Meta-Gest

A flexible and adaptable performance environment created in Max/MSP for real-time audio processing.



Konrad Kaczmarek November 2007

Meta-Gest: A flexible and adaptable performance environment created in Max/MSP for real-time audio processing.

l. Interfa	ce Overview:	page 1-4
	Channel Strip Effects Window Aux Assign DSP Monitor Sync Signal Multi-Channel Recording	1 2 3 3 4
ll. The Pr	ocesses:	5-12
	GrainLt Granular ffilter Convolve Morph Sample Sync Freezer Line_in Looper	5 6 7 8 8 9 10 11 11 11
III. MG_C	Control:	13-19
	Overview Sig_Control Keyboard Routing Footy Sequencer Faders MIDI in Rhythms Delta_Preset	13 14 15 16 17 18 18 19 19
IV. Exam	ples	20-27
	32_grains - main interface 32_grains - routing	20 21
	Freezing_pitch - main interface	22
	Feeding_pitch - main interface Feeding_pitch - effects window	23 24
	Combing_beats - main interface Combing_beats - sequencer Combing_beats - effects window	25 26 27



—Individual channel slots 1-16 (see description below)

Channel_strip:

load processing modules, set input signals, and adjust volume and pan



1. Choose from direct ADC input, pre or post fader signals from the individual channels, or one of the aux send signals.

2. Select from the menu to load one of the processing modules into the channel slot.

. Signal input list

1. Signal input list						
None	adc1					
1Pre	adc2					
1Post	adc3					
2Pre	adc4					
2Post	adc5					
3Pre	adc6					
3Post	au×1					
4Pre	aux2					
4Post	out					
5Pre	out2					
5Post						
6Pre						
6Post						
7Pre						
7Post						
8Pre						
8Post						

2. Process list

None 🔷 🍸
None
GrainLt
Granular
Filter
convolve
morph
sample
sync
linein
freezer
looper

The effects window:

The effects window allows you to apply various signal processing effects to the channel output signal. It uses a matrix mixer to allow flexible routing of the signal, including the ability to feed a process back into itself. Inputs are represented by rows, outputs are in the columns.



The aux assign window:

Adjust the volume of up to four input signals for each aux send channel. Apply compression and save as a preset.



Choose from the list of input signals (see list to the righ
 Volume adjustments



DSP monitor window:

Monitor input and output signal levels, access the aux send assignments, turn on/off signal processing, load in an overall preset and access the multichannel recording patch.



Sync Signal:

Set the project tempo sync signal and calculate measure or beat sub-division durations in ms.



- calculated durations of various measure groupings (assuming 4/4 time)

Multi-Channel Recording:

Create discrete audio files for various input and output channels.



1. Inputs files are automatically named with _in1.aiff, _in2.aiff, etc. extensions. Outputs file have _out1.aiff, _out2.aiff, etc. extensions.

GrainLt:

a sample playback module that uses a single-windowed granular process to allow independent control of time and pitch as well as many other fun granlar-based effects (less CPU expensive).



- 1. This parameter tunes all of the grains to the same pitch.
- 2. Defines the lower limit of the pitch spread when random (r) pitch mapping is selected.
- 3. Defines the upper limit of the pitch spread when random (r) pitch mapping is selected.
- 4. Defines the number of equal stages between the lower and upper limits of the random pitch spread.
- 5. Defines the maximum number of pitch increments a grain can move when in random mapping mode.
- 6. Force the playhead to follow the global sync signal (speed variable becomes rate multiplier).
- 7. This parameter determines the size of the secondary windowing function.
- 8. Determines the size of the primary windowing function.
- 9. Introduce bandlimited noise to the playhead position.
- 10. Define the type of chord to use when in chordal pitch mapping mode.
- 11. Define the type of scale map to use when in scalar pitch mapping mode.
- 12. Select from random (see notes 2. and 3.), scalar, or chordal pitch mappings.

Granular:

a sample playback module that uses a double-windowed granular process to allow independent control of time and pitch as well as many other fun granlar-based effects.



1. This parameter tunes all of the grains to the same pitch.

- 2. Defines the lower limit of the pitch spread when random (r) pitch mapping is selected.
- 3. Defines the upper limit of the pitch spread when random (r) pitch mapping is selected.
- 4. Defines the number of equal stages between the lower and upper limits of the random pitch spread.
- 5. Defines the maximum number of pitch increments a grain can move when in random mapping mode.
- 6. Toggle on to automatically scale the window sizes to maintain original relationship.
- 7. This parameter determines the size of the secondary windowing function.
- 8. Determines the size of the primary windowing function.
- 9. Introduce bandlimited noise to the playhead position.
- 10. Define the type of chord to use when in chordal pitch mapping mode.
- 11. Define the type of scale map to use when in scalar pitch mapping mode.
- 12. Select from random (see notes 2. and 3.), scalar, or chordal pitch mappings.

ffilter:

A bank of filters based on the fiddle~ and fffb~ objects. Allows you to tune the filters according to the spectral information of an input signal with independent gain and frequency control.



