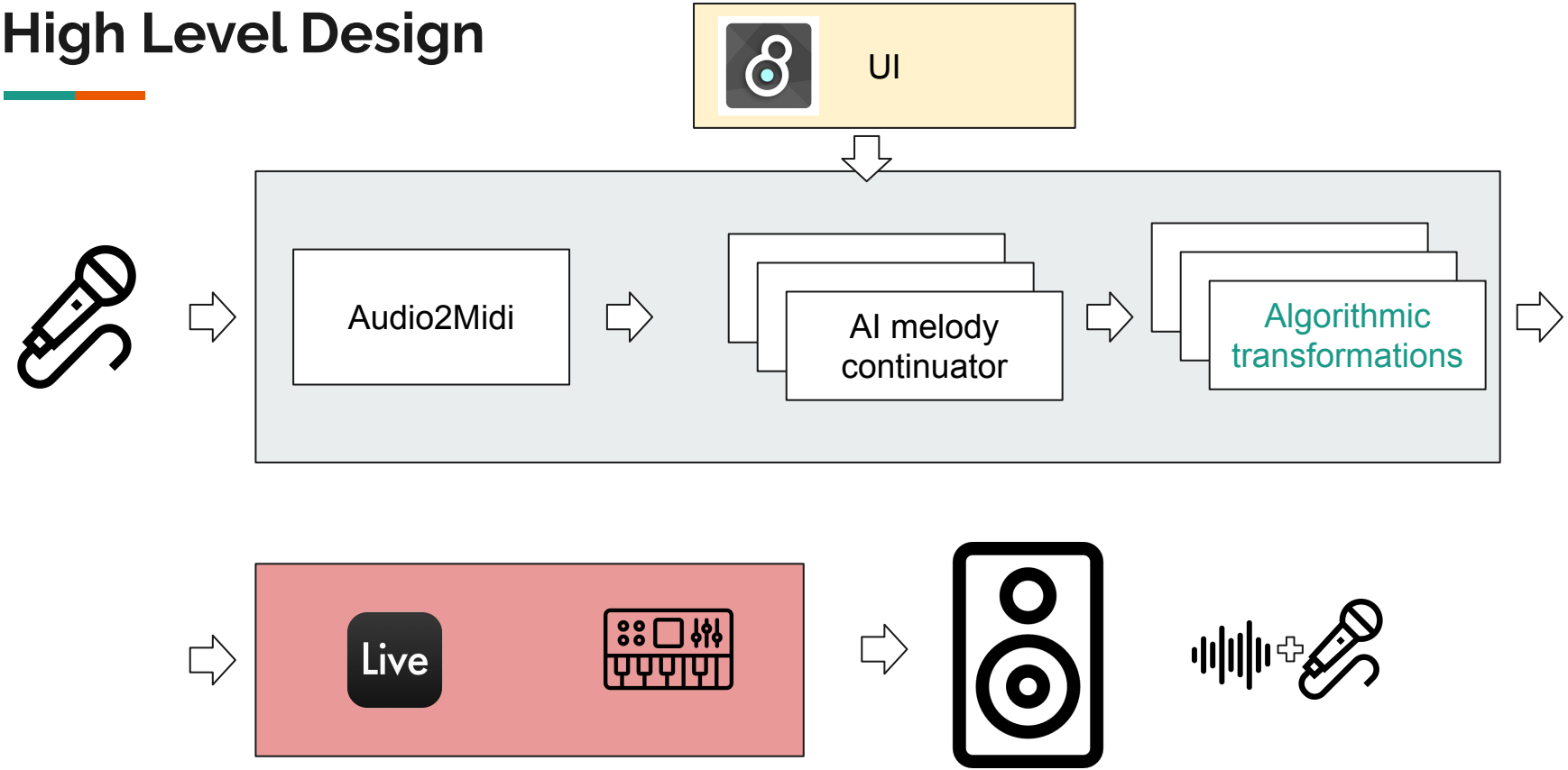
DUETable

Jonathan David
Jerzy Mizgiert
Dmitry Lamanov

Our goal

- Allow artist to improvise with AI in a conversational approach (question-answer)
- Answer is generated based on input melody
- Real Time interactive system
- Personalised and controllable pipeline for melody continuation

