MUS 3460 FINAL PROJECT

John Brittain

5/12/2025

INTRODUCTION

For my final project in MUS 3460, I set out to build a complete modular and expandable MIDI-based music interface in Max/MSP. This project expands significantly on the work I began in Project 3, refining the usability, musical flexibility, and design of my patching system. The main objective was to create a tool that empowers musicians, especially those with little to no coding experience—to engage creatively with MIDI sequencing, generative music, and effects processing through an intuitive interface.

PROBLEM STATEMENT

Max patchers are powerful, but often inaccessible to musicians unfamiliar with programming. Many generative or MIDI-based patches are cluttered, hard to expand, and lack standardization. My goal was to design a cleaner, more modular, and musician-friendly system with plugin-style architecture and intuitive musical controls.

OVERVIEW

Below is the overview of the main patcher. It contains multiple larger sections. Three Live Step Sequencers which are an intuitive way to store and generate sequence data to be played. Below that is the Plugin section where there are a multitude of hardwired and modular plugins which can change and add effects to the MIDI data generated by the Live sequencers. Master Control on the right allows for the storing/recalling of master presets along with independent channel, instrument, and volume control over 5 master outputs. Each section works together to form a cohesive, flexible environment for music creation and experimentation.

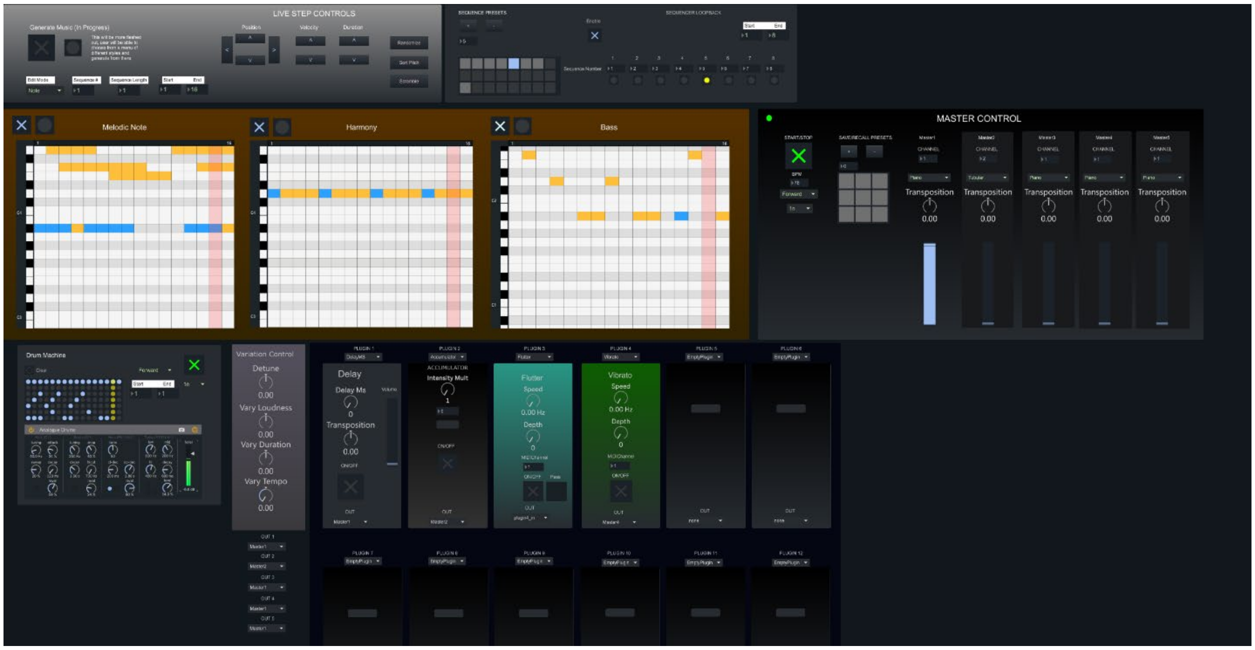


Figure 1 Overview of Main Patch

CONTENTS

LIVESTEP SEQUENCING

This section of the patch contains three minor sections, the LiveStep objects where the user can generate/create notes to be played sequence in a grid. LiveStep Controls which allows a user to have more direct control on the live step objects. And a sequence looper, which can playback defined and saved sequences in order and allows the user to create programmable songs.

