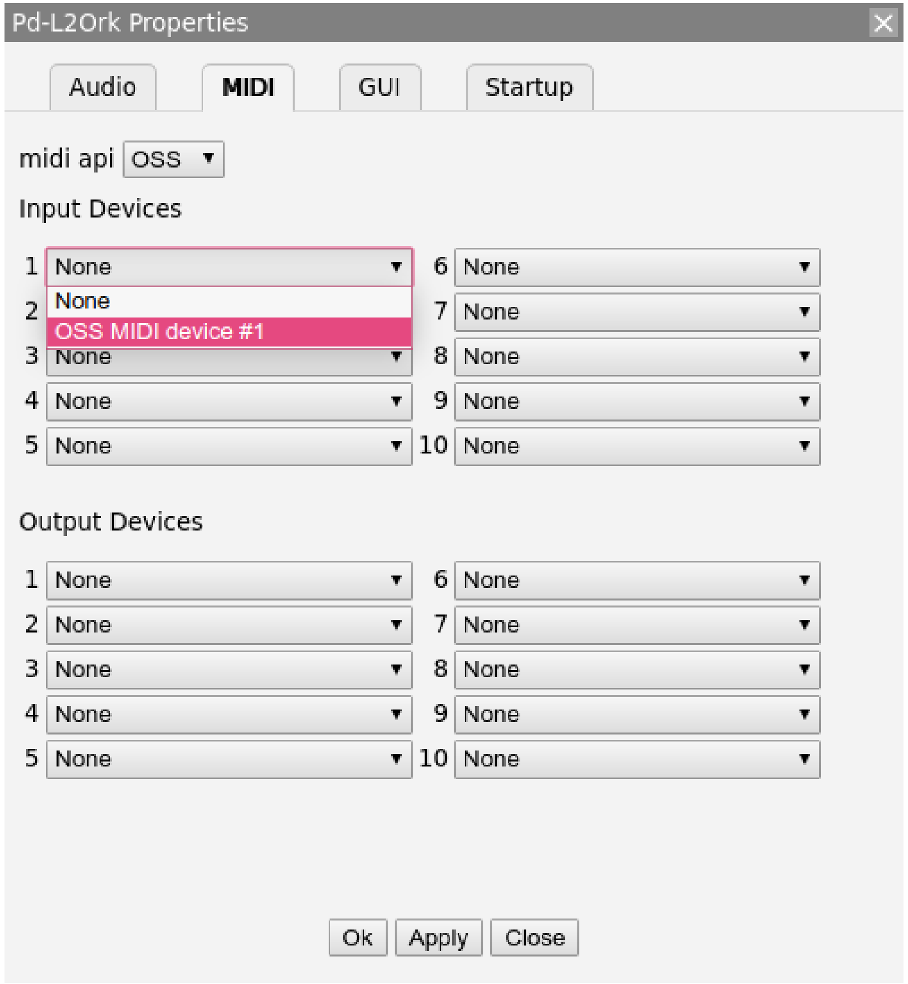
delay 69

Input Devices	channels
HDA Intel PCH (hardware)	2
None HDA Intel PCH (hardware) HDA Intel PCH (plug-in)	0
	0
None	0

Output Devices	channels
HDA Intel PCH (hardware)	2
None	0
None	0
None	0

Ok Apply Close

Also you can set up in this preferences popup (edit menu > preferences) the midi devices in case you are using it.