- 1. Choose which signal will be analyzed by the fiddle~ object.
- 2. Enable frequency information to be polled automatically at the defined time interval.
- 3. Poll the input signal to enter a new list of frequencies and relative gains.
- 4. Enter the time it takes for the frequencies to arrive at their new values once it has been polled.
- 5. Enter the time it takes for the grain values to arrive at their new values once it has been polled.
- 6. Adjust all frequency values in the list.
- 7. Adjust all gain values in the list.
- 8. View or adjust individual gain values with the multislider object.
- 9. View or adjust individual frequency values with the multislider object.
- 10. View or adjust individual frequency multipliers.
- 11. Enable or disable individual filters.

Convolve:

cross-synthesis of two signals using amplitude convolution. The module also includes two pannable fft-based spectral filters, a linear bin-shift effect, and a slider to control how much the signals are convolved.



- 1. Choose a pre-made buffer for filter one.
- 2. Choose a pre-made buffer for filter two.
- 3. Linearly shift the output bins to perform a non-harmonic pitch shift.
- 4. Click and drag on the spectrum to voom into a particular area.
- 5. Draw on the waveform object to shape the first filter.
- 6. Draw on the waveform object to shape the second filter.
- 7. Pan between the first and second filter states.
- 8. Pan from the unprocessed input signal on the left to a completely convolved output on the right.

Morph:

cross-synthesis of two signals using complex convolution. The module includes two pannable fft-based spectral filters, a linear bin-shift effect, and a slider to spectrally 'morph' between the two signals.



- 1. Choose a pre-made buffer for filter one.
- 2. Choose a pre-made buffer for filter two.
- 3. Linearly shift the output bins to perform a non-harmonic pitch shift.
- 4. Click and drag on the spectrum to zoom into a particular area.
- 5. Draw on the waveform object to shape the first filter.
- 6. Draw on the waveform object to shape the second filter.
- 7. Pan between the first and second filter states.
- 8. Pan from an unprocessed signal on the left or right to a completely convolved output in the middle.

Sample:

a basic sample playback module built around the MSP groove~ object. It includes various scale-based and tone-row pitch mappings as well as a selection quantizing and a windowed looping function.



1. Change the pitch/rate of playback by sample rate conversion.

- 2. Create a glissando pitch effect by smoothly moving from this pitch value to the base pitch value.
- 3. Sets the interpolation time for the glissando pitch effect.
- 4. Activate the sync-based triggering of pitch mapping.
- 5. Shift all pitch mappings up one octave.
- 6. Choose from various preset pitch mappings or enter in your own tone row.
- 7. Use this menu to set the base pitch or enter in a new tone row.
- 8. Determines the rate of triggering by setting the number of subdivisions of the sync signal.
- 9. Select from Scale (up or down the selected scale), Random (randomly play selected scale), or Pulse (uses the glissando pitch effect to create a continuous stream of pitch ramps).
- 10. Determines the range of pitch ramps in 'Pulse' pitch mapping mode.
- 11. Step forward or backwards through the quantized slices.
- 12. Choose between random and sequential slice ordering.
- 13. Displays the currently playing slice. You can also enter a number here to jump to that slice directly.

14. Determines the number of slices to create from the current selection.

Sync:

A sample loop player that is slaved to the main sync signal.



- 1. Enter in the number of beats (quarter notes) to set the sync signal rate multiplier.
- 2. Adjust the pitch (time-independant).
- 3. When initiated, the beats value will adjust itself according to the current selection's fractional relationship to the overall selection.
- 4. Displays the BPM value of the current selection according to the sync signal's BPM value and then number of beats of the selection.
- 5. Enter in a fraction or multiple of a quarter note to set the selection grid resolution.
- 6. Reset the position playhead to the sync signal.
- 7. Prevent the position playhead from becoming out of sync with the sync signal (disables the trigger button).
- 8. When initiated, the beats value will adjust itself according to the current quantization setting.
- 9. Enter the number of equal subdivisions of the current selection.
- 10. How many times each subdivision should be repeated before moving to the next.
- 11. Move through the sample in forward, reverse, or ping-pong motion.
- 12. Step through the slices linearly or randomly.
- 13. Displays the currently playing slice. You can also enter a number here to jump to that slice directly.
- 14. Turn on/off selection quantization (see note 5).

freezer:

A phase vocoder module that records spectral data into a buffer to allow independent pitch, time, and frame resynthesis order manipulations.



