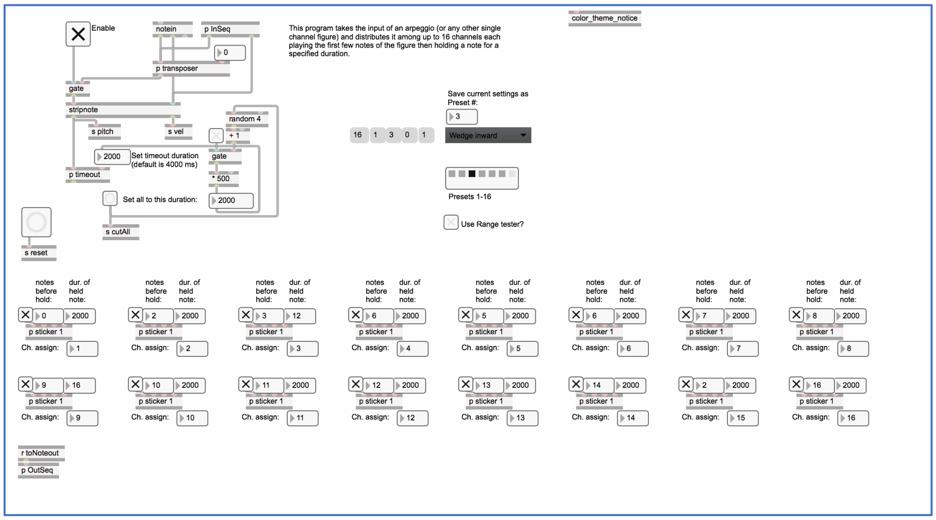
Thumbnails of composition utilities

Arp. orchestrator

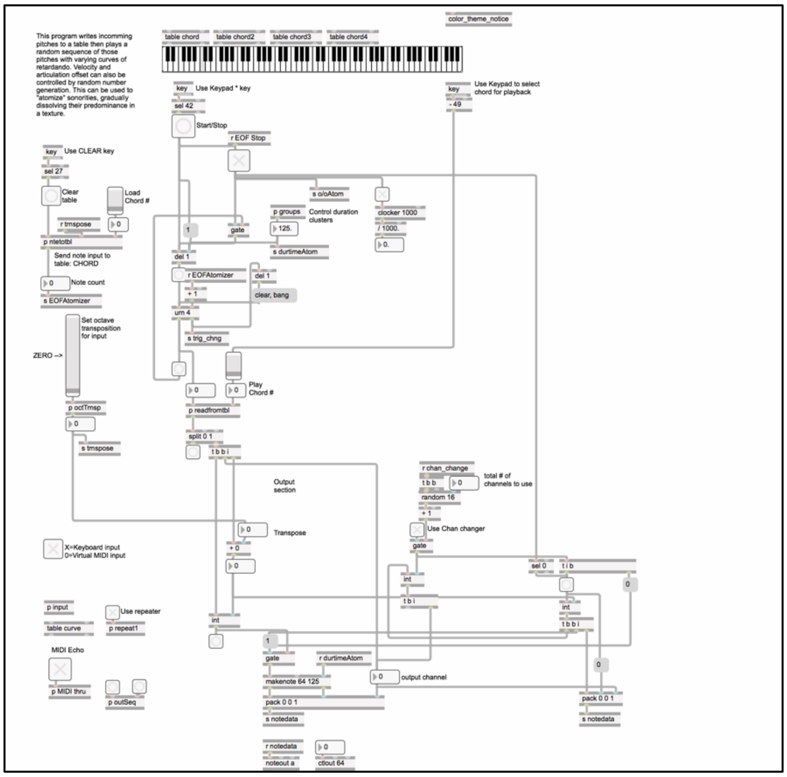
This program takes the input of an arpeggio (or any other single channel figure) played on a MIDI keyboard or from a MIDI sequence and distributes it among up to 16 channels each playing the first few notes of the figure then holding a note for a specified duration to build sustaining ensemble chords from shared arpeggiations.



This program takes the input of an arpeggio (or any other single channel figure) and distributes it among up to 16 channels each playing the first few notes of the figure then holding a note for a specified duration.

Atomizer_v3

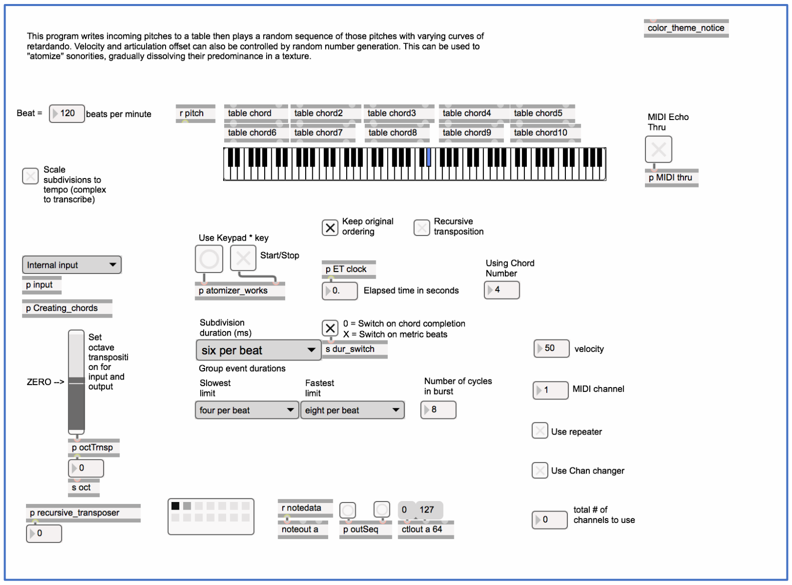
This program writes incoming pitches to a table then plays a random sequence of those pitches with varying curves of retardando. Velocity and articulation offset can also be controlled by random number generation. This can be used to "atomize" sonorities, gradually dissolving their predominance in a texture.



This program writes incomming pitches to a table then plays a random sequence of those pitches with varying curves of retardando. Velocity and articulation offset can also be controlled by random number generation. This can be used to "atomize" sonorities, gradually dissolving their predominance in a texture.

Burst.v.1

This program writes incoming pitches to a table then plays a random sequence of those pitches with varying curves of retardando. Velocity and articulation offset can also be controlled by random number generation. This can be used to "atomize" sonorities, gradually dissolving their predominance in a texture.



This program writes incoming pitches to a table then plays a random sequence of those pitches with varying curves of retardando. Velocity and articulation offset can also be controlled by random number generation. This can be used to "atomize" sonorities, gradually dissolving their predominance in a texture.

Beat = 120 beats per minute

☒ Keep original ordering

☐ Recursive transposition

Using Chord Number 4

Subdivision duration (ms): six per beat

Slowest limit: four per beat

Fastest limit: eight per beat

Number of cycles in burst 8

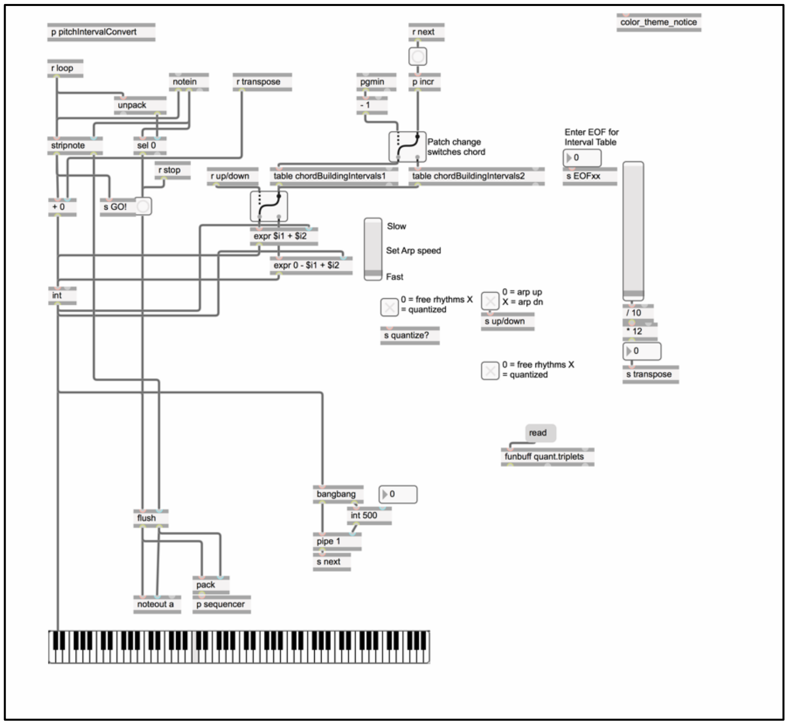
50 velocity

1 MIDI channel

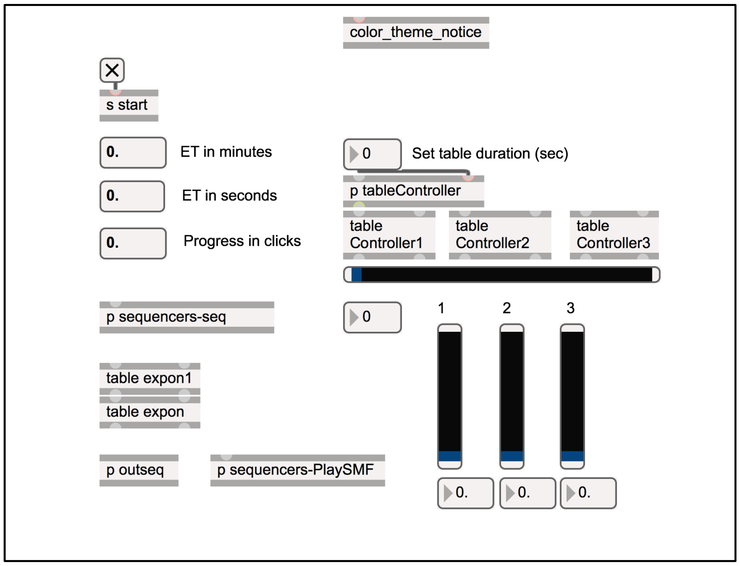
0 total # of channels to use

Chord Builder 3-1

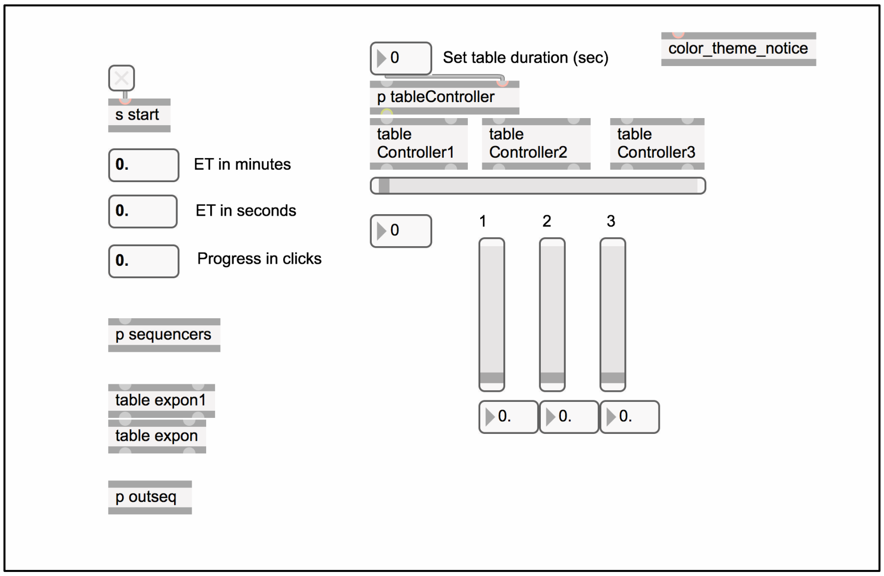
This program builds a chord from an input pitch using stored tables of intervals defining the chords to be built.



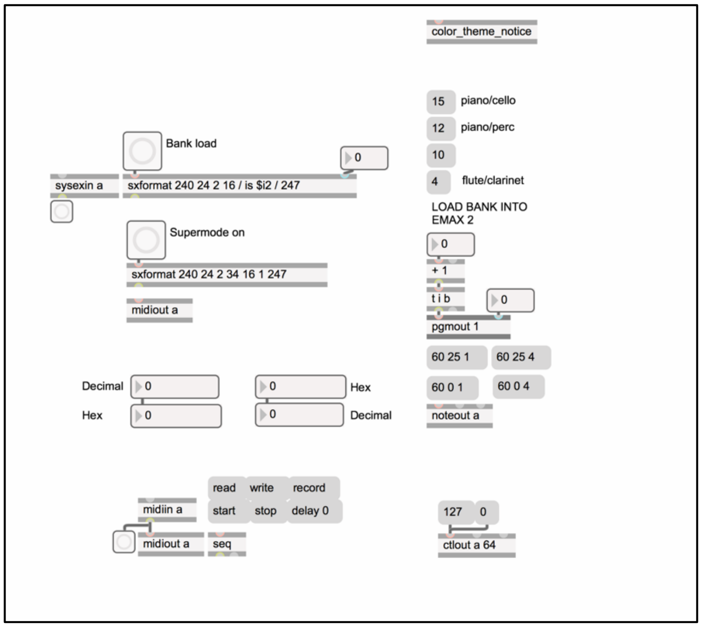
Controlling sequence playback with tables



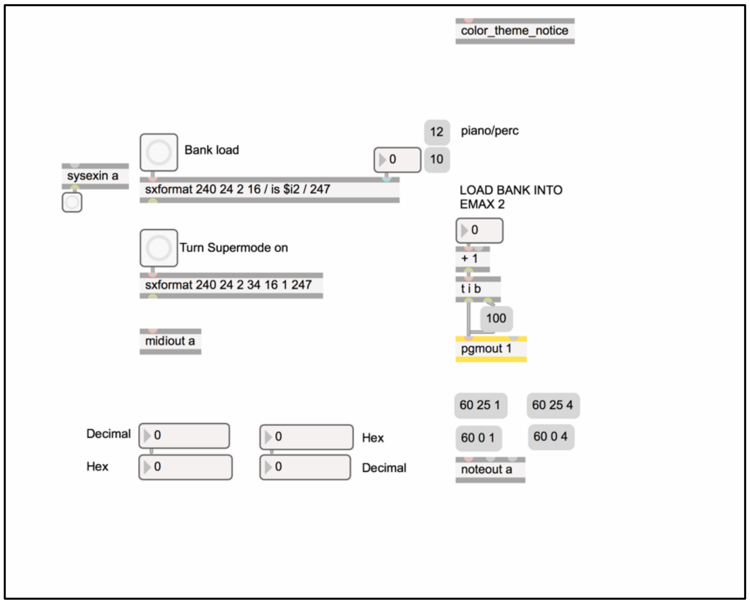
Dynamic ratios



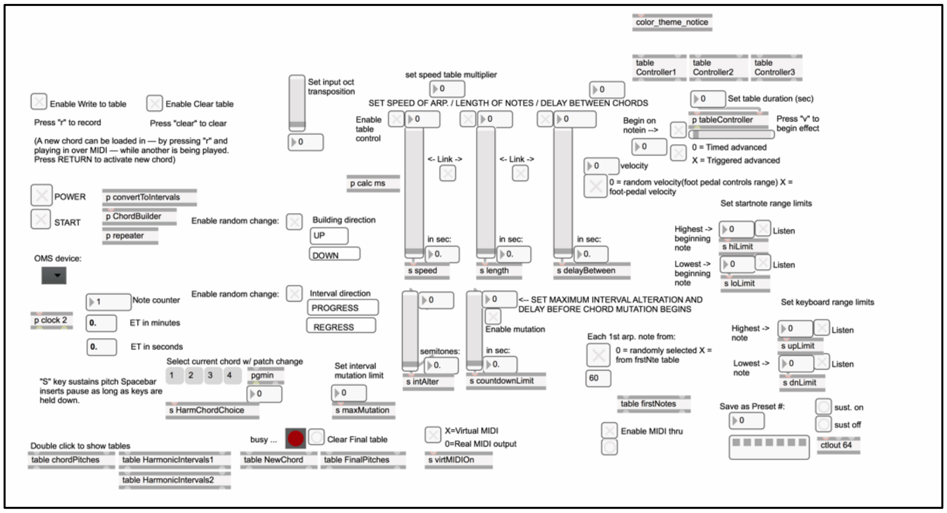
EMAX control



EMAX sysex control



Harmony Tree gen.v5.6



Press "r" to record

Press "clear" to clear

(A new chord can be loaded in — by pressing "r" and playing in over MIDI — while another is being played. Press RETURN to activate new chord)

