

# Philip Meyer MIDI Tools

**Blocks**

Notes 2 Quantize



Pitch C3 Vel. 100 Dur. 100%

**Divs**

Sliders 4 Quantize



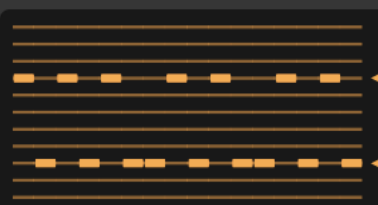
Velocity Pitch Rng. [110] [80]

Shape 0.6 Skew 0.5 Dir <

**Develop**

Generations 2 Develop

Mode Probabilistic

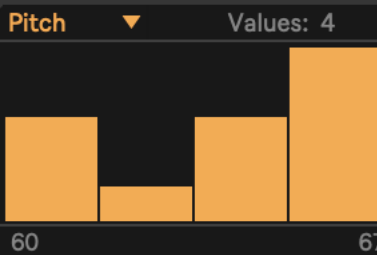


Generate

Density 41% Rotate 0%

**Condition Transform**

Pitch Values: 4



60 67

Modifier

Pitch +0

Accum : Inf Uni Wrap


**Pattern Transform**

Pattern

Mode Note Every 1

Density 4 Variant

Algo Eucl.




Modifier

Delete

**Draw**

Pitch Velocity Chance

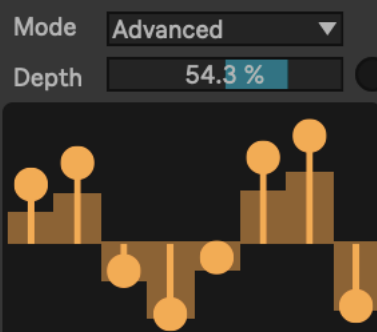


Reset Rel ± 12

**Swing**

Mode Advanced

Depth 54.3%



**Segment**

Div 4 Quantize Proportional



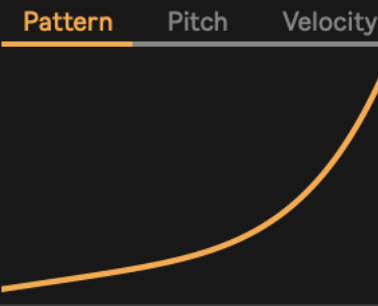
1/32 1/8

Velocity Pitch Rng. 100 100

Shape 0.0 Skew 0.5 Dir ↘

**Phase Pattern**

Pattern Pitch Velocity



0.00 0.13 Jitter 0%

Duration 100% Notes 8 Q

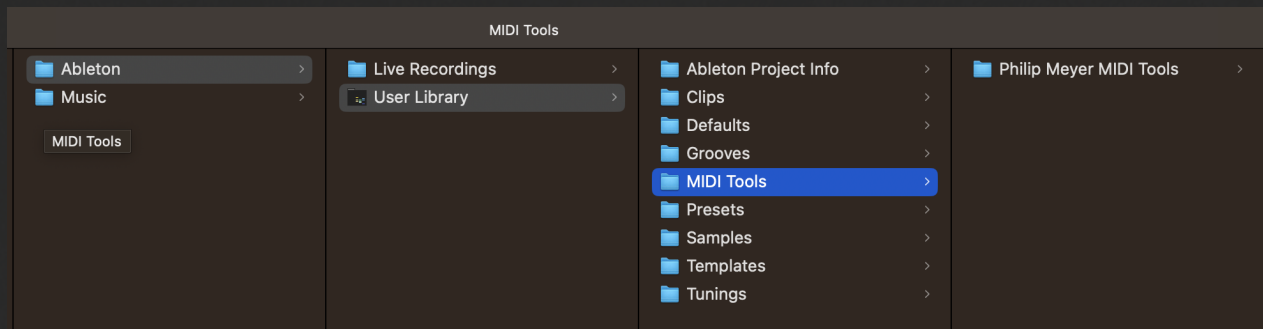
[philip-meyer.com](http://philip-meyer.com)

[philipmeyer.gumroad.com](http://philipmeyer.gumroad.com)

# Philip Meyer MIDI Tools

## Installation

To install MIDI Tools, simply drag the AMXD files into the MIDI Tools folder of your Ableton User Library. If there is no MIDI Tools folder, create one!



## Questions, Feedback, Ideas


Join Rhizomic Sequencing server on Discord: <https://bit.ly/rhizomic>

## Additional Resources

YouTube: [https://www.youtube.com/@p\\_\\_meyer](https://www.youtube.com/@p__meyer)

# Blocks Generator

A generator proportionally divides a clip to make rhythms. Inspired by Sam Tarakajian and Alex Van Gils' Nestup, a scripting language for creating nested tuplets and other complex rhythms.



**Number of Notes**  
The number of notes that will be created. This changes the number of Blocks sliders

**Blocks**  
Sets the lengths of the notes that will be created. The notes will fill the selected range of the clip, and the lengths are set proportionally depending on the slider values.

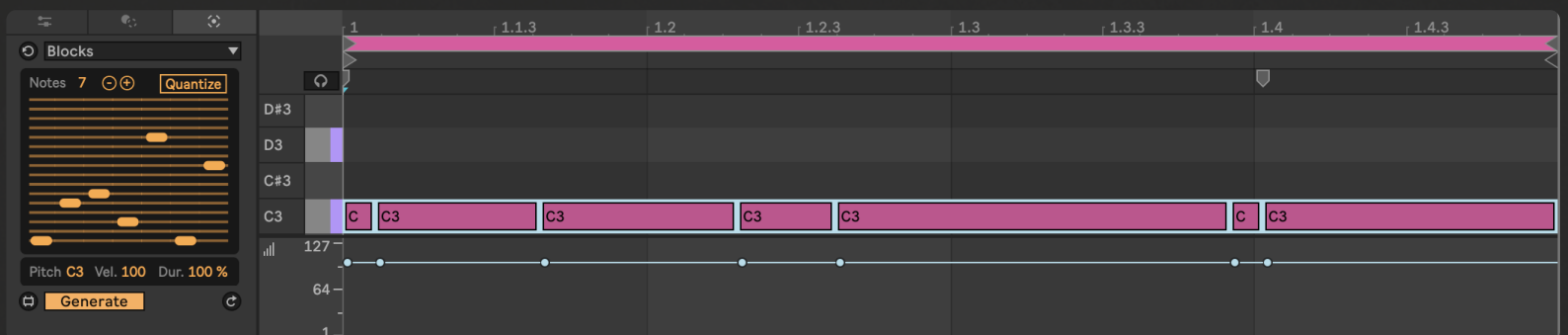
**Quantize**  
Quantizes the start position of the notes to the MIDI clip's grid

**Pitch**  
The pitch of the notes that will be generated.

**Velocity**  
The pitch of the notes that will be generated.

**Duration**  
Shifts the generated pitches up and down.

The interface shows a 'Blocks' panel with a 'Notes' count of 2, a 'Quantize' button, and a piano-roll visualization with two orange note blocks. Below the piano roll are controls for Pitch (C3), Velocity (100), and Duration (100%).



