Final Project Worksheet
Chord-following front-end to bitKlavier 2
Mike O'Connor – December 2016
(mike@prairiehaven.com www.prairiehaven.com)

Artists Statement

My goal in this project was to find a way that I could change the tuning of bitKlavier based on a live response to the chord-position of what I was playing on a keyboard.

I was interested in this because the kind of music I play is almost entirely based on chord progressions, which means that the "key" of the music is changing from section to section. Thus, a single tuning (eg. Just or Pythagorean) through a composition will be quite out of tune much of the time. I wanted to experiment with an instrument that would change its base tuning as I played, so that I could get a feel for what it would be like in a live/improv setting.

Technical description

Ver. 1 - HarmoTools

The first attempt of the project used a Max For Live plugin called HarmoChordJ74, which is part of a larger set of free plugins called HarmoTools. In essence, the plugin attempts to determine what chord is being played moment to moment and broadcasts that chord to its HarmoTools siblings on other tracks in Ableton Live.

I think this approach might work quite well for a certain style of piano-player. But my open-octave boogie bassline approach caused HarmoChord to jitter all over the map, throwing all sorts of spurious key changes at bitKlavier.

Ver. 2 – Yamaha S90SX

In this version, I'm using my Yamaha keyboard to detect chord changes and processing the MIDI stream out of the keyboard into a sequence that can be used to drive a tuning preparation in bitKlavier. Here's how it works.

Yamaha keyboard

I picked a preset that had a very fast uniform bass arpegiator (MC_StandardDisco_XS) that sends repeated 16th notes, always on the tonic of the chord that has been detected (notes vary by octaves, but always on the tonic). I've set the Yamaha to 300 bpm, the fastest tempo setting available, because I want to reduce the latency between when the Yamaha senses the chord and bitKlavier is told about it. The stream of notes from the arp is sent on a separate MIDI channel (channel 2) than the notes that I play as performer (channel 1).

DAW (Ableton Live) preprocessing

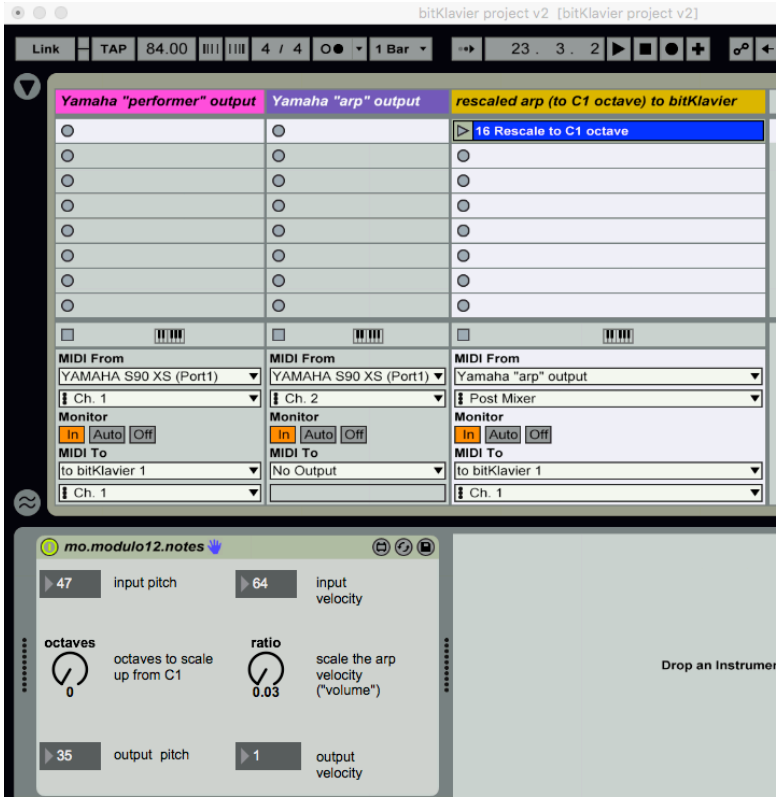
I route (channel 2) MIDI from the keyboard through a very simple patch, which normalizes the multi-octave arp output to a single octave before routing it on to bitKlavier. I also put octave-transposition and note-scaling in. My first-ever Max4Live patch.

The tracks:

Track 1 – live performance MIDI (channel 1) is passed directly out to the "to bitKlavier 1" destination

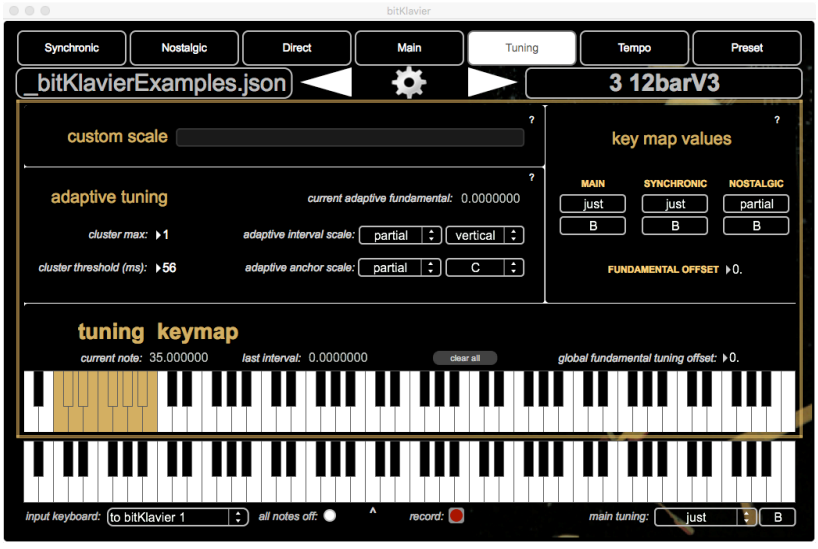
Track 2 – the rapid-fire, multi-octave tonic Disco bassline from the arp comes in and is rerouted to Track 3 for processing. NOTE: Tracks 2 and 3 can't be combined – doing so echoes the original multi-octave bass line out to bitKlavier, the whole thing we're trying to avoid.

Track 3 – Disco bassline is converted to a single octave (modulo 12), scaled up or down to a target octave and is velocity-trimmed (to quiet down the notes in bitKlavier – a to-do item would be to figure out a different way, because the notes still sound right now).



bitKlavier preparation

The C1 octave is dedicated to receiving the tuning notes from the arp. Each key is configured to set the base tuning of the scales to the note that is coming in ("B" in this example). Setting "input keyboard" to "to bitKlavier 1" is required to avoid echoing the original bass line directly from the keyboard.



Self Assessment	*Evaluation*	*Comparison to intention*
Art / aesthetics of result	*What's good and bad about the artistic output resulting from your project?* The Good - The patch kinda sorta works The Bad - It doesn't play very well unless the performer is willing to dramatically alter their technique	*How does this artistic output compare to what you imagined when you conceived the project?* My hope was to have fluid key changes based on chord changes. The target is the way the Yamaha keyboard plays – which is uncanny-fast/fluid. This is close, but not close enough to be any fun to play. The end of the demo video is a good example of a performance gone bad. I decided to leave it that way as a warning.
Programming	*What's good and bad about the way you programmed your project?* The Good - I learned a lot about both the Yamaha sequencer and bitKlavier through all the trial and error The Bad - I found lots of "wishlist" items for the bitKlavier developers.	*Did the project come out as you intended? How and why?* I think I came pretty close to my original intent. Especially after discovering how hard chord-detection is to do. Those Yamaha folks are wizard.