Press "v" to begin effect

0 = Timed advanced
X = Triggered advanced

0 = random velocity(foot pedal controls range) X = foot-pedal velocity

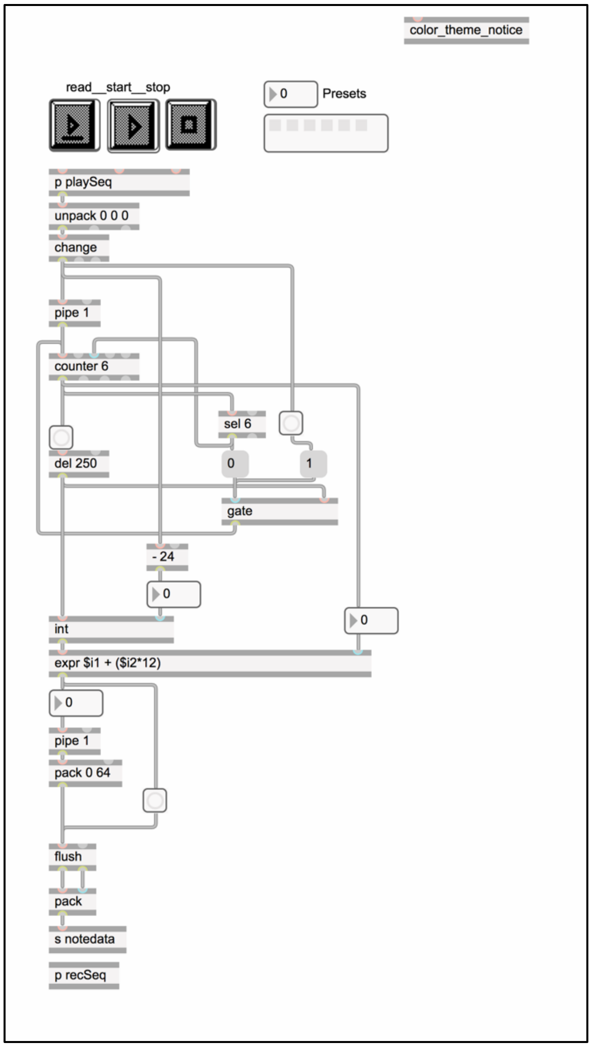
<-- SET MAXIMUM INTERVAL ALTERATION AND DELAY BEFORE CHORD MUTATION BEGINS

0 = randomly selected X = from frstNte table

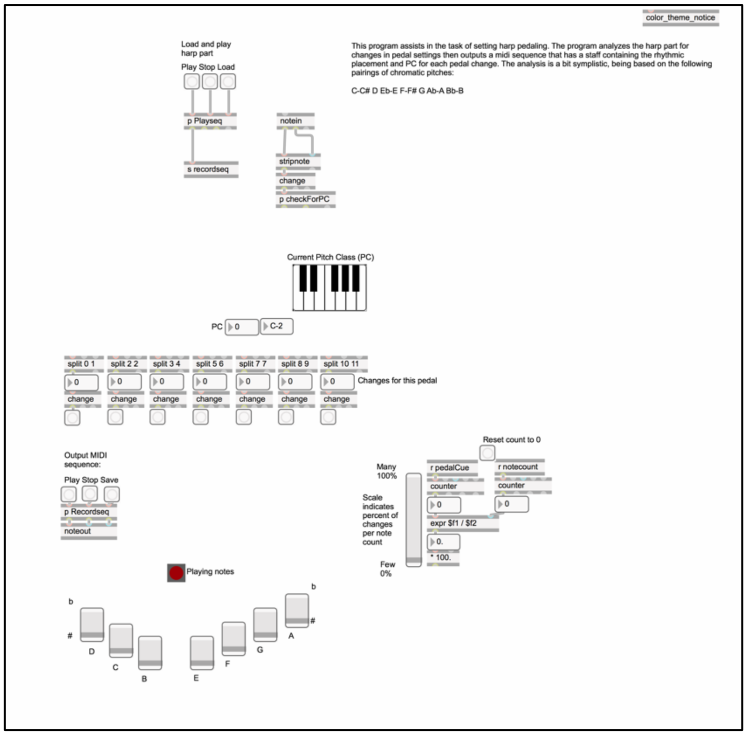
"S" key sustains pitch Spacebar inserts pause as long as keys are held down.

X=Virtual MIDI
0=Real MIDI output

Harp octave arp maker



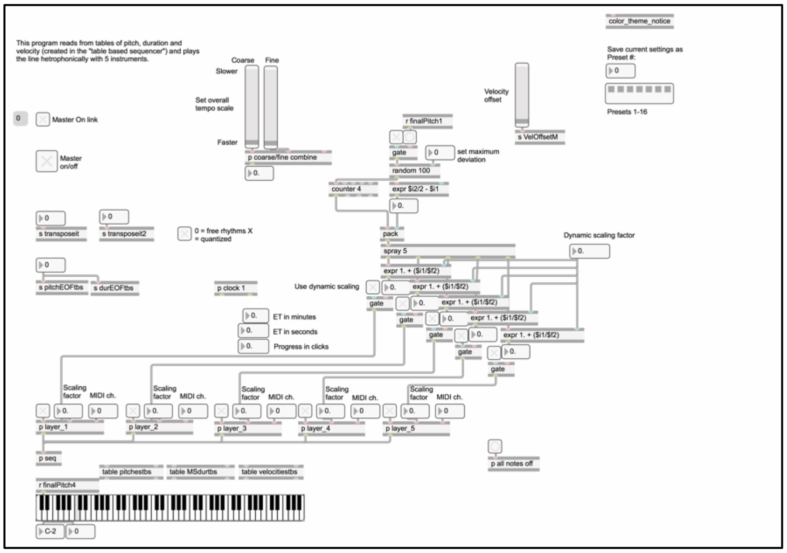
Harp pedaling assistant


This program assists in the task of setting harp pedaling. The program analyzes the harp part for changes in pedal settings then outputs a midi sequence that has a staff containing the rhythmic placement and PC for each pedal change. The analysis is a bit symplistic, being based on the following pairings of chromatic pitches:

C-C# D Eb-E F-F# G Ab-A Bb-B

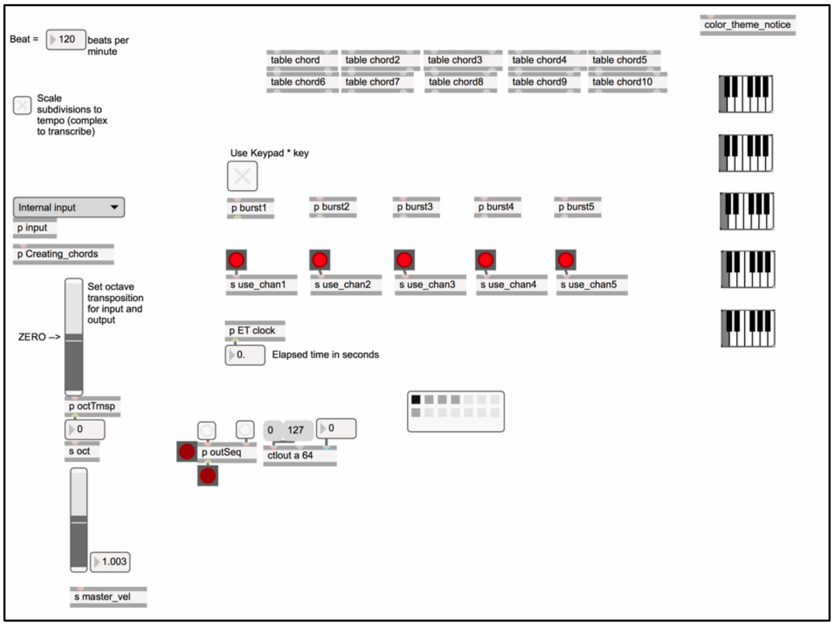
Hetrophonizer

This program reads from tables of pitch, duration and velocity (created in the "table-based sequencer") and plays the line hetrophonically with 5 instruments.


This program reads from tables of pitch, duration and velocity (created in the "table based sequencer") and plays the line hetrophonically with 5 instruments.

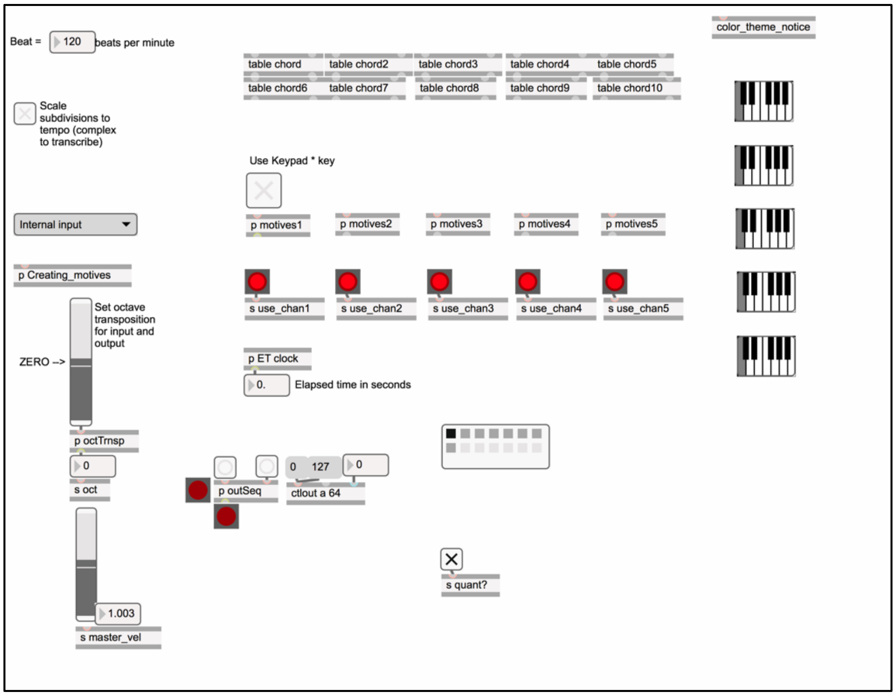
Multi_bursts.v.1

This program uses a set of five "burst" modules and writes incoming pitches to a table then plays a random sequence of those pitches with varying curves of retardando. Velocity and articulation offset can also be controlled by random number generation. This can be used to "atomize" sonorities, gradually dissolving their predominance in a texture.



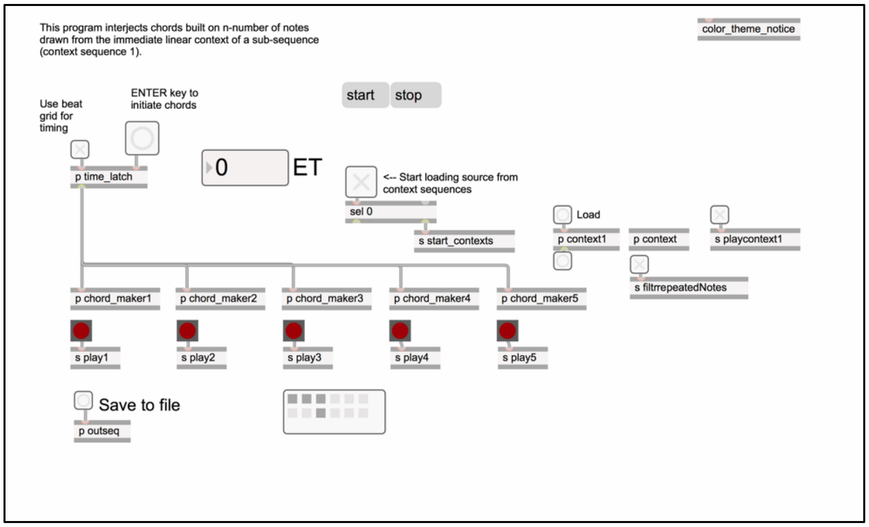
Multi_motives.v.1

This program uses a set of five "motive" modules and writes incoming pitches to a table then plays a random sequence of those pitches with varying curves of retardando. Velocity and articulation offset can also be controlled by random number generation. This can be used to "atomize" sonorities, gradually dissolving their predominance in a texture.



Multi.chords.from.context.v.1

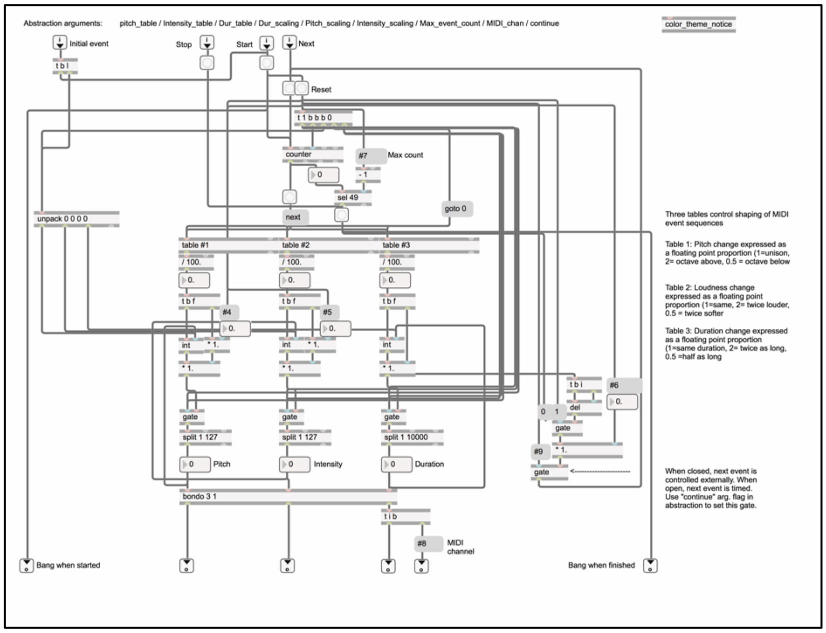
This program interjects chords built on n-number of notes drawn from the immediate linear context of a sub-sequence (context sequence 1).



This program interjects chords built on n-number of notes drawn from the immediate linear context of a sub-sequence (context sequence 1).

Nested_curves_proportions

Three tables control shaping of MIDI event sequences.
Table 1: Pitch change expressed as a floating point proportion (1=unison, 2= octave above, 0.5 = octave below
Table 2: Loudness change expressed as a floating point proportion (1=same, 2= twice louder, 0.5 = twice softer
Table 3: Duration change expressed as a floating point proportion (1=same duration, 2= twice as long, 0.5 =half as long
When closed, next event is controlled externally. When open, next event is timed. Use "continue" arg. flag in abstraction to set this gate.



