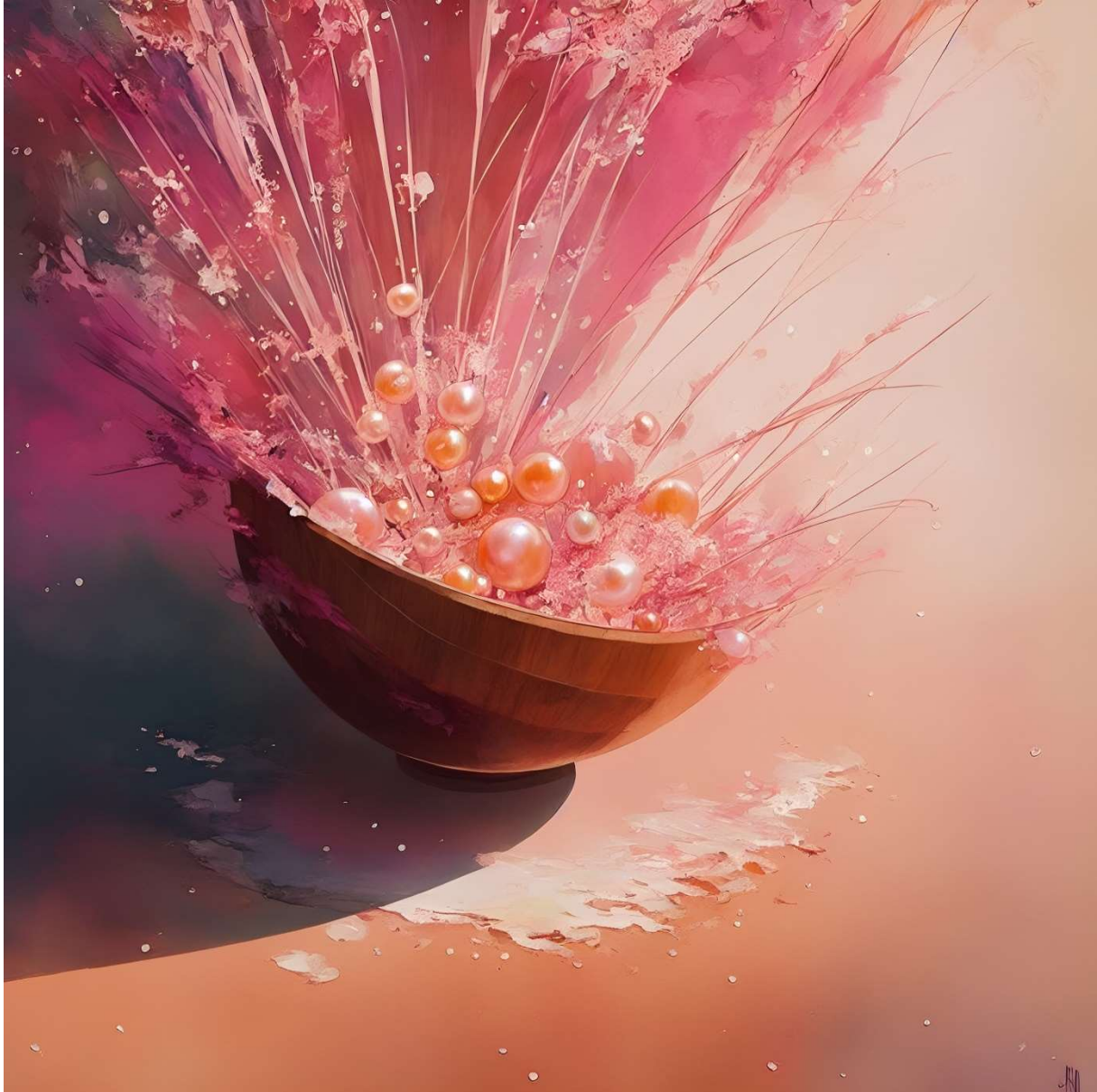


# PEARLS

Euclidean Collision Sequencer



Based on Euclidean Rhythms

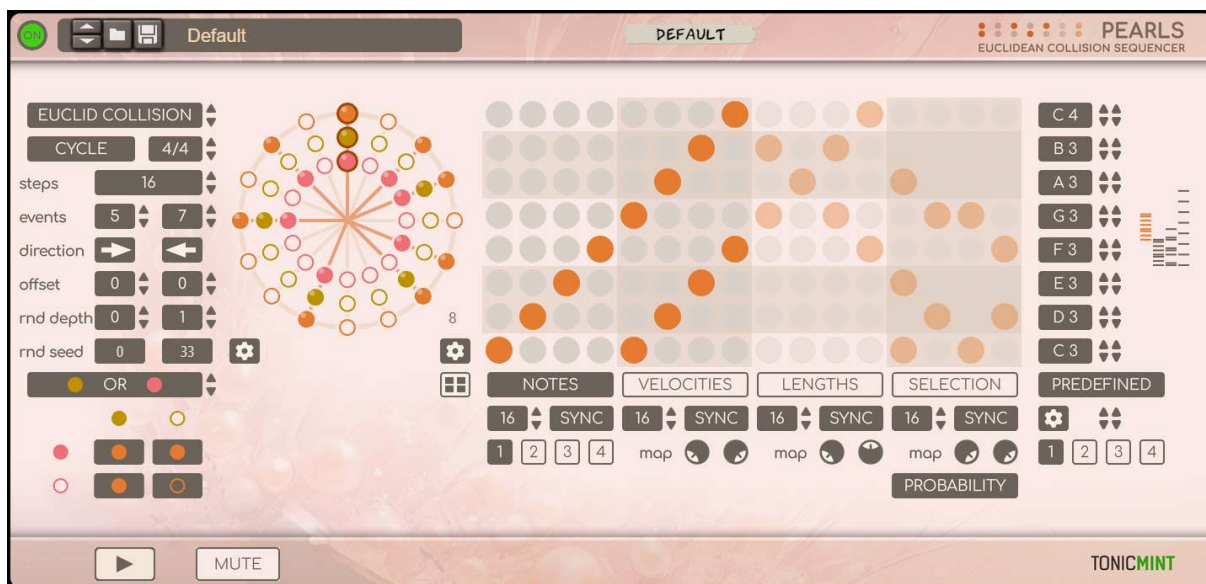
Independent sequences for note features

Sequencer and Arpeggiator

# CONTENTS

Intro.....	3
Architecture of Pearls.....	4
Rhythm generator .....	5
Type #1: Pulses .....	5
Type #2: Euclidean.....	6
Type #3: Altered Euclid .....	7
Type #4: Euclid Collision.....	8
Sequencers.....	9
Note sequences.....	9
Velocity sequence.....	10
Length sequence.....	11
Selection sequence.....	12
Note mapping .....	13
Predefined note mapping.....	13
Dynamic note mapping.....	13
UI of Pearls.....	14
Controls .....	14
Dark backgrounded controls with up and down arrows.....	14
Dark backgrounded controls with double up and down arrows.....	14
Dark backgrounded controls without arrows.....	15
Radio buttons.....	15
Gear menu buttons.....	15
NoteMap displays.....	15
Front panel.....	16
Header Section.....	16
Rhythm Section.....	16
Sequencer Section.....	23
NoteMap Section.....	26
Footer Section.....	27
Back panel .....	28
CV Inputs .....	28
CV Outputs.....	28
Tips and Tricks .....	29
n tuplets.....	29
Different rhythms for different drums/percussions with one instrument .....	29
Evolving Rhythms with probability .....	29
Evolving Rhythms with limits .....	30
Evolving Rhythms with the rhythm generator .....	30
Evolving Music.....	30
Arpeggiated chord progression.....	31