- 1. Generates ramped line-segments between successive frame selections.
- 2. Choose between manual, sequential, or random automation of frame selection.
- 3. When automation mode is on, choose between sync-signal or ms metro triggering.
- 4. Move slider to adjust vertical resolution.
- 5. Random slider determines the amount of deviation away from the current frame when in random mode.

Linein:

Used mainly for monitoring, routing, or applying effects to audio signals.



looper:

a buffer-based delay-line with variable input and feedback levels and an ADSR envelope for triggered recording.



MG Control: Managing control data for the processing modules in Meta_Gest player.

Overview:

navigating the main control interface. All control information is sent as a list of three elements, the first determines where the data should be routed, the second determines which parameter to control and the third is the actual control data.





Detail of buffer drawing window.

Sig_control:

Use an audio signal's envelope or attack as control data (listed as Amp#e or Amp#t respectively). The envelope follower uses the slide~ object for different smoothing of signal increases and decreases. The attack detector compares the current amplitude value with the average of a set number of preceding amplitudes.



Keyboard:

Use the computer's keyboard to send trigger control data. Sends of key-down, key-up, or both.



Routing:

The is where all the control data is received, scaled, and then sent to the processing modules.



1. Control Modules are listed with an abbreviation followed by the specific number (See below for abbreviation list).

2. Each controllable parameter has a unique number associated with it (See next page for full parameter list).

3. Different Mappings include floating point interpolation, MIDI note (according to a reference), and change (sends a bang whenever there is a change).

4. This variation allows any of the control module to adjust the scale low and high values on another controller map.

Control Module Abbreviation List.

B - Buffer-based automation control	BS - loop-based trigger control
M - MIDI control	BSs - Step Sequencer
R - Buffer-based record control	Amp#e - amplitude envelope
UC - Evolution UC33e fader control surface	Amp#t - amplitude threshold trigger
RH - Rhythm modules	Ky - computer keyboard control
BT - Beat trigger	Ft - Foot switch control
	Fc - Foot continuous controller

Full Parameter List:

0-none 1-target pitch 2-pulse speed 3-grain size 4-speed 5-range 6-step size 7-min pitch 8-max pitch 9-record 10-soundfile 11-convolution-pan 12-convolution-target 13-pulse on/off 14-trigger 15-gain 16-frequencyslewon/off 17-frequencyslewvalue 18-gainslewon/off 19-gain slewvalue 20-filtertype-pan

21-Channel fader 22- Channel pan 23-MIDI 24-multipan angle 25-multipan distance 26-multipan sub 27 multipan time 28 ePitchVol1 29 ePitchVol2 30 ePitchVol3 31 ePitchVol4 32 ePitchPitch1 33 ePitchPitch2 34 ePitchPitch3 35 ePitchPitch4 36 eFilterVol 37 eFilterFreq 38 eFilterGain 39 eFilterQ 40 eDelayVol 41 eDelayTime

42 eDelayFeedback 43 eDistortVol 44 eDistortInput 45 eDistortknee 46 eDistortcutoff 47 eRingVol 48 eRingVolRing 49 eRingModAmt 50 eRingIntTime 51 eRingOffset 52 ePlugVolL 53 ePlugVolR 54 ePlugP2 55 eReverbVol1 56 eReverbVol2 57 eReverbVol3 58 eReverbVol4 59 eReverbVol5 60 eReverbMute1 61 eReverbMute2 62 eReverbMute3

63 eReverbMute4 64 eReverbMute5 65 eReverbFre21 66 eReverbFre22 67 eReverbFre23 68 eReverbFre24 69 eReverbFre25 70 eReverbSize 71 eReverbDamp 72 edelay_trigger 73 buffer_clear 74 loopon_off

footy:

Input module for MIDI foot controller.