Abstraction arguments: pitch_table / Intensity_table / Dur_table / Dur_scaling / Pitch_scaling / Intensity_scaling / Max_event_count / MIDI_chan / continue

Three tables control shaping of MIDI event sequences

Table 1: Pitch change expressed as a floating point proportion (1=unison, 2= octave above, 0.5 = octave below

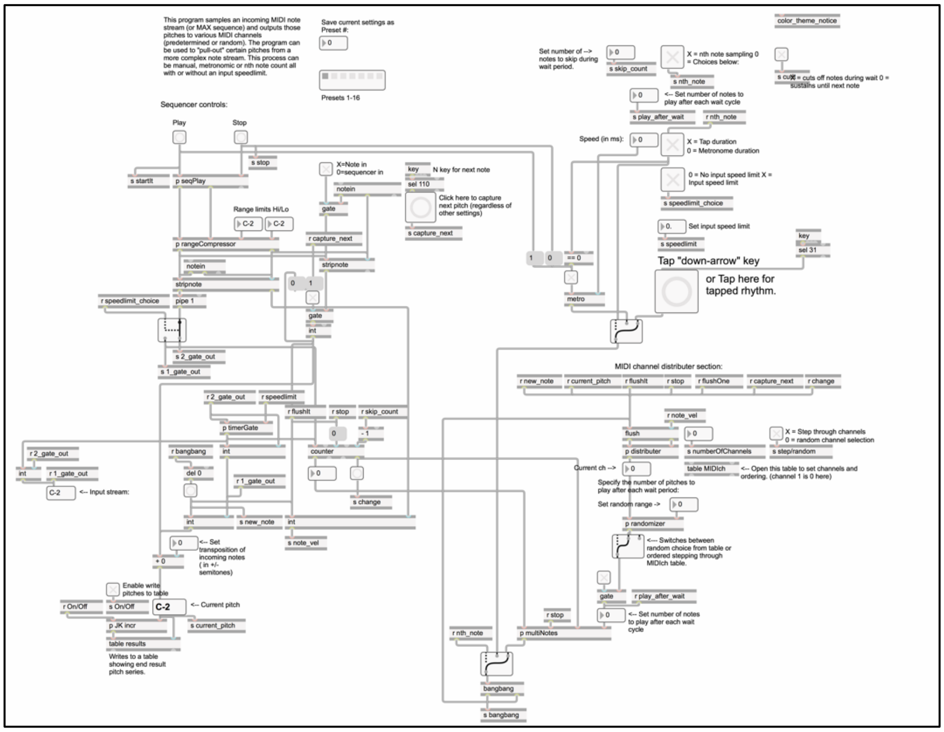
Table 2: Loudness change expressed as a floating point proportion (1=same, 2= twice louder, 0.5 = twice softer

Table 3: Duration change expressed as a floating point proportion (1=same duration, 2= twice as long, 0.5 =half as long

When closed, next event is controlled externally. When open, next event is timed. Use "continue" arg. flag in abstraction to set this gate.

Note sampler v.3

This program samples an incoming MIDI note stream (or MAX sequence) and outputs those pitches to various MIDI channels (predetermined or random). The program can be used to "pull-out" certain pitches from a more complex note stream. This process can be manual, metronomic or nth note count all with or without an input speed limit.

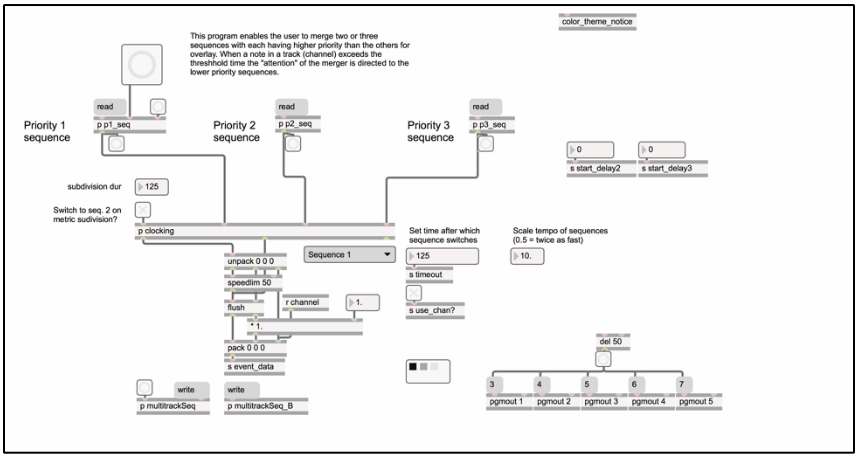


This program samples an incoming MIDI note stream (or MAX sequence) and outputs those pitches to various MIDI channels (predetermined or random). The program can be used to "pull-out" certain pitches from a more complex note stream. This process can be manual, metronomic or nth note count all with or without an input speedlimit.

Save current settings as Preset #:
Presets 1-16
Sequencer controls:
Play
Stop
s startIt
p seqPlay
s stop
X=Note in 0=sequencer in
notein
key
sel 110
N key for next note
Click here to capture next pitch (regardless of other settings)
s capture_next
Range limits Hi/Lo
C-2
p rangeCompressor
r capture_next
stripnote
pipe 1
r speedlimit_choice
gate
int
s 2_gate_out
s 1_gate_out
color_theme_notice
Set number of --> notes to skip during wait period.
s skip_count
X = nth note sampling 0 = Choices below:
s nth_note
X = cuts off notes during wait 0 = sustains until next note
<-- Set number of notes to play after each wait cycle
s play_after_wait
r nth_note
Speed (in ms):
X = Tap duration 0 = Metronome duration
0 = No input speed limit X = Input speed limit
s speedlimit_choice
Set input speed limit
s speedlimit
Tap "down-arrow" key
or Tap here for tapped rhythm.
sel 31
metro
MIDI channel distributer section:
r new_note
r current_pitch
r flushIt
r stop
r flushOne
r capture_next
r change
r note_vel
flush
p distributer
s numberOfChannels
X = Step through channels 0 = random channel selection
s step/random
Current ch -->
table MIDIch
<-- Open this table to set channels and ordering. (channel 1 is 0 here)
Specify the number of pitches to play after each wait period:
Set random range ->
p randomizer
<-- Switches between random choice from table or ordered stepping through MIDIch table.
r play_after_wait
<-- Set number of notes to play after each wait cycle
r stop
p multiNotes
r nth_note
bangbang
s bangbang
r 2_gate_out
r speedlimit
p timerGate
r flushIt
r skip_count
counter
s change
r bangbang
del 0
r 1_gate_out
s new_note
s note_vel
r 2_gate_out
r 1_gate_out
<-- Input stream:
<-- Set transposition of incoming notes (in +/- semitones)
+ 0
Enable write pitches to table
r On/Off
s On/Off
<-- Current pitch
p JK incr
s current_pitch
table results
Writes to a table showing end result pitch series.

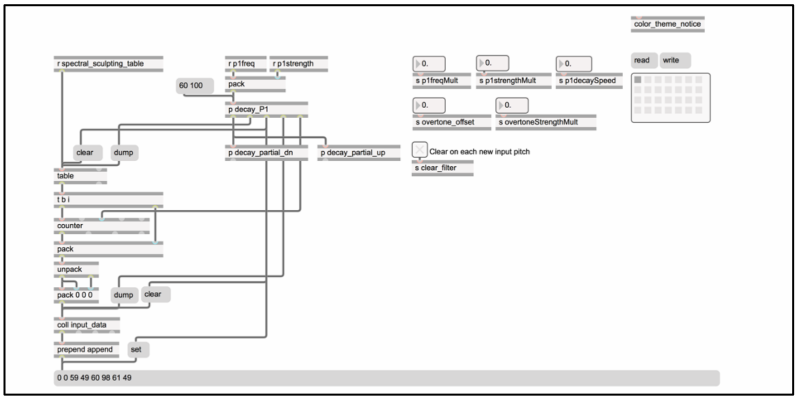
Overlay merging v3.0 (push-pull)

This program enables the user to merge two or three sequences with each having higher priority than the others for overlay. When a note in a track (channel) exceeds the threshold time the "attention" of the merger is directed to the lower priority sequences.



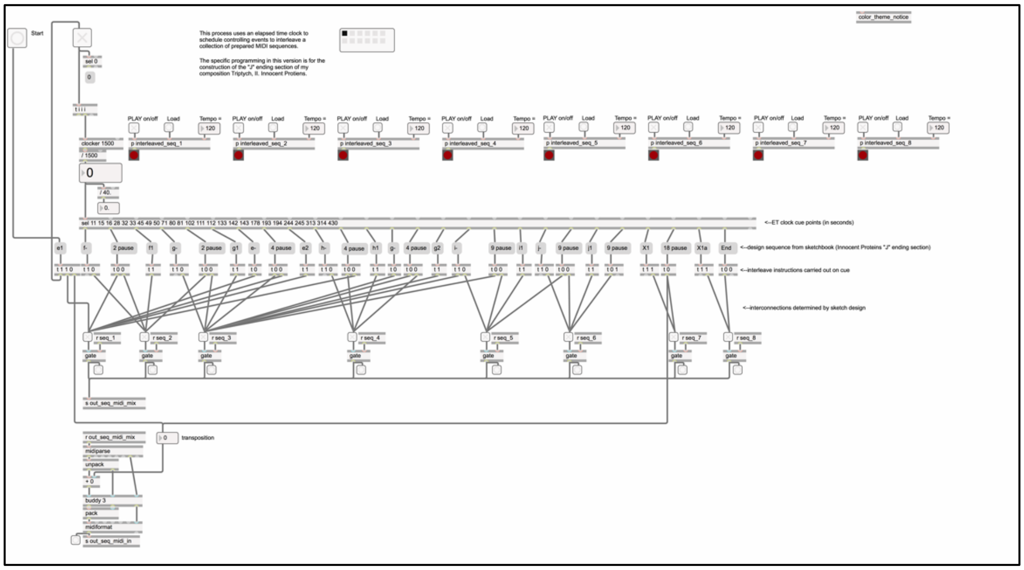
This program enables the user to merge two or three sequences with each having higher priority than the others for overlay. When a note in a track (channel) exceeds the threshhold time the "attention" of the merger is directed to the lower priority sequences.

Partial_sculpting



sequence_interleaver2

This process uses an elapsed time clock to schedule controlling events to interleave a collection of prepared MIDI sequences. The specific programming in this version is for the construction of the "J" ending section of my composition Triptych, II. Innocent Protiens.

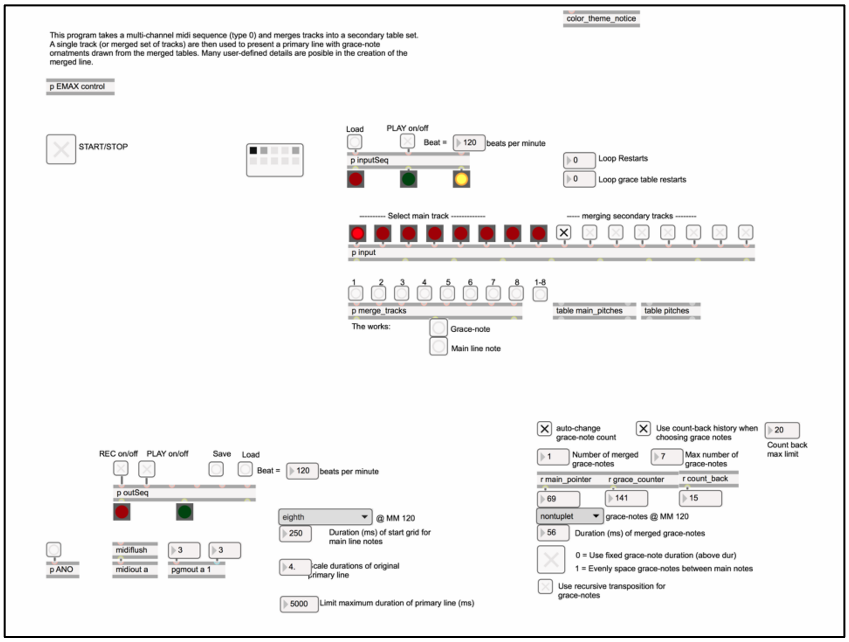


This process uses an elapsed time clock to schedule controlling events to interleave a collection of prepared MIDI sequences.

The specific programming in this version is for the construction of the "J" ending section of my composition Triptych, II. Innocent Protiens.

Track merging (grace-notes).v2

This program takes a multi-channel midi sequence (type 0) and merges tracks into a secondary table set. A single track (or merged set of tracks) are then used to present a primary line with grace-note ornatments drawn from the merged tables. Many user-defined details are posible in the creation of the merged line.

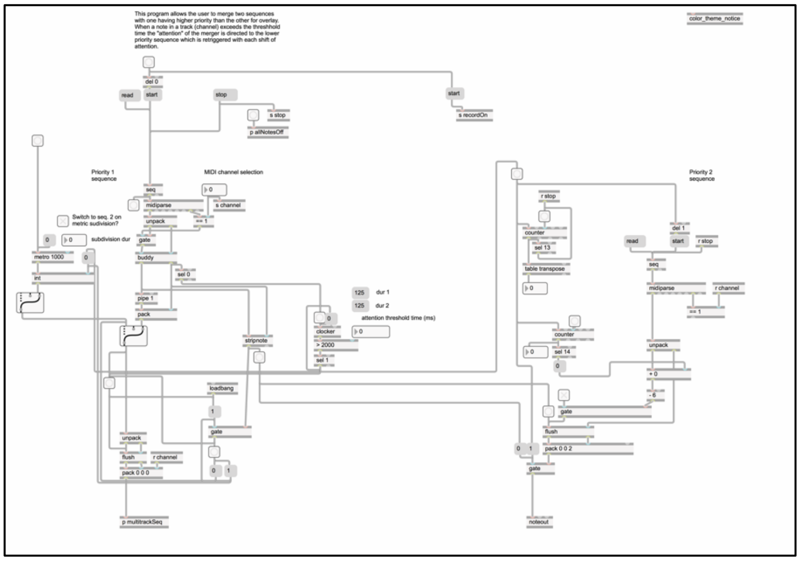


This program takes a multi-channel midi sequence (type 0) and merges tracks into a secondary table set. A single track (or merged set of tracks) are then used to present a primary line with grace-note ornatments drawn from the merged tables. Many user-defined details are posible in the creation of the merged line.

color_theme_notice
p EMAX control
START/STOP
Load
PLAY on/off
Beat =
120
beats per minute
p inputSeq
0
Loop Restarts
0
Loop grace table restarts
----------- Select main track -------------
------ merging secondary tracks --------
p input
1
2
3
4
5
6
7
8
1-8
p merge_tracks
table main_pitches
table pitches
The works:
Grace-note
Main line note
auto-change grace-note count
Use count-back history when choosing grace notes
20
Count back max limit
1
Number of merged grace-notes
7
Max number of grace-notes
r main_pointer
r grace_counter
r count_back
69
141
15
nontuplet
grace-notes @ MM 120
56
Duration (ms) of merged grace-notes
0 = Use fixed grace-note duration (above dur)
1 = Evenly space grace-notes between main notes
Use recursive transposition for grace-notes
REC on/off
PLAY on/off
Save
Load
Beat =
120
beats per minute
p outSeq
eighth
@ MM 120
250
Duration (ms) of start grid for main line notes
4.
Scale durations of original primary line
5000
Limit maximum duration of primary line (ms)
p ANO
midiflush
midiout a
3
3
pgmout a 1