The screenshot shows a MIDI piano roll with a pink piano roll area. The piano roll is divided into seven segments, each containing a note on the C3 pitch line. The segments are labeled with rhythmic values: 1, 1.1.3, 1.2, 1.2.3, 1.3, 1.3.3, 1.4, and 1.4.3. The piano roll is set to a C3 pitch and a velocity of 127. The 'Blocks' panel on the left shows 'Notes 7' and a 'Generate' button.

Moving the sliders in Blocks changes the relative spacing of the notes in the MIDI clip.

# Phase Pattern Generator

A generator that creates rhythms by bending time.

## Pattern Tab

**Function**  
Click and drag to modify the function, or use the number boxes. Its shape will control the note position. Where the curve is more steep, the note spacing will be wider.

**Duration**  
Scales the duration of the notes without changing their starting positions

**Number of Notes**  
Number of notes to create.

**Quantize**  
Quantizes the note start positions to the MIDI clip's grid.

A logarithmic curve creates a “bouncing ball” rhythm whose note durations shorten over the course of the pattern.

## Pitch/Velocity Tabs

**Function**  
Click and drag to modify the function, or use the number boxes. Its shape will control the pitch or velocity.

**Function Direction**  
Flips the function horizontally

**Range**  
The range of pitches or velocities generated by the function

**Link Range**  
Link the upper and lower bounds of the range, creating a static value

**Iterations**  
The number of times the pattern should loop over the course of the notes.

The pitch follows the function shape you've created. The Iterations sets the number of times the function is fit into the pattern.

# Turing Machine Generator

A generator based on the Music Thing Modular Turing Machine Eurorack sequencer module.

## Advance Section

Click these buttons to generate notes by adding one or more new bits to the shift register

- 0 adds an 'off' bit
- 1 adds an 'on' bit
- R adds a random bit (which is controlled by the Big Knob)
- R x Length is the same as clicking R Length number of times

## Big Knob

Determines how the new bits in the register are created. With the knob at the center 12 o'clock, the new bit's value will be random. Turning the knob right or left reduces the amount of randomness.

## Register Viewer

Displays the first 8 bits in the shift register. These are used to determine the pitch

## Length

The number of bits to loop within the shift register

## Clear

Clear the internal shift register and any notes that this device created

## Pitch Range

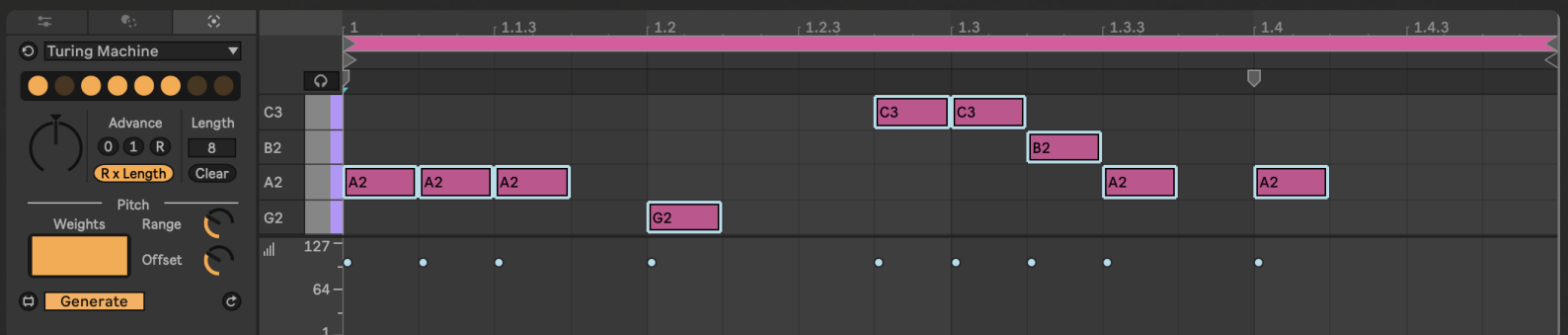
Scales the overall range of the generated pitches.

## Weights

Controls how the pitch is calculated from the bit values. In general, the greater the area under the sliders, the greater the pitch range will be.

## Pitch Offset

Shifts the generated pitches up and down.



Click the buttons in the Advance section to build the clip. Clicking 0, 1, or R will build the pattern 1 bit at a time. Clicking R x Length builds the entire pattern from Length bits.

# Polyrhythm Generator

A multi-track generator for creating polymetric and polyrhythmic patterns.

**Density**  
Number of notes in the pattern

**Length**  
Length of the pattern in steps

**Variant**  
Changes the pattern while keeping Steps and Duration constant.

**Velocity**  
Velocity of the notes in the track.

**Enable Track**  
When disabled, notes for this track won't be generated

**Clear**  
Resets all of the tracks to a default state.

**Track Transposition**  
Sets the pitch of each track by offsetting Base Pitch. If Scale is enabled for the MIDI clip, the offset is in scale degrees. If Scale is disabled, the offset is in semitones

**Base Pitch**  
The pitch from which each track will be transposed.

**Algorithm**  
The algorithm used to generate the rhythms.  
Euclidean creates patterns that evenly distribute the activated steps across the pattern. Omni allows you to choose any possible pattern with just three parameters.

**Distribution**  
Decides how the pattern is distributed within the MIDI clip. This setting allows you to toggle between polymetric and polyrhythmic patterns. Hug and Fill will give polymeters, while Stretch results in polyrhythms

+Sd	Len.	Den.	Var.	Vel.
5	16	0		100
4	16	0		100
3	9	5		100
2	27	5		100
1	16	7		100
0	13	7		100

Base: C3    Algo: Eucl.    Distrib.: Fill    C

A screenshot of a MIDI piano roll showing a polymetric pattern. The piano roll has a grid with a pink header bar. The pattern consists of several horizontal bars of varying lengths and positions, representing notes in different tracks. The distribution is set to 'Fill', resulting in a polymetric pattern.

A polymetric pattern using the Fill Distribution

A screenshot of a MIDI piano roll showing a polyrhythmic pattern. The piano roll has a grid with a pink header bar. The pattern consists of several horizontal bars of varying lengths and positions, representing notes in different tracks. The distribution is set to 'Stretch', resulting in a polyrhythmic pattern.

Changing the Distribution to Stretch makes the pattern polyrhythmic

# Condition Transform Transformer

A multi-function transformer that modifies certain notes according to a rule.

**Attribute**  
The attribute to be analyzed.

**Histogram**  
The graph shows the distribution of values across the selected attribute. Click and drag to highlight specific values and select notes to be transformed.

**Number of Values (View Only)**  
Total number of unique values in the range

**Bounds (View Only)**  
The maximum and minimum values in the range

**Modifier**  
The algorithm that will be used to transform the selected notes

In this example, we've used the Pitch attribute to select all of the notes that have the two highest pitches. Then, we subdivided them by 4.