LIVESTEP OBJECTS

An important object we learned about during this semester is the LiveStep object. Which is a powerful step-based sequencer which can store and play notes in sequence via a grid. Here is where the user will generate, create from scratch, and edit their MIDI sequences to later be played. Even better there are three of them, Melodic, Harmony, and Bass allowing the user to not just create single melodies, but full songs.

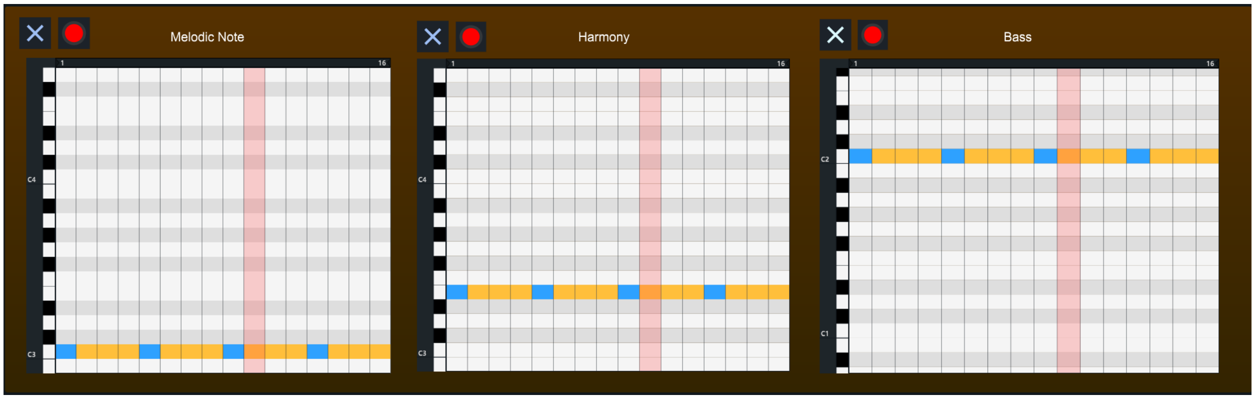
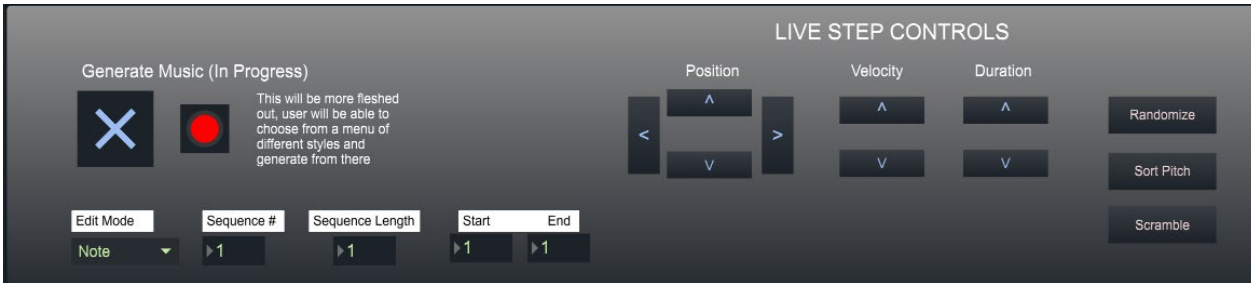


Figure 2 Three live step objects, Harmonic, Melodic, and Bass

LIVESTEP CONTROL

This minor section allows a user to control the LiveStep objects more directly using a variety of controls. First users are able to set the sequence length via the start/end inputs. Users can also control Edit mode, which changes which parameter, Pitch,Velocity,or Duration is visible and editable on the LiveStep object. Velocity up and down allows you to the control the general velocity of the LiveStep notes, Duration up and down allows you to control the overall duration of the notes. And the position controls allow a user to control the vertical/horizontal position of the notes within the LiveStep objects. Randomize, randomizes all of the data within the LiveStep objects, Scramble reorganizes the LiveStep information randomly(but does not create any new notes), and Sort sorts the LiveStep objects by pitch,velocity, or duration depending on the active view-mode.



