Overview

- Problem Identification
- Project Goals
- Team Breakdown
- DTW
- Pure Data
- Project Specifications
- Implementation
- Digital Effects
- Schedules
- Summary

Problem Identification

- Guitar effect pedals physically restrain the guitar player
- Effect pedals require presence of mind on the part of the performer
- Analog and digital effects pedals are costly

Project Goals

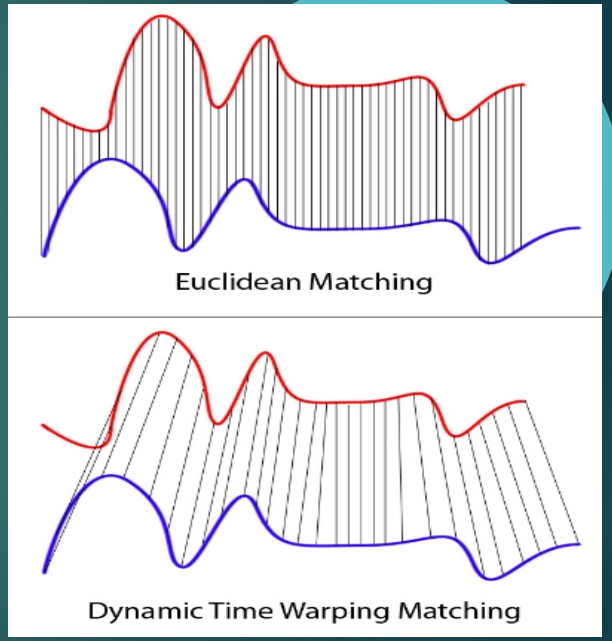
- Develop a software platform to analyze a time sequence of notes and trigger guitar effects as instructed
- Trigger on a designated instance that occurs multiple times throughout the song
- Mitigate the latency between a detected match and the triggering event
- Process sequences up to 10 note onsets per second

Team Breakdown

Bryan Guner	Team Lead - Develop a protocol for digital signal processing of the guitar signal in order to create a time-sequential record of the frequency content of the guitar signal and a comparison between pre recorded songs and live performances.
Ralph Quinto	Software Engineer - Research for possible programming platforms. Responsible for reading electric guitar signals, creating the signal analysis patches in pure data, and triggering digital guitar effects.
Haley Scott	Architectural Manager - Responsible for ensuring successful integration of project components, researching system methods, designing digital effects, project and organizational management.

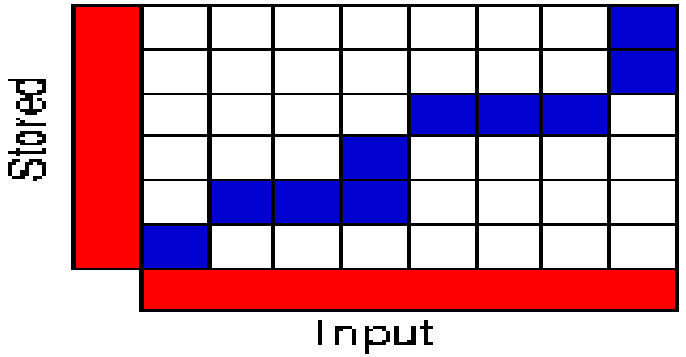
Dynamic Time Warping

- Algorithm for the measurement of similarity between two temporal sequences
- Calculates an optimal match in the form of a distance that is the sum of localized cost functions



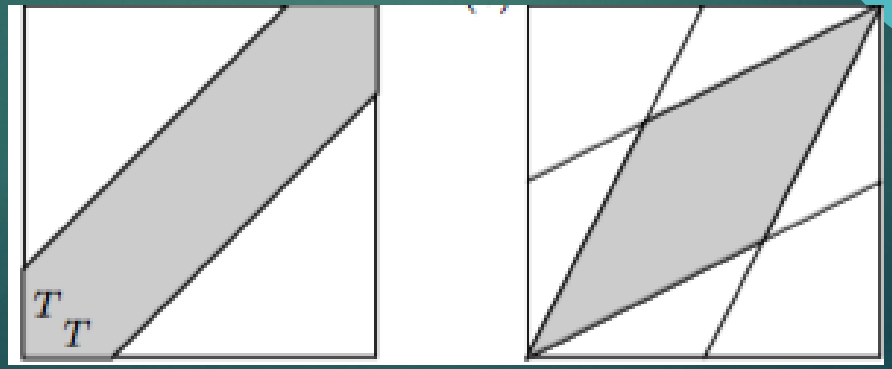
DTW Path

- The process can be thought of conceptually as arranging the two sequences on the sides of a grid
- Each cell within the grid will be filled in with a distance measure comparing the corresponding elements of the two sequences



