

MUMT 306 Final Project – The Lofi Hip-hop Machine

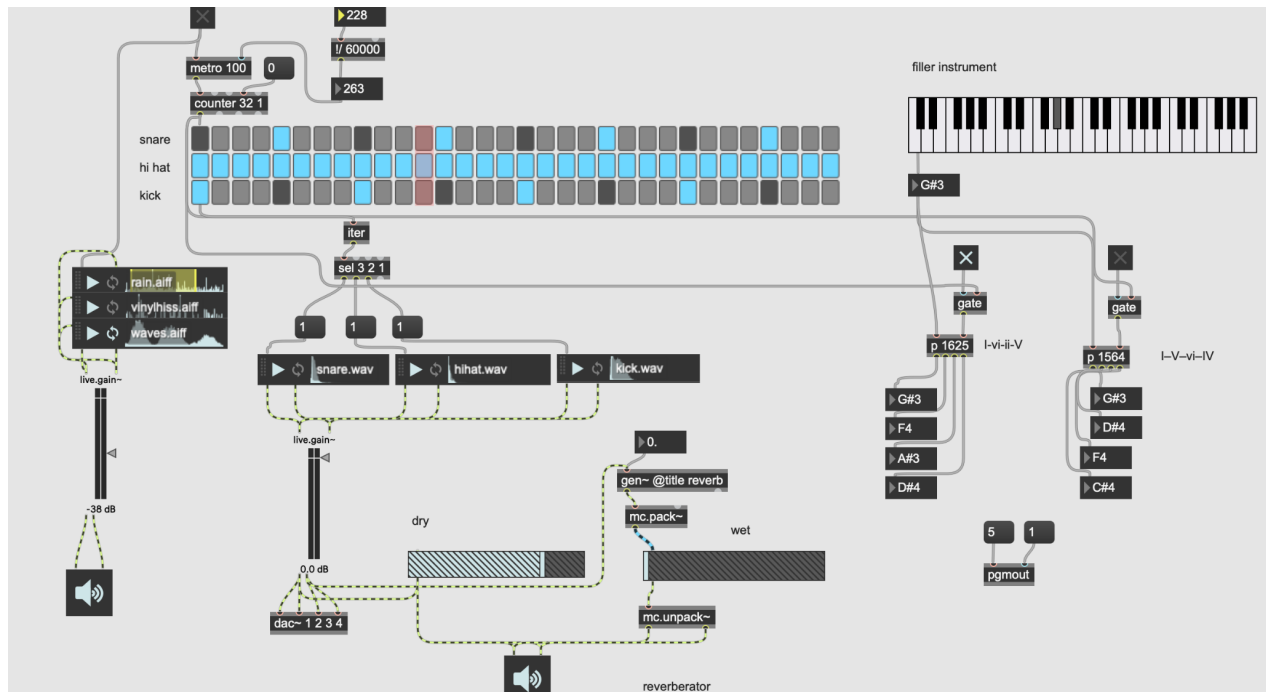
– Yanou Li

Introduction/motivation:

Have you ever come across this anime girl studying with a pair of headphones on YouTube? For me, this YouTube channel called lofi girl which livestreams lofi hip hop music has accompanied me for countless late night study sessions. Lofi hip hop is also called chill hop, it is a form of downtempo music that combines elements of hip hop and chill-out music. I like to have it on as background music while I'm studying because I found the perfect balance of the chill-out and upbeat vibes helps me stay awake and focused. Therefore, for this final project I decided to make a lofi hip hop beat maker so that I'll have a more interactive and fun way to play my study music.



Design

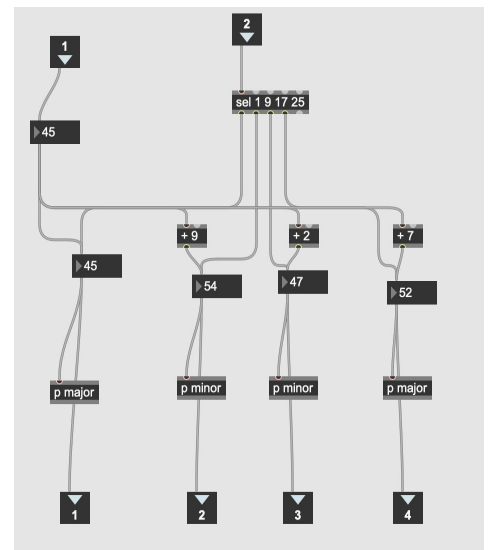
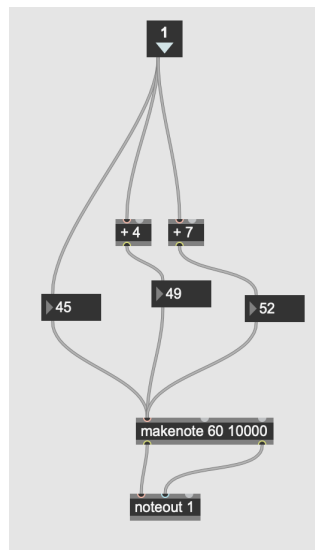