# INTRO



Pearls is a player type Rack Extension for Reason that incorporates a Euclidean rhythm generator, sequencer, and note mapping functionality.

The rhythm generator utilizes Euclidean rhythms and offers options for random exchanges between empty steps and steps with events. Additionally, it allows for the interaction of two Euclidean rhythms.

The sequencer operates independently and provides unique features for notes, velocities, note length, and note selection.

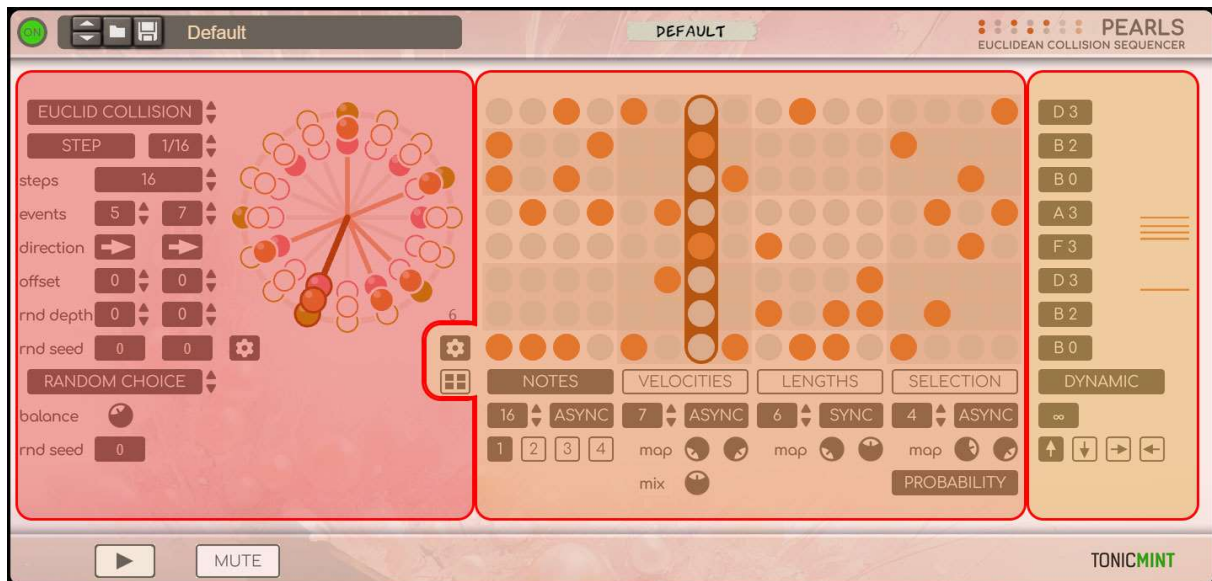
The note mapping can be configured as a general sequencer mode or an arpeggio mode, enabling the assignment of musical notes to each lane of the sequencer.

Overall, Pearls offers a diverse range of creative options for generating and sequencing rhythms and notes.

# ARCHITECTURE OF PEARLS

Pearls has three main parts:

- Rhythm generator
- Sequencers
- Note mapping



In this chapter we will go through broadly on these main parts.

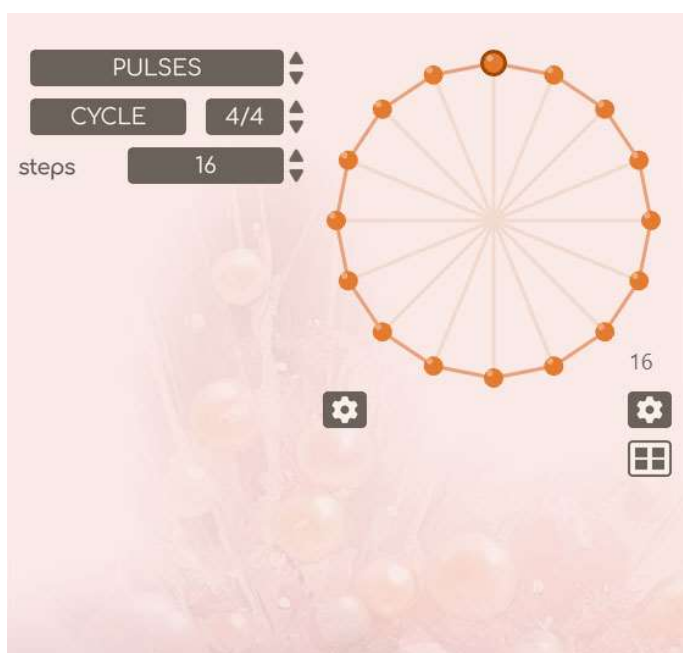
The detailed descriptions of the UI elements are in the UI of Pearls chapter.

# Rhythm generator

The rhythm generator consists of four types, ranging from simplest to most complex.

All of them can be cycle or step based.

## Type #1: Pulses

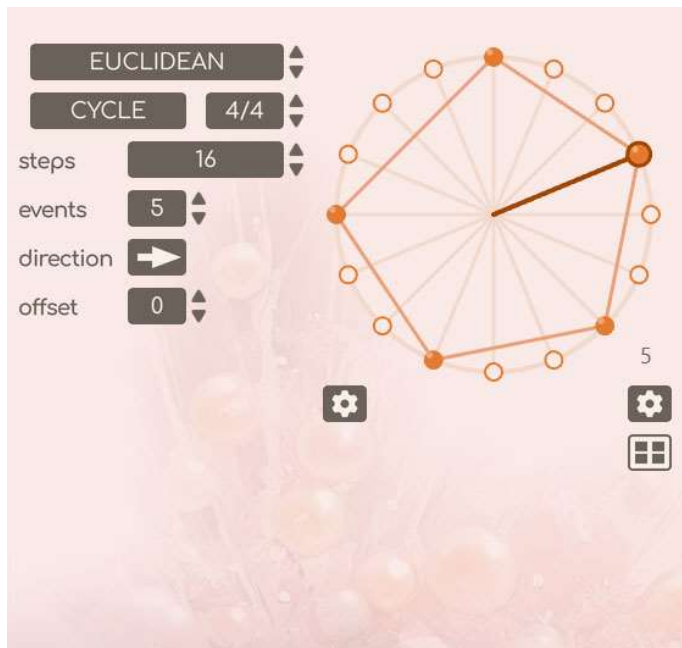


The rhythm can be defined by the length of steps or the length of the cycle, as well as the number of steps in a cycle. By stacking multiple Pearls units, interesting polyrhythms can be explored.