How DTW Works

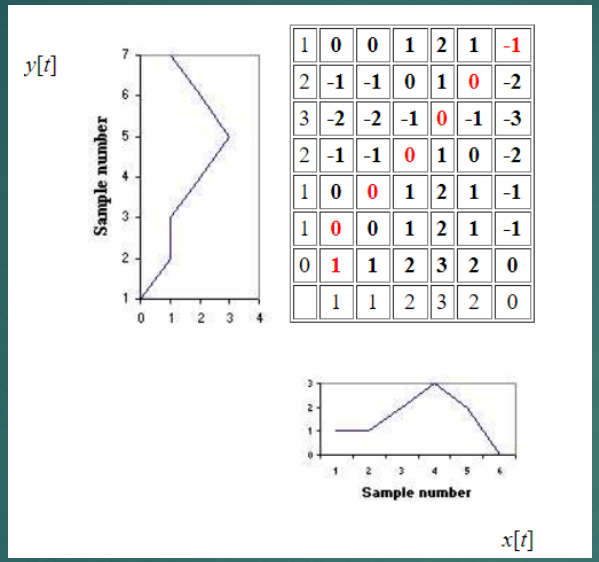
- To find the best path through the grid, we search for a path that minimizes the total distance between them
- Without optimizations, all possible paths through the grid are calculated and a minimum is selected



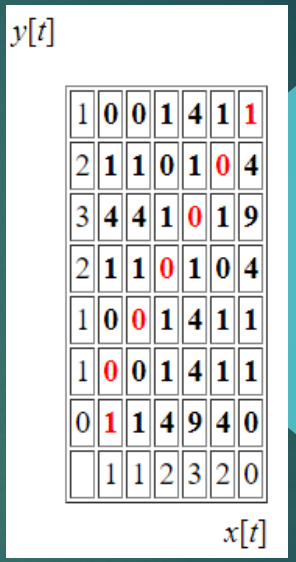
Why DTW?

- DTW is relatively insensitive to time-scale contraction or dilation in either the database or query signals
- A slightly erroneous performance will register a match as long as the section is the closest match to the database sequence relative to what's been processed so far
- The algorithm is commonplace in most speech dictation software and has a wide scope of applications such as gesture recognition

Basic Dynamic Time Warping



1	0	0	1	2	1	-1
2	-1	-1	0	1	0	-2
3	-2	-2	-1	0	-1	-3
2	-1	-1	0	1	0	-2
1	0	0	1	2	1	-1
1	0	0	1	2	1	-1
0	1	1	2	3	2	0
	1	1	2	3	2	0



1	0	0	1	4	1	1
2	1	1	0	1	0	4
3	4	4	1	0	1	9
2	1	1	0	1	0	4
1	0	0	1	4	1	1
1	0	0	1	4	1	1
0	1	1	4	9	4	0
	1	1	2	3	2	0

3	**0** **B: 1**	**0**	**1**
2	**0** **B: 1**	**0** **B: 2** **L: 1** **BL: 1**	**1** **B: 7** **L: 2** **BL: 2½**
1	**1**	**1** **L: 2**	**4** **L: 6**
	1	2	3