## Modifiers

### Increment Modifiers

Adds or subtracts from the selected notes' pitch, velocity, or chance.

**Accumulate**  
When enabled, adds Increment to a running total before adding to the attribute value

**Increment**  
Amount to add

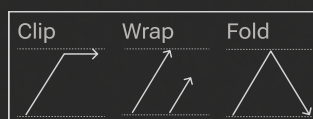
**Iterations**  
Number of times to add to the accumulation total before clipping, wrapping, or folding.

**Overflow Mode**  
Controls how values that cross the maximum or minimum are handled.

**Polarity**  
Sets the maximum and minimum for the accumulation value.\*

- Uni :  $0 \leftrightarrow \text{Iterations} \times \text{Increment}$
- Bi :  $-\text{Iterations} \times \text{Increment} \leftrightarrow \text{Iterations} \times \text{Increment}$

\* When Accumulate is enabled and the Overflow Mode is Wrap or Fold.



### Divide Modifier

Divides the selected notes.

**Divisions**  
Number of notes to divide the selected note into

**Quantize**  
Quantizes the start position of the divisions to the MIDI clip's grid

### Delete Modifier

Deletes the selected notes.

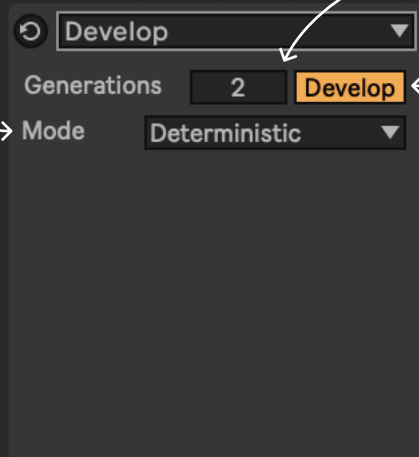
# Develop Transformer

A transformer that can make a pattern gradually appear or fade away.

## Mode

- **Deterministic:** Develops or degrades the pattern in a predictable fashion, using an evenly-spaced Euclidean algorithm and note muting.
- **Probabilistic:** Uses the chance attribute of notes to develop or degrade the pattern.

## Deterministic

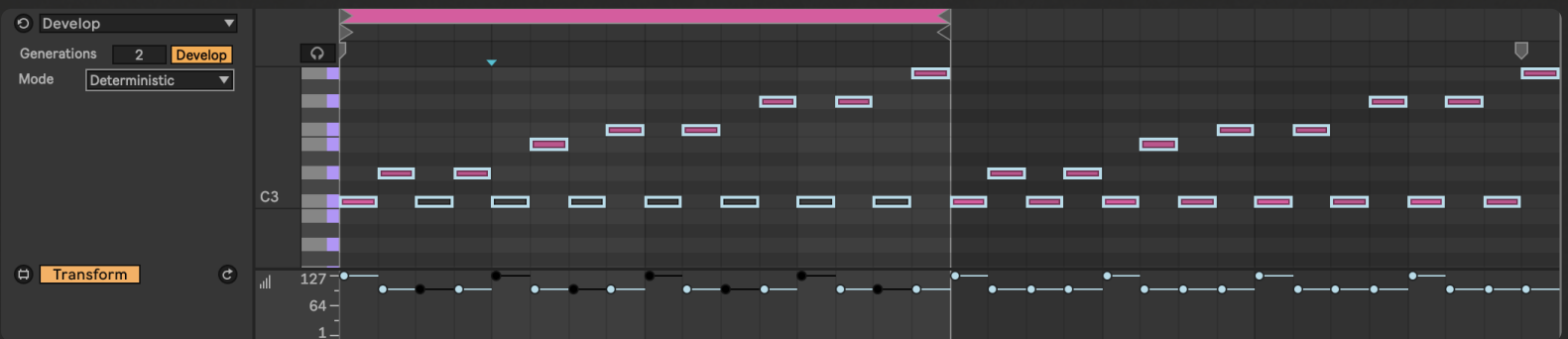


## Generations

The number of loops of the pattern over which the development/degradation occurs.

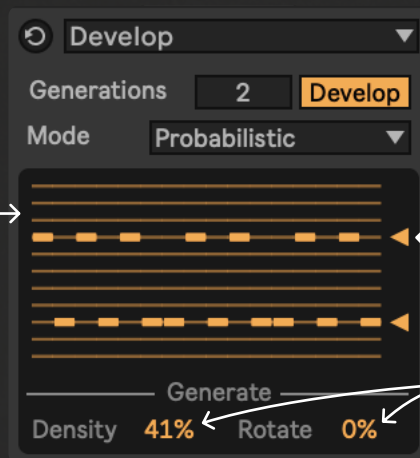
## Build Direction

Develop starts with a sparse pattern and builds to the complete one  
Degrade is the opposite!



**Deterministic mode.** The second bar is the original pattern, which has been duplicated to the first bar, and every other note in the first bar has been muted.

## Probabilistic



## Probabilities

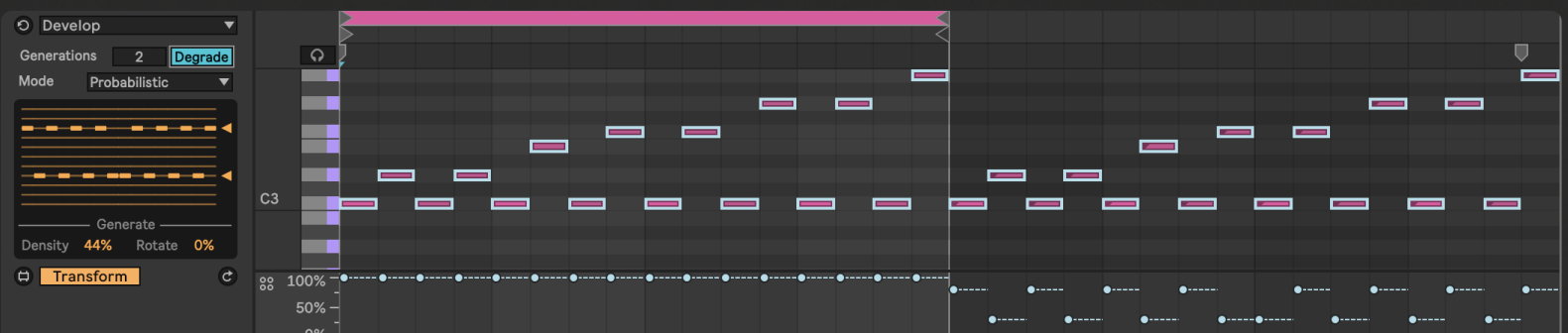
Sets the chance for each note at the 'most sparse' state, depending on the Build Direction. At the pattern's 'least sparse' state, the chance will be 100% for all notes.

## Range Control

Adjusts the maximum and minimum range by clipping the slider values.