Parameters:

- Number of steps

## Type #2: Euclidean



The rhythm generator is based on Euclidean rhythms. The previous working mode is a special case when the steps and events are the same, but with different values, a general Euclidean rhythm is created. This allows for the creation of familiar yet unusual rhythms.

Parameters:

- Number of steps
- Number of events
- Direction
- Offset

## Type #3: Altered Euclid

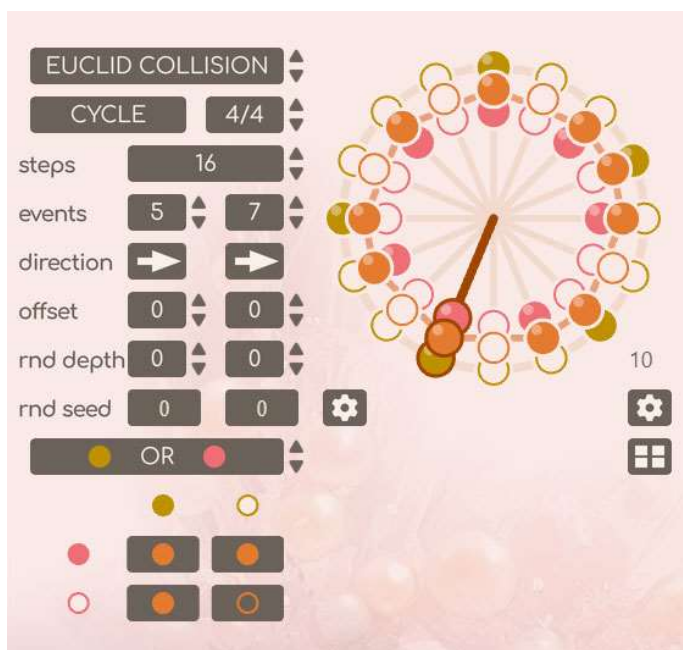


In addition to the usual parameters of Euclidean rhythms (number of steps, number of events, direction, offset), the random exchange of empty steps and steps containing events can be played with.

Parameters:

- Number of steps
- Number of events
- Direction
- Offset
- Number of random exchanges between empty steps and event steps
- Random seed for this exchange (per Euclidean rhythm)

## Type #4: Euclid Collision



The Pearls rhythm generator introduces the interaction of two Euclidean rhythms. This interaction can be a logical (bool) operation (AND, OR, XOR) or a weighted random choice between the two rhythms. This adds further creative possibilities.

Parameters:

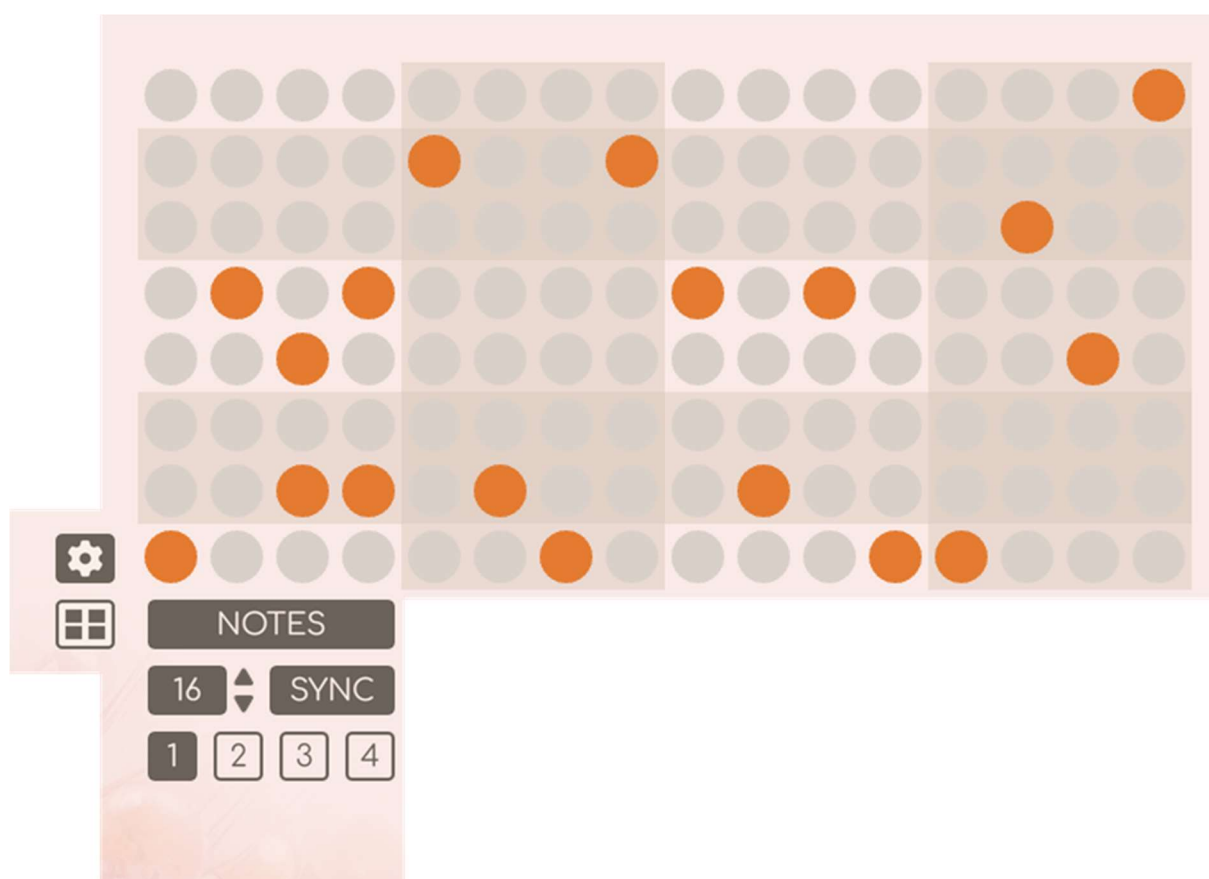
- Number of steps
- Number of events (per Euclidean rhythm)
- Direction (per Euclidean rhythm)
- Offset (per Euclidean rhythm)
- Number of random exchanges between empty steps and event steps (per Euclidean rhythm)
- Random seed for this exchange (per Euclidean rhythm)
- Interaction operation
- Random Choice Balance (if the operation is Random Choice)
- Random Choice Seed (if the operation is Random Choice)