Triggered merging

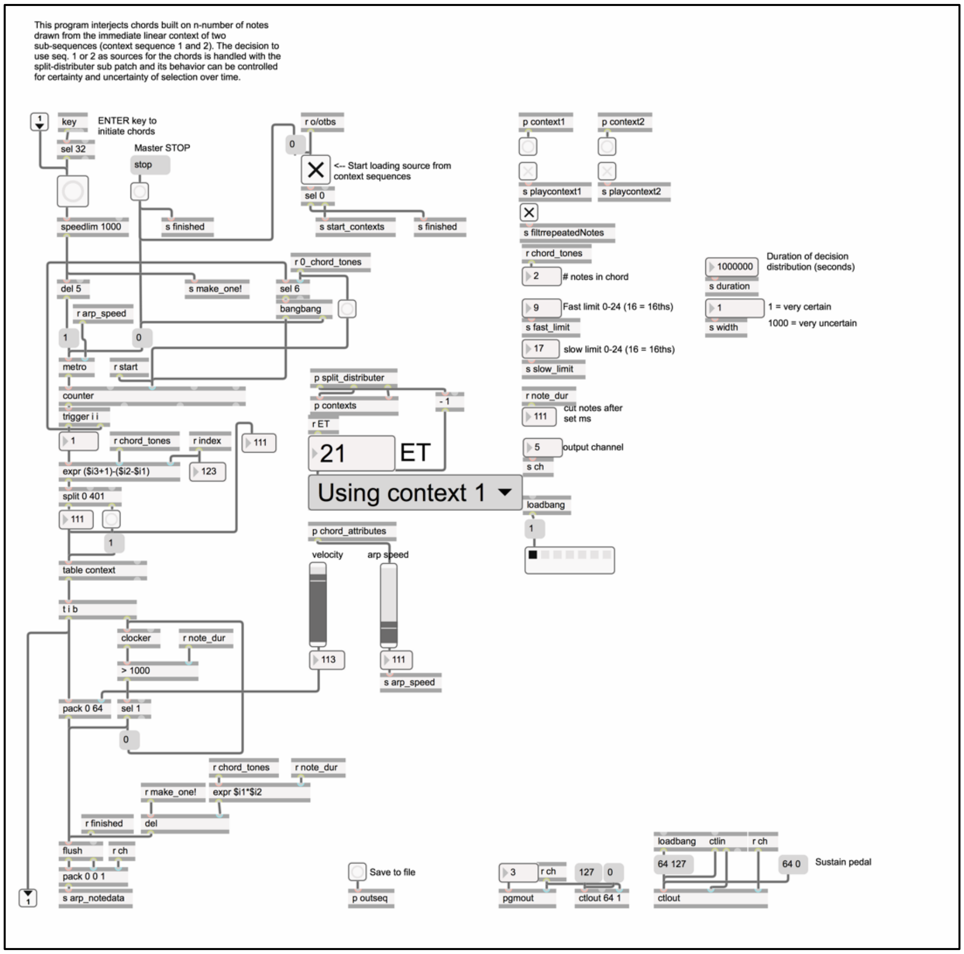
This program allows the user to merge two sequences with one having higher priority than the other for overlay. When a note in a track (channel) exceeds the threshhold time the "attention" of the merger is directed to the lower priority sequence which is retriggered with each shift of attention.



This program allows the user to merge two sequences with one having higher priority than the other for overlay. When a note in a track (channel) exceeds the threshhold time the "attention" of the merger is directed to the lower priority sequence which is retriggered with each shift of attention.

Triggered.arps.texture

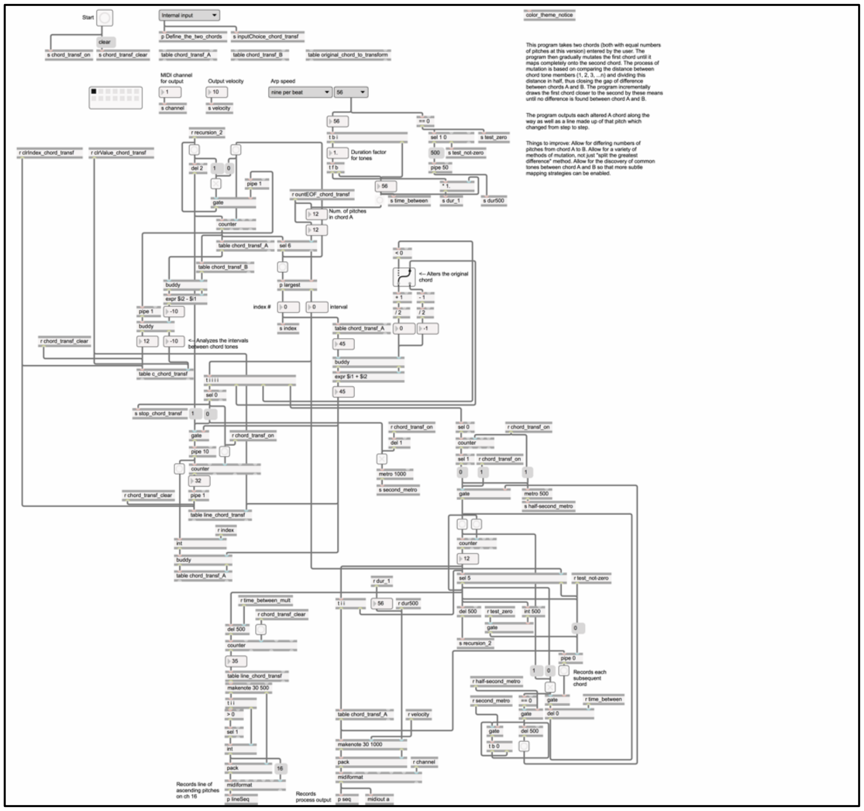
This program interjects chords built on n-number of notes drawn from the immediate linear context of two sub-sequences (context sequence 1 and 2). The decision to use seq. 1 or 2 as sources for the chords is handled with the split-distributer sub patch and its behavior can be controlled for certainty and uncertainty of selection over time.



This program interjects chords built on n-number of notes drawn from the immediate linear context of two sub-sequences (context sequence 1 and 2). The decision to use seq. 1 or 2 as sources for the chords is handled with the split-distributer sub patch and its behavior can be controlled for certainty and uncertainty of selection over time.

Chord transformation.v2

This program takes two chords (both with equal numbers of pitches at this version) entered by the user. The program then gradually mutates the first chord until it maps completely onto the second chord. The process of mutation is based on comparing the distance between chord tone members (1, 2, 3, ...n) and dividing this distance in half, thus closing the gap of difference between chords A and B. The program incrementally draws the first chord closer to the second by these means until no difference is found between chord A and B.


color_theme_notice
Start
clear
Internal input
p Define_the_two_chords
MIDI channel for output
Output velocity
Arp speed
nine per beat
Duration factor for tones
Num. of pitches in chord A
<-- Alters the original chord
<-- Analyzes the intervals between chord tones
index #
interval
Records each subsequent chord
Records line of ascending pitches on ch 16
Records process output

This program takes two chords (both with equal numbers of pitches at this version) entered by the user. The program then gradually mutates the first chord until it maps completely onto the second chord. The process of mutation is based on comparing the distance between chord tone members (1, 2, 3, ...n) and dividing this distance in half, thus closing the gap of difference between chords A and B. The program incrementally draws the first chord closer to the second by these means until no difference is found between chord A and B.

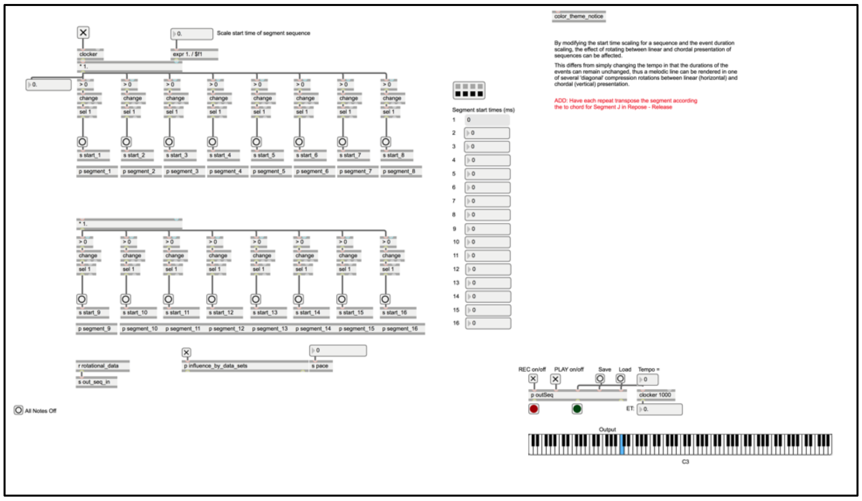
The program outputs each altered A chord along the way as well as a line made up of that pitch which changed from step to step.

Things to improve: Allow for differing numbers of pitches from chord A to B. Allow for a variety of methods of mutation, not just "split the greatest difference" method. Allow for the discovery of common tones between chord A and B so that more subtle mapping strategies can be enabled.

3D rotation of sequence fragments

By modifying the start time scaling for a sequence and the event duration scaling, the effect of rotating between linear and chordal presentation of sequences can be affected.

This differs from simply changing the tempo in that the durations of the events can remain unchanged, thus a melodic line can be rendered in one of several 'diagonal' compression rotations between linear (horizontal) and chordal (vertical) presentation.



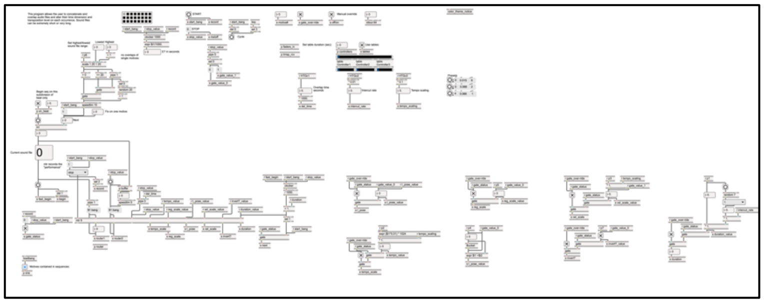
By modifying the start time scaling for a sequence and the event duration scaling, the effect of rotating between linear and chordal presentation of sequences can be affected.

This differs from simply changing the tempo in that the durations of the events can remain unchanged, thus a melodic line can be rendered in one of several 'diagonal' compression rotations between linear (horizontal) and chordal (vertical) presentation.

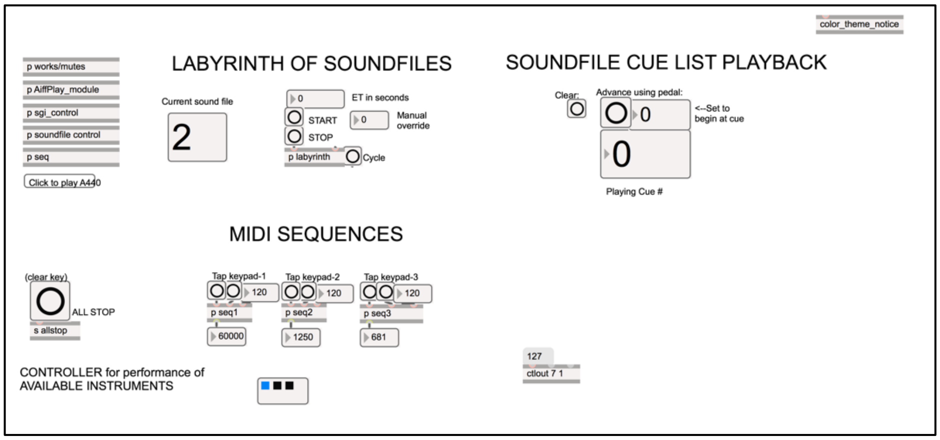
ADD: Have each repeat transpose the segment according the to chord for Segment J in Repose - Release

Labyrinth audio v.1 rev1 Max8

This program allows the user to concatenate and overlap audio files and alter their time dimension and transposition level on each occurrence. Sound files can be extremely short or very long.

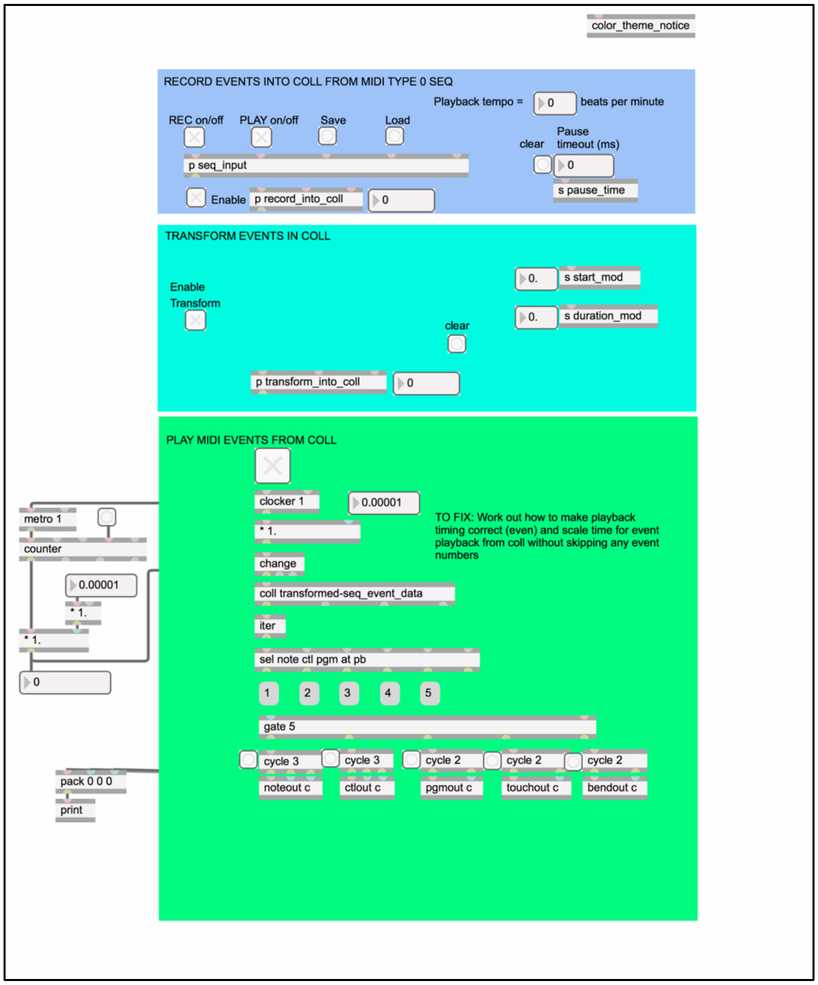


Labyrinth of SF - Available instr. control v.1.2 Max8



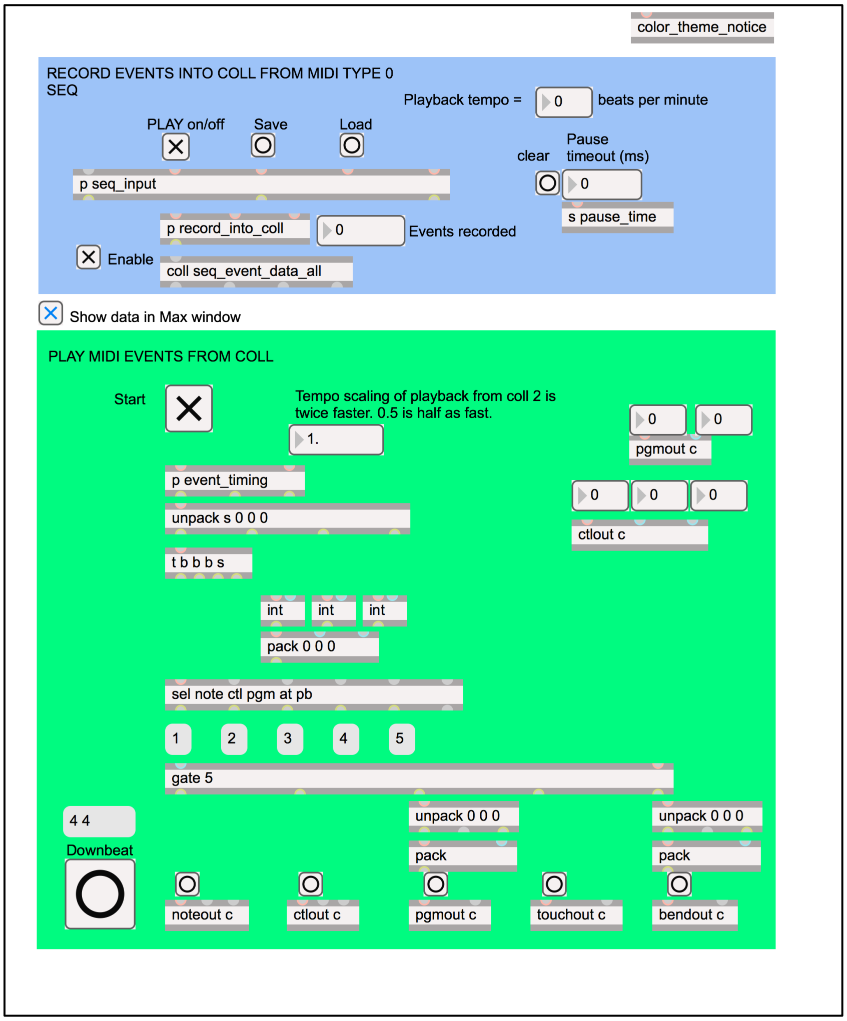
CONTROLLER for performance of
AVAILABLE INSTRUMENTS

