

Synths, Samplers, Drum Machines, Effects Processors, and Beyond: Programming in Pd

Faculty of Arts and Sciences, Music Department, NYU
Spring 2014

Instructor: Jaime E. Oliver La Rosa
ja.rosa@nyu.edu
268 Waverly, Office 266F
Mondays / Wednesdays 10:45am – 12:15pm

Description:

Technological devices have shaped the evolution of popular music. Imagine Hendrix playing on an acoustic guitar, Lucy in the Sky with Diamonds played on a piano, electronic dance music played by a rock band, hip hop without turntables and samplers. These technologies and the music made with them are inseparable. In this course, we will learn to program computers to make music using the open source programming environment Pure Data (Pd), by emulating some of the technologies that defined popular music in the second half of the 20th century. As we move through the course content, we will learn concepts in digital audio and acoustics, and explore the work of artists, their music, machines, and the ideas that fostered and resulted from them.

Specific Goals:

The main goal of this course is that you learn how to program sound applications in Pure Data and in doing so, learn basic computer music techniques. To do so, we will emulate some devices that provide us with an interactive experience and that explore some of the most significant techniques or analog and digital audio generation and manipulation.

We will first look at a subtractive synthesis techniques by emulating a *minimoog* as well as polyphonic synthesizers. We'll also take a brief look a FM based synthesizers. We will then turn our attention to sample based devices such as samplers. Finally we will build a sequencer to control either sampled or synthesized sounds.

While you will learn how to program known devices, you are welcome to apply the techniques learnt in them in whatever field of music creation you're most interested in.

Grading

There will be several exercises to learn specific properties of the Pure Data Programming Environment as well as basics of acoustics, digital sound, synthesis, and signal processing. These small assignments will be graded continuously throughout the semester.

Your final project consists in applying some of the concepts developed in class to produce a device of your own. Your project may expand on one or combine some of the examples we will develop as a Group in class, be a substantial modification of those techniques

Final Project	50%
Small Assignments	30%
Attendance & Participation	20%

Possible workshop

LibPd – A Library to program pd applications in mobile devices.

Tentative Outline:

Units:

I. Subtractive Synthesizers

1. Installing, running, and testing Pure Data.
2. Creating, editing and saving patches.
3. Different kinds of boxes in Pure Data. (objects, messages, number, symbol, and comment) {print}
4. Brief introduction to digital audio / Sampling Rate / Defining sine waves or sinusoids. {osc~, print~, tabwrite~, bang, dac~, array}
5. Learning to control the frequency of a sinusoid. Frequency and pitch. {mtof, ftom}
6. Learning to control the amplitude of a sinusoid. Amplitude units. {*~, dbtorms, rmstodb}
7. Ramp Generators: line segments, time varying amplitude, and clicking {line, line~}
8. introduction to sub-patches and abstractions: building an ADSR envelope generator. (Hot and Cold inlets) {trigger, delay, pack/unpack, float}
9. Connecting a MIDI keyboard. {notein, poly~, send/receive}
10. Adding signals and multiplying sinusoids and constants, controlling distortion. {+~, /~}
11. Basic concepts in additive synthesis. Introduction to the relationship between spectra and wave shapes.
12. Different kinds of oscillators: Square, triangle, sawtooth. {phasor~, abs~, clip~}
13. Introduction to Filters {noise~, hip~, lop~, bp~, vcf~}
14. control messages, and pitch-bend and modulation wheels {ctlin, bendin}
15. GUI, messages, and presets.
16. FM synthesis

II. Samplers

17. Arrays & sample playback {array, tabread~, soundfiler}
18. Playback speed and pitch.
19. Interpolation {tabread4~}
20. Control triggers for samples.

III. Drum Machines

21. Building a counter {metro}
22. Building a basic sequencer {sel}
23. Synthesizing noises.
24. Karplus Strong
25. More GUI elements.

IV. Effect

26. Amplitude Modulation
27. Phasing
28. Harmonizers
29. Delays & Reverb
30. Distortion and waveshaping
31. Wahs and Autowahs

Final Projects!