# Sequencers

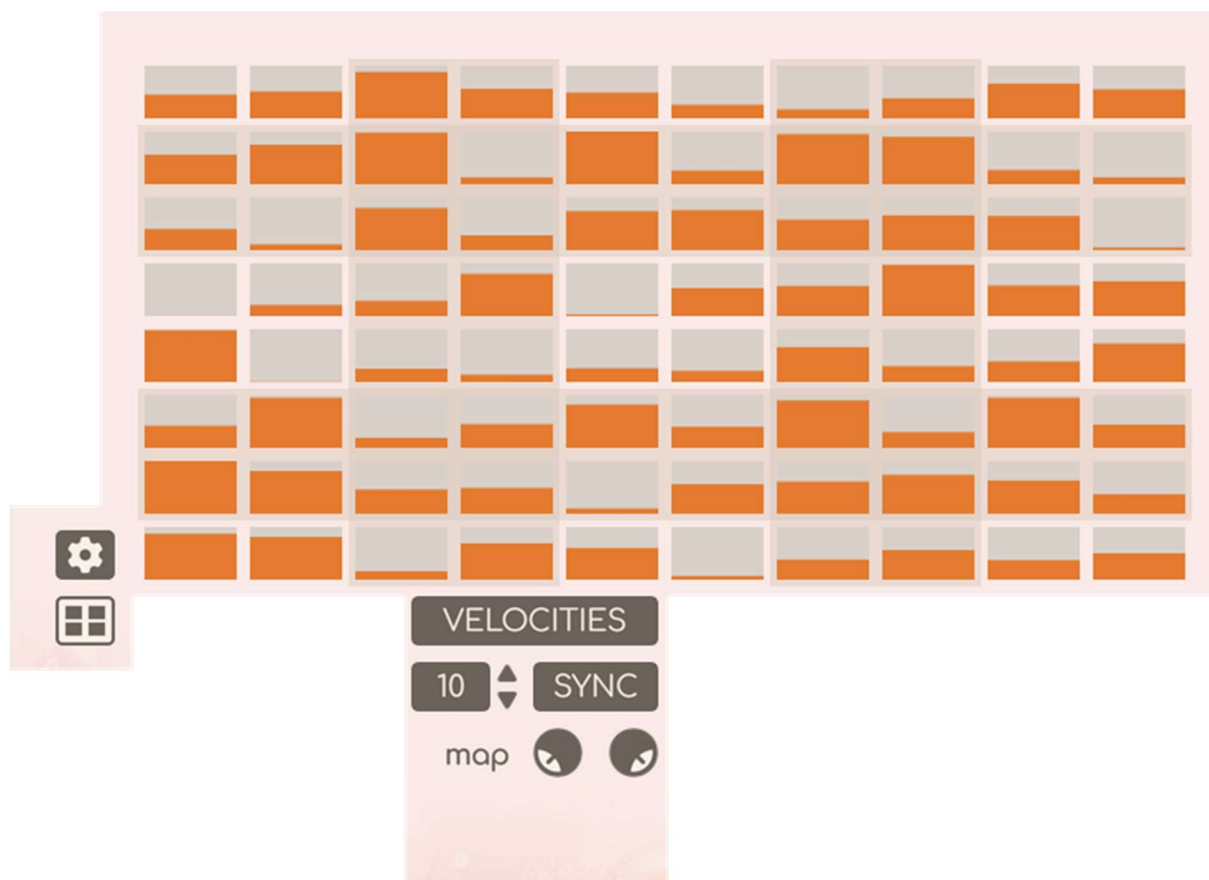
Pearls features independent sequencers for notes, velocities, note length, and note selection. The sequencers only advance on every event generated by the rhythm generator.

## Note sequences



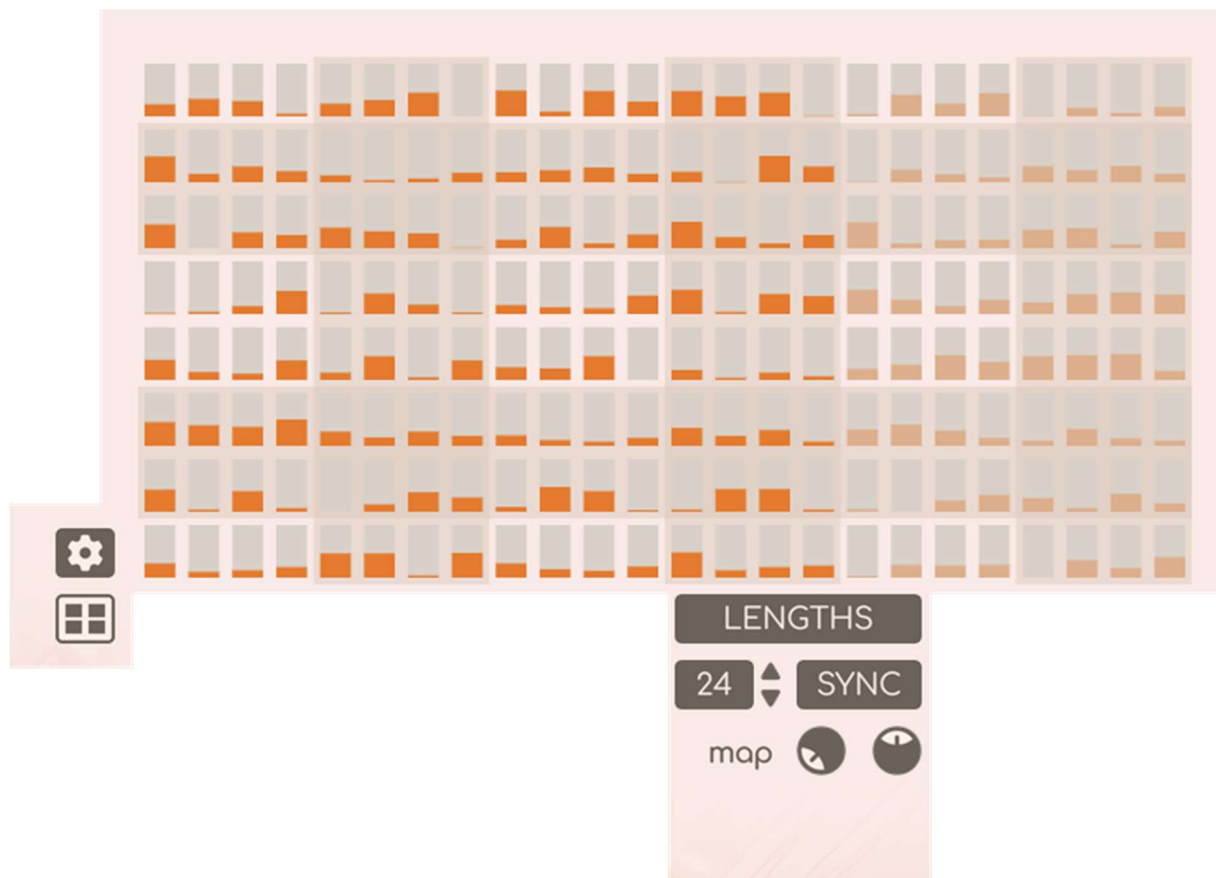
Four patterns can be defined for the notes, which can be switched between at any time during playback.