## Generate Density and Rotation

Modifying these parameter overwrites the slides with a Euclidean pattern

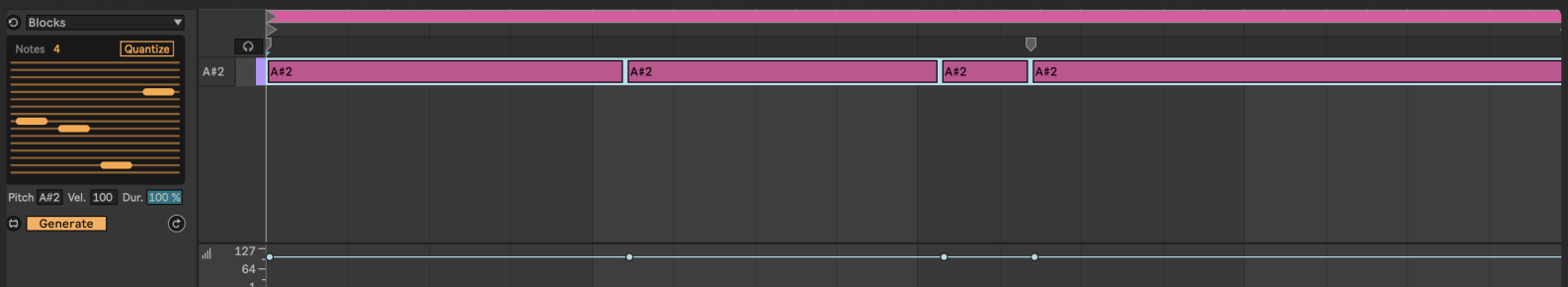
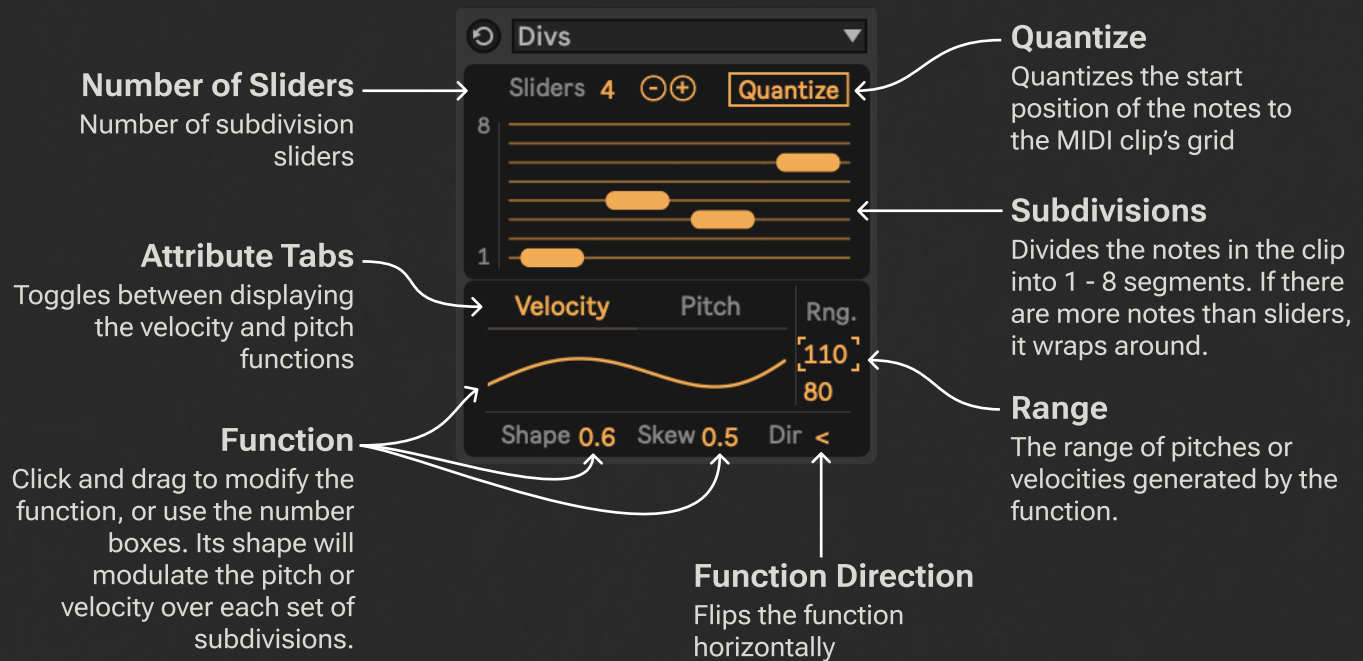


**Probabilistic mode.** Because the build direction is Degrade, the full pattern is the starting point, and the ending point is the same pattern with chance set by the sliders in the Develop device.

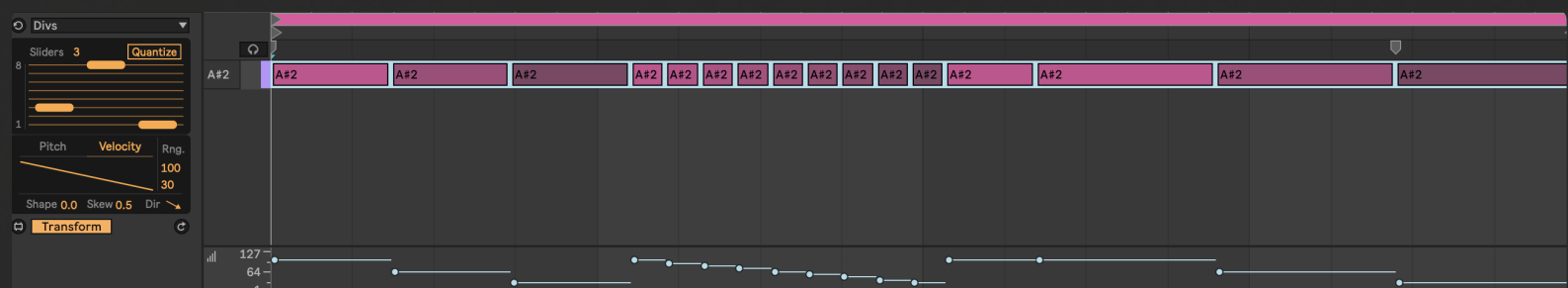


# Divs Transformer

Subdivide a list of notes. Designed to be used in conjunction with Blocks to make nested rhythms.