midi api OSS

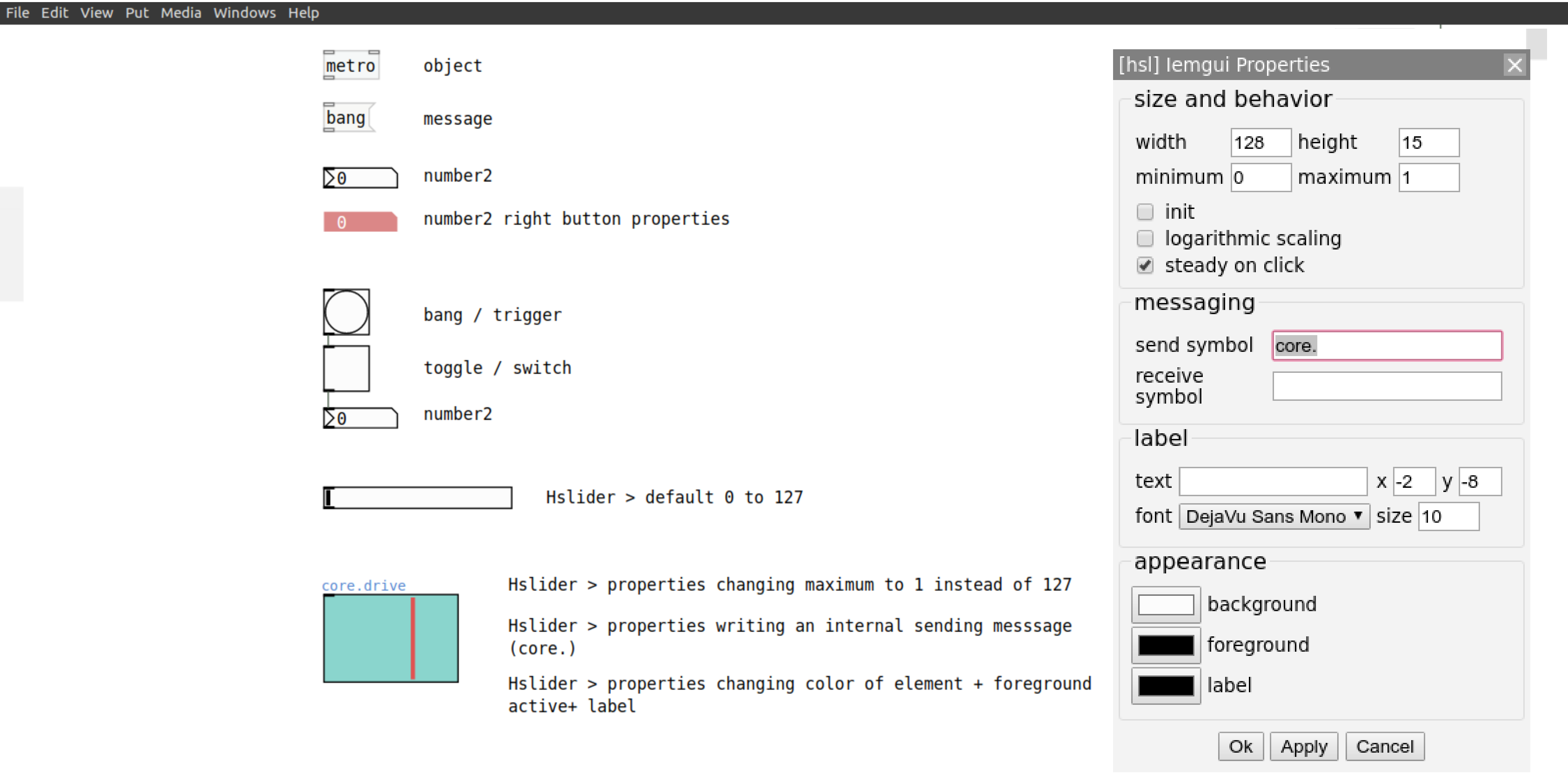
Input Devices

1 None
None
OSS MIDI device #1
2
3 None
4 None
5 None
6 None
7 None
8 None
9 None
10 None

Output Devices

1 None
2 None
3 None
4 None
5 None
6 None
7 None
8 None
9 None
10 None

6__Before start lets remind a bunch of pd syntax elements. ***All pd syntax elements are located in menú put***
Several gui elements like number2 / bangs / toggles / Hsliders / Vsliders /Hradio / Vradio can be tuned with different properties
(right button over the element > properties > appears a popup to change several parameters)



```
metro    object

bang     message

0        number2

0        number2 right button properties

         bang / trigger

         toggle / switch

0        number2

         Hslider > default 0 to 127

core.drive

Hslider > properties changing maximum to 1 instead of 127

Hslider > properties writing an internal sending messsage
(core.)

Hslider > properties changing color of element + foreground
active+ label
```

[hsl] Iemgui Properties

size and behavior

width 128 height 15
minimum 0 maximum 1

□ init
□ logarithmic scaling
☑ steady on click

messaging

send symbol: core.
receive symbol:

label

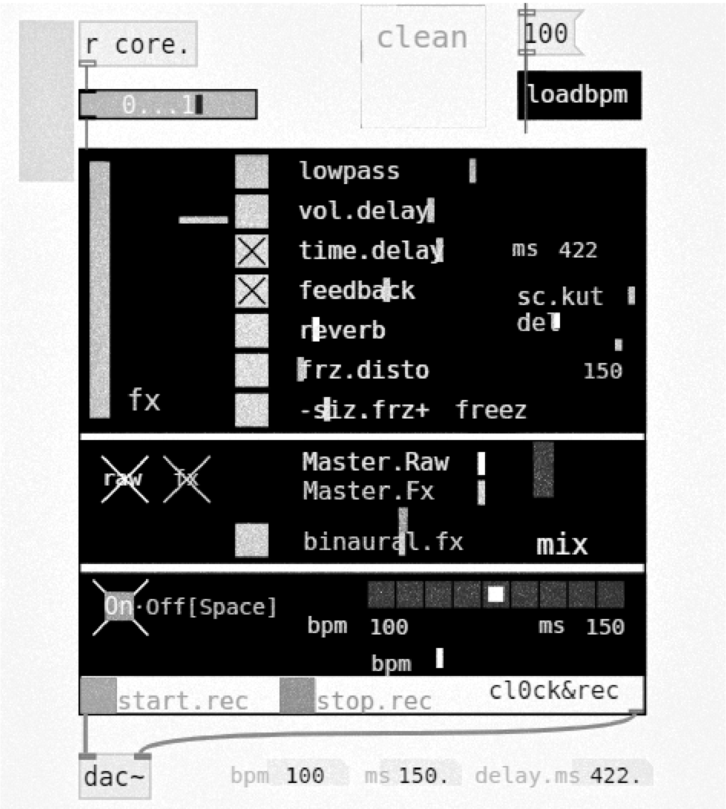
text: x -2 y -8
font: DejaVu Sans Mono size 10

appearance

background
foreground
label

***7**_In gnrtv.cells we have available a set of elements that are interconnectable between them (inspired by modular synths framework).*
All values are normalized to a range from 0 to 1
..therefore many parameters can be translated and interconnected between modules an sections in a easy way.