## Velocity sequence



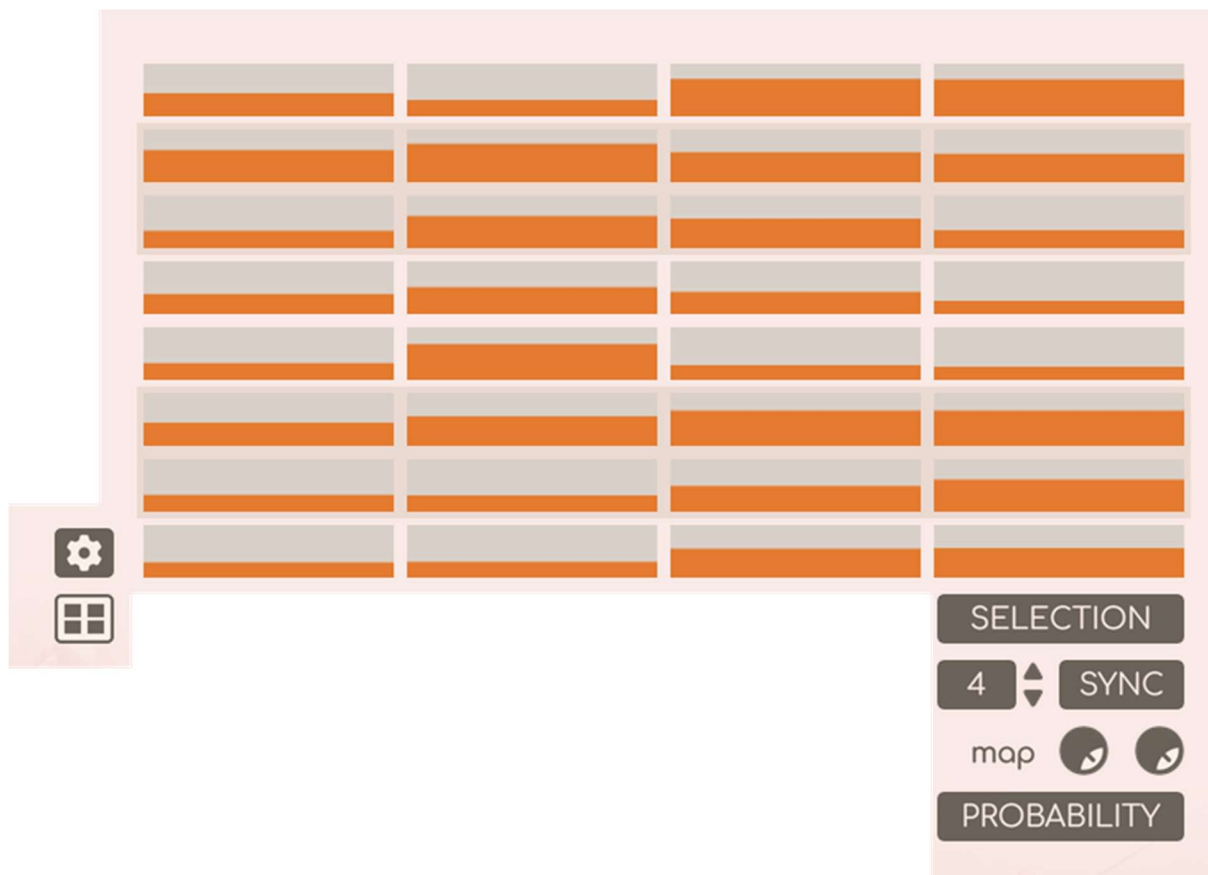
The values in the sequence range from 0 to 100. With the mapping knobs, these values can be interpolated, extrapolated, or projected into a desired velocity range. The target velocities can range from 1 to 127, matching standard MIDI velocity values. Velocity can also be adjusted during playback.

## Length sequence



The values in the sequence range from 0 to 100. Using the mapping knobs, these values can be interpolated, extrapolated, or projected into a desired length range. The target length can range from 6.25% to 1600% of the current step length. Length adjustments can also be made during playback.

## Selection sequence

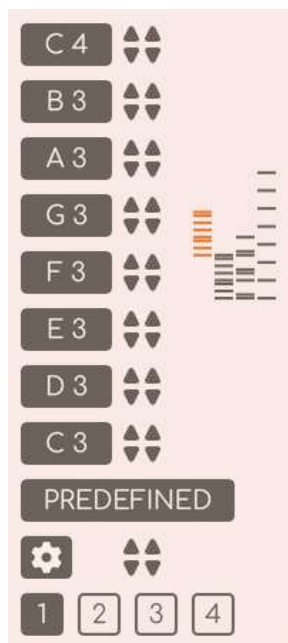


The values in the sequence range from 0 to 100. The selection can function as a probability. By using the mapping knobs, the sequence values can be interpolated, extrapolated, or projected into a desired probability range. The target probability can range from 0% to 100%, determining the chance/probability of playing the corresponding note (0% meaning the note never plays and 100% meaning the note always plays).

The selection sequence can also be used in limit mode, where lower and upper note limits are set, muting any notes outside the range.

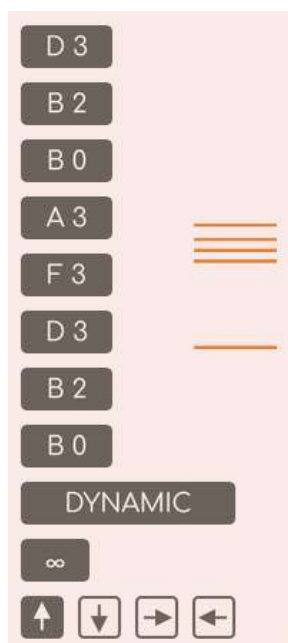
# Note mapping

## Predefined note mapping



This is a general sequencer mode where musical notes can be assigned to each lane of the sequencer. Four note mapping presets can be created and switched between at any time. In this mode, incoming MIDI notes pass through the device, allowing for stacking of multiple Pearls units.