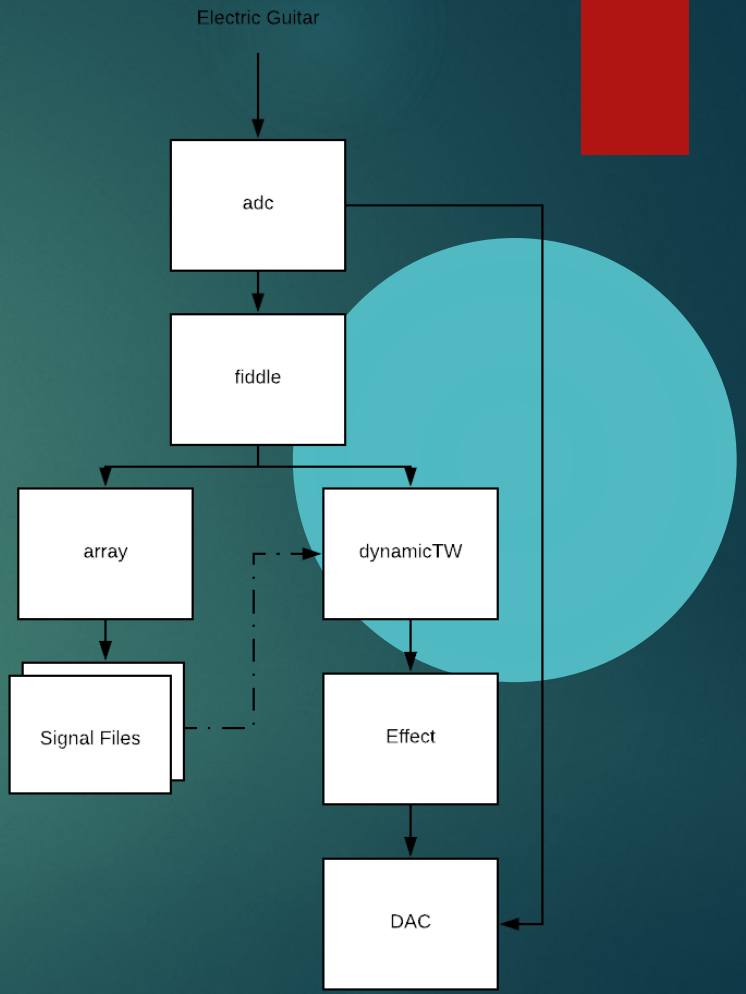
Background (Pure Data)

- Visual programming language (LabView)
  - Objects are linked together to model the flow of control and audio
  - Designed for creating interactive computer music and multimedia works
    - Generate Waveforms
    - Perform Signal Analysis
- Modular code base
  - Externals could be generated using C, C++, Python, Java, and many more
- Open source project listed under a modified BSD License
  - all distributed copies of the source code must contain the BSD license

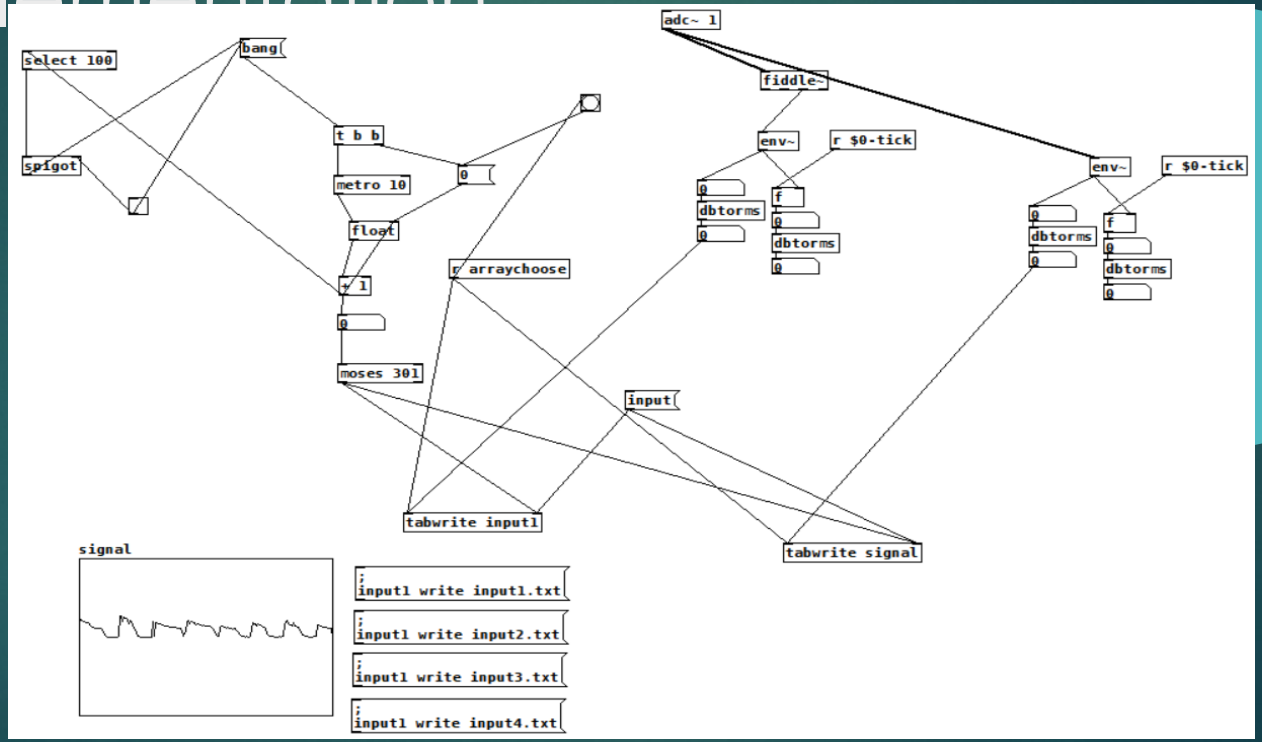
Project Specs (PD & Physical Components)