gnrtv
cells



gnrtv
cells

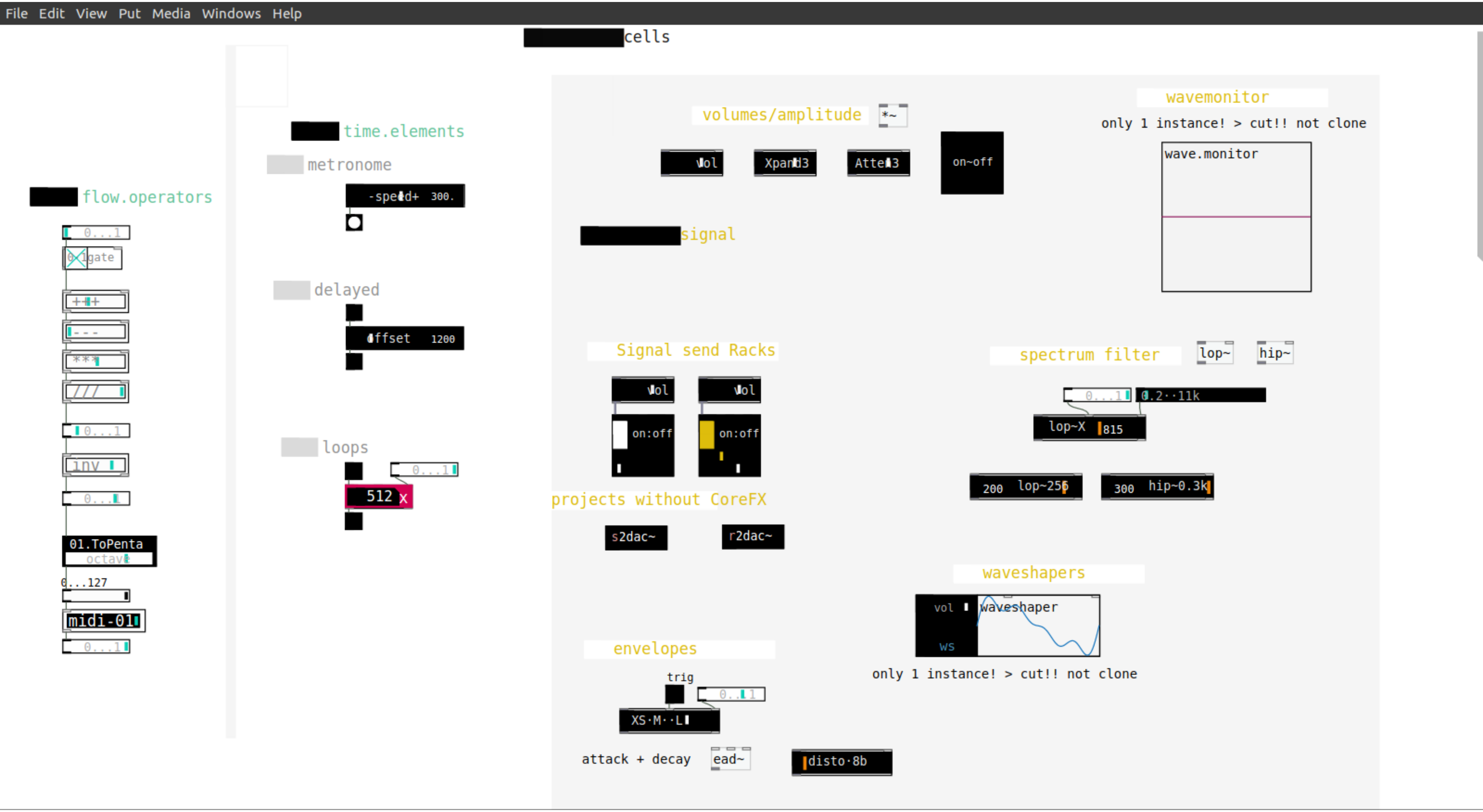
synths
cells

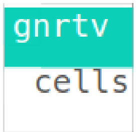
sampl
cells

inter
act

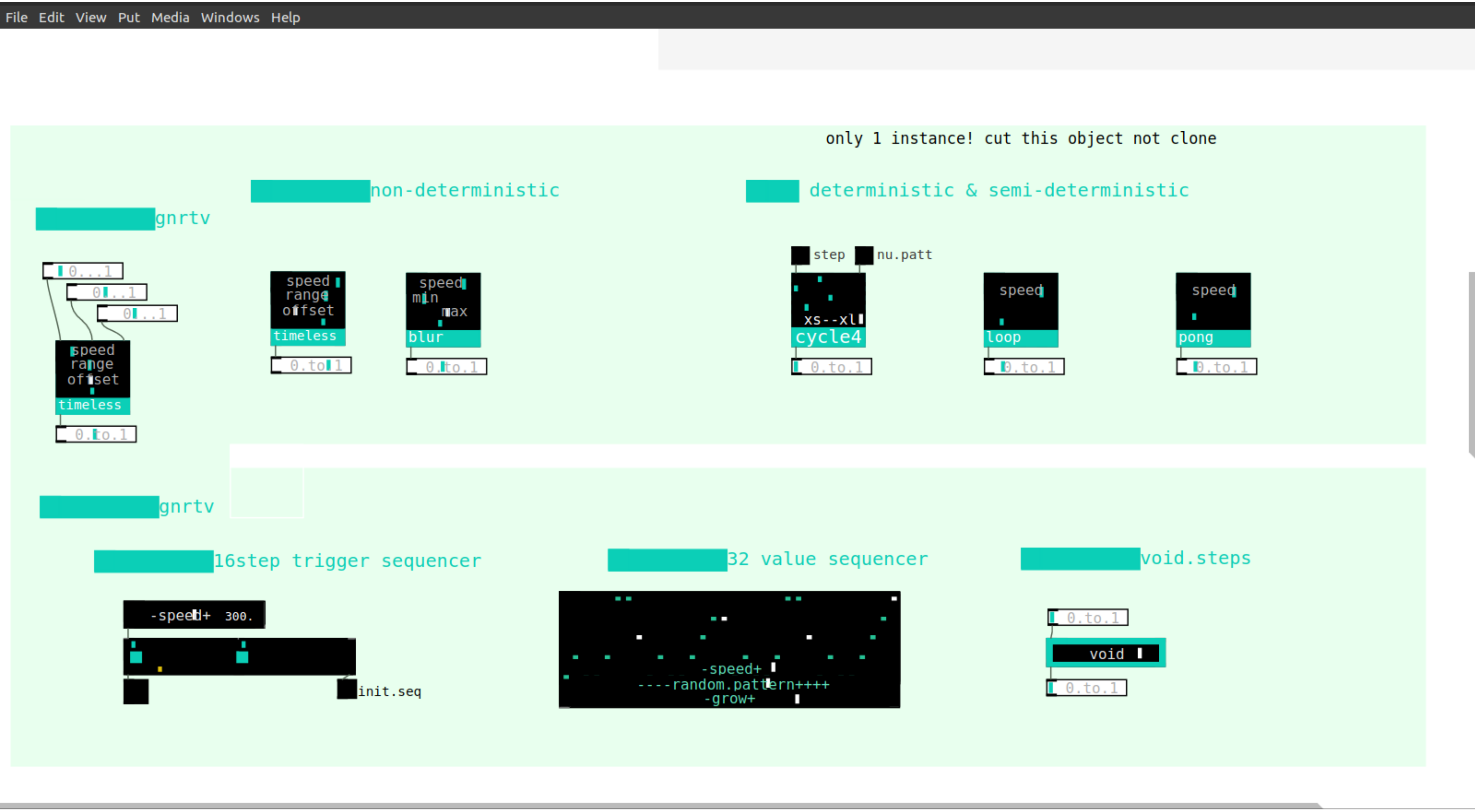
Clicking over gnrtv menú you'll find elements of math tweaks of the 0 to 1 values (flow operators)
elements of time (speed / offset / loops)
cicles (loops like 'for' in other languages)
vols / expanders / attenuators / on~off
filtering & control signal (signal send racks / spectrum filters / envelopes / waveshapers and 8bit distorsion

gnrtv
cells



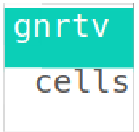


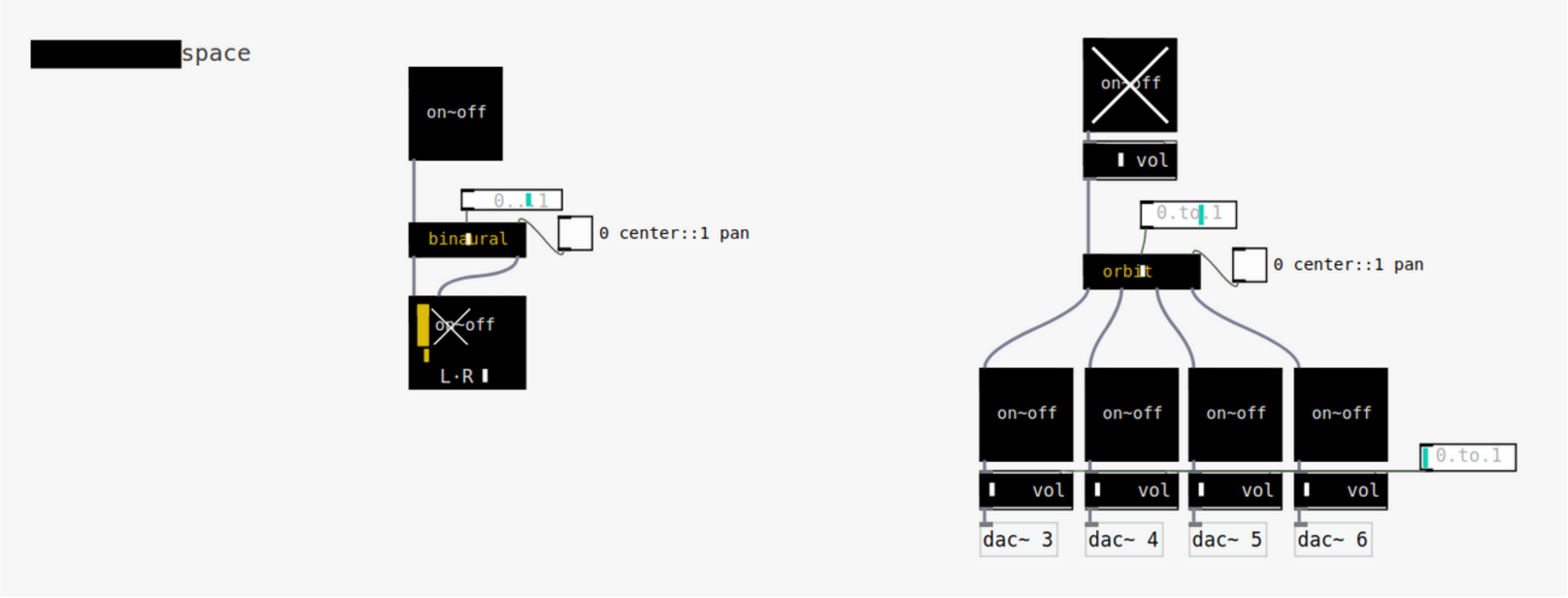
Clicking over gnrtv menú you'll also find generative elements which creates different strings of values from 0 to 1.
There are modules more undeterministic or randomic (timeless blur and void) and the rest which even are generatively created are more deterministic, structured or less randomic.