This will be more fleshed out, user will be able to choose from a menu of different styles and generate from there

Edit Mode: Note
Sequence #: 1
Sequence Length: 1
Start: 1
End: 1

Figure 3 UI of Live Step Controls

SEQUENCING LOOPER

Each LiveStep object is connected to a preset system where users can save/recall sequences they chose to save. From there you can input the preset number into 1 or 8 available number inputs. After this the sequence looper will automatically playback the defined sequences in order. The user can define how long the start and end point of the sequence (from 1 to 8, 3 to 7, etc.). The sequencing looper can be enabled/disabled if the user does not want to loop through presets yet.

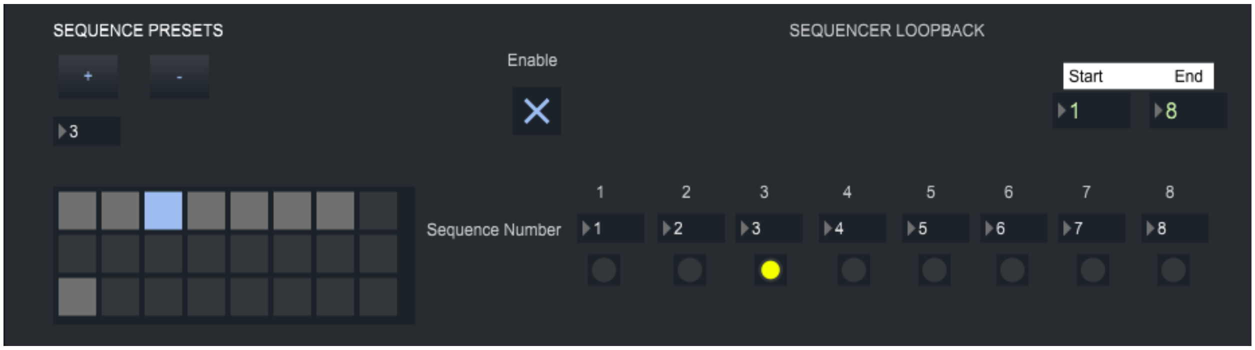


Figure 4 UI of Sequencing Looper

ALGORITHMIC MUSIC GENERATION

A more experimental and complicated feature of this patch is the ability to generate sequences randomly using an algorithm. However, this part of the project was not as fleshed out as the others. However, a functional version does exist to show off it’s potential integration. Future expansion includes support for multiple styles, and more musically coherent results.

MASTER CONTROL

This section allows direct user control over the main patch. Users can start/stop the patch via a toggle object, set the BPM, and control the counter object directly. Users can also save/recall master presets which will recall the entire state of the patch, Plugins, settings, outputs etc. On top of those controls users also have control over the channel, volume, and instrument of 5 independent Master Out tracks.

