



PERSONAL DRUMS SEQUENCER

Culmination Project Proposal

by

Daytona van Eijck

New York City College of Technology
Spring 2019
ENT4499

Table of Contents

| | page |
|---------------------------------------|------|
| 1. Project Description | 3 |
| 2. Methods | 4 |
| i. Basic Representation of Instrument | 5 |
| 3. Materials | 6 |
| 4. Budget | 7 |
| 5. Deliverables | 8 |
| 6. Calendar | 9 |

Project Description

As an Emerging Media student at City Tech, I have always been interested in combining my life-long interest of music and sound with it's technical aspects. In order to push the boundaries of our understanding of modern day, you have to understand the technology behind it and see where the opportunities lie for inventive ideas. Technology opens up new doors for creating music in original ways, for my Culmination Project therefore I want to create a software instrument to begin thinking outside the realm of traditional instrumentation.

My instrument will function as a drum sequencer and sampler which takes in analog input (i.e. vocals, live recordings). Instead of preset sounds you will make a completely personal sequencer. This project will be both a functional instrument which can be used in the creation of my own music, as well as that of my peers, and a fun experiment in a field that I am passionate in.

Methods

In order to create my instrument I will be using the program Max/MSP/Jitter by Cycling'74. This is a programming language specifically intended for music and multi-media which allows you to visually create programs using objects. This software will be most compatible for my project, considering it allows for the manipulation and use of .wav files and my instrument will record live input using in .wav format.

As for the interface of my instrument, I will use the iconic Roland TR-808 as inspiration. In my opinion, this is the best organized layout for a drum sequencer and is used for decades as the standard for electronic drum machines. You will be able to record up to 16 individual sounds which can be manipulated separately. In its basic mode you can program in 4 measures of 4, meaning 16 steps per sound.

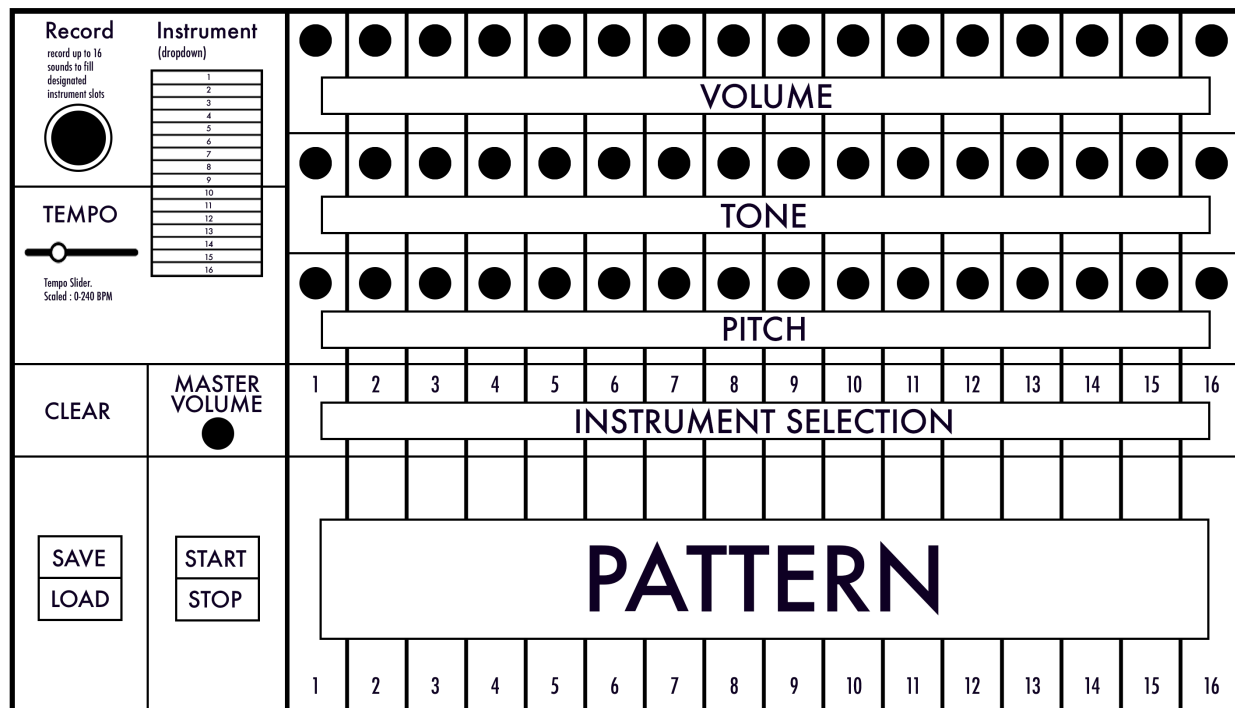
The sound input will be recorded live through an audio interface (i.e. Focusrite Scarlett) which converts analog input to a digital format, in this case a .wav files. The sounds recorded in my instrument will be able to be manipulated in the dimensions of:

- individual sound level
- individual pitch
- individual tone (lo-pass filter/hi-pass filter)
- master level
- master tempo (bpm)

After creating a full drum patch, users will be able to save their patch as a preset and load it at any moment. The amount of presets will be determined during the making of the instrument as constraints will occur on CPU and program storage size. The instrument will eventually be able to be exported as a VST (virtual studio technology) and be compatible with most Digital Audio Workstations.

I have a basic understanding of Max/MSP/Jitter, but will need to research the methods of putting together this instrument/sequencer, how to communicate with an audio interface, store/load sounds, and manipulate sounds. The graphics of the interface will be made in Adobe Illustrator and can be imported into Max/MSP/Jitter.

Basic Representation of Instrument



Materials

Software:

- Max/MSP/Jitter
- Adobe Illustrator

Hardware:

- Audio Interface (analog to digital converter)
- USB Type B cable
- Microphone
- XLR cable
- Headphones/Monitors

Budget

| | |
|----------------------------------|------------------|
| Max/MSP/Jitter software | \$ 200 |
| Max/MSP/Jitter for Music (ebook) | \$ 15.80 |
| | <hr/> |
| <u>Total</u> | <u>\$ 215.80</u> |

Deliverables

- Project Description Poster
- Schematics/Drawings
- Production Calendar
- Functioning Prototype I (with Sequencer)
- Functioning Prototype II (with sound manipulation features)
- Interface Graphics
- Instrument exported a VST
- OpenLab Portfolio

Calendar

| | |
|----------------|--|
| <u>Week 1</u> | Research on Max/MSP/Jitter. Methods on audio conversion and integration. |
| <u>Week 2</u> | Start coding for back-end of instrument. Basic drum-sequencer. |
| <u>Week 3</u> | Prototype functioning: sequencer with preloaded .wav files. Continue coding for back-end. Communication of Max/MSP/Jitter with live audio from interface. |
| <u>Week 4</u> | Continue coding for back-end. Implementation of live sounds into sequencer. |
| <u>Week 5</u> | Functioning basic Prototype: Sequencer with live sounds. |
| <u>Week 6</u> | Continue coding for back-end. Manipulation of individual sounds (level, tone, pitch, potential other functions) |
| <u>Week 7</u> | Continue coding for back-end. Manipulation of master (level, bpm, potential other functions) |
| <u>Week 8</u> | Functioning basic Prototype: Sequencer with live sounds and manipulation. |
| <u>Week 9</u> | Continue coding for back-end. Patch storage for presets and re-loading. |
| <u>Week 10</u> | Complete functioning back-end. Start Interface Graphic sketches. |
| <u>Week 11</u> | Finish Interface Graphics Assemble final Instrument. |
| <u>Week 12</u> | Export Instrument in VST format, compatibility with DAW's. |
| <u>Week 13</u> | Presentations |