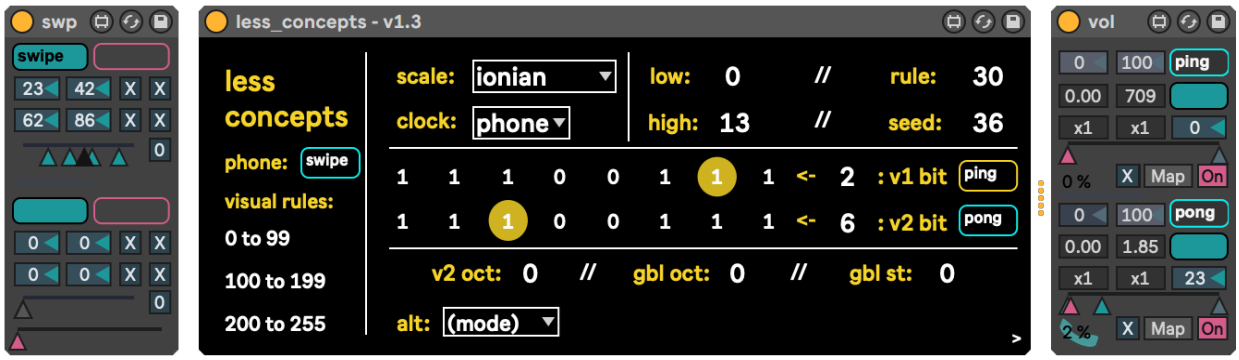
less concepts

max for live



v1.3

updated: 200916

less concepts is a generative sequencer that uses elementary Cellular Automata (CA) to scan sets of notes and determine which ones to play + when. It rewards simple choices with complex results.

requirements:

- Ableton Live Suite (10.1.9+)
- Max for Live (8.1.3+)
- optional: smartphones
- Works on Mac and Windows

This guide will be updated as folks ask questions, challenge my assumptions, and share their work.

Table of contents:

Have questions? Want to share what you've made with 'less concepts'? Visit the discussion thread on lines.

The core functionality of 'less concepts' is adapted from Ezra Buchla's binary-centric implementation of CA in monome's Teletype eurorack module.

To learn more about elementary cellular automata (also, image sources):
http://mathworld.wolfram.com/ElementaryCellularAutomaton.html
https://en.m.wikipedia.org/wiki/Elementary_cellular_automaton

'less concepts' is also an album: https://dndrks.bandcamp.com/album/less-concepts
Album artwork by Racquel Cable (IG: @biscotti.dippin)

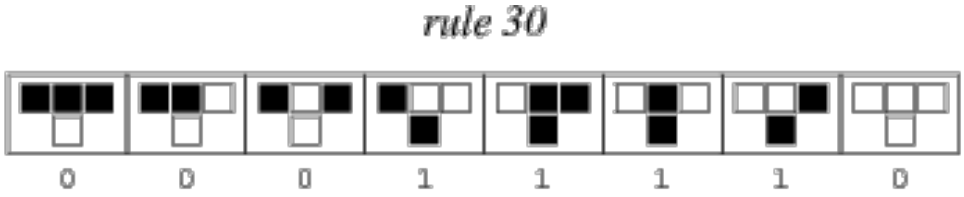
seek. think. discover.

Cellular Automata are collections of patterns that evolve by comparing an initial value to a fixed set of combinations, and assigning an output based on a given rule. They are most clearly represented visually:

The best way to read that image is to look at each set of three squares like this:

Left Neighbor | Current | Right Neighbor

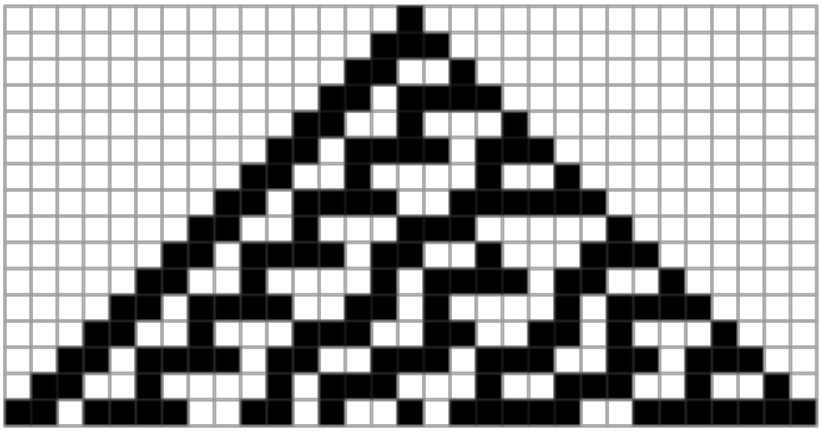
Rules come into play as the *output* of these combinations:



The first trio can be read as: "When a black square's Left Neighbor and Right Neighbor are also black squares, then it should output a white square." In elementary Cellular Automata, black = 1 and white = 0.

This example is known as rule 30 because the 8-bit binary **0 0 0 1 1 1 1 0** is equal to **30.**

This is what rule 30 looks like after multiple generations:



If we start at the top-middle and iterate the rule, we see how this pattern was made:

W | B | W
B

Similarly, less concepts generates values by comparing trios of 1's and 0's against a fixed set of combinations:

[111] [110] [101] [100] [011] [010] [001] [000]

When you enter a rule it is converted into 1's and 0's (8-bit binary integers):

rule 30 = 0 0 0 1 1 1 1 0

And this becomes the output of each combination:

[111]	[110]	[101]	[100]	[011]	[010]	[001]	[000]
0	0	0	1	1	1	1	0

When you enter an initial seed, it's also converted into 8-bit binary integers:

seed 36 = 0 0 1 0 0 1 0 0

And these binary bits are then grouped into trios (with the edges wrapped):

group 1 (0 0 1 0 0 1 0 0) = [000]
group 2 (0 0 1 0 0 1 0 0) = [001]
group 3 (0 0 1 0 0 1 0 0) = [010]
group 4 (0 0 1 0 0 1 0 0) = [100]
group 5 (0 0 1 0 0 1 0 0) = [001]
group 6 (0 0 1 0 0 1 0 0) = [010]
group 7 (0 0 1 0 0 1 0 0) = [100]
group 8 (0 0 1 0 0 1 0 0) = [000]

less concepts compares these trios against the fixed combinations and builds a new number out of the output:

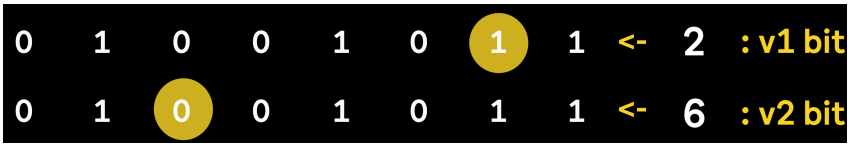
[111]	[110]	[101]	[100]	[011]	[010]	[001]	[000]
0	0	0	1	1	1	1	0

[000] -> 0
[001] -> 1
[010] -> 1
[100] -> 1
[001] -> 1
[010] -> 1
[100] -> 1
[000] -> 0

0 1 1 1 1 1 1 0 = 126

…and that's the next state, which will be re-seeded to be processed the same way. Sometimes, things will loop (if we re-seed the original seed). Sometimes, it'll terminate (if we re-seed a 0).

[bits = gates]



As less concepts iterates seeds through rules, it presents the 8-bit binary for each step. These 1's and 0's make perfect rhythm generators, so let's use them as gates!

To get started:
· choose rule 30
· choose seed 36
· start Live's transport

The 'bit' section will come alive with 1's and 0's.

Use the number boxes to the left of `v1 bit` and `v2 bit` to select a gating bit. Whenever a `1` appears, a note will occur. `0`s are rests. If you choose 'bit 9', then notes will occur more frequently (when any bit is 1).

As you play, you'll find that some rules start off as very active and then terminate to silence (eg. rule 60). This is expected and intentional -- simply re-seed the rule and you'll get another burst of activity (great for flourishes). less concepts is full of parameters that can be tweaked and modulated to create complex sequences out of very simple decisions.