multiple seq player with chrd sieves split Max8



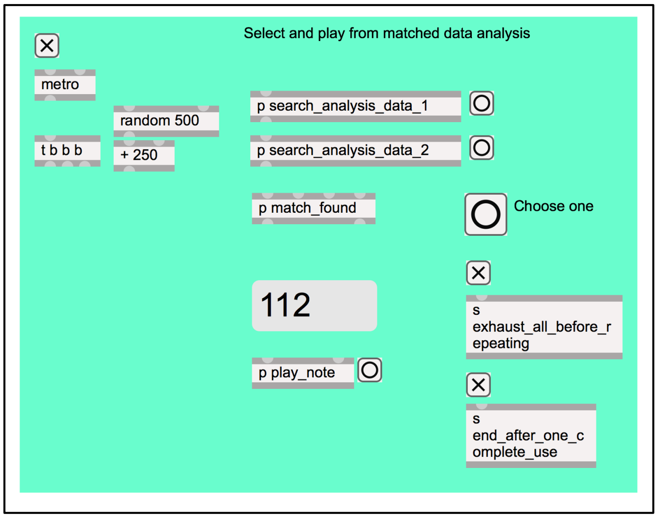
TO FIX: Work out how to make playback timing correct (even) and scale time for event playback from coll without skipping any event numbers

Play seq from coll random Max8

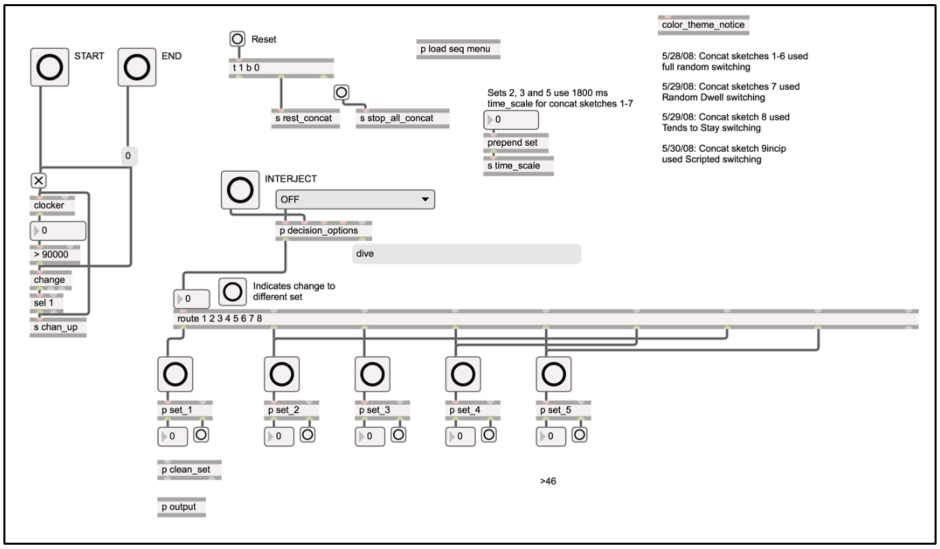


Tempo scaling of playback from coll 2 is twice faster. 0.5 is half as fast.

Segment database module Max8



Seq. concatenator 2a Max8



Sets 2, 3 and 5 use 1800 ms time_scale for concat sketches 1-7

5/28/08: Concat sketches 1-6 used full random switching

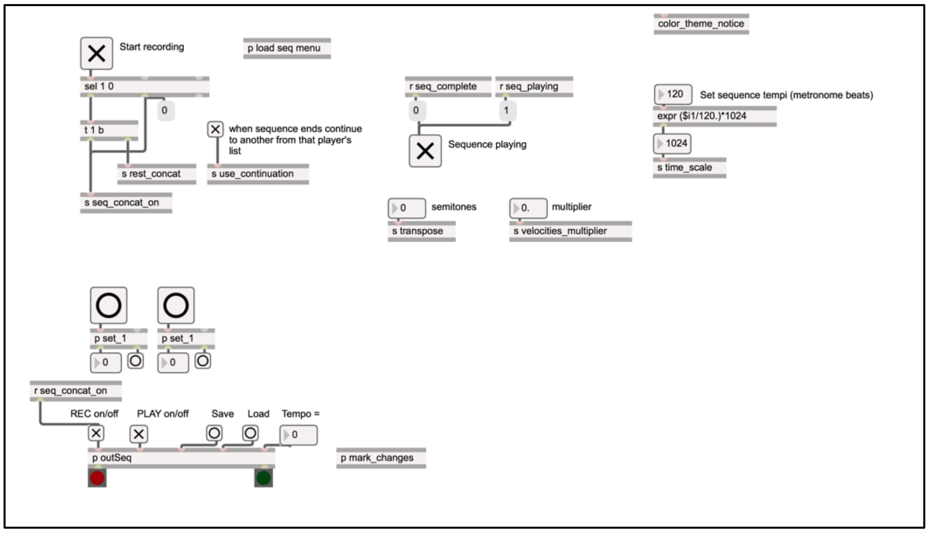
5/29/08: Concat sketches 7 used Random Dwell switching

5/29/08: Concat sketch 8 used Tends to Stay switching

5/30/08: Concat sketch 9incip used Scripted switching

Indicates change to different set

sequence assembler Max8



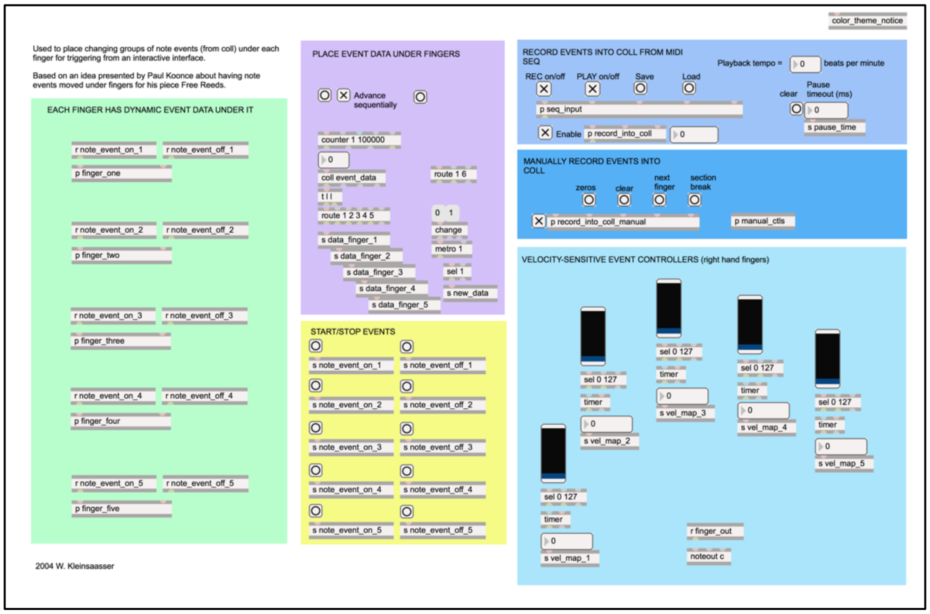
when sequence ends continue to another from that player's list

Set sequence tempi (metronome beats)

under_fingers v. 1.5 Max8

Used to place changing groups of note events (from coll) under each finger for triggering from an interactive interface.

Based on an idea presented by Paul Koonce about having note events moved under fingers for his piece *Free Reeds*.



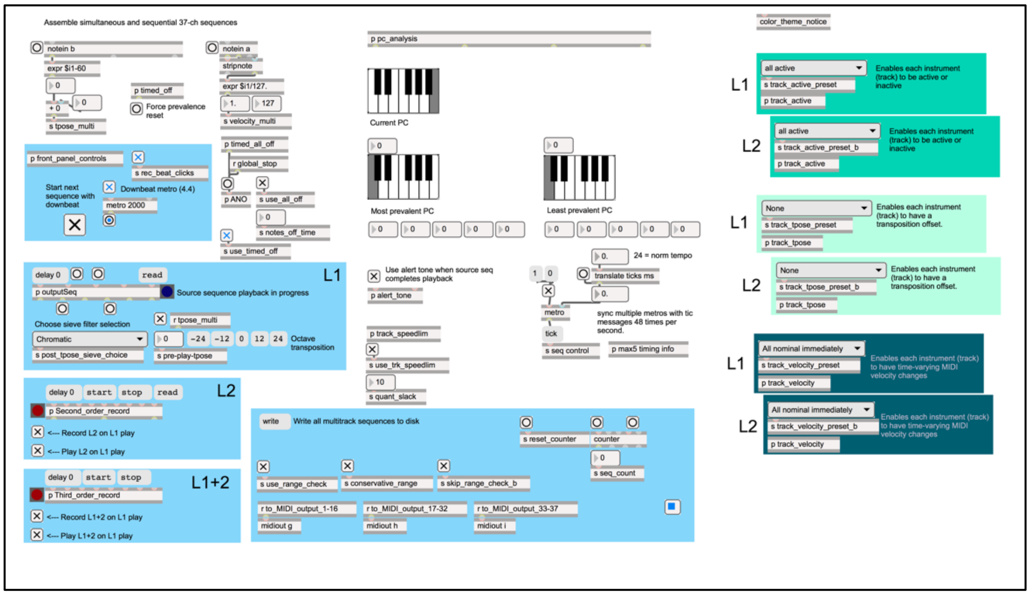
Used to place changing groups of note events (from coll) under each finger for triggering from an interactive interface.

Based on an idea presented by Paul Koonce about having note events moved under fingers for his piece Free Reeds.

2004 W. Kleinsaasser

24_V second order assembler_5 sync Max8

Assemble simultaneous and sequential 37-ch sequences



Assemble simultaneous and sequential 37-ch sequences

Source sequence playback in progress

Use alert tone when source seq completes playback

sync multiple metros with tic messages 48 times per second.

Write all multitrack sequences to disk

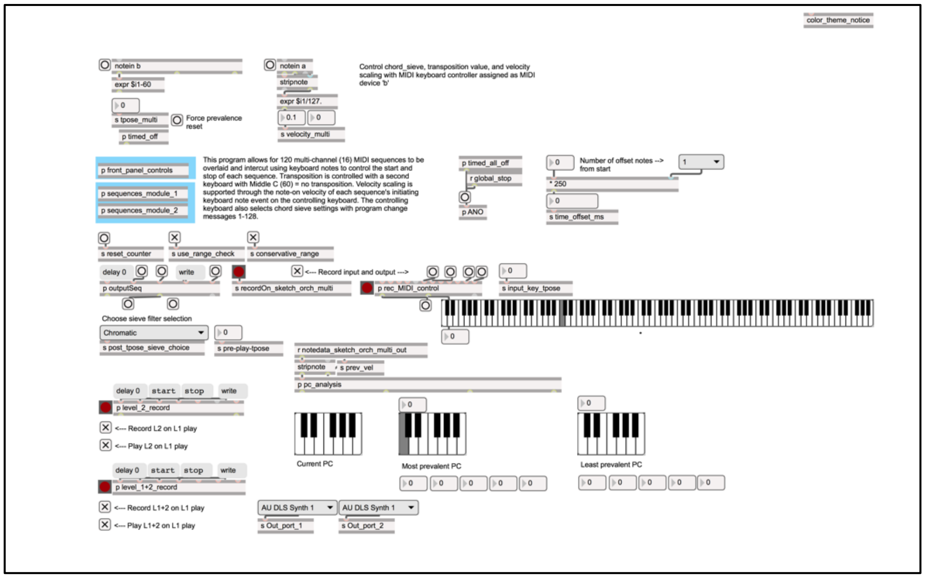
Enables each instrument (track) to be active or inactive

Enables each instrument (track) to have a transposition offset.

Enables each instrument (track) to have time-varying MIDI velocity changes

interstitial concac Max8

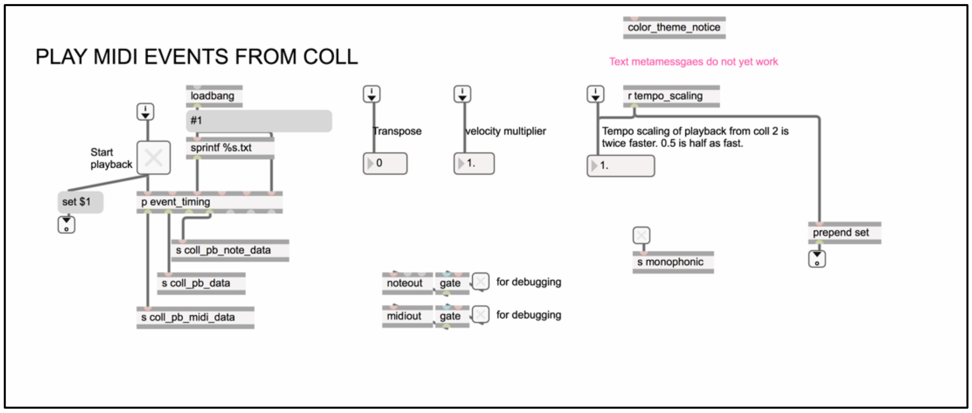
This program allows for 120 multi-channel (16) MIDI sequences to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.



Control chord_sieve, transposition value, and velocity scaling with MIDI keyboard controller assigned as MIDI device 'b'

This program allows for 120 multi-channel (16) MIDI sequences to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.