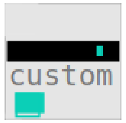
only 1 instance! cut this object not clone

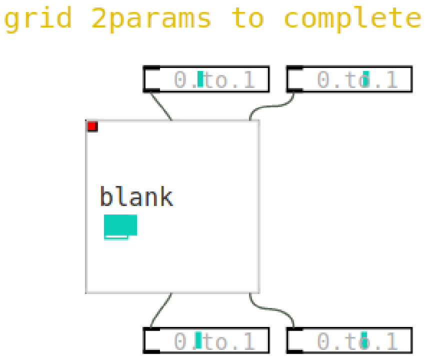
*Clicking over gnrtv menú you'll also find sonic space elements like **binaural** and orbit which can be used to control an L+R or stereo outcome or **'orbit'** which can be used for a Quadraphony multifocal sound system.*

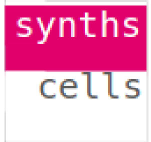




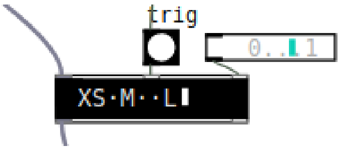
*Also in this menú you have a **blank-grid** to control 2 parameters at once and a custom-blank abstraction if you want to create yours.*

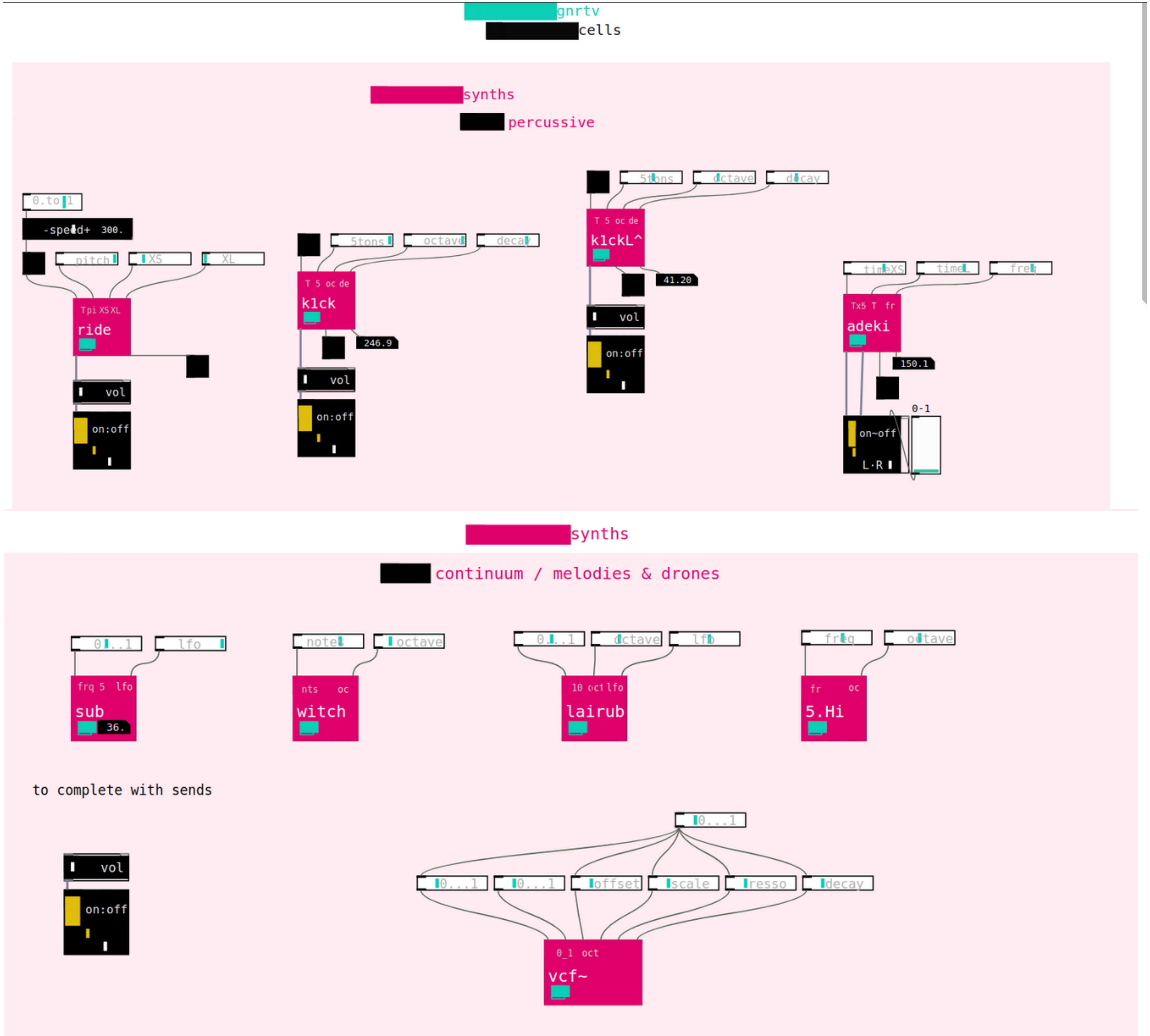


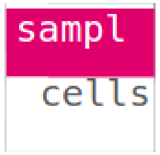




*Clicking over synths menú youll find synthesizers..divided in percussive synths and continuum synths. Anyways those which are continuous can be converted in percussive controlling its envelope. In **gnrtv.menu** section you can find the element **XS M L** to change the sonic decay*