- Pure Data Specs
  - Sampling rate of 44,100 Hz
  - Fiddle uses 1024 most recent samples to produce midi data
- Electric Guitar: Ibanez RG5EX1
  - Bridge Pickup: Infinity 4
    - Magnet: Ceramic
    - DC Resistance: 15.6 KΩ
  - Gauges: .009/.011/.016/.024/.032/.042

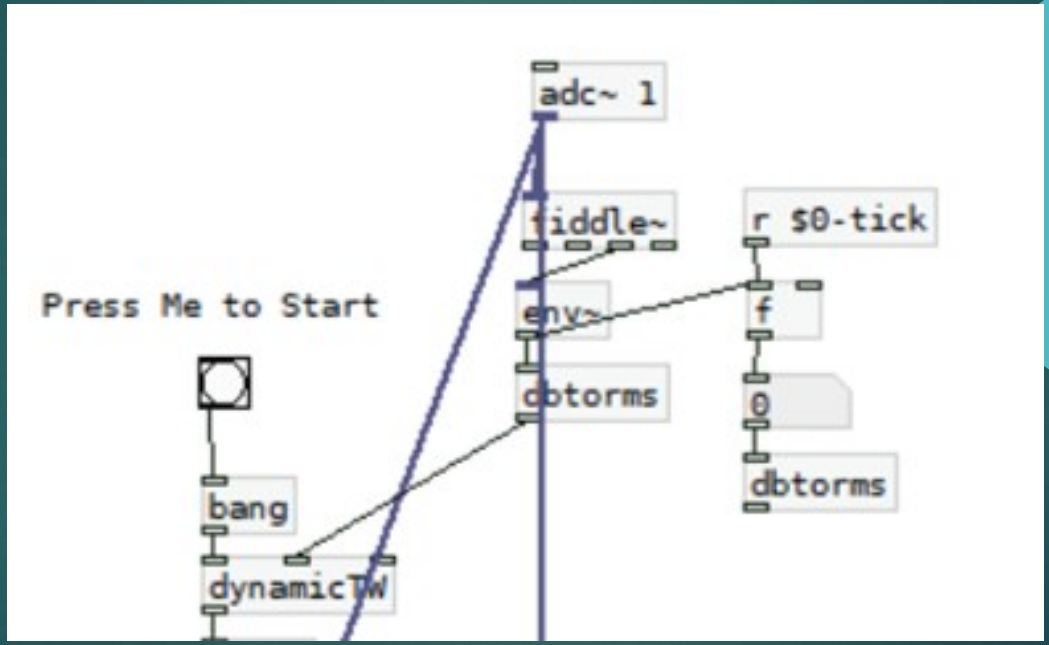
Block Diagram of Simplified PD Patch



Current PD Recording Implementation



Current PD Implementation



Testing and Validation

- Tested different sequences
  - Single Notes
    - Consistent triggering
  - Chords
    - Inconsistent triggering
- Record system clock at input and triggering
  - Took difference to measure latency
    - 1-2 ms
- 44100/1024 = 44 notes per second