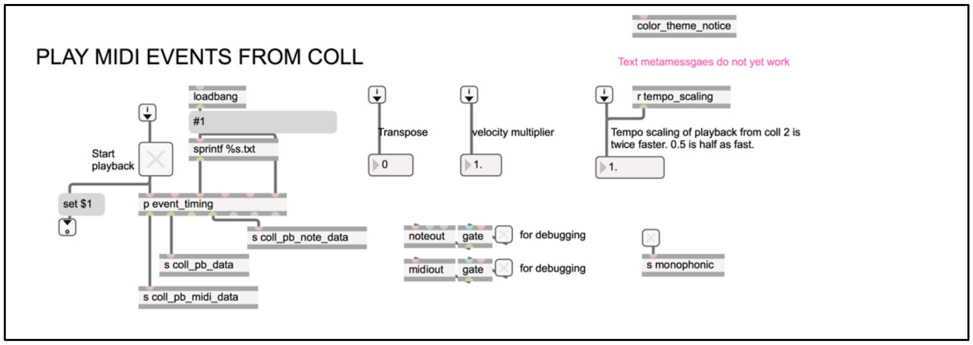
play_from_coll_metaD Max8



Text metamessgaes do not yet work

Tempo scaling of playback from coll 2 is twice faster. 0.5 is half as fast.

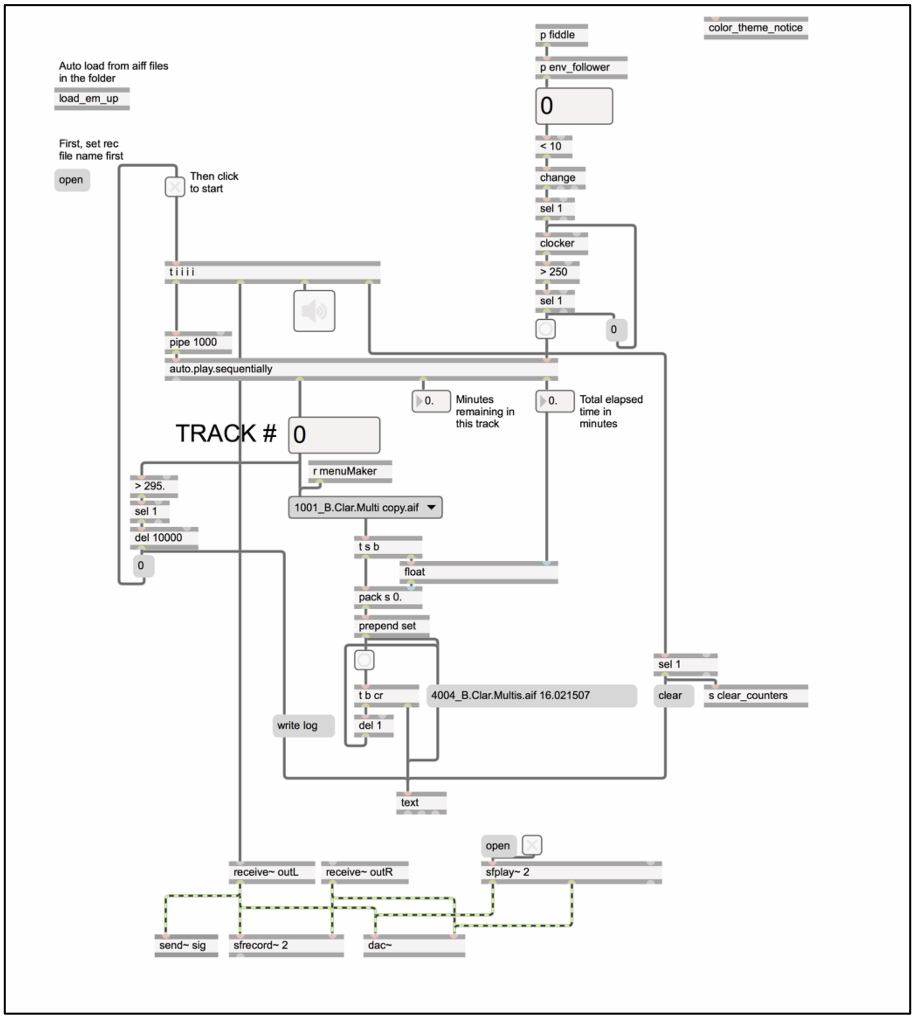
play_from_coll_metaD cputimer Max8



Text metamessgaes do not yet work

Tempo scaling of playback from coll 2 is twice faster. 0.5 is half as fast.

sf concatenator Max8

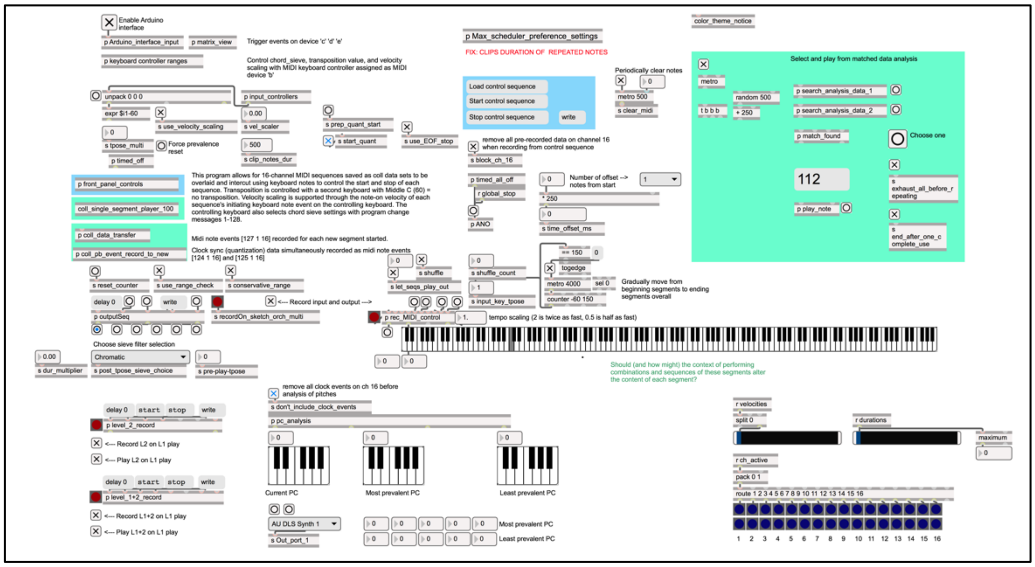


Coll Segment Player

Single-16-chan version

coll_sgmnt_plyr_mult_60 [1-31-16 + 12-16-16] 1-11-17 Single-16 Max8

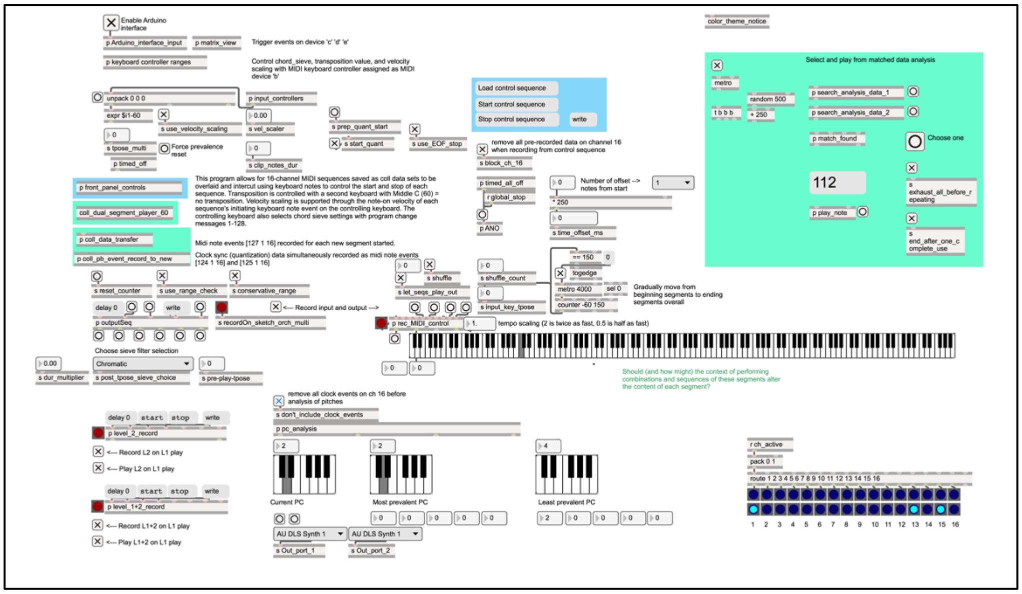
This program allows for 16-channel MIDI sequences saved as coll data sets to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.



Dual-32-chan version

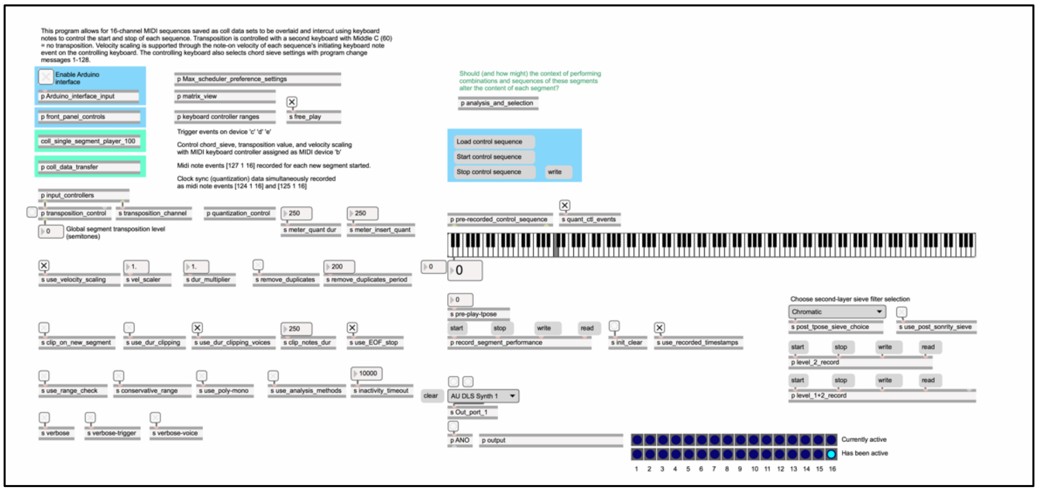
coll_sgmnt_plyr_mult_301 1-31-16 Dual-32 Max8

This program allows for 16-channel MIDI sequences saved as coll data sets to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.



Coll segment player 10-16-17 Auduino matrix interface clean

coll_sgmnt_plyr_mult_60 [1-31-16 + 12-16-16] 4-16-17 Single-16 midifile-out timing solved 10-20-17 Aurduino matrix controller Max8



This program allows for 16-channel MIDI sequences saved as coll data sets to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.

Should (and how might) the context of performing combinations and sequences of these segments alter the content of each segment?

Trigger events on device 'c' 'd' 'e'

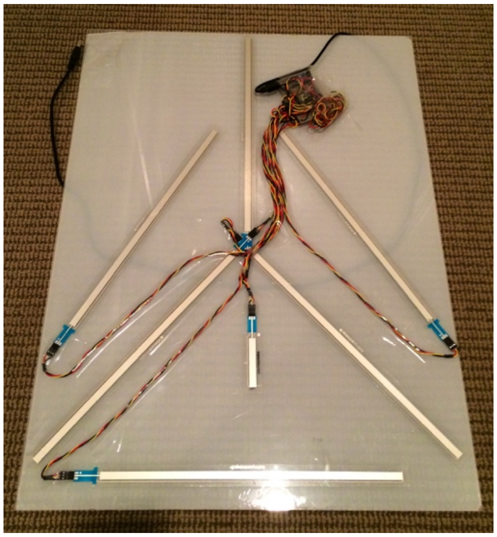
Control chord_sieve, transposition value, and velocity scaling with MIDI keyboard controller assigned as MIDI device 'b'

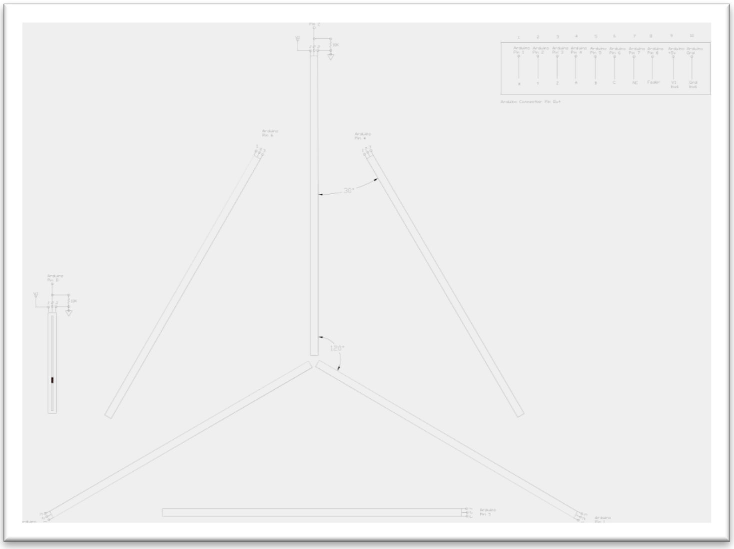
Midi note events [127 1 16] recorded for each new segment started.

Clock sync (quantization) data simultaneously recorded as midi note events [124 1 16] and [125 1 16]

Global segment transposition level (semitones)

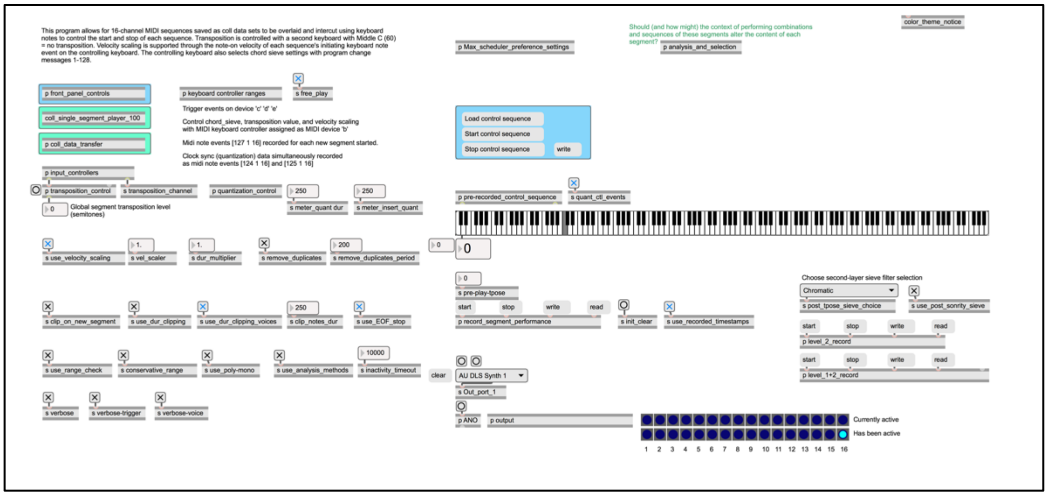
Choose second-layer sieve filter selection





Coll segment player 10-16-17 Max8 clean

coll_sgmnt_plyr_mult_60 [1-31-16 + 12-16-16] 4-16-17 Single-16 midifile-out timing solved 10-20-17 Max8



This program allows for 16-channel MIDI sequences saved as coll data sets to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.

Trigger events on device 'c' 'd' 'e'

Control chord_sieve, transposition value, and velocity scaling with MIDI keyboard controller assigned as MIDI device 'b'

Midi note events [127 1 16] recorded for each new segment started.