## Dynamic note mapping



This is an arpeggio mode where incoming MIDI notes are utilized as the notes of the sequencer lanes. The notes can be assigned to the lanes based on four modes: oldest to newest, newest to oldest, lowest to highest, and highest to lowest. If the number of incoming notes is less than 8, recursion can be enabled to repeat the current pattern for the remaining lanes.

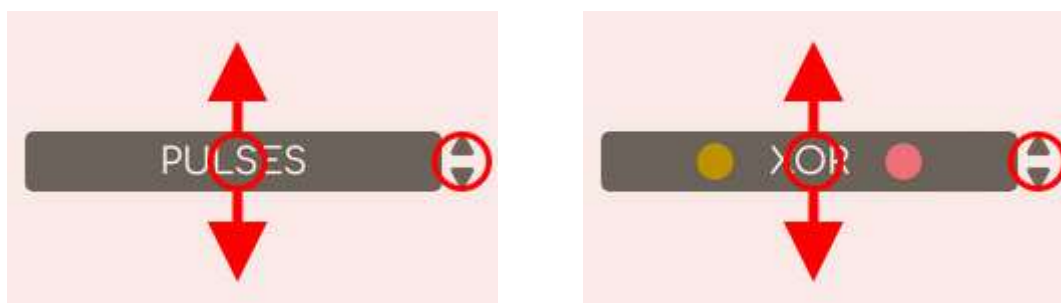
## UI OF PEARLS

More detailed description of the UI elements of Pearls.

### Controls

#### Dark backgrounded controls with up and down arrows

Use the arrow buttons for increasing/decreasing or switching the value or click and drag the big dark area to scroll through the values. The big dark areas working as a standard knob in Reason.



#### Dark backgrounded controls with double up and down arrows

Similar controls, but for note mappings you can step the values not just by 1, but by 12 (octave).



## Dark backgrounded controls without arrows

These controls are two stated buttons. Click on them to change the value.



## Radio buttons

The active element of the radio button group is displayed with dark background, and other values are displayed with light background and dark borders.



Sequencer selector buttons are also radio buttons. Click on one of them to set the displayed sequencer. The first in the list with four square is the overview display mode.



## Gear menu buttons

Buttons with gear icon are gear menus. They contain context sensitive functions for the main sections of Pearls.



## NoteMap displays



These controls don't receive any click events, they are just showing the current NoteMap settings.

In Predefined mode you can see all the presets, and the current preset is highlighted by orange.

# Front panel

## Header Section

- On/Off button
- Patch Browser

## Rhythm Section

The Rhythm Section is on the left side of the main body. Parameters on the left side of this are from top to bottom, and the right side of this section is the Rhythm display.

### *Rhythm type switch*

This has 4 options, from simply to more complex types:

- Pulses
- Euclidean
- Altered Euclid
- Euclid Collision

Use the small arrow buttons in the right side to step the value or click and hold down your mouse on the button and move up or down to change the value. It is working like a knob, and this is the behavior of most controls.

### *Step or cycle switch*

- To determine the base of the length of the rhythm.
- Click on the switch to change the value.

### *Step Length*

- If the previous switch is in Step mode
- values are: 1/2 ... 1/64T



## *Cycle Length*

- If the previous switch is in Cycle mode
- values are: 2/4 ... 64/4

## *Steps*

- number of steps

## *Events*

- number of events
- visible when we have Euclidean type or more complex.

## *Direction*

- you can flip or mirror the Euclidean rhythm.
- visible when we have Euclidean type or more complex.

## *Offset*

- visible when we have Euclidean type or more complex

## *RND depth*

- you can swap/flip/exchange two steps in a rhythm, where one step is empty, and another has event.
- The steps for this operation are selected randomly.
- The value of RND depth says, how many exchange will be applied.
- values are from 0 to 32
- 0 means: the rhythm not changed; no exchange applied.
- the used random numbers are reproducible random numbers.
- visible when we have Altered Euclid type or more complex.

### *RND seed*

- seed value for the previously mentioned operation
- same seed: same random values, different seed: different random values
- values are from 0 to 10000
- visible when we have Altered Euclid type or more complex

### *Events B*

- for the secondary Euclidean rhythm
- same as the previously mentioned Events
- when we have Euclid Collision type then Events param become Events A
- visible when we have Euclid Collision type

### *Direction B*

- for the secondary Euclidean rhythm
- same as the previously mentioned Direction
- when we have Euclid Collision type then Direction param become Direction A
- visible when we have Euclid Collision type

### *Offset B*

- for the secondary Euclidean rhythm
- same as the previously mentioned Offset
- when we have Euclid Collision type then Offset param become Offset A
- visible when we have Euclid Collision type

### *RND depth B*

- for the secondary Euclidean rhythm
- same as the previously mentioned RND depth
- when we have Euclid Collision type then RND depth param become RND depth A
- visible when we have Euclid Collision type

### *RND seed B*

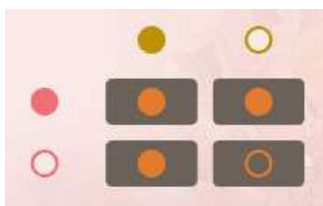
- for the secondary Euclidean rhythm
- same as the previously mentioned RND seed
- when we have Euclid Collision type then RND seed param become RND seed A
- visible when we have Euclid Collision type

## *Operation*

- visible when we have Euclid Collision type.
- when we are in Euclid Collision type, the output of the rhythm generator is the result of an operation between two Euclidean rhythms. With this parameter you can define the operation
- you can switch between 17 operation type.
- we have 17 operation types. 16 Boolean operation and 1 random choice operation
- list operations:
  - NONE
  - A AND B
  - NOT A AND B
  - B
  - A AND NOT B
  - A
  - A XOR B
  - A OR B
  - NOT A AND NOT B
  - NOT A XOR B
  - NOT A
  - NOT A OR B
  - NOT B
  - A OR NOT B
  - NOT A OR NOT B
  - ALL
  - Random choice

## *Operation Boolean editor*

- visible when we have Euclid Collision type and Operation value is in the first 16 operation (the Boolean operations)
- you can switch between the first 16 operation type (the Boolean operations)
- this is a visual editor so easier to understand.