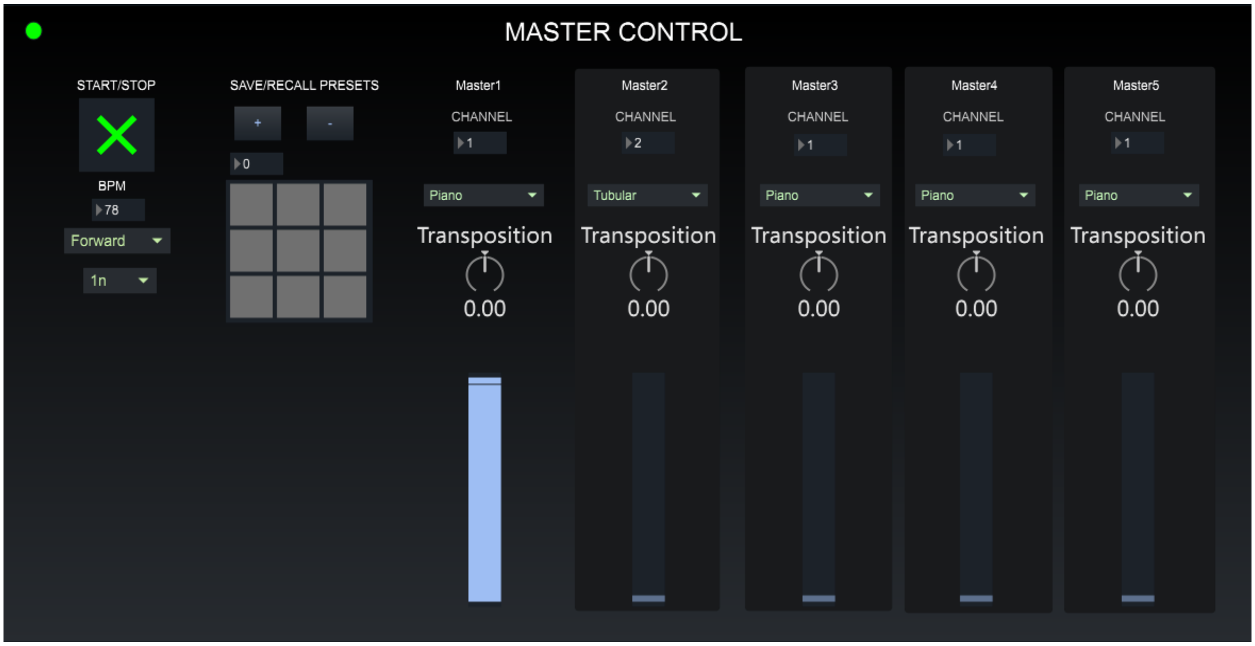
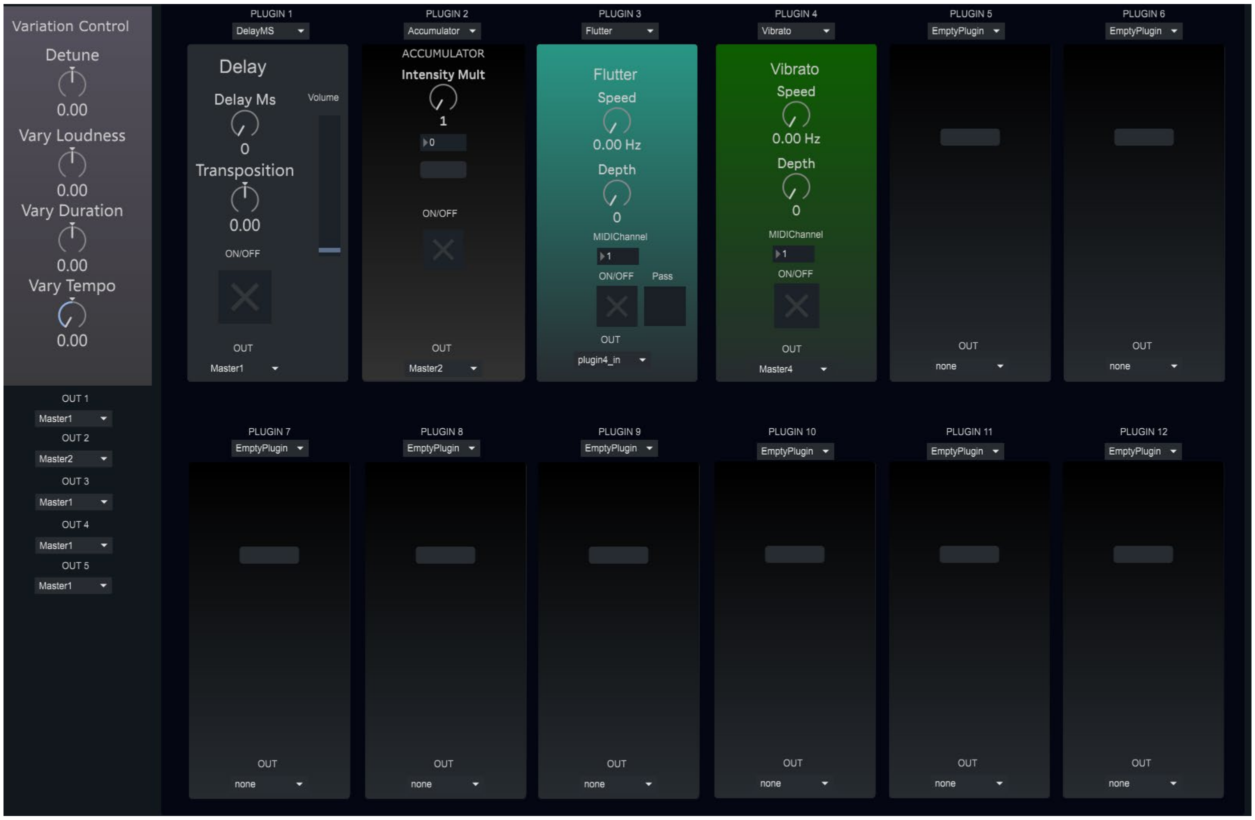