									 Individual foot switch buttons/display
									- Expression Pedal display
L I									
· •	input	output type	channe	1	out-low	out-hiat	_		 Quick routing module
	8	O-none	0	Float	Þ0.	⊳1.	DC-2	і Іг	 Preset save, recall, rename
Ft6 Ft7 Ft8 Ft9 Ft10	8	O-none	0	Float	Þ0.	⊳1.	⊳c-2		
	ø	O-none	0	Float	⊳ 0.	▶1.	▶C-2		
	8	0-none	0	Float	Þ0.	≥1.)C-2	└─┤	
Ft1 Ft2 Ft3 Ft4 Ft5	ø	O-none	0	Float	>0.	≥1.)¢-2		
	ø	O-none	0	Float	Þ0.	⊳ 1.)C-2		
	8	O-none	0	Float	Þ0.	≥1.)C-2		
FUIDFUI/FUIDFUIDFUZD	ø	O-none	0	Float	Þ0.	≥1.)C-2		
	ø	0-none	0	Float	≥o.	≥1.)C-2		
E+11 E+12 E+13 E+14 E+15	8	0-none	0	Float	Þ0.	Þ1.)C-2		
	8	0-none	0	Float	<u></u> >0.	▶1.) C-2		
	ø	0-none	0	Float	>0.	▶1.	▶C-2		
Ft26 Ft27 Ft28 Ft29 Ft30	8	O-none	0	Float	>0.	≥1.)C-2		
	8	O-none	0	Float	>0.	>1.	▶C-2		
	8	U-none	. 0	Float	>0.	<u>>1.</u>	▶C-2		
Ft21 Ft22 Ft23 Ft24 Ft25	8	U-none	U 0	Float	20.	<u>≥1.</u>	PU-2		
	8	U-none	0	Float	20.	P1.	pu-2		
	<i>»</i>	O-none		Float	20.				
FT36 FT37 FT38 FT39 FT40	8	U-none	0	Float	<u>20.</u>	<u>P1.</u>	PU-2		
	<i>n</i>	0-none 0-none	0	Float	<u>ko</u> .		0-2		
F+31 F+32 F+33 F324 F+35	<i>»</i>	0-none 0-none	0	Float	<u>No.</u>	K1	0-2		
	a a	0 none 0-none	0	Float	50.	51	DC-2		
ALL DLS Synth 1	м я	0-none	0	Float	<u>b</u> n	51	0.0-2		
default (л Я	0-none	0	Float	b0.	≥ 1 .)) C-2		
save	8	0-none	0	Float	>0.	▶1.))C-2		
	8	O-none	0	Float	>0.	▶1.	>C-2		
	8	O-none	0	Float	⊳ 0.	▶1.)2		
	ø	0-none	0	Float	⊳ 0.	▶1.	▶C-2		
	8	0-none	0	Float	⊳ 0.	⊳ 1.)C-2		
	8	0-none	0	Float	⊳ 0.	≥1.)C-2		
	8	0-none	0	Float	Þ0.	⊳1.)C-2		
	8	O-none	0	Float	Þ0.	≥1.)C-2		
	ø	O-none	0	Float	>0.	≥1.)C-2		
	8	0-none	0	Float	>0.	≥1.)C-2		
	8	O-none	0	Float	Þ0.	▶1.) C-2		
	ø	O-none	0	Float	⊳ 0.	▶1.)>C-2		
	ø	O-none	0	Float	Þ0.	≥1.)¢C-2		
	ø	O-none	0	Float	⊳ 0.	▶1.)C-2		
	ø	O-none	0	Float	>0.	≥1.)C-2		
	8	O-none	0	Float	<u>}0.</u>	▶1.)C-2		
	ø	O-none	Ö	Float	<u>}0.</u>	<u>⊳1.</u>) C-2		

Sequencer:

Various control modules to organize and trigger events in time. Modules include loop-based sequencers, sync signal-based sequencers, step sequencers, and a few idiosyncratic pitch/velocity generators.

	reset 8. beats	1 Pitch	1 Velocity
BS_2	reset 8. beats		
BS_3	reset 8. beats		
BS_4	reset 8. beats		
BS_5	reset 8. beats	m reset C rate 1. octave 0 scale	major Trigger X send_ve
BS_6	reset 8. beats	2 Pitch	2 Velocity
BS_7	reset 8. beats		
BS_B m	reset 8. beats		
BS_9	reset 8. beats		-
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1 Steps 1-16 0.25 rate/temp 16 number route res	6 et m reset C rate 1. octave 0 scale	major Trigger X send_ve
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1	⁶ 3 Pitch	3 Velocity
	Steps 1-16 0.25 rate/temp 16 number route res 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1	6	
m	Steps 1-16 0.25 rate/temp 16 number route res 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1	et 6	
m	Steps 1-16 0.25 rate/temp 16 number route res	t	
m	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1 Steps 1-16 0.25 rate/temp 16 number route rese	6 ^{et} m reset C rate 1. octave 0 scale	major Trigger X send_ve
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1 Stars 1-16 0 25 rste/term 16 number Fourte rest	⁶ 4 Pitch	4 Velocity
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 11	6	
m	Steps 1-16 0.25 rate/temp 16 number route rese 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1	et	
m	Steps 1-16 0.25 rate/temp 16 number route resu	et	
Syn		m reset C rate 1. octave 0 scale	major Trigger X send_ve
Beat-trig Restart a Step Seq Loop-ba	ger module. Enter in a rate relative to the global II the step-sequence modules. uencer module - enter parameters for number o sed sequencer - set the duration of the loop acco and then enter trigger impulses by clicking on t	sync-signal. f steps and resolution. ording to sync-signal beats, the square at the left.	
1 Pito	ch 1 Velocity	<	
		And and a second second second second	

Raise these sliders to adjust the amount each pitch or volume value can randomly deviate per cycle.
 Enter pitch and velocity values (intended for use with the 'Sample' module).