High Level Design



Audio2Midi

- Different implementations tested
  - basic pitch
  - aubio
  - essentia
- Not easy to use for real time application
- **Aubio** did the best results regarding processing time VS quality, ~72% accuracy

Melody continuation

- MuPT: LLM model trained on ABC notation
  - Controls: melody key, meter, n of bars, note length
  - Rich melodies
  - Hard to personalize
- Markov chain melody generator
- Mixed: executes and memorizes output from MuPT and then apply Markov chain on top
- Dummy: continuously listen environment to produce melody

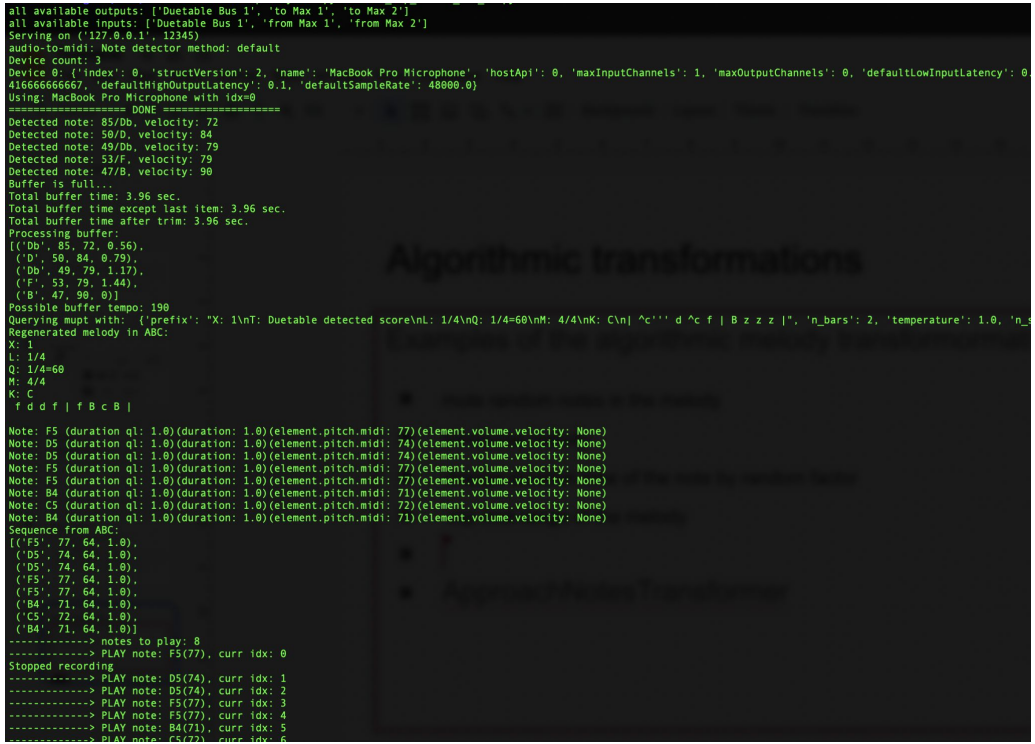
Algorithmic transformations

Examples of the algorithmic melody transformation:

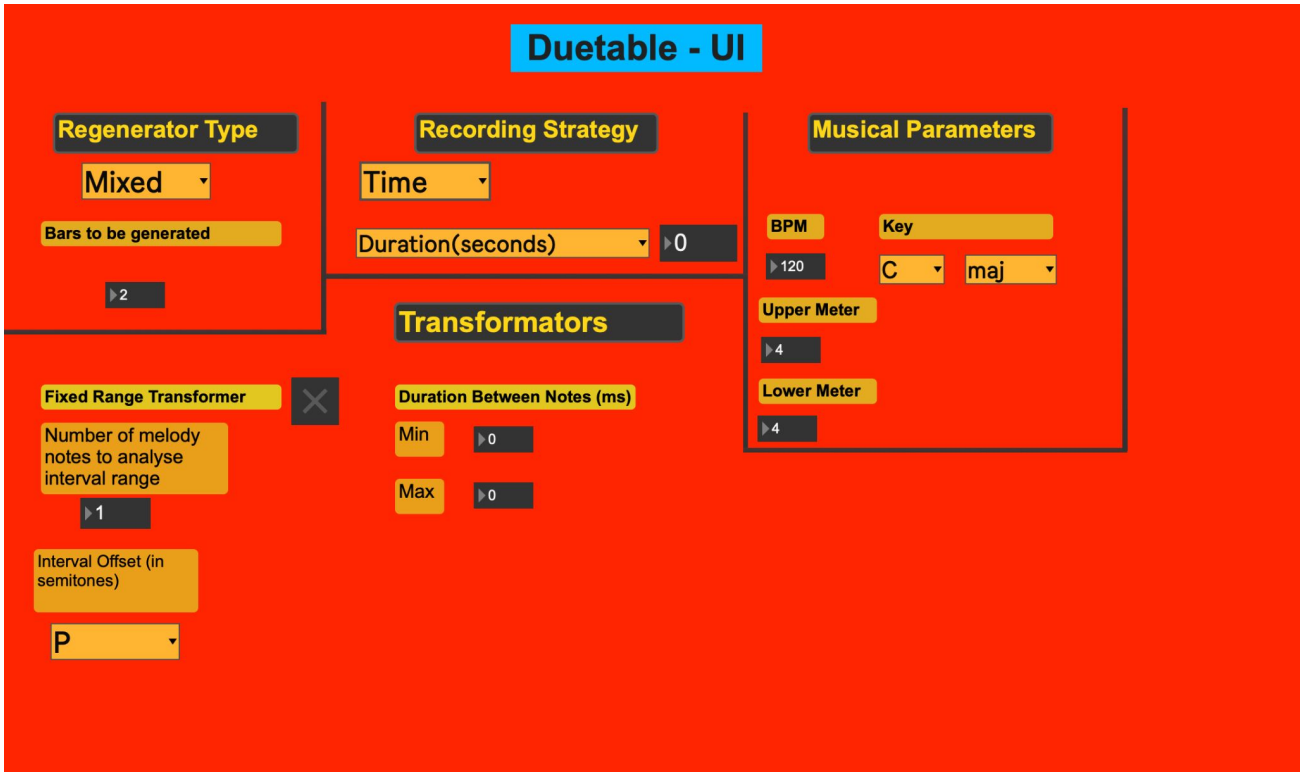
- mute random notes in the melody
- transpose notes in melody
- change duration time of the note by random factor
- clip midi range of the melody, or define ranges for parts of it
- create a half step ascend to each note in melody

Implementation

- Python 3.11
- Poetry
- PyAudio
- NumPy
- MidiToolKit
- Music21
- Mido
- Python Osc
- Max MSP


all available outputs: ['Duetable Bus 1', 'to Max 1', 'to Max 2']
all available inputs: ['Duetable Bus 1', 'from Max 1', 'from Max 2']
Serving on ('127.0.0.1', 12345)
audio-to-midi: Note detector method: default
Device count: 3
Device 0: {'index': 0, 'structVersion': 2, 'name': 'MacBook Pro Microphone', 'hostApi': 0, 'maxInputChannels': 1, 'maxOutputChannels': 0, 'defaultLowInputLatency': 0.
416666666667, 'defaultHighOutputLatency': 0.1, 'defaultSampleRate': 48000.0}
Using: MacBook Pro Microphone with idx=0
================== DONE ==================
Detected note: 85/Db, velocity: 72
Detected note: 50/D, velocity: 84
Detected note: 49/Db, velocity: 79
Detected note: 53/F, velocity: 79
Detected note: 47/B, velocity: 90
Buffer is full...
Total buffer time: 3.96 sec.
Total buffer time except last item: 3.96 sec.
Total buffer time after trim: 3.96 sec.
Processing buffer:
[('Db', 85, 72, 0.56),
('D', 50, 84, 0.79),
('Db', 49, 79, 1.17),
('F', 53, 79, 1.44),
('B', 47, 90, 0)]
Possible buffer tempo: 190
Querying mupt with: {'prefix': "X: 1\nT: Duetable detected score\nL: 1/4\nQ: 1/4=60\nM: 4/4\nK: C\n| ^c''' d ^c f | B z z z |", 'n_bars': 2, 'temperature': 1.0, 'n_s
Regenerated melody in ABC:
X: 1
L: 1/4
Q: 1/4=60
M: 4/4
K: C
f d d f | f B c B |