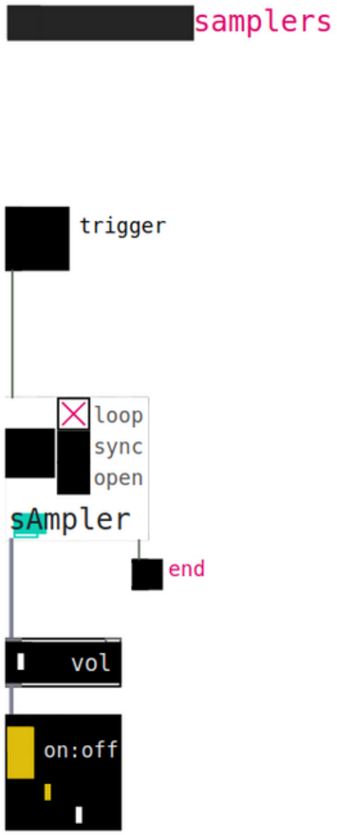


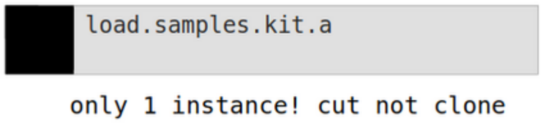


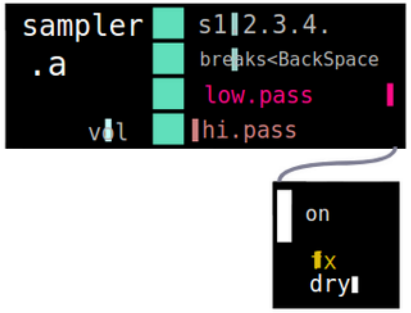


Clicking over *sampl menú* *you'll find sampling modules*

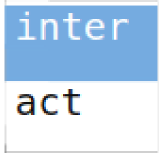
In ***sampler.a*** *block sounds are called and stored in memories (arrays) with the sounds contained in folder* ***SND.a***







(Optional & Advanced)
Advanced method to build more samples > In case you want more samplers, you can copy ***sampler.b.pd (inside EXAMPLES folder****)* *content within your project. You ll also need to create a folder called* ***SND.b*** *(in the same level of the cells default folders) and copying inside 4 .wav files. This will create a set of 4 extra audio samples in your project.*
The previous operation is recursive, which means that if you need more samplers you have to clone the file ***sampler.b.pd*** *with for example* ***sampler.c.pd****, and opening with an editor like sublime. There, you can search within the file all* ***.b*** *and replace it with* ***.c***
Save it and create the folder ***SND.c*** *with 4 more .wav files. ....And so forth.*



clicking Inside interact menú you'll find different elements and methods useful for interaction.

Therefore elements like :
mic // which captures the line in / microphone input and analizes envelope peaks,frequency detected and envelop dynamics.

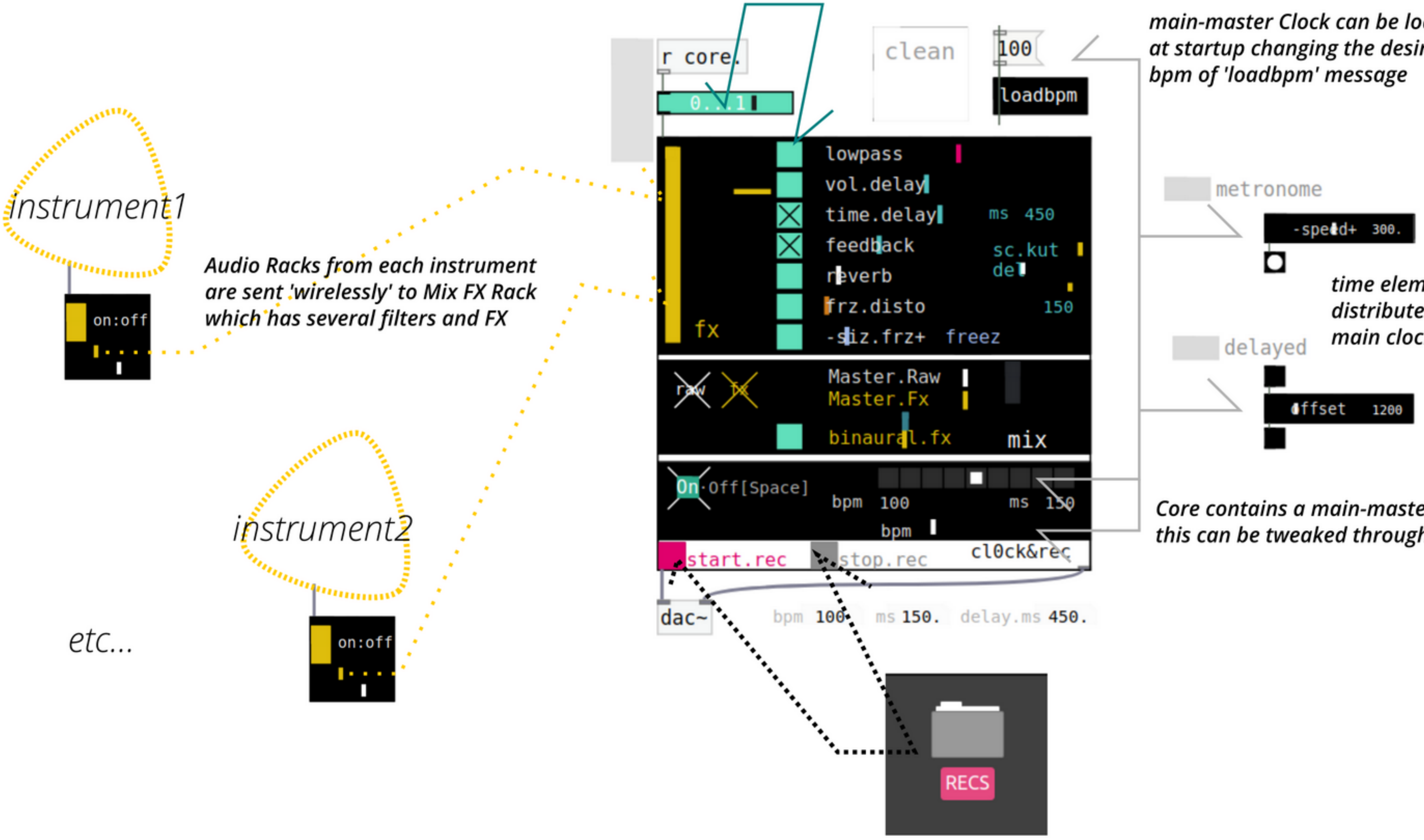
Grains // element which reprocess any sonic signal with granular synthesis technique.
Midi // objects to capture midi input messages / devices
OSC // different elements of listening and sending OSC messages.
And other objects which can be useful with another devices and platorms :
comport *(arduino & serial connectivity)*
hid *(devices like joysticks or tablets)*
shell *in order to execute external scripting outside pd.*

Notice that in this case, this menú is a 'minimal' set of those elements.

If you need more details , methods, and examples you can check the folders :

CODE.Interact.MIC
CODE.Interact.MIDI
CODE.Interact.OSC

*In addition **CORE.template** contains the CORE block :*



Gnrtv Slider can automat any of the toggled FX
switch with cross means active driven by the top left main slider

main-master Clock can be loaded
at startup changing the desired
bpm of 'loadbpm' message

Audio Racks from each instrument
are sent 'wirelessly' to Mix FX Rack
which has several filters and FX

time elements like speed and offset
distributes quantized times related to the
main clock

Core contains a main-master Clock
this can be tweaked through the bpm slider.