Faders:

Direct input for MIDI control surfaces such as the Evolution UC-33e.



Display for knob values.

MIDI_in:

A MIDI input module for note, velocity, pitchbend, and controller inputs.

M1 none 🔻 keyboard range	M2 none 🔻 keyboard range	M3 none 🔻 keyboard range	Toggle for expanded keyboard view (see below).
control Nb. 0 A0 <> B7	control Nb. 0 AO <> B7	control Nb. 0 AO	Select the type of MIDI input.
control num/val. clear select all	control num/val.	control num∕va clear select all ←	Select a range of note input values (see below).
▶0 / ▶0 input ▶0 ▶127	▶0 / ≥0 input ≥0 ≥127	0 / ≥0 input ≥0 ≥127	In Controller mode, set the controller number.
M4 none 🔻 keyboard range	M5 none 🔻 keyboard range	M6 none 🔽 keyboard range	
control Nb. 0 AO <> B7	control Nb. 0 AO <> B7	control Nb. 0 AO <> B7	
control num/val. clear select all	control num/val.	control num/val.	
▶0 / ▶0 input ▶0 ▶127	▶0 / ≥0 input ≥0 ≥127	▶0 / ▶0 input ▶0 ▶127	
default			
save			

M1 none 🔽 keyboard range	M2 none 🔻 keyboard range M3 none 🔻 keyboard range
control Nb. 0 AO <> B7	control Nb. 0 C3 <> F4 control Nb. 0 A0 <> B7
control num/val.	clear select all control num/val.
▶0 / ▶0 input ▶0 ▶127	▶0 / ▶0 input ▶0 ▶127 ▶0 / ▶0 input ▶0 ▶127
M4 none 🔽 keyboard range	
control Nb. 0 AO <> B7	
control num/val. clear select all	control num/val.
▶0 / ▶0 input ▶0 ▶127	▶0 / ▶0 input ▶0 ▶127 ▶0 / ▶0 input ▶0 ▶127
default	
save	

-Select the note values that you want to monitor, (selecting 'MIDI' in the Routing Module will output direct MIDI note values, otherwise the output will automatically be scaled from 0.0 to 1.0).

Rhythms:

Another control module for holding trigger events. This module, however, is not slaved to the master sync signal.



Trigger the rhythmic sequence.

- click on the 'T' button to enter in a new sequence

- Erase the current sequence and start a new one. Times are not recorded until the first trigger is received.

Delta_preset:

Use the Footy input to change a preset setting in any of the other control modules.



32_grains:

This pieces uses four granular and two sample record/playback modules. The sequencer triggers recording in the granular and sampler modules, and can be turned on and off with the foot controller.



Channel 1 - Monitor for ADC channel 1.

Channel 2 - Granular process with speed parameter set to '0' and randomized playhead position within the recorded sample.

Channel 3 - Granular process with speed/rate parameter set to 0.06.

Channel 4 - Sync Module for sample playback down one octave.

Channel 5 - Monitor for ADC channel 2.

Channel 6 - Granular process with speed parameter set to '0' and randomized playhead position within the recorded sample (larger grain size than Ch. 2 for variation in texture)

Channel 7 - Granular process with speed/rate parameter set to -0.06.

Channel 8 - Sync Module for sample playback down two octaves.

Detail of the Routing Preset for 32_grains.