Note: F5 (duration ql: 1.0)(duration: 1.0)(element.pitch.midi: 77)(element.volume.velocity: None)
Note: D5 (duration ql: 1.0)(duration: 1.0)(element.pitch.midi: 74)(element.volume.velocity: None)
Note: D5 (duration ql: 1.0)(duration: 1.0)(element.pitch.midi: 74)(element.volume.velocity: None)
Note: F5 (duration ql: 1.0)(duration: 1.0)(element.pitch.midi: 77)(element.volume.velocity: None)
Note: F5 (duration ql: 1.0)(duration: 1.0)(element.pitch.midi: 77)(element.volume.velocity: None)
Note: B4 (duration ql: 1.0)(duration: 1.0)(element.pitch.midi: 71)(element.volume.velocity: None)
Note: C5 (duration ql: 1.0)(duration: 1.0)(element.pitch.midi: 72)(element.volume.velocity: None)
Note: B4 (duration ql: 1.0)(duration: 1.0)(element.pitch.midi: 71)(element.volume.velocity: None)
Sequence from ABC:
[('F5', 77, 64, 1.0),
('D5', 74, 64, 1.0),
('D5', 74, 64, 1.0),
('F5', 77, 64, 1.0),
('F5', 77, 64, 1.0),
('B4', 71, 64, 1.0),
('C5', 72, 64, 1.0),
('B4', 71, 64, 1.0)]
------------> notes to play: 8
------------> PLAY note: F5(77), curr idx: 0
Stopped recording
------------> PLAY note: D5(74), curr idx: 1
------------> PLAY note: D5(74), curr idx: 2
------------> PLAY note: F5(77), curr idx: 3
------------> PLAY note: F5(77), curr idx: 4
------------> PLAY note: B4(71), curr idx: 5
------------> PLAY note: C5(72), curr idx: 6


Duetable - UI
Regenerator Type
Mixed
Bars to be generated
2
Recording Strategy
Time
Duration(seconds)
0
Musical Parameters
BPM
120
Key
C
maj
Upper Meter
4
Lower Meter
4
Transformators
Fixed Range Transformer
Number of melody notes to analyse interval range
1
Interval Offset (in semitones)
P
Duration Between Notes (ms)
Min
0
Max
0

Demo

Resources

Github repo: https://github.com/l11ama/duetable/

Configuration

Multiple parameters can be provided on runtime to configure the system. These are:

- `recording_strategy` - how the midi buffer is filled. Options are:
  - NOTES - recording till amount of notes is reached and equal to `buffer_length`
  - TIME - recording till amount of time is reached and equal to `buffer_time`
  - NOTES_ONCE - recording only once till amount of notes is reached and equal to `buffer_length`
  - TIME_ONCE - recording only once till amount of time is reached and equal to `buffer_time`
- `record_when_playing` - if the system should record when playing
- `append_to_play_buffer` - if set will append to newly detected notes to previous ones
- `upper_meter` - upper meter for the regenerated melody
- `lower_meter` - lower meter for the regenerated melody
- `bpm` - tempo for the regenerated melody
- `sleep_with_note` - specify if the midi player has to sleep with the note duration or time calculated from tempo and meter
- `regenerator` - specify method for regenerating recorded midi notes, possible values are:
  - `HttpMuptRegenerator` - regenerates melody using MUPT API
  - `MuptWithMarkovChainRegenerator` - first regeneration is executed with MUTP API and all other iterations with MarkovChain
  - `MarkovChainRegenerator` - regenerates melody using MarkovChain
  - `DummyRegenerator` - returns the same melody as input
- `n_bars` - number of bars to generate with Mupt API