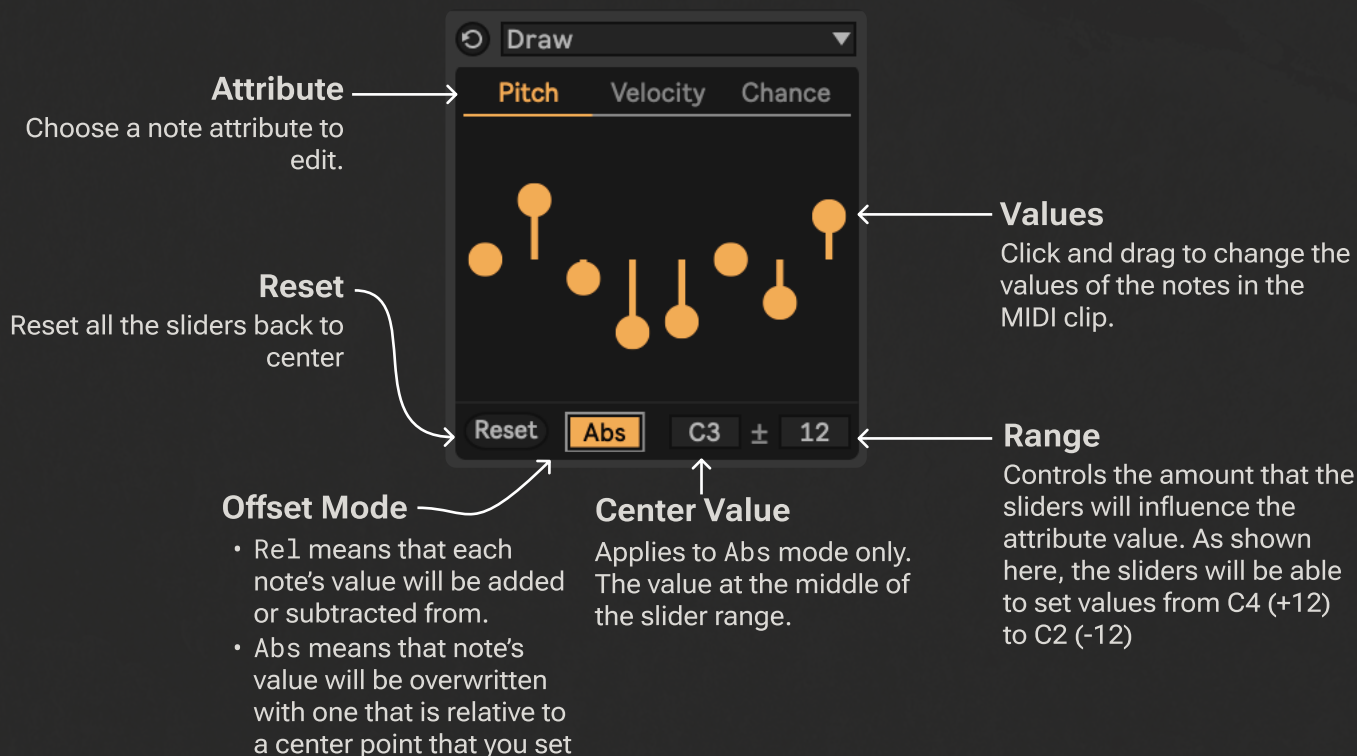
We'll start with a pattern generated by blocks.



Next, we'll use three dividers to divide the first, second, and fourth note with a falling velocity. The third note is not divided because the third slider's value is 1.

# Draw Transformer

Quickly manually edit note pitch, velocity, and chance.



**Attribute** → Choose a note attribute to edit.

**Reset** → Reset all the sliders back to center

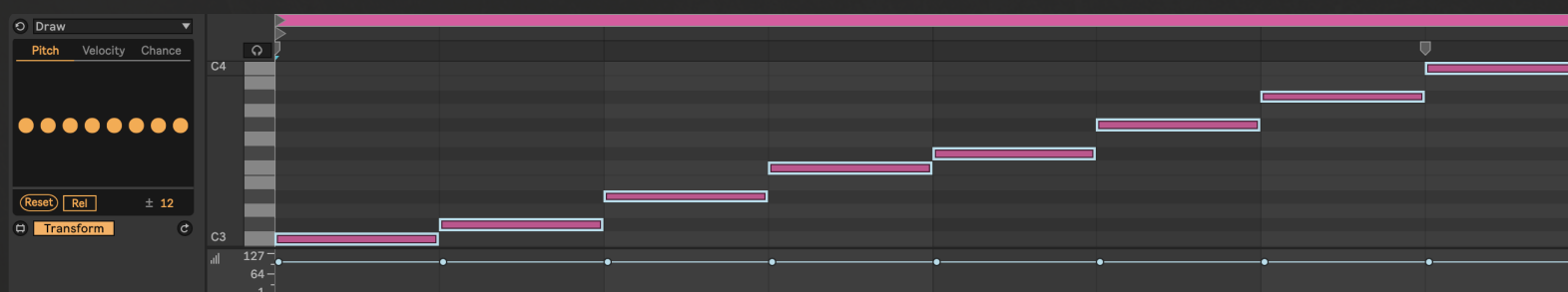
**Values** → Click and drag to change the values of the notes in the MIDI clip.

**Range** → Controls the amount that the sliders will influence the attribute value. As shown here, the sliders will be able to set values from C4 (+12) to C2 (-12)

**Offset Mode** →

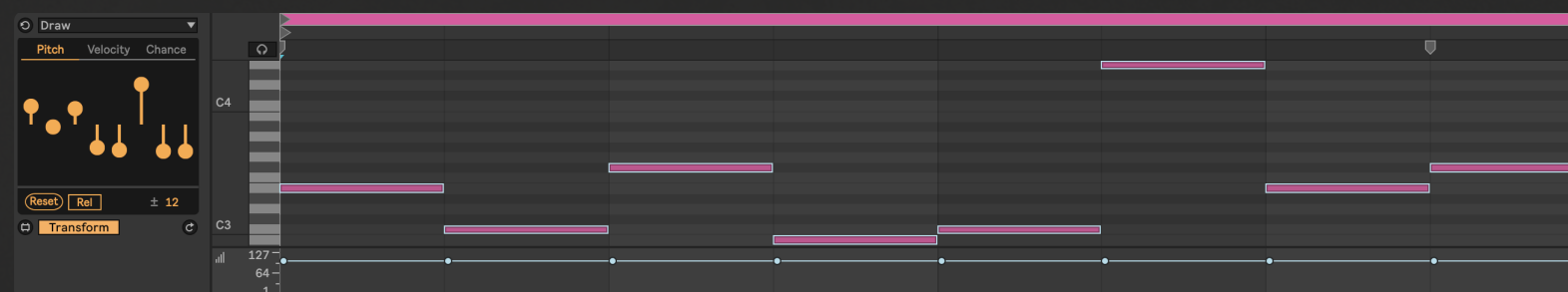
- Rel means that each note's value will be added or subtracted from.
- Abs means that note's value will be overwritten with one that is relative to a center point that you set

**Center Value** → Applies to Abs mode only. The value at the middle of the slider range.



The screenshot shows a MIDI piano roll with a single track. The vertical axis is labeled with C4 and C3. The horizontal axis represents time. A series of notes are placed on a grid, showing a simple rising pitch sequence from C3 to C4. The Draw Transformer interface is visible on the left, with the 'Pitch' attribute selected and the 'Rel' mode button highlighted.

To start, we have a clip with a simple rising pitch. Next, we'll transform it with Draw.



The screenshot shows the same MIDI piano roll as before, but the notes have been transformed. The Draw Transformer interface on the left now shows the 'Abs' mode button highlighted, and the sliders for the notes are visible, indicating that the pitches have been adjusted relative to their original positions.