B1	24-multinan angle	3	Float	b0.05	bn 49	DC-2	D
B2	24-multinan angle	7	Float	0.00	0.19	bc-2	Routina
BZ	22- Channel nan	T T	Float	5-100	N100	NC-2	
D4	22 Channel pan	4	Float	K-100	K100	Ko-2	sarc
D5	57 oBlueVelD	- -	Float	K. 100.	150	Ko-2	save
DU DC1	50 eP lugvolk 59 ePeverbVel5	0	Float	Ko.	K 1 JO.	NC-2	
DOT DCO	60 eReverbyute1	0	Float	Ko.	K¦:	NO-2	
B52 D07	60 eReverbinuter	U O	Float	Ko.	K!		
855	61 eReverbinuteZ	U	Float	NO.	KI:		
854	62 eReverbMutes	U	Float	20.	R1.	pu-z	
855	63 eReverbMute4	U	Float	20.	R1.	20-2	
B11	9-record	2	Float	PU.	P1.	PC-2	
BT2	9-record	3	Float	20.	P1.	≥C-2	
BT3	9-record	4	Float	≥0.	₽1. 	¢C-2	
UC1	21-Channel fader	0	Float	≥ 0.	▶158.	>C-2	
UC2	21-Channel fader	0	Float	Þ0.	≥158.	¢C-2	
UC3	21-Channel fader	3	Float	Þ0.	158.	¢C-2	
UC4	21-Channel fader	4	Float	⊳ 0.	▶158.	DC-2	
BSs1	9-record	6	Float	Þ0.	⊳ 1.	¢C-2	
BSs2	9-record	7	Float	Þ0.	⊳ 1.	DC-2	
BSs3	9-record	8	Float	⊳ o.	⊳ 1.	DC-2	
UC5	21-Channel fader	0	Float	⊳ o.	▶158.	DC-2	
UC6	21-Channel fader	0	Float	⊳ o.	▶158.	DC-2	
UC7	21-Channel fader	7	Float	⊳ o.	▶158.	DC-2	
UC8	21-Channel fader	8	Float	⊳o .	▶158.	DC-2	
UC11	1-target pitch	2	Float	bo	b1.	bc-2	
UC12	1-target pitch	3	Float	5o.	b1.	bc-2	
UC15	1-target pitch	6	Float	⊳ o.	⊳ 1.	>C-2	
UC16	1-target pitch	7	Float	⊳ 0.	⊳ 1.	>C-2	
UC35	25-multipan distance	3	Float	⊳ 0.	b1.	¢C-2	
UC10	25-multipan distance	7	Float	5o.	6 1.)C-2	
Kv1	73 buffer_clear	2	Float	50.	b10.	bc-2	
Kv2	73 buffer_clear	3	Float	50.	b10.	bc-2	
KVZ	73 buffer clear	4	Float	Kn.	b10	5C-2	
Ku4	73 buffer clear	6	Float	Ko.	b 10	5C-2	
Ku5	73 buffer clear	7	Float	50	510	50-2	
KU6	73 buffer clear	e	Float	50. 50	510.	50-2	
a		ů n	Float	50. 50	N10	bc-2	
a		o o	Float	Ko.	K10.	No-2	
a a	0-none O-popo	0	Float	Ko.	K10.	K0-2	
8	O-none O-none	0	Float	Ko.	K10.	NO-2	
8	O-none O-none	0	Float	Ko.	K10.	NC-2	
8	O-none O-none	0	Float	Ko.	K10.	NO-2	
<i>N</i>	0-1010	Ö	Float	K	K 10.		
<i>N</i>	O-none	U O	Float	Ko.	K 10.		
8	O-none	0	Float	Ko	K10.		
	U-none	U	Float	PU.	P 10.	PU-2	
8	O-none	0	Float	20.	K 10.		
	U-none	0	Float	K0.	K!	20-2	
8	U-hone	0	Float	PU	<u> </u>	20-2	
8	U-none	0	Float	20.	<u>[]</u>	PC-2	
8	U-none	0	Float	20.	P1	PC-2	
8	U-none	0	Float	Þ0.	P1.	¢C-2	
ø	O-none	0	Float	Þ0.	≥1	>C-2	
8	O-none	0	Float	> 0.	≥1.	>C-2	
ø	O-none	0	Float	Þ0.	▶1.	DC-2	

Control buffers 1-4 determine the angle and
distance for 8 channel panning.
BT modules 1-3 trigger recording on channels
2, 3, and 4.
BSs(Beat step-sequencer) modules trigger
recording on channels 6, 7, and 8.
UC33e faders 1-8 control volume on channels

1-8. Keyboard controls 1-6 clear buffer contents. UC33e rotary knobs 11, 12, 15, and 16 adjust

pitch on granular processes.

Freezing_pitch:

This piece uses the freezer module to create rhythmic patterns out of melodic material recorded in real-time. There are also two granular modules that generate a drone that slowly shifts between a perfect 5th, minor 3rd, and a unison relative to the melody being played.



Channel 1 - Granular drone process.

Channel 2 - Freezer Module with position parameter determined by sync signal triggering.

Channel 3 - Freezer Module with position parameter determined by sync signal triggering.

Channel 4 - Granular drone process.

Channel 5 - Freezer Module with position parameter determined by sync signal triggering.

Channel 6 - Freezer Module with position parameter determined by sync signal triggering.

Channel 7 - Monitor for ADC channel 1.

Channel 8 - Monitor for ADC channel 2.

Feeding_pitch:

This piece uses a live input signal with pitch-shifting and feedback, convolution with a pre-recorded sample, and a granular drone.



Channel 1 - Line_in module performs the initial pitched feedback (see effects detail on next page).

- Channel 2 Another Line_in module performs a second pitch shifting process.
- Channel 3 Convolution of the input signal with a audio from Channel 5.
- Channel 5 Pre-recorded sample playback.
- Channel 6 Granular Drone.



Feeding_pitch effects channel detail.

Audio input is routed first to a filter module, then to the pitch shift, which is routed both to the compressor and back into itself to create a controlled feedback loop. The Compressed signal is then routed to the output and into a reverb module, which is also routed to the output.

combing_beats:

This is an example of a real-time multi-channel diffusion of a sequenced rhythmic pattern.