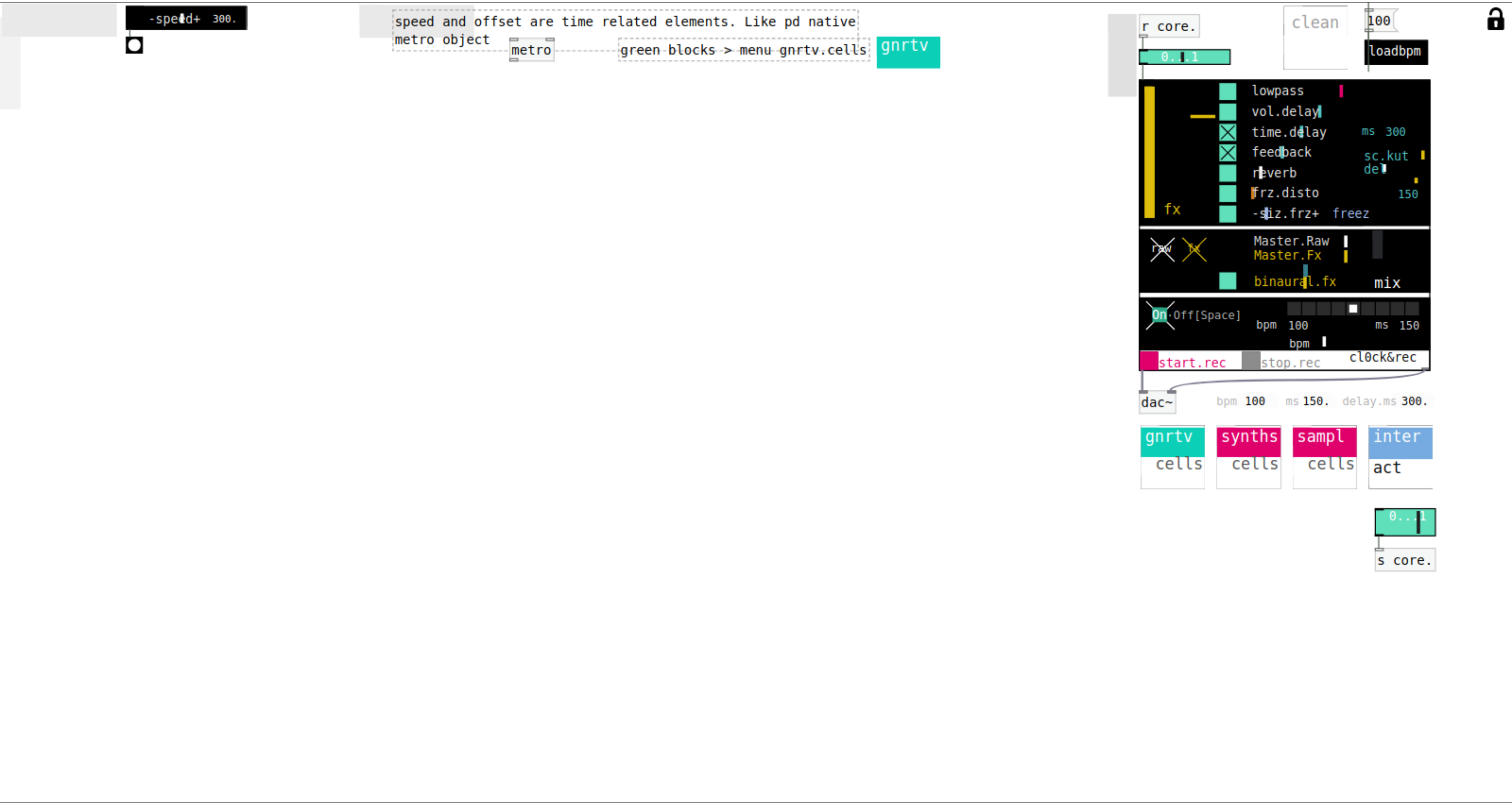
Any live session performed with cells can be recorded :.
Recordings are stored directly at RECS folder as a temporary name in .wav.

***8**_The previous pages are related to the **CORE.template** elements, but how to start building generative instruments?*

In order to build generative sonic algorythms usually can be folllowing the next the structure :

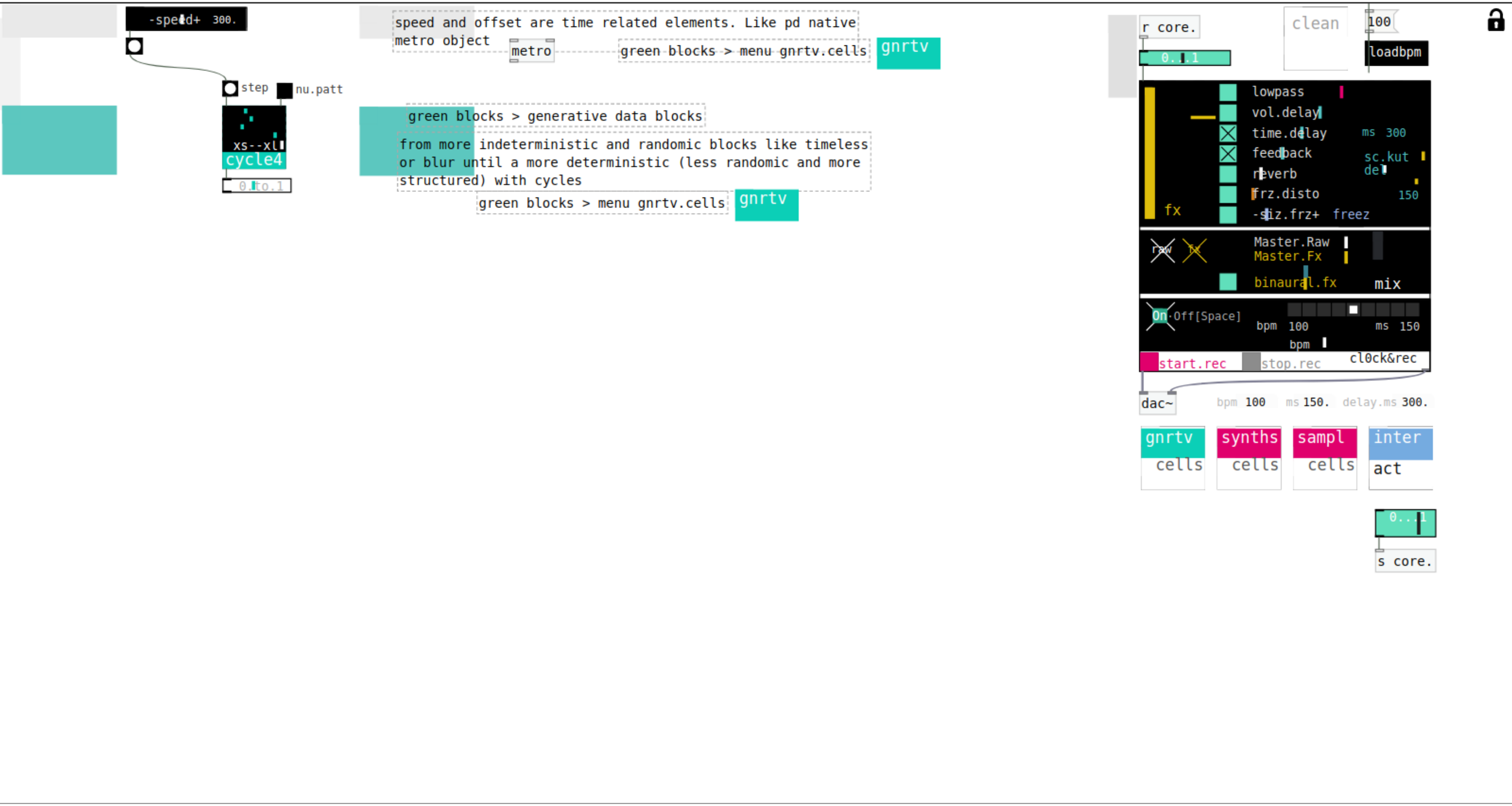
time
|
sequences / gnrtv elements
|
sonic instruments (oscillators, noise, synths, samplers etc)
|
filtering (optionally)
|
amplitude control (volume)
|
Signal Rack (Sends to dac~ and Core's FX Rack)