Issues With Approach

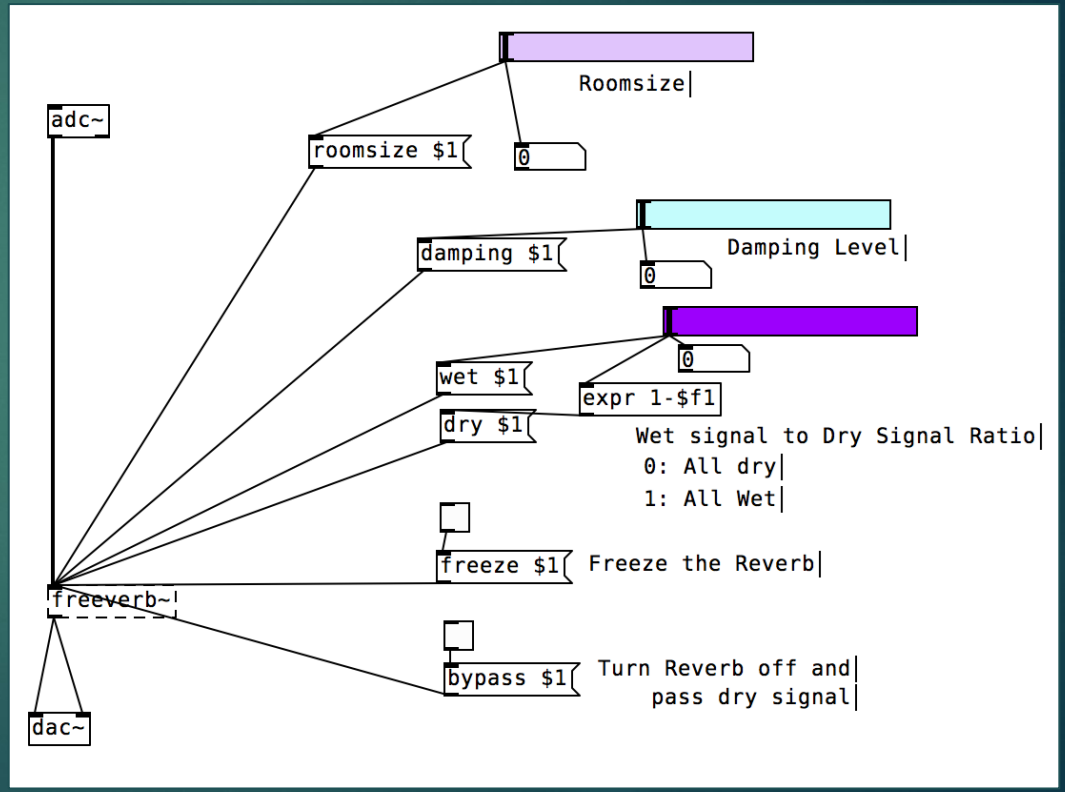
- Currently triggers on the first instance of trigger sequence
- Accuracy in chord detection (trigger sequence length)
- Match detection versus actual desired trigger point
- User familiarity with PD / ease of use

Future Improvements

- Subsequence tracking over DTW matching
- Application to control actions other than guitar effects
- Interface kit for analog effect pedals
- User GUI external to PD patch
- Hybrid of other candidate techniques to serve as false trigger fail safe

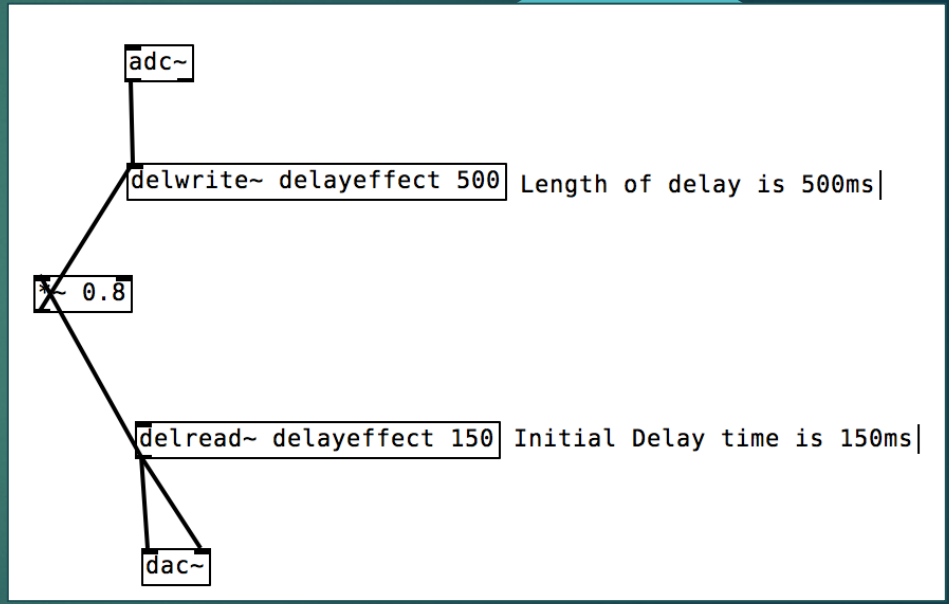
Reverb Effect

- Creates the sound of a performance in a concert hall
- Mirrors a large number of reflections to build up and then decay