None	LOOP low_clamp	2.aif new folder	aux1 🔹	LOOD bang	g.aif	new folder	None 💌 🔻	
sample 🔻	twom	loop 5063 quant 5063	sample ▼ 5	twom save A	loop 237	quant 237	None 🔻	
2ch	2000. L record rT	ם ۲ ۲	2ch	2000. L record rT		0 × F	2ch	
5	8.94 -12 1. 0. 900 chr	pulse 0. 8. Fw rom up 1 0 step	s	0.84 -12 0. 900	1. pulse O. chrom up 1	8. Fw 0 step	5	
ado1	LOOD low_clamp	2.aif new folder	aux1 🔻	Sync ⁷⁰ ₅	Ballad Drums 01.aif	new folder	None 🔻	
sample	twom	loop 5063 quant 5063	sync •	twom T save A	ر 100p 8726	quant 2908	None	
Sch	2000. L		8ch	8727. S			2ch	
	212 1.	pulse 0. 8. Fw		4. beats	link 1. res au	3 8 Rev		
5	0. 900 chr	rom up 1 0 step	5	1. pitch	27.5 bpm quant none	0 step	5	
aux1 🔹	LOOD knocking.ai	f new folder	aux2 🔻	Sync 70s	Ballad Drums 01.aif	new folder	None 🔻	
3 8ch	twom save A 2000. L record rT	loop 881 quant 6	Sync 7	twom save A 4363. S record SI		quant 545	11 2ch	
•	112 1. 0. 900 chr	pulse 0. 0.83 Rev rom up 128 0 step	• 5	1.5 beats 1. pitch	link 1. res au 20.6 bpm quant none	8 8 Fw 0 step	s	
aux1 🔷 🔻	LOOD low_clamp	2.aif new folder	adc3 🔷 🔻	Svnc ^{70s}	Ballad Drums 01.aif	new folder	None 💌 💌	
sample 🔻	twom T save A	loop 5063 quant 39	sync ▼ 8	default save A	loop 1999	quant 249	None None	
8ch	2000. L		Sch	2000. S record SI			2ch	
	412 1. 0. 900 chr	pulse 0. 0.83 Rev.		1. beats 0. pitch	link 1. res au 30.0 bpm quant none	8 8 Fw	s	
1234	567891011	1213141516 12	3456	78	Tem	no/Sync		
		74			BPM 110.	JU/ Sync		
		audio			∆T 545.4546	14 7.4 1994		
dac~ 1 2 3	4 5 6 7 8 aux_send	twomelodiesb save	reco	rd	Δt 2m 4363.636	Δt 3m 854 572 Δt 4m 872	7.27344 =	dur 0.

Channels 1-5 - Sample playback modules for the individual audio files.

Channel 6 - Sample playback module that records channels 1-3 and pitches the result up two octaves.

Channel 7 - Sample playback module that records channels 4 and 5 and pitches the result up two octaves.

Channel 8 - Extra sample playback module for recording live audio into the sequence.

Combing_beats sequencer detail:

BeatSync twomelodies		
BS_1 reset 8. beats	1 Pitch	1 Velocity °
BS_2 m reset 8. beats		-
BS_3 m reset 8. beats		
BS_4		_
BS_5		
BS_6	m reset C rate 0.25 octave 7 sea	le pentatonic (min) trigger x send_velocity 2 Volocity 0
BS_7		2 Velocity
m reset 8. beats		-
reset 8. beats		1
m reset 8. beats		
	2 13 14 15 16	
m Steps 1-16 0.125 rate/temp 48 number	2 13 14 15 16 3 Ditch	le pentatonic (maj) trigger X send_velocity 3 Velocity 0
m Steps 1-16 0.5 rate/temp 32 number	route reset	3 Velocity
	2 13 14 15 16	-
1 2 3 4 5 6 7 8 9 10 11 17	2 13 14 15 16	
m Steps 1-16 0.5 rate/temp 16 number	route reset	-
1 2 3 4 5 6 7 8 9 10 11 12 Steps 1-16 0 25 rate/temp 48 pumber	2 13 14 15 16	
	2 13 14 15 16 4 Pitch	4 Velocity
m Steps 1-16 0.25 rate/temp 16 number	route reset	1 Velocity
m Steps 1-16 0.5 rate/temp 32 number	2 13 14 15 16	-
	2 13 14 15 16	
m Steps 1-16 0.25 rate/temp 16 number	route reset	
32. 16. 22. 1. 1. 1.	1. 1.	re pentatonio (maj) · j drigger A send_veločity

Pitch modules 2-4 adjust pitch on Channels 2-4. See next page for more details on the routing.

Combing_beats effects detail:



Audio from this Channel (2) is routed to a tap delay module, then to a comb filter tuned to a major chord. The result it send to a filter, then compressed and finally pitched down two octaves.