For example....



speed and offset are time related elements. Like pd native metro object

green blocks > menu gnrtv.cells



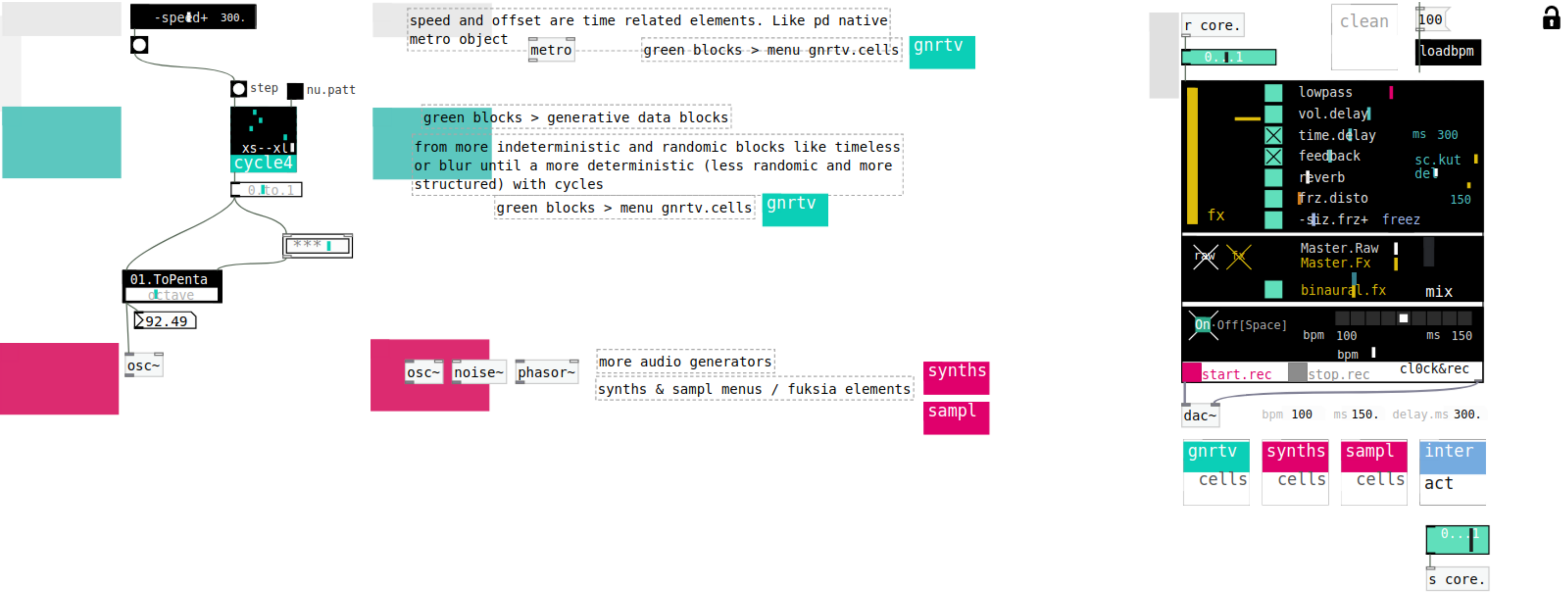
speed and offset are time related elements. Like pd native metro object

green blocks > menu gnrtv.cells

green blocks > generative data blocks

from more indeterministic and randomic blocks like timeless or blur until a more deterministic (less randomic and more structured) with cycles