Adjusting the sliders in Draw in Rel mode offsets the pitches of the notes relative to their original position.

# Pattern Transform Transformer

A multi-function transformer that uses a pattern to choose which notes to modify.

**Advance Mode**  
Determines whether the pattern's steps are determined by a count of notes (Note) or an interval the clip's grid (Grid)

**Pattern Density**  
Percentage of the steps in the pattern that are active

**Pattern Algorithm**  
Determines how the pattern is generated. Eucl. always distributes the events evenly. Omni is a more flexible algorithm.

**Every**  
Determines the size of each step. In Note mode, this is a number of notes. In Grid mode, it's a timing interval

**Variant**  
Rearranges the active steps.  
• Eucl. mode: rotates the pattern.  
• Omni mode: selects from among all possible patterns for the density.

**Pattern (View Only)**  
The pattern you've created.

**Modifier**  
The algorithm that will be used to transform the selected notes

Here, we started with a 16th notes at a constant pitch of C3. Using Pattern Transform, we applied a Euclidean pattern and increased the pitch by 3 of any notes that fall into the active steps of the pattern.

## Modifiers

### Increment Modifiers

Adds or subtracts from the selected notes' pitch, velocity, or chance.

**Modifier**  
Pitch: +0  
Accum: Inf Uni Wrap

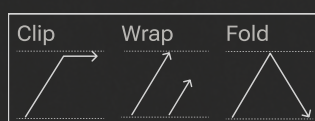
**Increment**  
Amount to add

**Modifier**  
Velocity: +1  
Accum: x4 Uni Wrap

**Iterations**  
Number of times to add to the accumulation total before clipping, wrapping, or folding.

### Overflow Mode

Controls how values that cross the maximum or minimum are handled.



### Polarity

Sets the maximum and minimum for the accumulation value.\*

- Uni :  $0 \leftrightarrow \text{Iterations} \times \text{Increment}$
- Bi :  $-\text{Iterations} \times \text{Increment} \leftrightarrow \text{Iterations} \times \text{Increment}$

\* When Accumulate is enabled and the Overflow Mode is Wrap or Fold.

### Delete Modifier

Deletes the selected notes.

**Modifier**  
Delete

### Fuse Modifier

Joins the selected notes that fall within the pattern step

**Modifier**  
Fuse

### Divide Modifier

Divides the selected notes.

**Modifier**  
Divide: 2  
Quant.

**Divisions**

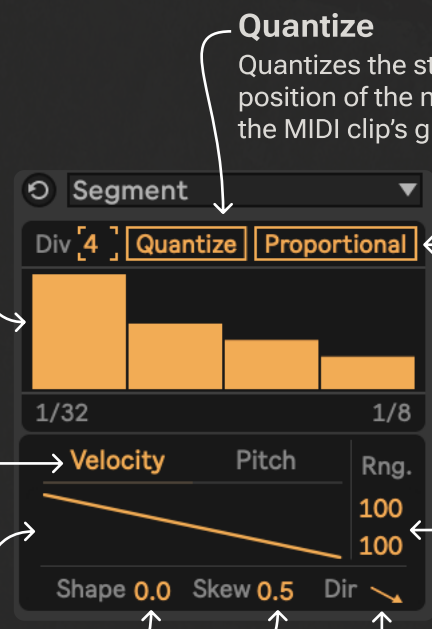
Number of notes to divide the selected note into

### Quantize

Quantizes the start position of the divisions to the MIDI clip's grid

# Segment Transformer

Subdivide conditionally based on note duration.



**Quantize**  
Quantizes the start position of the notes to the MIDI clip's grid

**Histogram**  
The graph shows the distribution of note durations. Click and drag to highlight specific values and select notes to be transformed.

**Division Mode**  
By default (Proportional), Segment will divide each note into even segments. But sometimes, you want the resulting notes to all be the same length. Switch this to Fixed to do that!

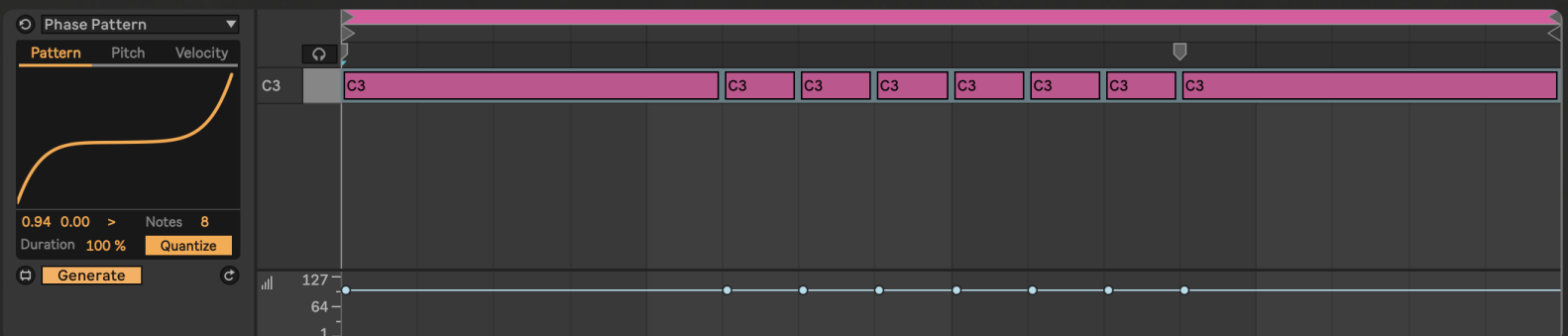
**Attribute Tabs**  
Toggles between displaying the velocity and pitch functions

**Range**  
The range of pitches or velocities generated by the function.

**Function**  
Click and drag to modify the function, or use the number boxes. Its shape will modulate the pitch or velocity over each set of subdivisions.

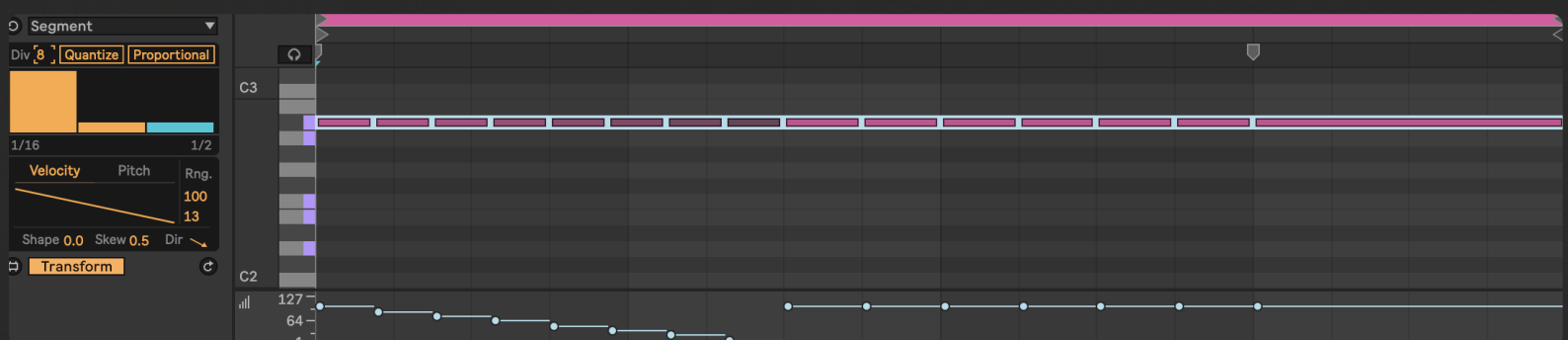
**Function Direction**  
Flips the function horizontally