Clock sync (quantization) data simultaneously recorded as midi note events [124 1 16] and [125 1 16]

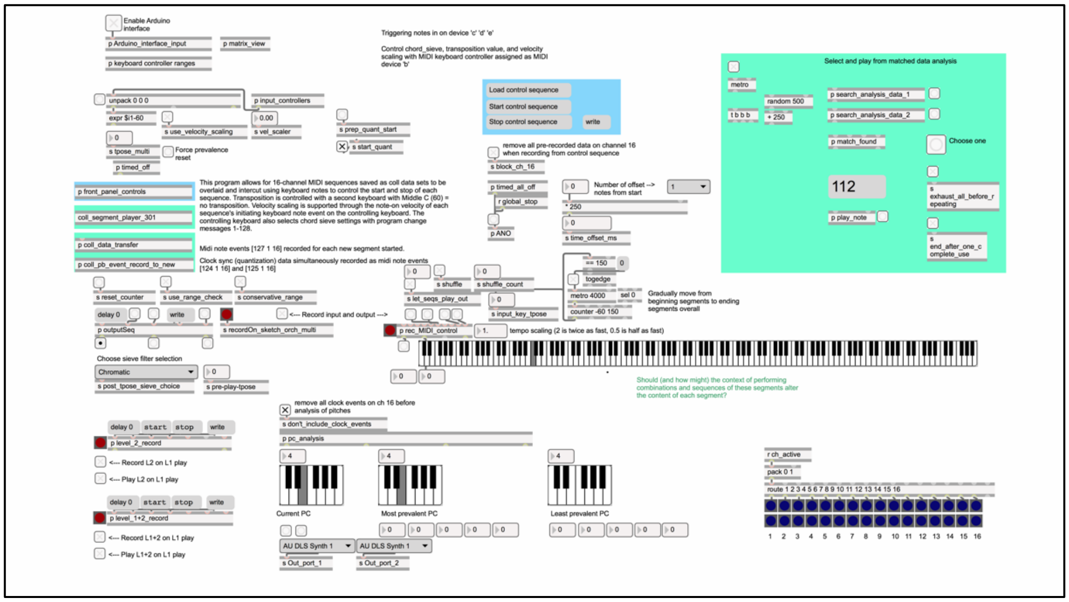
Global segment transposition level (semitones)

Should (and how might) the context of performing combinations and sequences of these segments alter the content of each segment?

This program allows for 16-channel MIDI sequences saved as coll data sets to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.

Coll segment player Arduino matrix interface

coll_sgmnt_plyr_mult_301 Arduino matrix Max8



Triggering notes in on device 'c' 'd' 'e'

Control chord_sieve, transposition value, and velocity scaling with MIDI keyboard controller assigned as MIDI device 'b'

This program allows for 16-channel MIDI sequences saved as coll data sets to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.

Midi note events [127 1 16] recorded for each new segment started.

Clock sync (quantization) data simultaneously recorded as midi note events [124 1 16] and [125 1 16]

remove all pre-recorded data on channel 16 when recording from control sequence

Gradually move from beginning segments to ending segments overall

tempo scaling (2 is twice as fast, 0.5 is half as fast)

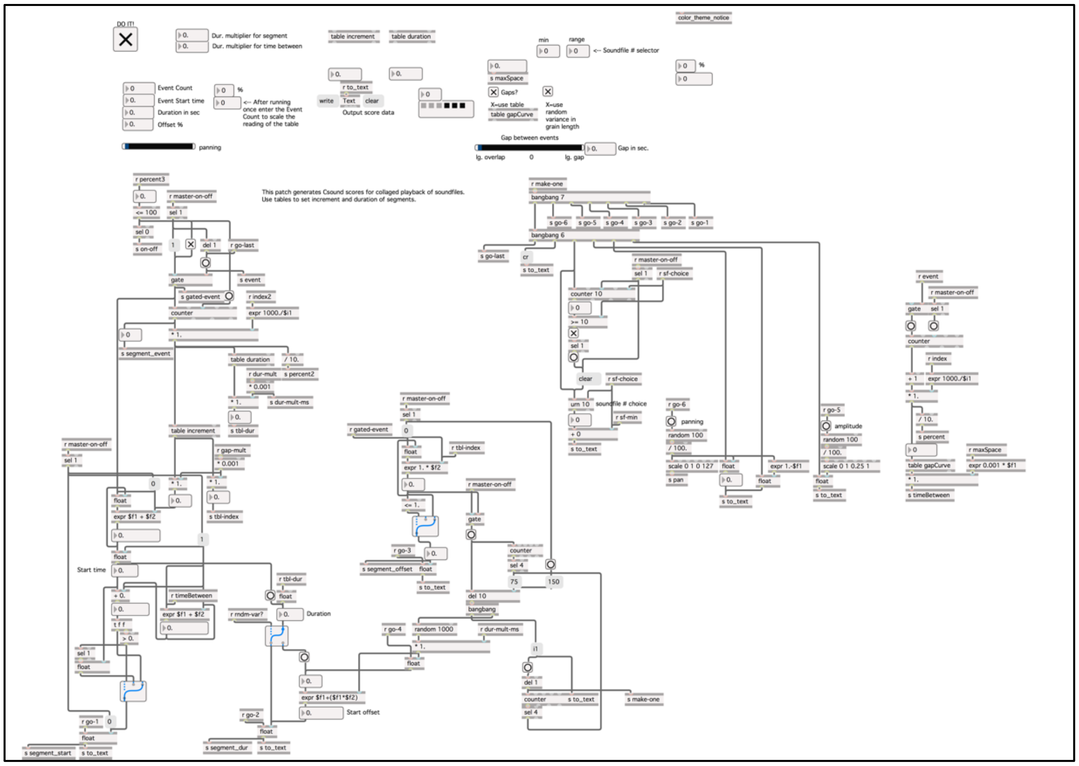
remove all clock events on ch 16 before analysis of pitches

Should (and how might) the context of performing combinations and sequences of these segments alter the content of each segment?

This program allows for 16-channel MIDI sequences saved as coll data sets to be overlaid and intercut using keyboard notes to control the start and stop of each sequence. Transposition is controlled with a second keyboard with Middle C (60) = no transposition. Velocity scaling is supported through the note-on velocity of each sequence's initiating keyboard note event on the controlling keyboard. The controlling keyboard also selects chord sieve settings with program change messages 1-128.

Csound collage score maker Max8

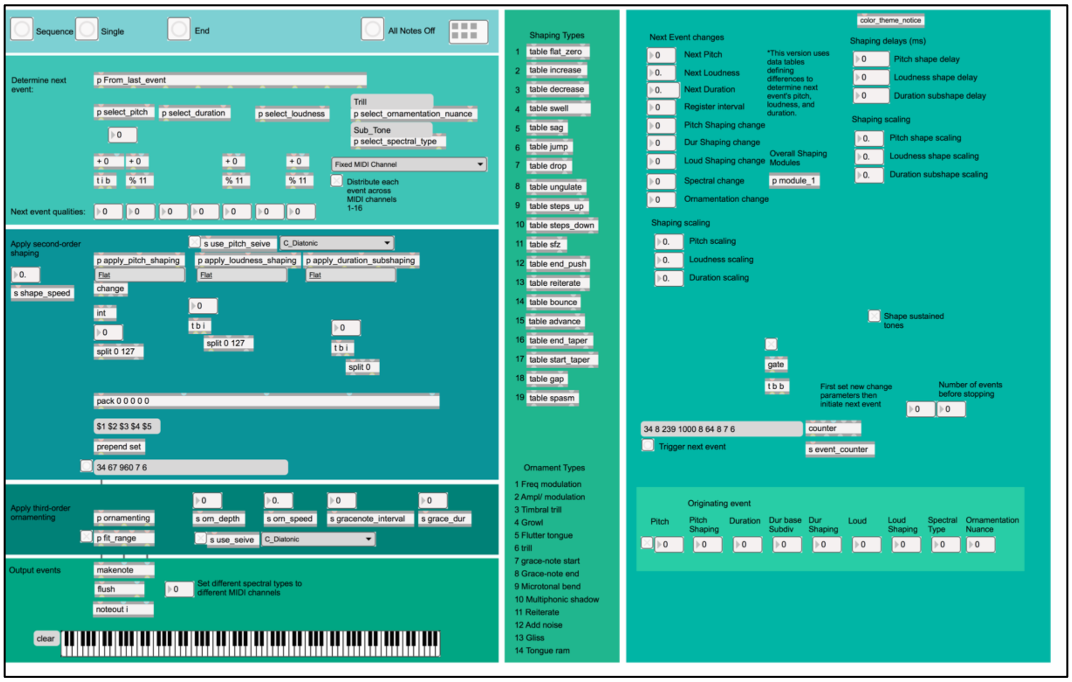
This patch generates Csound scores for collaged playback of soundfiles. Use tables to set increment and duration of segments.



This patch generates Csound scores for collaged playback of soundfiles. Use tables to set increment and duration of segments.

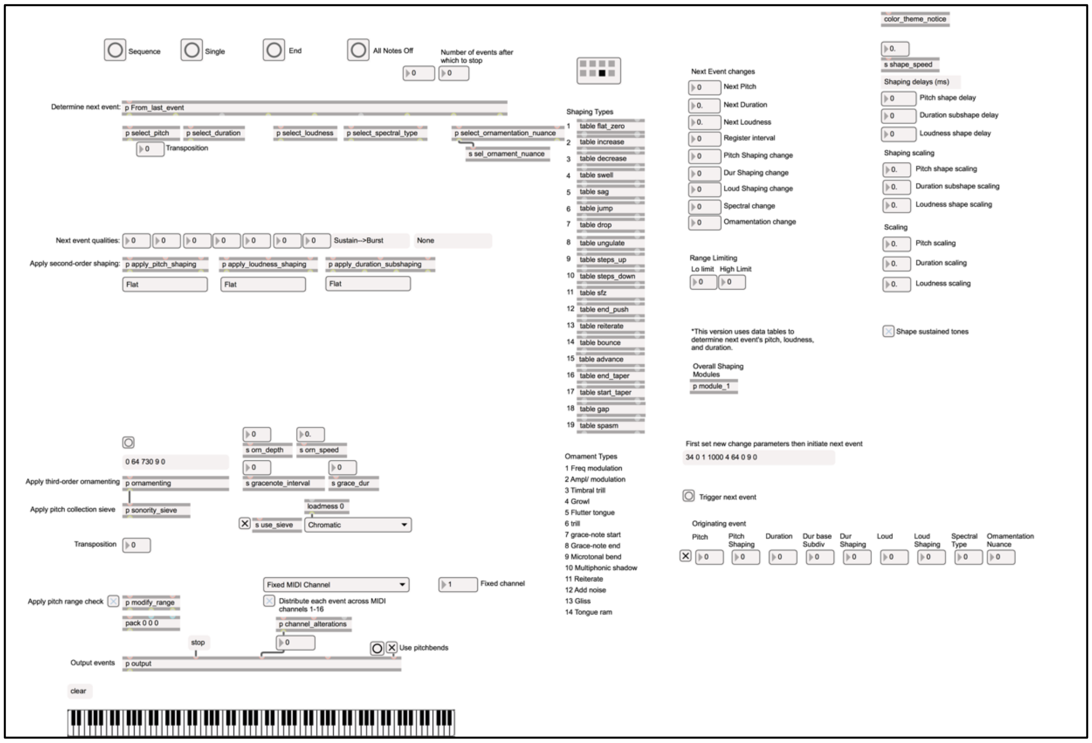
Lines-Curves of Influence

Line - Curves of Influence 7.29 (diffs) rev Max8



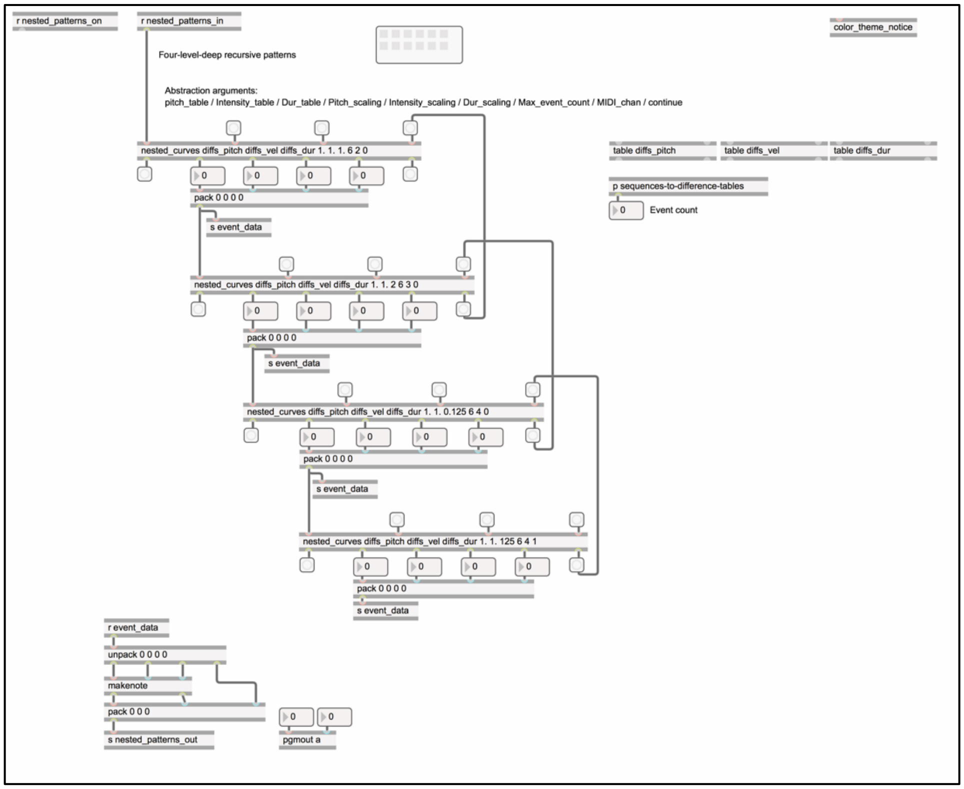
*This version uses data tables defining differences to determine next event's pitch, loudness, and duration.