green blocks > menu gnrtv.cells



speed and offset are time related elements. Like pd native metro object

green blocks > menu gnrtv.cells

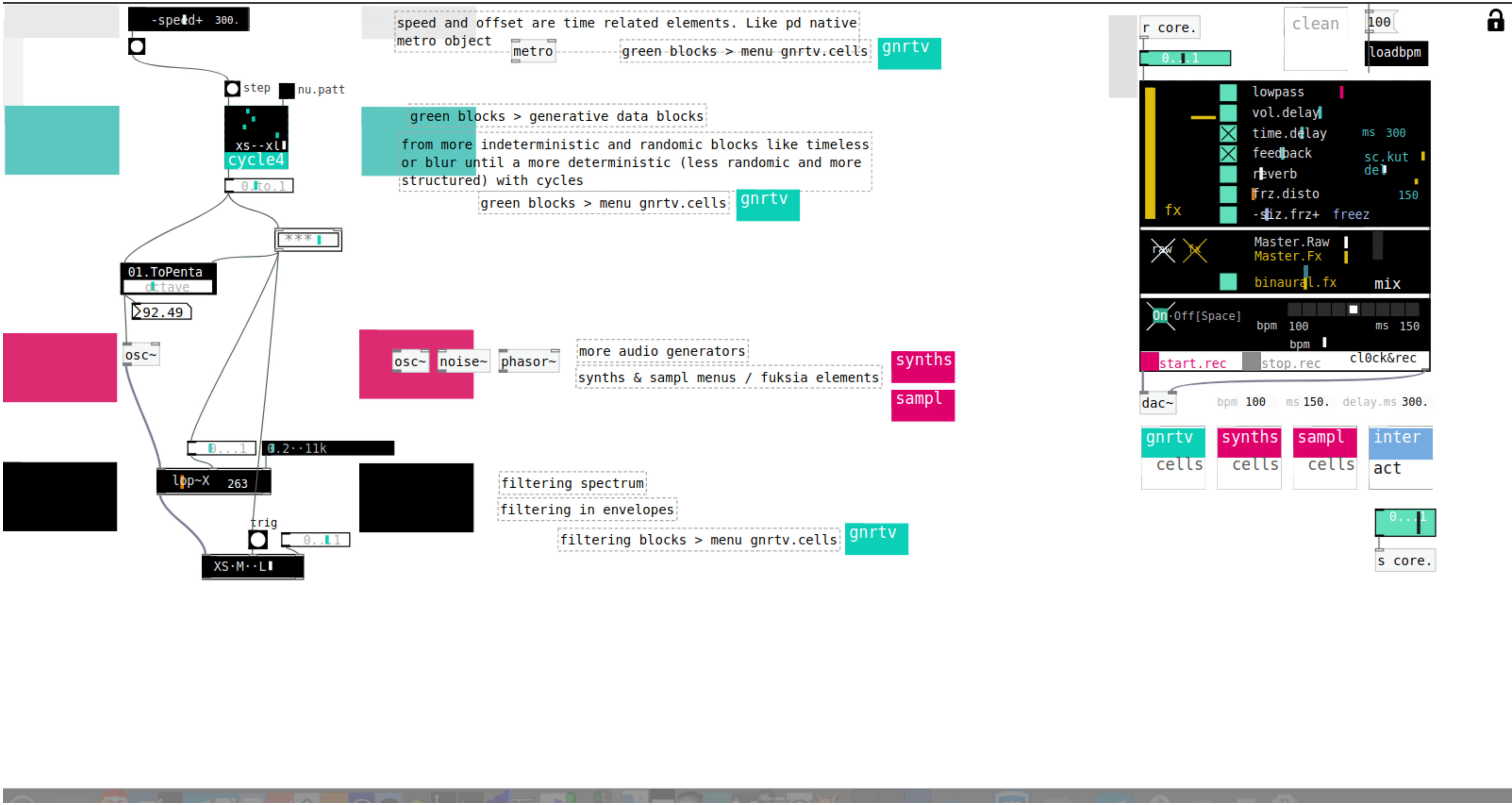
green blocks > generative data blocks

from more indeterministic and randomic blocks like timeless or blur until a more deterministic (less randomic and more structured) with cycles

green blocks > menu gnrtv.cells

more audio generators

synths & sampl menus / fuksia elements



speed and offset are time related elements. Like pd native metro object

metro

green blocks > menu gnrtv.cells

green blocks > generative data blocks

from more indeterministic and randomic blocks like timeless or blur until a more deterministic (less randomic and more structured) with cycles

green blocks > menu gnrtv.cells

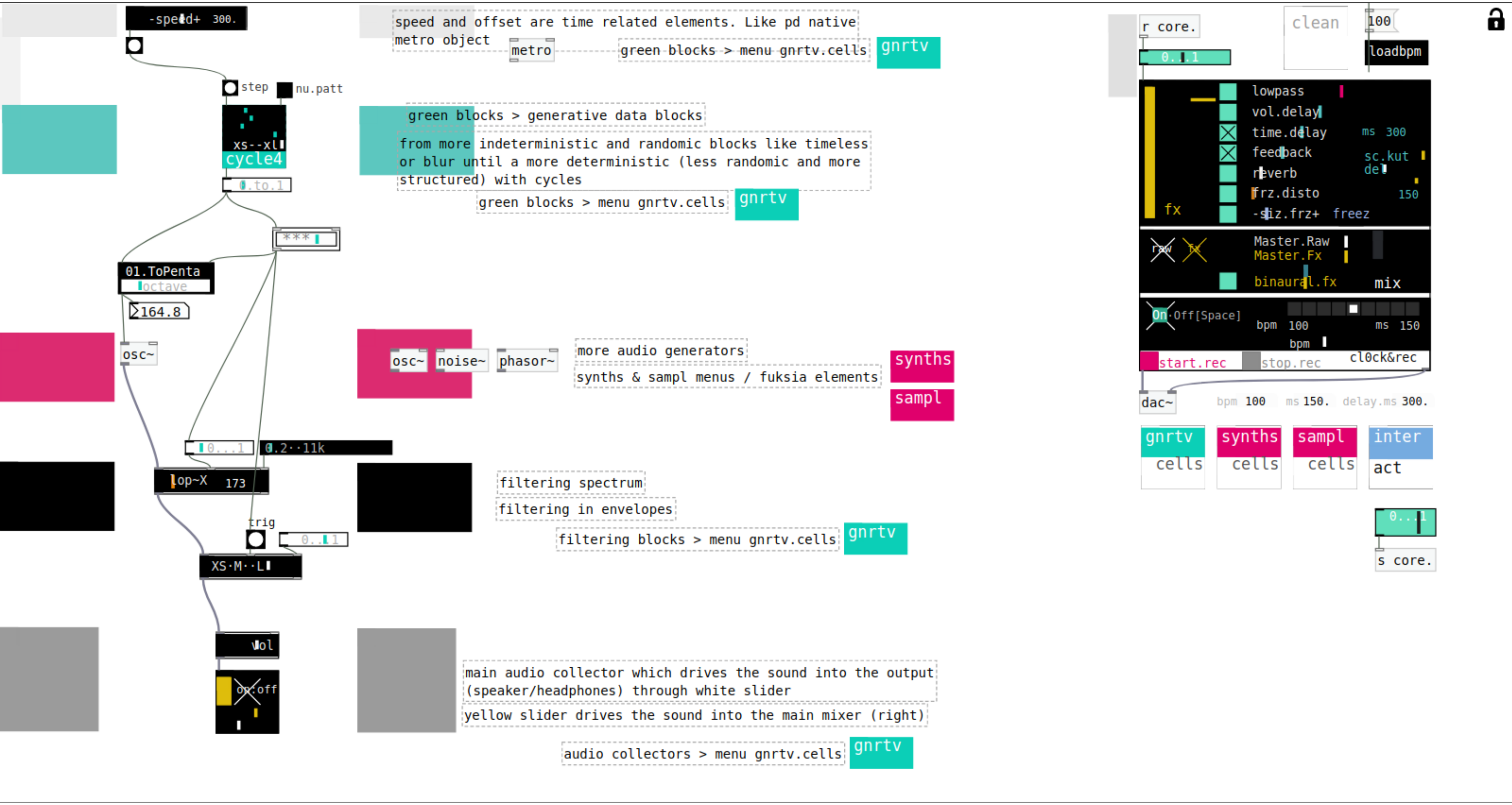
more audio generators

synths & sampl menus / fuksia elements

filtering spectrum

filtering in envelopes

filtering blocks > menu gnrtv.cells



speed and offset are time related elements. Like pd native metro object

green blocks > menu gnrtv.cells

green blocks > generative data blocks

from more indeterministic and randomic blocks like timeless or blur until a more deterministic (less randomic and more structured) with cycles

green blocks > menu gnrtv.cells

more audio generators

synths & sampl menus / fuksia elements

filtering spectrum

filtering in envelopes

filtering blocks > menu gnrtv.cells

main audio collector which drives the sound into the output (speaker/headphones) through white slider

yellow slider drives the sound into the main mixer (right)

audio collectors > menu gnrtv.cells

Check it out this example in CODE folder > TUTOS > gnrtv.cells__tutorial2_cells.structure.pd

N·Joy Generative Algoryhtms ^___^)))

gnrtv.cells by Xa.Manzanares 2024
+info github & IG // ***@xamanza***