Using the *visual rules* popups, you should be able to predict what the activity of a particular rule might be. To help guide your guesses, here's a bit more about the four classes of rules:
· Class 1: rapidly converge to stasis (eg. rule 32)
· Class 2: rapidly converge to repetition (eg. rule 108)
· Class 3: longer-running + repeating (eg. rule 182)
· Class 4: complex structure that has varying convergences (eg. rule 110)

[scales, low and high]

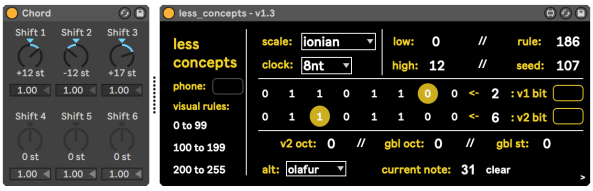
Use the `scale` dropdown to select a scale. Scales have 28 values (which equals 4 octaves for the diatonic scales).

`low` and `high` fence the values you can generate notes from – you can think of them as the range of scale degrees. As such, 0 to 13 is a full 2 octaves (for diatonic scales) and a nice place to start.

Toward the bottom of the device, there are some additional adjustments that can be made. You can offset the octave of the second bit stream (v2 oct), offset the octave of both streams at the same time (gbl oct), or offset both streams by semitones (gbl st). *nb. the base note, without offsets, is MIDI 48 (C3)*

alt modes:

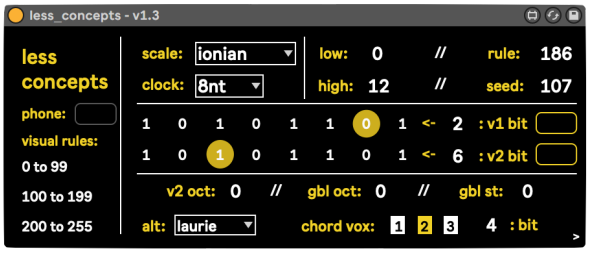
[olafur]



Inspired by the self-playing pianos of Olafur Arnalds, [olafur] mode will bypass the chosen scale and will build note pools from direct keyboard input. Just arm the track less concepts is on and play a chord – you'll see `high` map to the number of keys you're holding down.

Personally, I find that placing Live's *Chord* device ahead of less concepts really helps round out the shape of a keyed-chord.

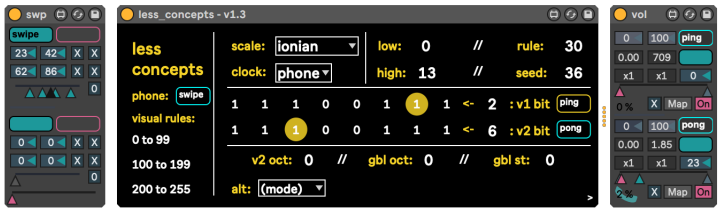
[laurie]



Inspired by Laurie Spiegel's Music Mouse software, [laurie] mode will add chord voicing to your bit streams. Choose a gating bit and enable '1' '2' or '3' for unique harmonies. *nb. [laurie] mode only functions with the diatonic scales.*

- '1' is the root
- '2' adds a fifth above (will switch between major/minor/aug depending on scale)
- '3' adds a ninth above (will switch between major/minor/aug depending on scale)

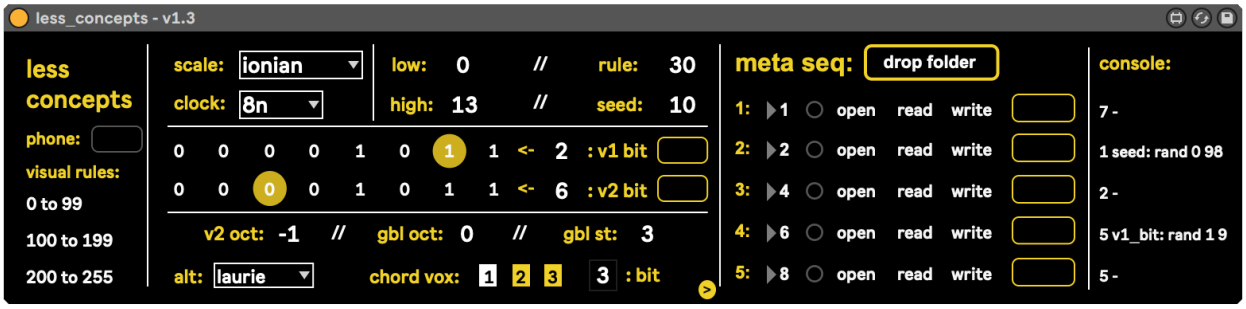
phone:



Inspired by jr's smartphones, which create a network of responsive triggers using text boxes and physical gesture, you can break less concepts away from Live's clock.