The interface shows a 'Segment' panel with a 'Div' of 4, 'Quantize' and 'Proportional' buttons, a histogram, and tabs for 'Velocity', 'Pitch', and 'Rng.'. The 'Velocity' tab is active, showing a function curve and a 'Range' of 100. Parameters include 'Shape 0.0', 'Skew 0.5', and 'Dir'.



The Phase Pattern interface shows a 'Pattern' tab with a function curve and a 'Generate' button. The 'Notes' field is set to 8. The MIDI piano roll shows a single note on C3 divided into 8 equal segments.

We'll start with a quantized pattern generated by Phase Pattern.

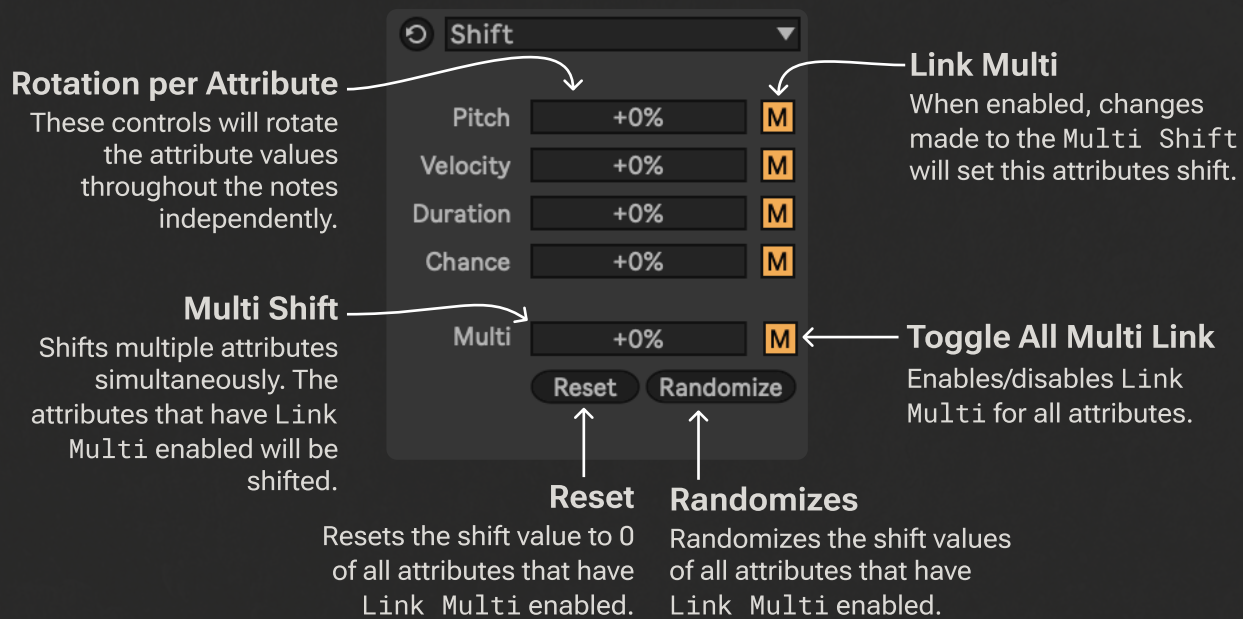


The Segment Transformer interface shows a 'Div' of 8, 'Quantize' and 'Proportional' buttons, a histogram, and tabs for 'Velocity', 'Pitch', and 'Rng.'. The 'Velocity' tab is active, showing a function curve and a 'Range' of 100. Parameters include 'Shape 0.0', 'Skew 0.5', and 'Dir'. The MIDI piano roll shows a single note on C3 divided into 8 segments with a declining velocity, creating an echo effect.

Then, we'll select the longest note note by clicking on the histogram in Segment. Segment will divide only this note into 8 segments with a declining velocity, creating an echo effect.

# Shift Transformer

Subdivide a list of notes.



**Rotation per Attribute**  
These controls will rotate the attribute values throughout the notes independently.

**Multi Shift**  
Shifts multiple attributes simultaneously. The attributes that have Link Multi enabled will be shifted.

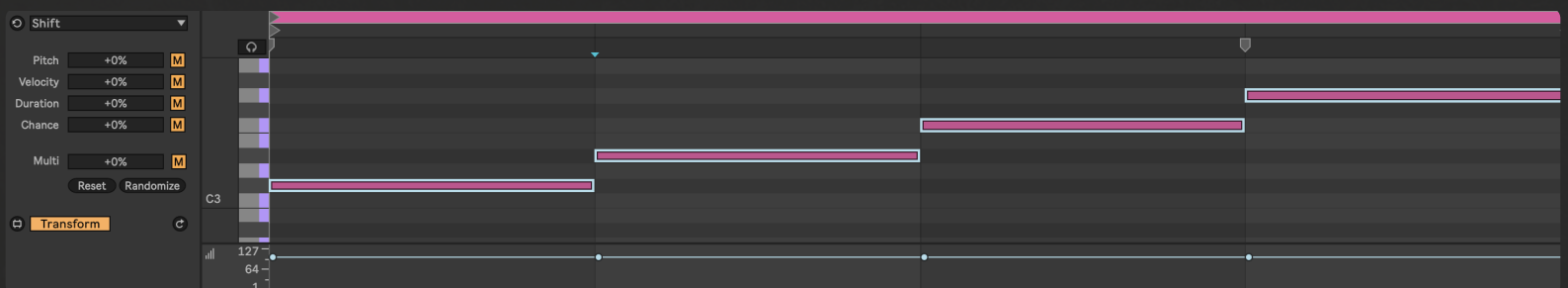
**Link Multi**  
When enabled, changes made to the Multi Shift will set this attributes shift.