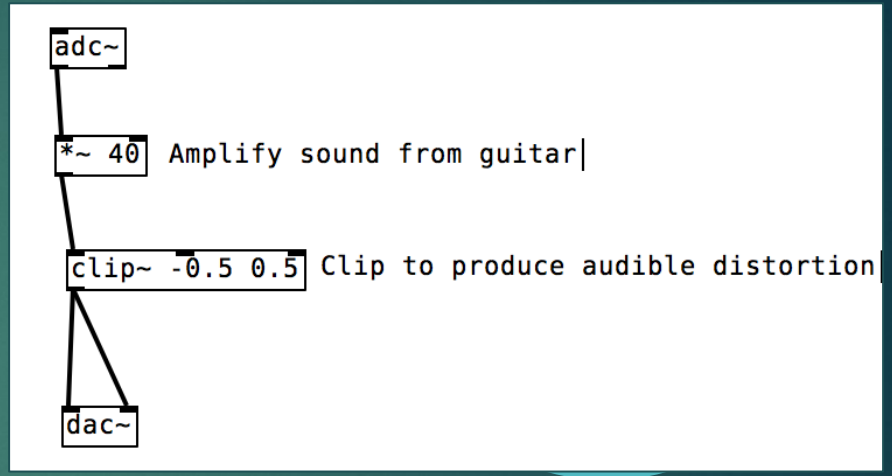
Delay Effect

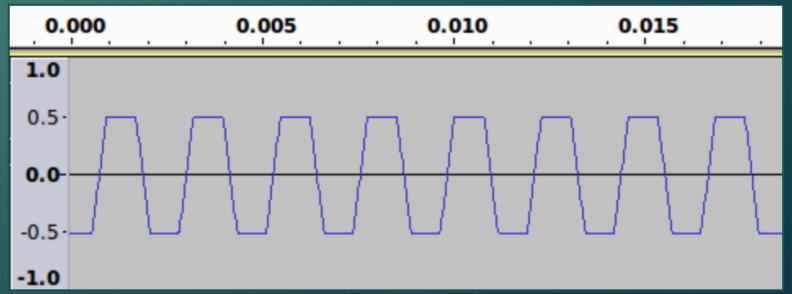
- Creates the sound of a repeating, decaying echo
- Delwrite block allocates memory for a delay line
- Delread block reads the signal from a delay line



Fuzz Effect

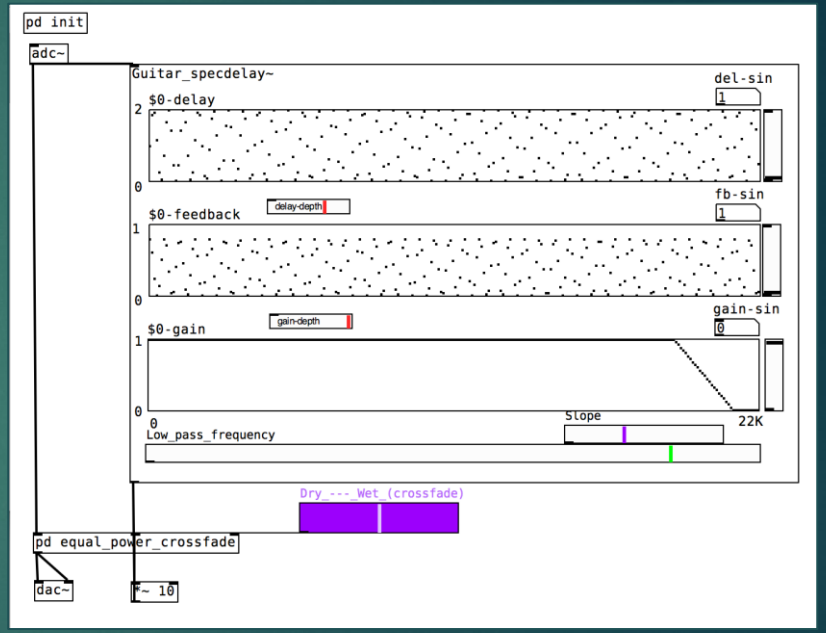
- Creates the sound of a distorted, heavier guitar
- Clip block restricts a signal to lie between two limits