Figure 5 UI Of Master Control section

MODULAR PLUGINS

I created a series of modular plugins which can do a variety of MIDI effects processing. They are BPATCHER-based plugins with standardized size, and input/outputs. Meaning they can be loaded in and routed anywhere in the patch. The user dynamically loads plugins into 12 predefined slots through a dropdown menu. From there, the user selects where they want to output the data. In can be passed into another plugin, or routed into one of 5 master outs.



Detune 0.00
Vary Loudness 0.00
Vary Duration 0.00
Vary Tempo 0.00

OUT 1 Master1
OUT 2 Master2
OUT 3 Master1
OUT 4 Master1
OUT 5 Master1

PLUGIN 1 DelayMS
Delay
Delay Ms 0
Transposition 0.00
OUT Master1

PLUGIN 2 Accumulator
ACCUMULATOR
Intensity Mult 1
0
OUT Master2

PLUGIN 3 Flutter
Flutter
Speed 0.00 Hz
Depth 0
MIDIChannel 1
OUT plugin4_in

PLUGIN 4 Vibrato
Vibrato
Speed 0.00 Hz
Depth 0
MIDIChannel 1
OUT Master4

PLUGIN 5 EmptyPlugin
OUT none

PLUGIN 6 EmptyPlugin
OUT none

PLUGIN 7 EmptyPlugin
OUT none

PLUGIN 8 EmptyPlugin
OUT none

PLUGIN 9 EmptyPlugin
OUT none

PLUGIN 10 EmptyPlugin
OUT none

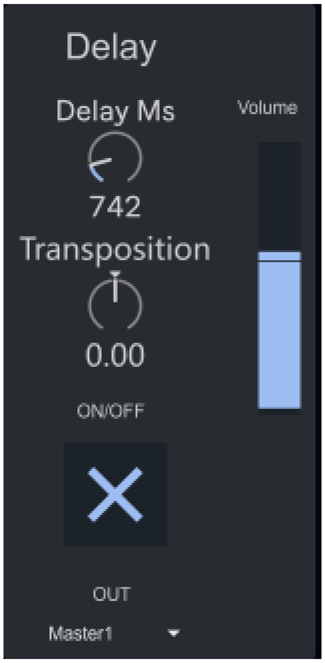
PLUGIN 11 EmptyPlugin
OUT none

PLUGIN 12 EmptyPlugin
OUT none

Figure 6 UI Showing 12 Plugin Slots

DELAYMS

Adjustable delay set by milliseconds, also allows transposition and independent volume control of the MIDI data.



Delay Ms 742

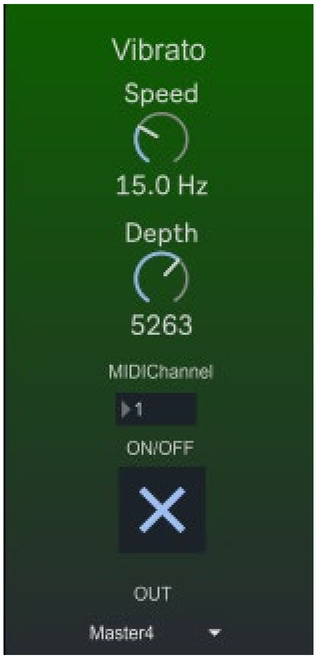
Transposition 0.00

Master1

Figure 7 DelayMS Plugin

VIBRATO

Standard Vibrato effect which uses the Xbend attributes. Users can also wire the Xbend information to other plugins or/also choose the MIDI channel effected.