## *Random Choice Balance*

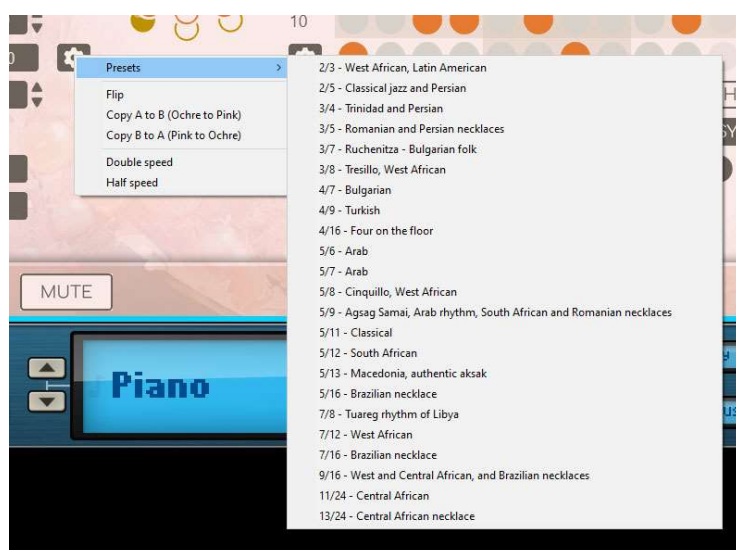
- visible when we have Euclid Collision type and Operation value is "Random choice"
- in "Random choice" mode the output of the rhythm generator is chosen by rolling dice and the chance of A or B rhythm is balanced by this value.
- the used series of random numbers is a reproducible random series (dice not rolled step by step or cycle by cycle.)

## *Random Choice Seed*

- visible when we have Euclid Collision type and Operation value is "Random choice"
- seed value for the random values for the Random Choice functionality
- same seed: same random values, different seed: different random values
- values are from 0 to 10000

## *Rhythm generator Gear Menu*

- actions for Rhythm generator parameters
- presets, half speed, double speed, copy, etc...
- Presets available when we are using a Euclidean rhythm type, when we are using Euclid Collision, then the first (A) (Ochre) rhythm can be set by the preset action
- Flip and copy available when we are using Euclid Collision. this action flips or copy the values between the (A) (Ochre) and (B) (Pink) rhythm.
- context sensitive menu, the actions are based on the current rhythm type.



## *Rhythm display*

- displays/visualize the rhythm
- dots in a circle
- filled dots: steps with event
- empty dots: empty steps
- orange is always the color of the output rhythm
- when we have Euclid Collision type, we see three concentric rings for the A, B and the result rhythm
- ochre is the A rhythm, pink is the B rhythm
- click on the Rhythm display to the visualization options
- you can turn on/off the followings:
  - Ring
  - indicator spoke
  - step spokes
  - step shape
  - event spokes
  - event shape
  - overlapped dots (for Euclid Collision type) (otherwise discrete dots)
  - specular highlight (just a visual fun)



## Sequencer Section

The sequencer section is in the middle part of the main body. Sequencer editors are in the top part of this area and additional controls for the four sequencers are below this in four group.

Pearls have 4 sequencers, but they are working together. They are individual sequencers for notes, velocities, note lengths and selection (which is some kind of filtering of the outputs)

Each sequencer has its own lengths and other parameters.

The sequencers have 8 lanes. The sequencers produce note on/off information and velocities together for these 8 lanes.

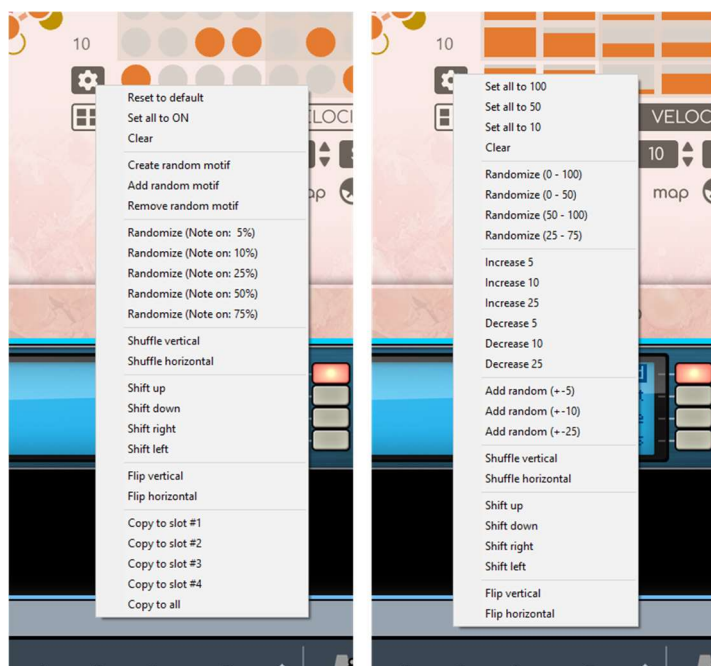
### *Sequencer editor*

- The display and behavior of this editor depends on what is the display mode of this control.
- The current step of a sequencer indicated by a dark red background
- Display modes:
  - Notes
  - Velocities
  - Lengths
  - Selections
  - Overview
- Display modes - Notes
  - You can select/unselect notes by mouse
- Display modes - Velocities, Lengths, Selections
  - You can set values between 0 and 100 for each individual cell (steps and lanes) by mouse
  - With shift key you can finetune a value by mouse
  - With control/command key you can edit all lane value together by mouse
- Display modes - Overview
  - You can see all sequencers at once. You can't edit here, but you can see how sequencers look like, how they are working together.

- 
- Sequencer editor display mode selector buttons
    - Below the Sequencer editor you can find 4 buttons: Notes, Velocities, Lengths, Selections
    - Left from these but in the same row a smaller button with four rectangles is the Overview display mode
  - Sequence Length controls
    - Below the Notes Velocities, Lengths, Selections display mode selector buttons
    - Set the length of the related sequencer
  - Sequence Sync/Async switch
    - Below the Notes Velocities, Lengths, Selections display mode selector buttons, right from the related Sequence Length control
    - Toggle between the Sync and Async mode of the related sequencer
  - Note Sequence Pattern selector
    - Below the Sequence Length and Sync/Async switch controls of the Note Sequence
    - 4 radio button in a row to switch between 4 patterns
  - Velocity map controls
    - Below the Sequence Length and Sync/Async switch controls of the Velocity Sequence
    - 2 knobs to set the from value and to value for the sequence mapping
    - From and to values can be set between 1 and 127, just like a standard MIDI velocity range
    - The current sequence value will be projected from the 0..100 interval to the given from to value range
  - Velocity mix
    - Visible when the NoteMap is in the Dynamic (Arpeggio) mode
    - You can mix the velocity of the incoming velocity and the sequencer produced velocity
  - Length map controls
    - Below the Sequence Length and Sync/Async switch controls of the Length Sequence
    - 2 knobs to set the from value and to value for the sequence mapping
    - From and to values can be set between 6.25% and 1600%, in the percent of the current step length of the rhythm generator
    - The current sequence value will be projected from the 0..100 interval to the given from to value range
  - Probability map/Limits controls
-



- Below the Sequence Length and Sync/Async switch controls of the Selection Sequence
- 2 knobs to set the from value and to value for the sequence mapping or the limits
- If the selection mode feature is Probability, then the label of the controls is map
- If the selection mode feature is Limits, then the label of the controls is limit
- From and to values can be set between 0% and 100%
- If the selection mode feature is Probability, then the current sequence value will be projected from the 0..100 interval to the given from to value range, and will be applied as probability of the current note
- If the selection mode feature is Limit, then the current note will be filtered out if the sequence value is outside of the range of the limit values
- Selection Mode switch
  - Below the Probability map/Limits controls of the Selection Sequence
  - Switch between the Probability mode and the Limit mode
- Sequencer Gear Menu
  - actions for sequencer parameters
  - presets, randomizations, shift, flip, copy and other actions
  - context sensitive menu, the actions are based on the currently selected display mode of the Sequencer editor.