Combing_beats routing detail:

B1	1-target pitch	1	Float	>3.5	⊳ 14.	DC-2	Dautina
B2	1-target pitch	2	Float	⊳-18.	87.	DC-2	Routing
B3	22- Channel pan	3	Float	⊳-100.	▶100.	DC-2	twomolodios
B4	22- Channel pan	4	Float	>-100.	▶100.	DC-2	save
B5	53 ePlugVolR	0	Float	Þ0.	▶158.	DC-2	Sare
BS1	59 eReverbVol5	0	Float	⊳ o.	⊳ 1.	DC-2	
BS2	60 eReverbMute1	0	Float	Þ0.	⊳ 1.	DC-2	
BS3	61 eReverbMute2	0	Float	Þ0.	⊳ 1.	DC-2	
BS4	62 eReverbMute3	0	Float	Þ0.	⊳ 1.	DC-2	
BS5	63 eReverbMute4	0	Float	⊳ o.	⊳ 1.	DC-2	
BT2	9-record	7	Float	Þ0.	⊳ 1.	DC-2	
BT1	9-record	6	Float	⊳ o.	⊳ 1.	DC-2	
Ky1	5-range	BS	Float	⊳ 16.	▶16.)C-2	
Ky2	5-range	BS	Float	⊳ o.	⊳ o.	DC-2	
КуЗ	5-range	BS	Float	≥16.	▶16.	DC-2	
Ky4	5-range	BS	Float	⊳ o.	⊳ o.	DC-2	
КуБ	5-range	BS	Float	36.	≥36.	DC-2	
Куб	5-range	BS	Float	⊳ 40.	▶40.	DC-2	
BSs2	4-speed	0	Float	>-3.	≥20.	DC-2	
BSs3	4-speed	0	Float	⊳-3.	Þ20.	DC-2	
UC1	21-Channel fader	1	Float	⊳ o.	▶158.	DC-2	
UC2	21-Channel fader	2	Float	Þ0.	▶158.	DC-2	
UC3	21-Channel fader	3	Float	Þ0.	▶158.	DC-2	
UC4	21-Channel fader	4	Float	⊳ o.	▶158.	DC-2	
UC5	21-Channel fader	5	Float	⊳ o.	▶158.	DC-2	
UC6	21-Channel fader	6	Float	Þ0.	▶158.	DC-2	
UC7	21-Channel fader	7	Float	⊳ o.	▶158.	DC-2	
UC8	21-Channel fader	8	Float	Þ0.	▶158.	DC-2	
UC35	37 eFilterFreq	3	Float	502.	6068.	DC-2	
UC10	24-multipan angle	1	Float	Þ0.	⊳ 1.	DC-2	
UC11	24-multipan angle	2	Float	⊳ o.	⊳ 1.	DC-2	
UC12	24-multipan angle	3	Float	⊳ o.	⊳ 1.	DC-2	
UC13	24-multipan angle	4	Float	⊳ 0.	⊳ 1.	¢C-2	
UC14	24-multipan angle	5	Float	⊳ 0.	⊳ 1.	DC-2	
UC15	24-multipan angle	6	Float	⊳ o.	⊳ 1.	DC-2	
UC16	24-multipan angle	7	Float	⊳ o.	⊳ 1.	DC-2	
UC17	24-multipan angle	8	Float	⊳ 0.	⊳ 1.	DC-2	
UC18	25-multipan distance	1	Float	⊳ o.	⊳ 1.	DC-2	
UC19	25-multipan distance	2	Float	⊳ 0.	⊳ 1.	DC-2	
UC20	25-multipan distance	3	Float	Þ0.	⊳ 1.	DC-2	
UC21	25-multipan distance	4	Float	Þ0.	⊳ 1.	DC-2	
UC22	25-multipan distance	5	Float	Þ0.	⊳ 1.	DC-2	
UC23	25-multipan distance	6	Float	Þ0.	⊳ 1.	DC-2	
UC24	25-multipan distance	7	Float	Þ0.	⊳ 1.	DC-2	
UC25	25-multipan distance	8	Float	Þ0.	⊳ 1.	DC-2	
8	O-none	0	Float	⊳ 0.	▶10.	DC-2	
8	O-none	0	Float	Þ0.	≥ 1.	¢C-2	
ø	O-none	0	Float	>0.	≥1.	>C-2	
ø	O-none	0	Float	Þ0.	≥ 1.)C-2	
ø	O-none	0	Float	Þ0.	≥1.	DC-2	
ø	O-none	0	Float	Þ0.	≥ 1.	¢C-2	
ø	O-none	0	Float	> 0.	≥ 1.)C-2	
8	O-none	0	Float	> 0.	≥1.)C-2	
ø	0-none	0	Float) 0.	₽ 1.	DC-2	