**Toggle All Multi Link**  
Enables/disables Link Multi for all attributes.

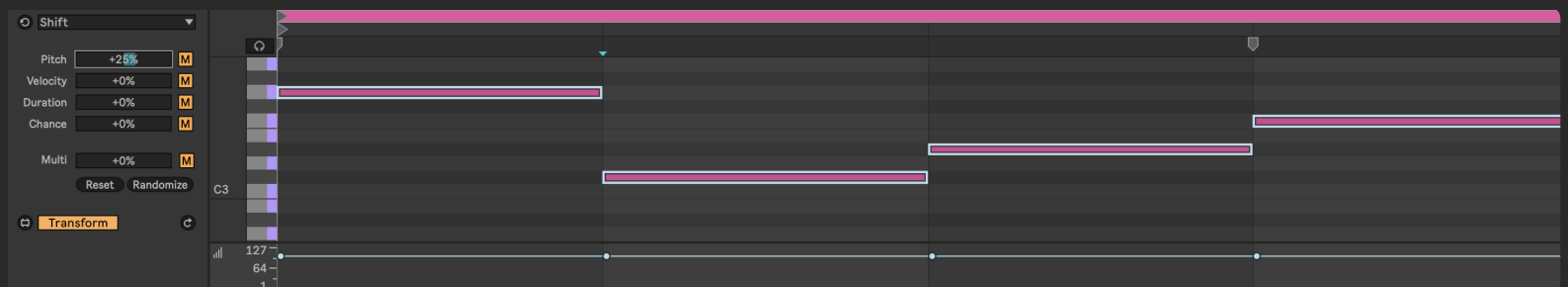
**Reset**  
Resets the shift value to 0 of all attributes that have Link Multi enabled.

**Randomizes**  
Randomizes the shift values of all attributes that have Link Multi enabled.

Shift  
Pitch +0% M  
Velocity +0% M  
Duration +0% M  
Chance +0% M  
Multi +0% M  
Reset Randomize



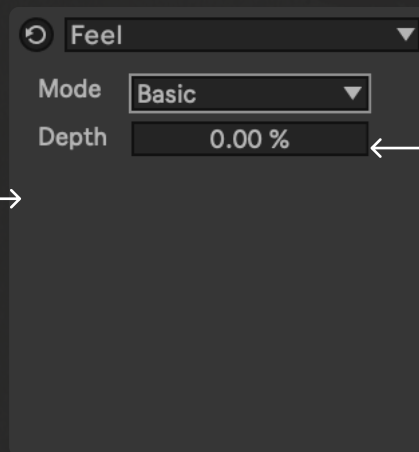
In this original, untransformed clip, we have both a rising pitch and a rising velocity.



Here, with the Pitch Rotation set to 25%, the pitches have shifted rightward and wrapped around, but the velocities have not. This is because each attribute can be shifted independently of the other attributes!

# Feel Transformer

A transformer for creating swing, grooves, and microtiming

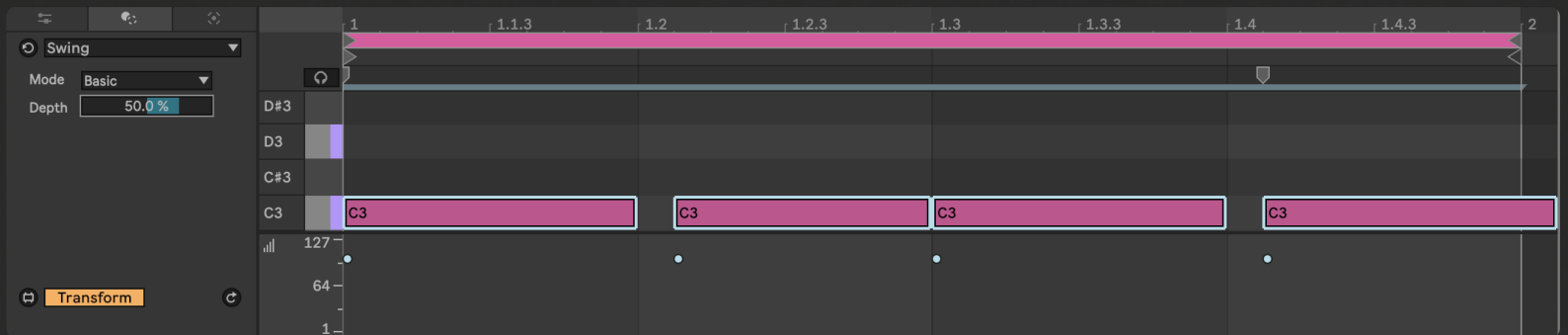


## Swing Depth

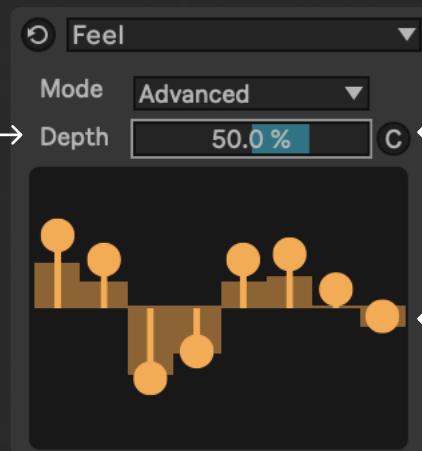
Amount of swing to apply to the pattern. Swing is applied using the clip's grid. Positive values shift every second note back, negative values shift them forward

## Mode

- Basic is the typical swing we all know and love.
- Advanced allows you to fine-tune the timing for each step in the grid independently.



Basic swing



## Swing Depth

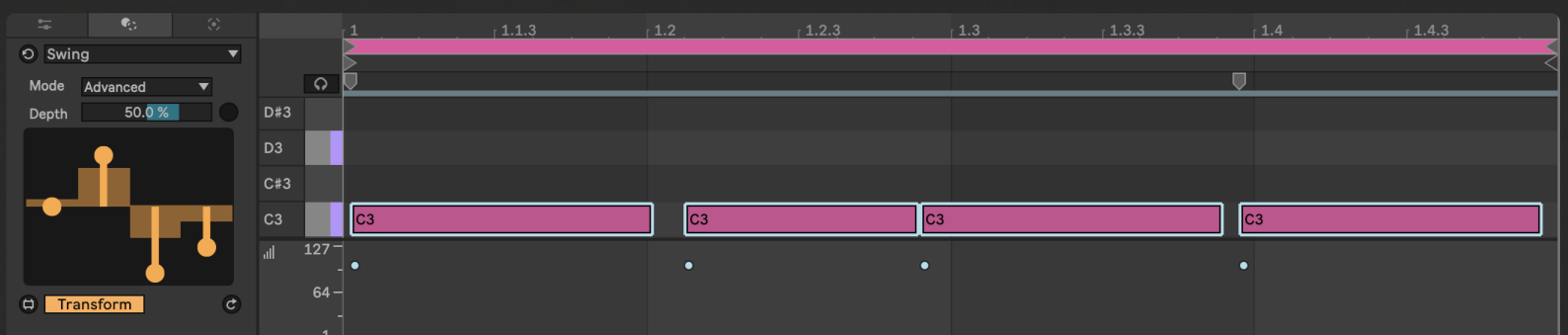
Controls the extent to which the sliders shift the notes.

## Clear Microtiming

Sets all Microtiming sliders to 0.

## Microtiming

There's one slider for each grid line in the MIDI clip. Dragging the slider in the positive direction (upward) shifts the note back (to the right). The highlighted bars show the actual adjustment, after normalizing and applying Depth



Advanced swing