Line - Curves of Influence 7.32 (tbls) Max8



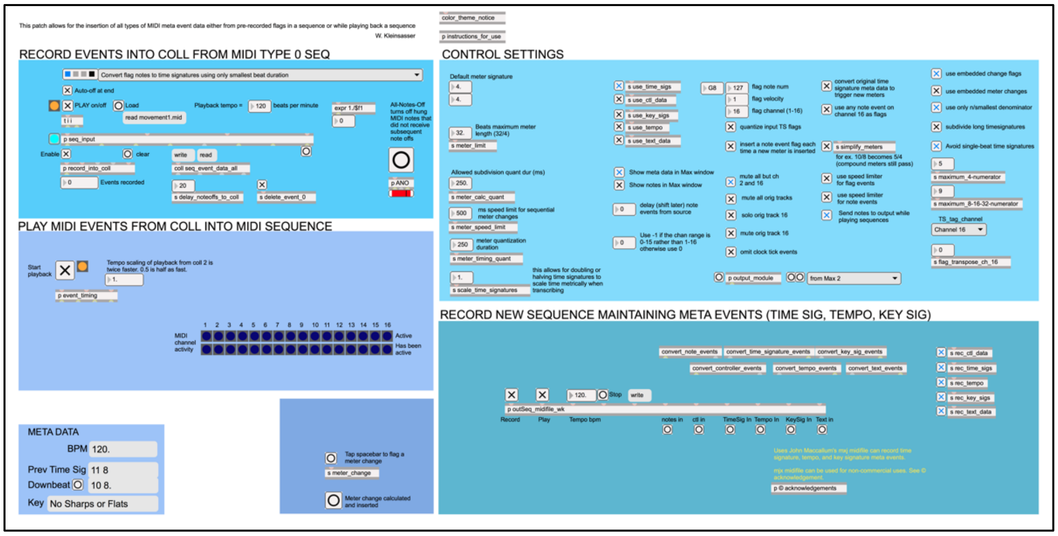
Nested_patterns Max8

Four-level-deep recursive patterns.


r nested_patterns_on
r nested_patterns_in
color_theme_notice
Four-level-deep recursive patterns
Abstraction arguments:
pitch_table / Intensity_table / Dur_table / Pitch_scaling / Intensity_scaling / Dur_scaling / Max_event_count / MIDI_chan / continue
nested_curves diffs_pitch diffs_vel diffs_dur 1. 1. 1. 6 2 0
table diffs_pitch
table diffs_vel
table diffs_dur
p sequences-to-difference-tables
Event count
pack 0 0 0 0
s event_data
nested_curves diffs_pitch diffs_vel diffs_dur 1. 1. 2 6 3 0
pack 0 0 0 0
s event_data
nested_curves diffs_pitch diffs_vel diffs_dur 1. 1. 0.125 6 4 0
pack 0 0 0 0
s event_data
nested_curves diffs_pitch diffs_vel diffs_dur 1. 1. 125 6 4 1
pack 0 0 0 0
s event_data
r event_data
unpack 0 0 0 0
makenote
pack 0 0 0
s nested_patterns_out
pgmout a

Rec seq to coll w meta v14B 1 11-12-17 Max8

This patch allows for the insertion of all types of MIDI meta event data either from pre-recorded flags in a sequence or while playing back a sequence



This patch allows for the insertion of all types of MIDI meta event data either from pre-recorded flags in a sequence or while playing back a sequence

W. Kleinsasser

RECORD EVENTS INTO COLL FROM MIDI TYPE 0 SEQ

Convert flag notes to time signatures using only smallest beat duration

☒ Auto-off at end

☒ PLAY on/off

Playback tempo = 120 beats per minute

All-Notes-Off turns off hung MIDI notes that did not receive subsequent note offs

Enable ☒

Events recorded 0

PLAY MIDI EVENTS FROM COLL INTO MIDI SEQUENCE

Start playback ☒

Tempo scaling of playback from coll 2 is twice faster. 0.5 is half as fast.

1.

META DATA

BPM 120.

Prev Time Sig 11 8

Downbeat ○ 10 8.

Key No Sharps or Flats

Tap spacebar to flag a meter change

Meter change calculated and inserted

CONTROL SETTINGS

Default meter signature 4. 4.

32. Beats maximum meter length (32/4)

Allowed subdivision quant dur (ms) 250.

500 ms speed limit for sequential meter changes

250 meter quantization duration

1. this allows for doubling or halving time signatures to scale time metrically when transcribing

☒ s use_time_sigs

☒ s use_ctl_data

☒ s use_key_sigs

☒ s use_tempo

☒ s use_text_data

☒ Show meta data in Max window

☒ Show notes in Max window

0 delay (shift later) note events from source

0 Use -1 if the chan range is 0-15 rather than 1-16 otherwise use 0

G8 127 flag note num

1 flag velocity

16 flag channel (1-16)

☒ quantize input TS flags

☒ insert a note event flag each time a new meter is inserted

☒ mute all but ch 2 and 16

☒ mute all orig tracks

☒ solo orig track 16

☒ mute orig track 16

☒ omit clock tick events

☒ convert original time signature meta data to trigger new meters

☒ use any note event on channel 16 as flags

☒ s simplify_meters for ex. 10/8 becomes 5/4 (compound meters still pass)

☒ use speed limiter for flag events

☒ use speed limiter for note events

☒ Send notes to output while playing sequences

☒ use embedded change flags

☒ use embedded meter changes

☒ use only n/smallest denominator

☒ subdivide long timesignatures

☒ Avoid single-beat time signatures

5 s maximum_4-numerator

9 s maximum_8-16-32-numerator

TS_tag_channel Channel 16

0 s flag_transpose_ch_16

from Max 2

RECORD NEW SEQUENCE MAINTAINING META EVENTS (TIME SIG, TEMPO, KEY SIG)

Tempo bpm 120.

☒ s rec_ctl_data

☒ s rec_time_sigs

☒ s rec_tempo

☒ s rec_key_sigs

☒ s rec_text_data

Uses John Maccallum's mxj midifile can record time signature, tempo, and key signature meta events.

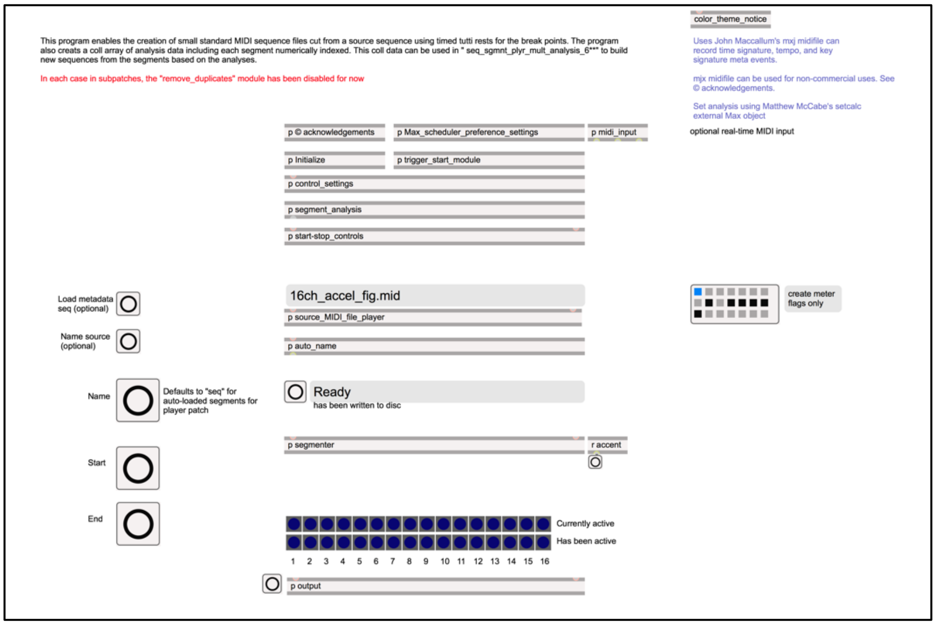
mjx midifile can be used for non-commercial uses. See © acknowledgement.

Sequence Segmenter

SINGLE SOURCE
sequence segmenter with event contextual data Max6 full-segmentation 5-19-17 colls

Seq_segmntr12 w evnt cntxtl 3-6-17 metaData TS timestamp single-source full-segmentation no-analysis modular origTimeStamp minimal speedlim voice_clip 3-14-17g - 5-19-17 colls Max8

This program enables the creation of small standard MIDI sequence files cut from a source sequence using timed tutti rests for the break points. The program also creates a coll array of analysis data including each segment numerically indexed. This coll data can be used in " seq_sgmnt_plyr_mult_analysis_6**" to build new sequences from the segments based on the analyses.



This program enables the creation of small standard MIDI sequence files cut from a source sequence using timed tutti rests for the break points. The program also creats a coll array of analysis data including each segment numerically indexed. This coll data can be used in " seq_sgmnt_plyr_mult_analysis_6**" to build new sequences from the segments based on the analyses.

In each case in subpatches, the "remove_duplicates" module has been disabled for now

Uses John Maccallum's mxj midifile can record time signature, tempo, and key signature meta events.

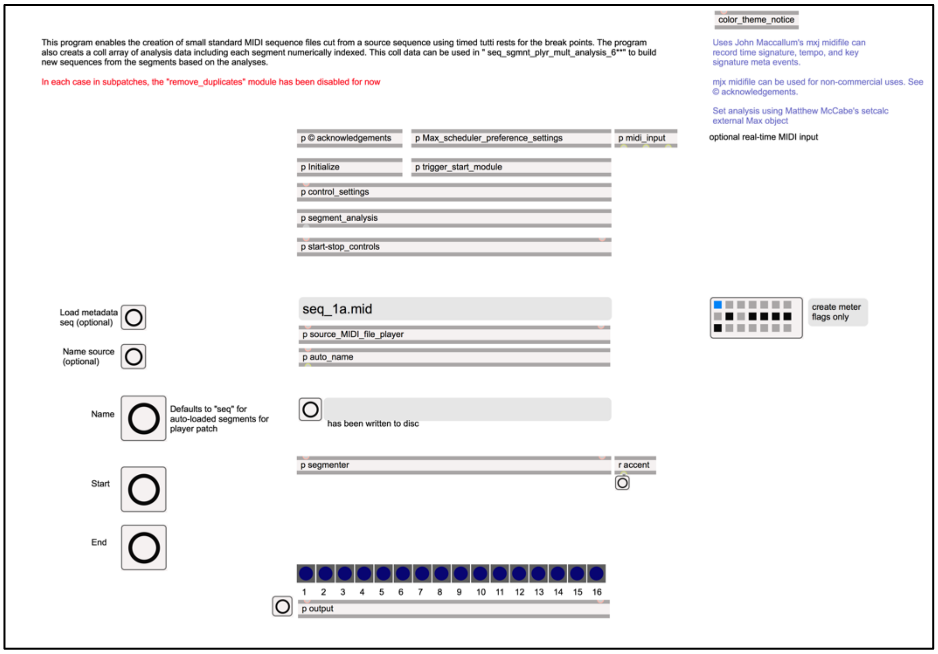
mjx midifile can be used for non-commercial uses. See © acknowledgements.

Set analysis using Matthew McCabe's setcalc external Max object

BATCH SOURCE
(attmpts to fix from single source mods) sequence segmenter with event contextual data Max6 Max8

Seq_segmntr12 w evnt cntxtl 1-12-17 metaData TS timestamp single-source full-segmentation no-analysis modular origTimeStamp minimal BATCH Max8

This program enables the creation of small standard MIDI sequence files cut from a source sequence using timed tutti rests for the break points. The program also creates a coll array of analysis data including each segment numerically indexed. This coll data can be used in " seq_sgmnt_plyr_mult_analysis_6**" to build new sequences from the segments based on the analyses.



This program enables the creation of small standard MIDI sequence files cut from a source sequence using timed tutti rests for the break points. The program also creats a coll array of analysis data including each segment numerically indexed. This coll data can be used in " seq_sgmnt_plyr_mult_analysis_6**" to build new sequences from the segments based on the analyses.

In each case in subpatches, the "remove_duplicates" module has been disabled for now

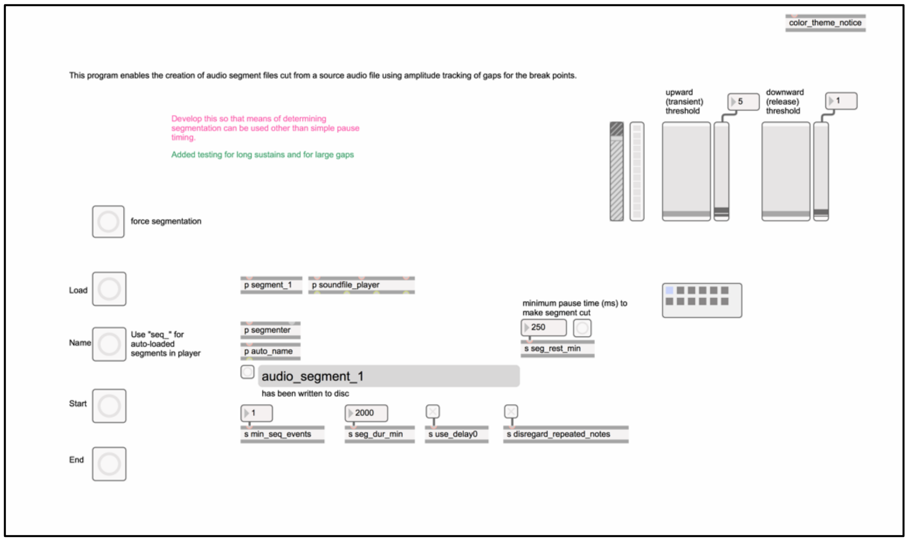
Uses John Maccallum's mxj midifile can record time signature, tempo, and key signature meta events.

mjx midifile can be used for non-commercial uses. See © acknowledgements.

Set analysis using Matthew McCabe's setcalc external Max object

Soundfile_segmntr Max8

This program enables the creation of audio segment files cut from a source audio file using amplitude tracking of gaps for the break points.



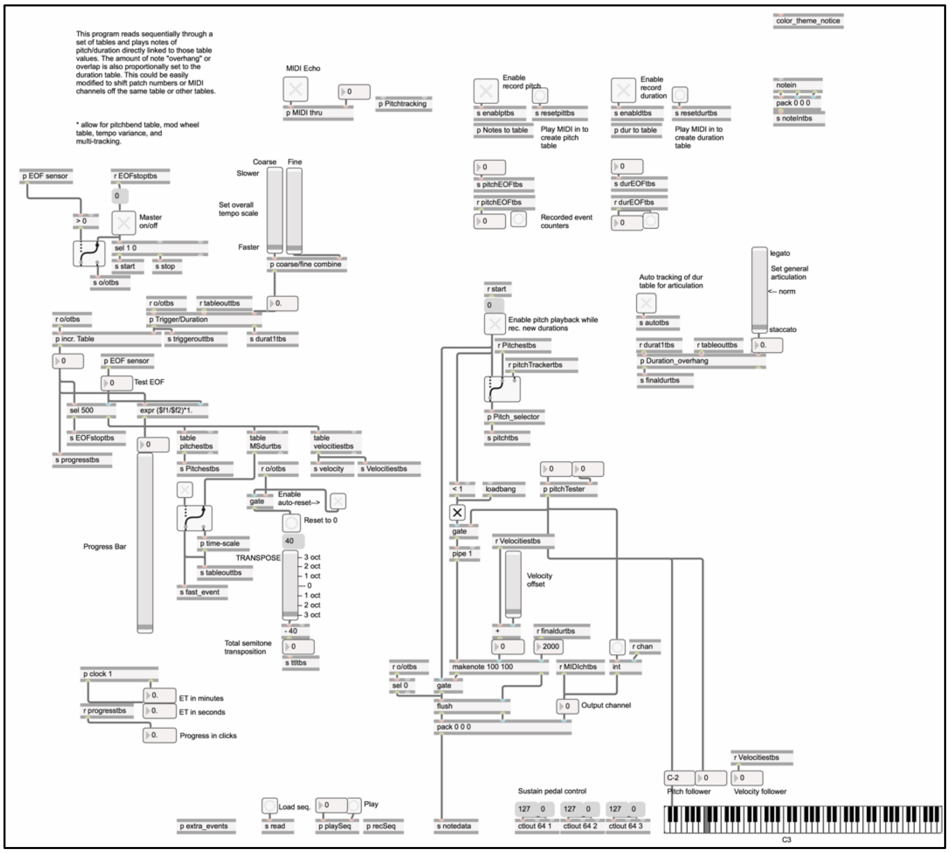
This program enables the creation of audio segment files cut from a source audio file using amplitude tracking of gaps for the break points.

Develop this so that means of determining segmentation can be used other than simple pause timing.

Added testing for long sustains and for large gaps

Table based sequencer.v2.2 Max8

This program reads sequentially through a set of tables and plays notes of pitch/duration directly linked to those table values. The amount of note "overhang" or overlap is also proportionally set to the duration table. This could be easily modified to shift patch numbers or MIDI channels off the same table or other tables.



This program reads sequentially through a set of tables and plays notes of pitch/duration directly linked to those table values. The amount of note "overhang" or overlap is also proportionally set to the duration table. This could be easily modified to shift patch numbers or MIDI channels off the same table or other tables.

* allow for pitchbend table, mod wheel table, tempo variance, and multi-tracking.