Speed 15.0 Hz

Depth 5263

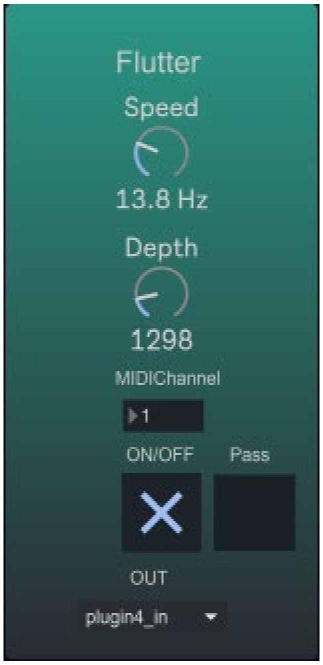
MIDIChannel 1

Master4

Figure 8 UI Vibrato Plugin

FLUTTER

Standard Flutter effect which uses the Xbend attribute. Users can also pass the Xbend information to other plugins or output the Xbend info to the chosen MIDI channel.

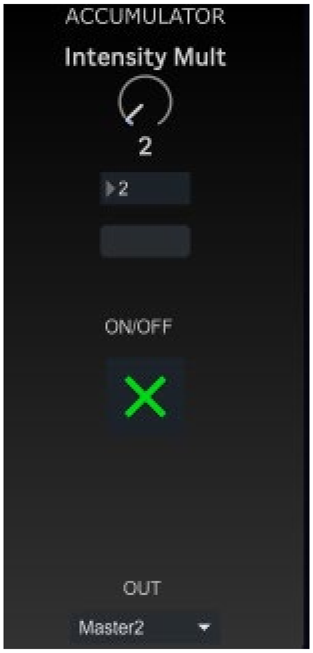


Flutter
Speed 13.8 Hz
Depth 1298
MIDIChannel 1
ON/OFF ☒
Pass □
OUT plugin4_in

Figure 9 UI Flutter Plugin

ACCUMULATOR

Accumulates a value over time(synced up with the master clock), users can accumulate BPM, pitch, velocity, or any other data. The intensity of the accumulation can be controlled via a dial.