---

## NoteMap Section

NoteMap Section is in the right side of the main body. NoteMap has two modes, Predefined (classic sequencer behavior) or Dynamic (arpeggiator behavior)

- NoteMap values
  - 8 control to see and/or edit the MIDI note value for the related sequencer lane
  - aligned to the lanes of the Sequencer editor, so easy to find out which lane related to a given NoteMap value
  - If up down arrows are visible, we are in Predefined NoteMap mode, in this case the values are editable, otherwise not
- NoteMap overview display
  - Thin lines in the right side of the NoteMap values
  - Represents the NoteMap presets and the selected notemap (highlighted with orange) in the Predefined mode
  - Represents the NoteMap values by the incoming notes in the Dynamic (arpeggio) mode
- NoteMap Gear Menu
  - actions for NoteMap
  - presets by devices (like Redrum, Kong, Rytmik, Umpf, Mimic, Dr. OctoRex, ScaleMatrix)
  - presets by scales (like Major, Minor, Lydian, Mixolydian, Spanish, Dorian, Phrygian, Harmonic Major, Harmonic Minor, Melodic Major, Melodic Minor, Whole tone, Blues V, Pentatonic Major, Pentatonic Minor etc...)
  - presets by intervals (like Minor seconds, Major seconds, Minor thirds, Major thirds, Fourths, Tritones, Fifths)
  - random notes
  - Ascending, descending or random order (Shuffle)
  - Shifting up or down
  - copy
  - visible when we are in predefined mode
- NoteMap mode switch
  - switch between the Predefined (classic sequencer behavior) and Dynamic (arpeggiator behavior) modes
  - in the Dynamic (arpeggio) mode the NoteMap filled by the incoming notes

- Arpeggio recursion switch
  - visible when we are in dynamic mode
  - off displayed as 1, on displayed as infinity on this switch
  - when we are in dynamic mode, and the number incoming MIDI notes are less than the number of lanes (8) then all of the lanes can be filled with the currently incoming MIDI notes by repeating them. For this feature turn on this switch.
- Predefined NoteMaps
  - visible when we are in predefined mode
  - 4 radio buttons to select the current NoteMap
- Arpeggio mode
  - visible when we are in dynamic mode
  - 4 radio buttons to select the arpeggio NoteMap, represented by arrows
    - lowest to highest
    - highest to lowest
    - oldest to newest
    - newest to oldest

## Footer Section

- Run button
  - Toggle between Run and Stop
  - Pearls start automatically when the main sequencer started.
- Mute button
  - When Pearl is playing, you can bypass the generated notes by this button.
- Patch info, customizable text for notes about the patch

## Back panel



### CV Inputs

Pearls has 3 CV input for modulating the current velocity, length, and selection sequencer values.

If a cable is connected to a CV input, the corresponding Pearls sequencers will be modulated with the CV value.

### CV Outputs

Pearls has 5 CV output per channel ( $5 \times 8 = 40$ )

- Notes
- Gates
- the current value in the velocity sequencer
- the current value in the length sequencer
- the current value in the selection sequencer

CV input has no effect on this CV output values. You can use Pearls as a 3x8 channel 32 step long CV sequencer ;)

---

## TIPS AND TRICKS

### n tuplets

In various music subgenres unusual tuplets are very popular, and with Pearls you can achieve this in an easy way. For example if you want to create a 19 tuplets, just set the rhythm type to Pulses, and set the Cycle mode, and set the cycle length to matching your time signature (eg 4/4), and after that just set the steps to 19.

### Different rhythms for different drums/percussions with one instrument

(Like Kong, Redrum etc.)

Pearls can be stacked when you are using it in Predefined Notemap mode, and this is what we want in this case. You just must set the appropriate channel settings for each Pearls device.

### Evolving Rhythms with probability

This is just one example; you can achieve this in many other ways.

Go to the selection sequence and set the sequencer values to 100 for the steps, when you always want to trigger a note. And set a smaller value, where you want to give a chance to skip the note. Set the "Selection Map To" value to 100, and a smaller value for the "Selection Map From" parameter.

Probability is actually a chance to play a note, and if the sequencer arrives to the same step again, this chance will be "evaluated" again, and the note played or not played by the given chance.

## Evolving Rhythms with limits

Go to the selection sequence and set the sequencer values to 100 for the steps, when you always want to trigger a note. And set smaller and smaller values, where you want to skip notes during your performance. Set the "Selection Map From" value to 0 and set the "Selection Map To" value to 100. During the performance change the "Selection Map From" value, and when a current sequencer value is smaller than the "Selection Map From" value, the note will be skipped.

Compared to the probability mode, in this case you have strict control over the notes.

## Evolving Rhythms with the rhythm generator

Choose the Euclid collision rhythm type and set different Euclidean rhythms for the A (Ochre) and B (Pink) rhythms. Choose the Random choice operation. (Click on the upper arrow of the operation selector until "RANDOM CHOICE" is selected or click on the operation selector and drag your mouse up until "RANDOM CHOICE" is selected.)

With the Balance knob you can smoothly change the rhythm from A to B. If you are not satisfied with the transition, just change the rnd seed value.

## Evolving Music

Try Pearls on an instrument, which is sensitive enough for the velocity and can produce short and long/sustained notes also.

Set any rhythm in the rhythm generator.

Set a melody sequence in the Notes sequencer. (Actually, can be a random melody)

Set random values in the Velocities, Lengths, and Selections sequencers.

Set your favorite scale in the NoteMap.

Set different lengths and sync/async modes for the sequencers.

Start Pearls and play with the map knobs of the sequencers.

It has a big chance; you are hearing evolving music ;)

## Arpeggiated chord progression

For this, you must use 2 instances of Pearls. And one more player for generating chords.

Set a slow rhythm with long notes in Predefined mode in the first Pearls. This will generate progression. For example, in Cycle mode set 8/4 long cycle and set 4 step long rhythm and with the length sequencer set the values to producing 100% long notes. In the note sequencer draws a monophonic sequence.

Put a Scales & Chords or other chord player after this Pearls and set up the chords.

And finally put another Pearls after it. But in this instance of Pearls use it in Dynamic (Arpeggio) mode. Set an interesting and fast rhythm here and draw something in the note sequencer.