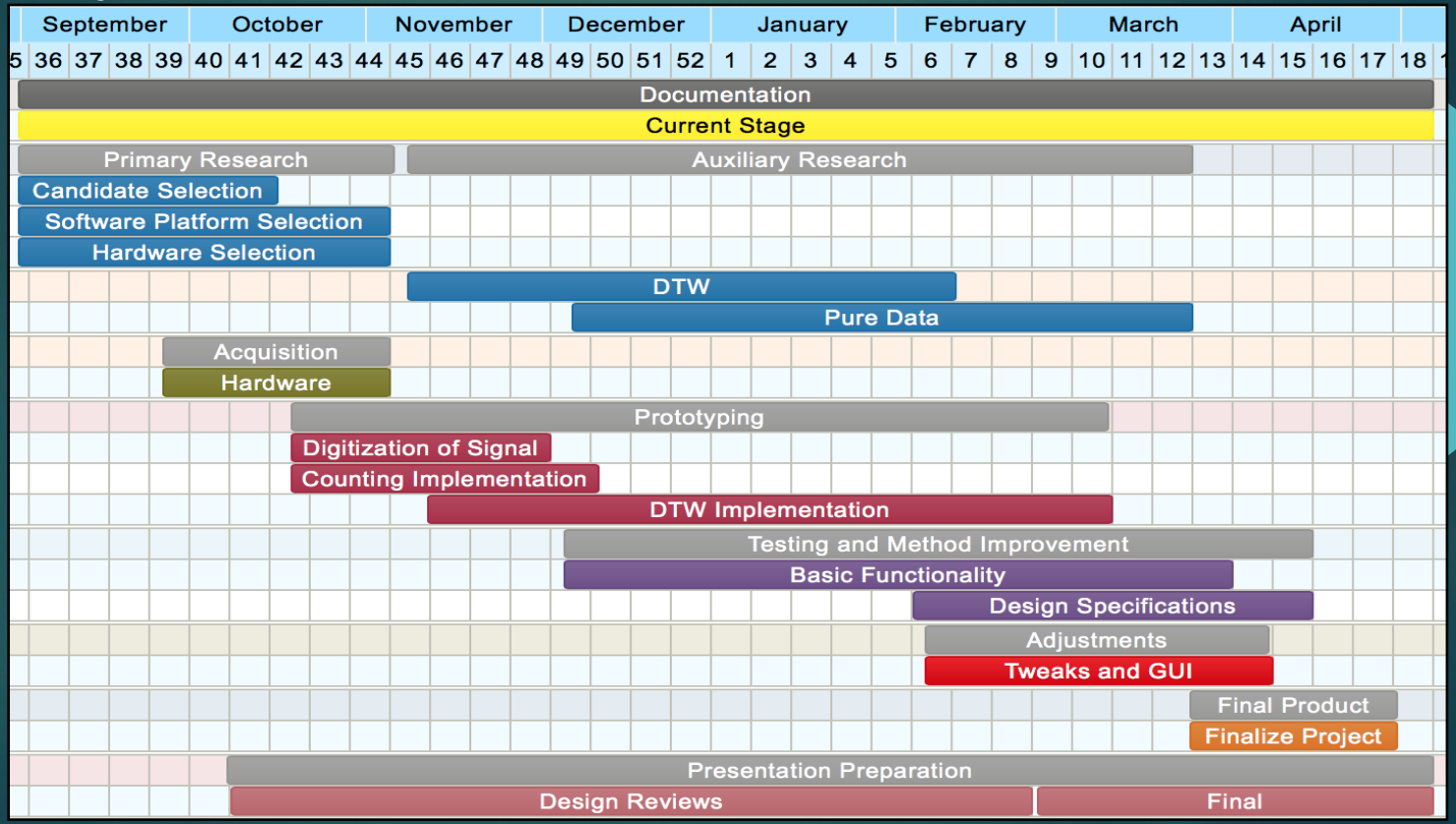


Spectral Delay Effect

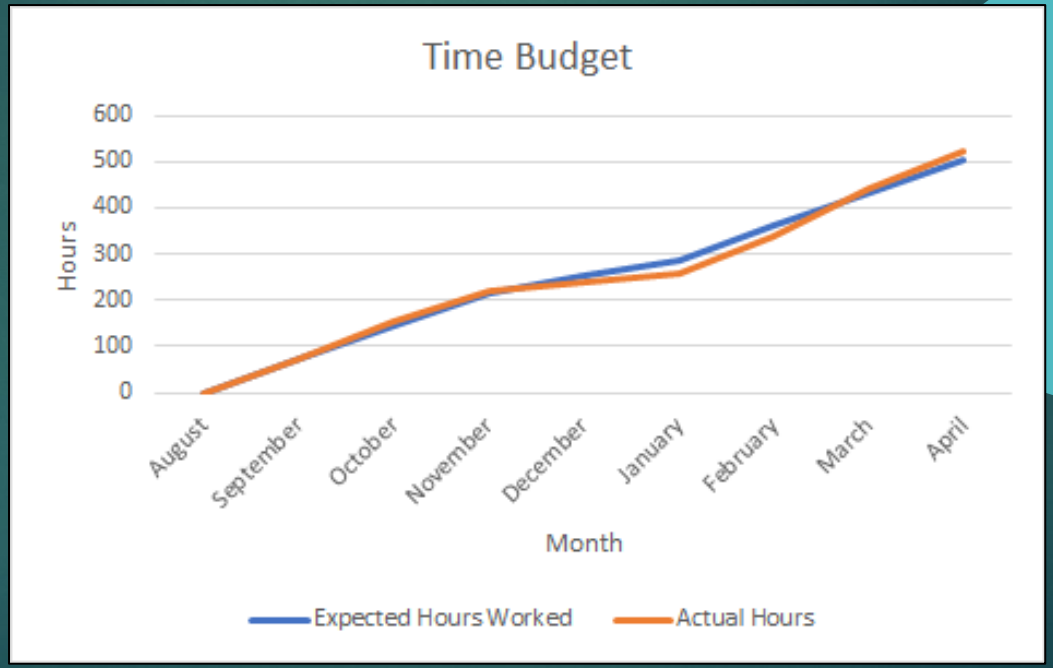
- Creates the sound of a repeating echo, with harmonics ringing at different times
- FFT divides frequencies into smaller bins, which each have a different delay applied



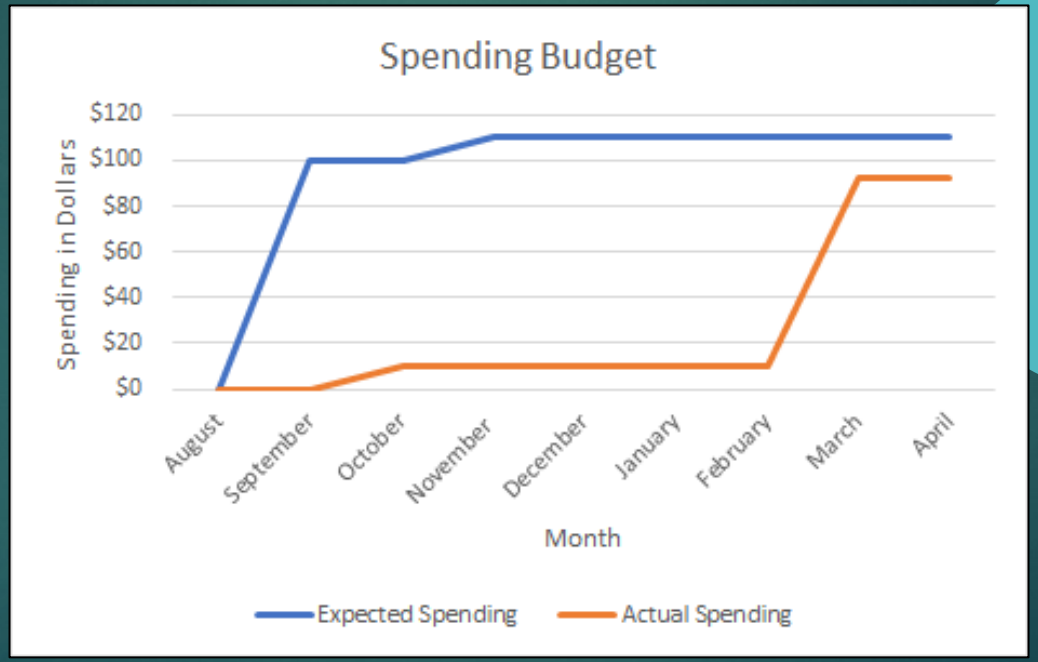
Project Schedule



Time Budget


Time Budget
600
500
400
300
200
100
0
Hours
August
September
October
November
December
January
February
March
April
Month
Expected Hours Worked
Actual Hours

Spending Budget



Summary

- Created an automatic guitar effect trigger system using DTW
  - Capable of triggering any digital effect
- Design criteria met:
  - Trigger latency of <= 1 second (2 ms)
  - Minimum note onset separation of 10 notes per second
    - 43 notes / sec (detect a new note every 23 ms)
  - Concurrent effects triggering

Questions?