- select [phone] from the clock menu
- enter the same text identifier in the textbox on the far-left of the device and in one of the smartphone transmitters
- less concepts will *only* advance the rule/seed generation when it receives a trigger from one of the smartphones
- enter additional identifiers next to `v1` and `v2` to send pulses on active bits

Click the `>` in the bottom right corner to reveal the meta sequencers.



There are 5 of them, each with their own beat divisor (the rate of execution is indicated by the blinking pulse). These divisors are based on the `clock` rate, so an `8n` clock rate and a `1` meta seq divisor will execute on every 8th note. Similarly, a `4n` clock rate and a `3` meta seq divisor will execute on every 3rd quarter note.

Use the sequencers to script clocked changes and modification to less concepts.

To try it out:
- 'open' a text window
- enter the following on line 1: **seed: 30**
- enter the following on line 5: **seed: 193**
- use your ENTER key to add blank lines, all the way down to line 10
- close the text window to commit the changes (*nb. each meta sequencer will not execute new commands until its individual text editor is closed*)

With Live's transport running, the `console` will show which line is being executed in each sequencer. You'll notice they're like trackers, executing each line in order and looping when they reaches the end of the file.

Here is what you can control (and how):

rule: 30	**seed: 36**	**v1_bit: 7**	**v2_bit: 2**
v2_oct: -1	**gbl_oct: 1**	**gbl_st: -7**	**low: 0**
high: 6	**chord_bit: 1**	**chord_2: 1**	**chord_oct_2: 0**
clock: 4nt	**scale: aeolian**	**seq_1: 4**	

nb. chord_[x] commands toggle chord voices on/off with 1/0.

There is also a `rand` operator, to help inject some unpredictability, eg:

seed: rand 62 94 *<~~~ chooses a seed between 62 and 94, inclusive*
v2_bit: rand 3 7 *<~~~ chooses a bit for v2 between 3 and 7, inclusive*

`rand` is a particularly useful technique when working with terminating / static rules.

nb: `rand` supports all of the above parameters except clock and scale.

Each meta sequencer can also send a smartphones pulse at every line read, which can create a clock-divided network of signals.

[vertical + horizontal composition]

The meta sequencers execute both vertically and horizontally, allowing for multiple commands to be performed at once in a single line (so long as they are separated by commas). eg:

v1_bit: rand 1 6, seed: 30, clock: 8nt *<~~~ executes all three commands at once*

To sync + manipulate the sequencers' read-heads, utilize the **seq_x** command, replacing **x** with the sequencer you wish to affect. eg:

seq_3: 2 *<~~~ sets sequencer 3 to line 2 on its next tick*
seq_1: 1, seq_2: 1, seq_3: 1 *<~~~ sets sequencers 1,2,3 to line 1 on their next tick*

[**saving + re-loading sequences]**

meta sequences can be saved independent of the Live Set:
· press 'write' in the `meta seq`
· a dialog box will pop up
· save your sequence as a text file

If you want to save/recall your sequences with your Live Set:
· save your Live Set normally
· press 'write' in the `meta seq`
· create a folder inside your Live Set (name it whatever you want)
· save your sequences as X.TXT (where X is the sequencer number)
· IMPORTANT: after you save your sequences for the first time, drag the folder you saved them all in back the 'drop folder' area. this creates a symbolic link between the folder and less concepts, so less concepts knows where to look for sequences when you load your Live Set.
· IMPORTANT: save your Live Set again
· *nb: the recall function will restore based on position in the folder -- so if there are three files (regardless of their names), they will be restored to the first three sequencers*

If you want to load sequences from other sessions:
· press 'read` in the `meta seq`
· a dialog box will open
· choose your file
· unless you perform the steps up above to save/recall WITH your Live Set, this file association will not save.