ACCUMULATOR
Intensity Mult 2
2
ON/OFF ☒
OUT Master2

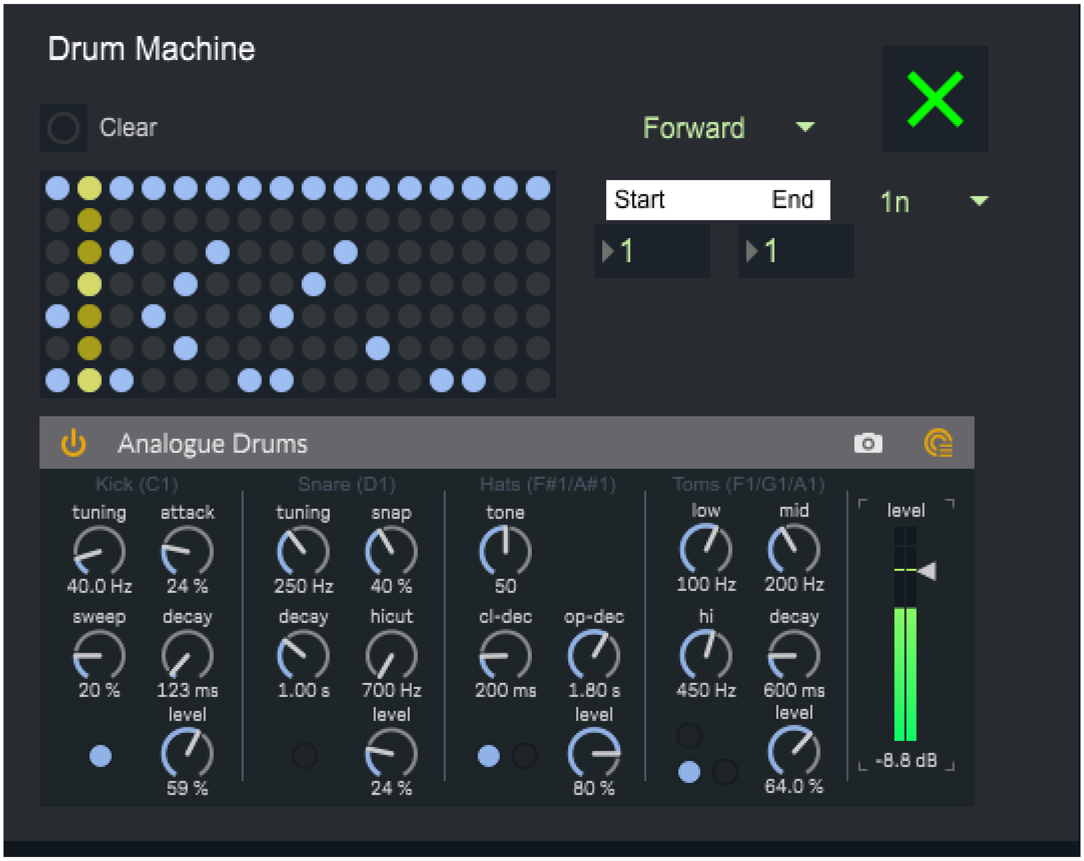
Figure 10 UI Accumulator Plugin

EXPANDABLE

Because of the standardized nature of these plugins, they can be expanded upon infinitely. Other users/programmers can create their own plugins simply and easily integrate them into the main patch.

HARDWIRED PLUGINS

Not all of the plugins within the patch are modular, some are hardwired directly into the main patch.



Forward

Start 1 End 1

1n

Kick (C1)
tuning 40.0 Hz
attack 24 %
sweep 20 %
decay 123 ms
level 59 %

Snare (D1)
tuning 250 Hz
snap 40 %
decay 1.00 s
hicut 700 Hz
level 24 %

Hats (F#1/A#1)
tone 50
cl-dec 200 ms
op-dec 1.80 s
level 80 %

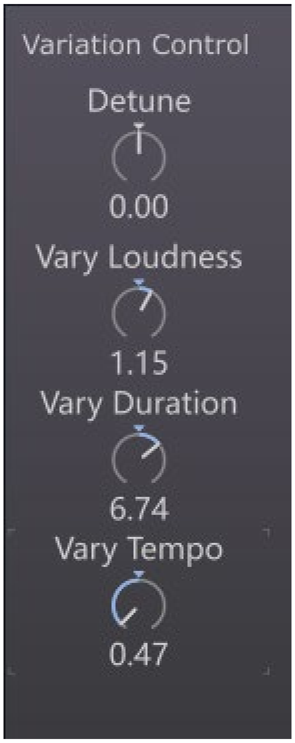
Toms (F1/G1/A1)
low 100 Hz
mid 200 Hz
hi 450 Hz
decay 600 ms
level 64.0 %

level -8.8 dB

Figure 11 UI Drum Machine Plugin

RANDOM VARIATION

Takes the MIDI data generated from the LiveStep sequencers and applied a random twist to each attribute, Pitch, Velocity, Duration, and even tempo can be varied randomly. Controlled via dials. This when used subtly can introduce a more human-like and imperfect performance, or a complete train wreck of random noises.



Variation Control

Detune 0.00

Vary Loudness 1.15

Vary Duration 6.74

Vary Tempo 0.47

Figure 12 UI Random Variation Plugin

DRUM MACHINE

Standard Matrix controlled drum machine using one of max's available plugins to generate the actual drum sounds.

FUTURE DIRECTIONS

Though this patch is finalized, there are still many areas for improvement and sections to be expanded upon. Such as;

- Support for VST plugins into modular chain
- Expand algorithmic music generator for other genre's / uses
- Fully custom in-house drum machine
- More plugin and effects, tremolo, reverb etc.
- Add support for raw signals on top of MIDI data.
- Ability to save/recall entire sequence loops

ULTIMATELY

This project represents the culmination of my work in Max/MSP this semester. It combines interface design, modular plugin architecture, algorithmic composition, and sequencing into a single unified tool. The resulting patch is approachable, powerful, and designed to grow as my needs evolve. Whether used for live performance, composition, or experimentation, it offers a strong foundation for music-making and invites further expansion by others.