The project is fully implemented in Max/Msp and it includes a 32-step drum sequencer and also provides additional features for other lofi hip hop elements. The Max patch can be divided into 3 sections. A drum sequencer, a filler instrument channel and a reverberator. To add more lofi vibe, there is also a collection of various white noises that the user can choose from to play in the background. The patch is designed to be as interactive as possible. The user can control the tempo of the beat by selecting a BPM value. Then it will be converted to milliseconds and be fed into a metro object. The drum sequencer simulates a real life drum sequencer and allows the users to create any pattern they want. A live.grid object is used to make the drum sequencer interface neater and easier to use. However, due to the variation and complication of the melody part, the filler instrument is only semi-interactive. The filler instrument panel allows the user to choose from some predefined chord progression patterns, a key from the piano keyboard interface, and will play that chosen chord progression in the selected key. The reverberator consists of different kinds of filters that are implemented as subpatches. In the main user interface, the level of the reverb effect can be controlled through a slider.

Subpatches

The Chord progression generator

The arpeggiator exercise we did for homework is very helpful when implementing this. This patch takes the midi number of the input note and computes the rest of the chords in the progression.



The Schroder Reverberator

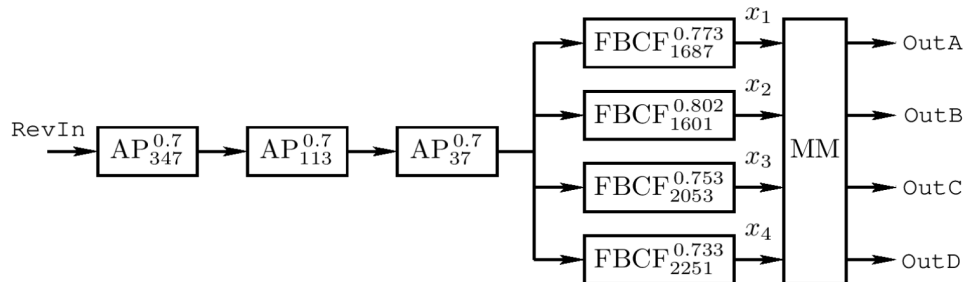
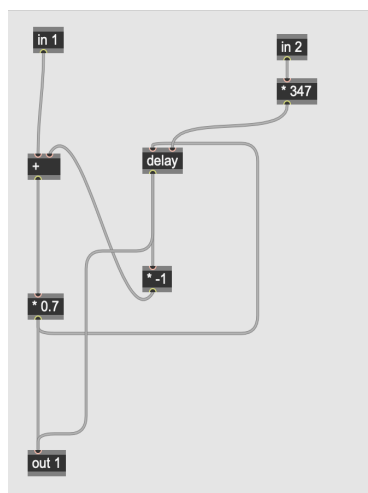
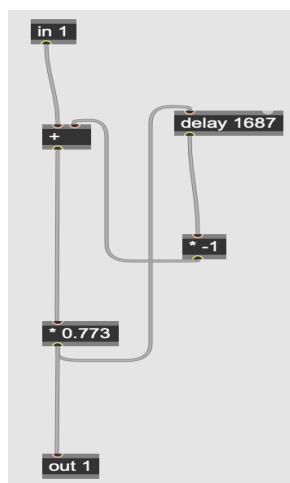


Figure: A Schroeder reverberator we will call JCREV developed by Prof. John Chowning, founding director of CCRMA (drawn from a 1972 MUS10 software listing, where MUS10 was an "acoustic compiler" language descended from Music V [308]).

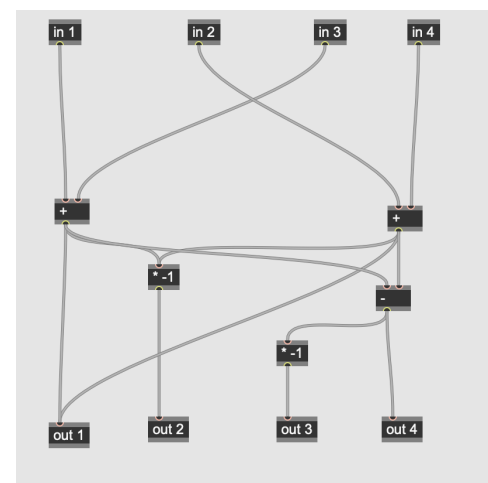
To implement the reverberation effect, I followed the model developed by Prof. John Chowning from the CCRMA. As the figure shows, this model consists of a series connection of several allpass filters, a parallel bank of feedback comb filters and a mixing matrix. An allpass filter is a signal processing filter that passes all frequencies equally in gain, but changes the phase relationship among various frequencies. A comb filter is a filter implemented by adding a delayed version of a signal to itself. Here are the subpatches I made for each of the elements shown in the figure.



Allpass filter



Comb filter



signal mixer

Limitations

The reverberator doesn't work for the filler instrument since Schroeder reverberator only works for audio signals and the filler instrument is implemented with built in Max/msp objects.

Conclusion

Overall, I'm satisfied with the result I got. I was able to make some decent sounding beats with the Max patcher. However, there is definitely still a lot of room for improvements. For example, an additional Arduino interface can be made for the control panels (tempo, level of reverberation, etc..) to make the project more interactive and interesting. The filler instrument can also be improved to make more various and complex melodies.