WHO NEEDS MAX ANYWAYS?: LEARNING HOW TO MAKE WIIMOTE-BASED INSTRUMENTS WITH PD-L2ORK

Brandon Hale

Institute for Creativity, Arts, and Technology (ICAT)
Virginia Tech, United States
bthaleproductions@gmail.com

ABSTRACT

Max msp is used in many research environments for doing complex audio and data manipulations, as well as creating electronic instruments. I will be showing the Linux community how one can create electronic instruments using the extended free software pd-l2ork. The basic format of the workshop will consist of first explaining where I work and how pd-l2ork is relevant to me. Then, I will focus on getting the audience up to speed with the basics of using pd-l2ork (and pd, since pd-l2ork is based on pd). After that, I will show the audience how to create some basic unit generators and playback and recording of audio samples. We will go through basic digital signal processing one can build in pd-l2ork, and further combine everything that we have learned to make a synthesizer with a Wiimote controller. The last part of the workshop will consist on free building and sharing of code. I will then go around the room and help individuals to program what they want and answer questions on how to do certain things.

Required software will be pd-l2ork or purr-data. If unsure, purr-data will work fine. They can be installed from here:
https://build.opensuse.org/project/show/home:aggraef.

1. DESCRIPTION

I plan on executing the workshop with these ideas in mind:

1.1. Introduction

I will state a little about where I work (ICAT at Virginia Tech and Linux Laptop Orchestra) and show pictures of the spaces, and how it is relevant to the workshop. Then, I will give links out to the workshop (if needed) to show where to install both pd-l2ork 1.0 (for Linux systems only) and Purr-data (for all systems, including Linux). I am hoping that most machines will be running some form of Linux, or BSD. I will introduce the basics of Pure Data: showing how to place objects, messages, numbers (there are only floats), comments, GUI sliders, toggles, bangs, show pd-l2ork's help patches, mention the differences of pd-l2ork and pd, and look for help in the help browser of pd-l2ork; all with keyboard shortcuts to help get people moving along with pd-l2ork quickly.

1.2. Basic Programming

The first programming I intend to show is the "Hello World" of computer music: making a basic sine wave. From here, I will show how the [dac~] object works, and the difference between black and blue cables and objects with and without "~" (the difference being data and audio rate data). Then, I will show how to change the frequency of the sine wave using a basic counter (built without the [counter] object) using [metro] and [toggle]. Then, I will show how to scale audio properly in pd, without zippering of the audio, and build a simple mixer using the [line~] object. This should give the most fundamental audio processing needed to start experimenting more.

1.3. More Unit Generators

This part of the workshop will introduce how to make some basic unit generators, (using [osc~], [phasor~], and [clip~] objects for sine, sawtooth, and square waves respectively). Then, I will introduce the concept of an array, how it contains numbers in a sequence, show how to record audio to it with [adc~] and [tabwrite~], and then playback the array as a sample with [tabread4~] and [phasor~]. [readsf~] will also be mentioned to playback audio from disc. Lastly, we will then use [random] and [drunk] objects to make randomly occurring synths, and use [legacy_mousemotion] to make mouse-driven synthesizers.

1.4. DSP Effects

Using everything we have learned so far, I will show off some audio processing effects that one can use. We will go through [freeverb~], [distortion~], and [vd~] with [delwrite~] for delay. We will then build tremolo/ring modulation with [osc~] to [*~]. I will show how to make [pd] subpatches and abstractions to containerize these dsp units. Lastly, we will combine these individually to create the preferred timbre and color of each person’s instrument.

1.5. Combining Everything with Wiimote

I will demonstrate my workflow with [disis_wiimote] and how to start mapping our synth (that should all be different at this point) to Wiimote controls and motion. We will take the [disis_wiimote] help patch and use it as the basis for our Wiimote control unit.

1.6. **Free Building**

At this point, depending on how much time the group has left, I will answer questions around the room, see how the audience is doing, and if everyone does not have questions, experiment, showing some other processing pd-l2ork can do. The goal is to encourage code sharing and collaborate.

2. CONCLUSIONS

I would like to keep the time I spend on all of these points flexible, depending on how the audience is doing. If the audience is fluent already in pd, I will try to spend time showing some of the interesting objects and code pd-l2ork has to offer, and answer questions from the audience. I hope to help the audience develop unique and interesting instruments that could be used in a future piece. The community's participation will determine what everyone focuses on. Also, the workshop would ideally be community-based, if people find interesting sounds, I want to encourage that and ask the audience to share with the broader community. I will make sure to bring a flash drive to send pd patches around, and am willing to post my patch, and maybe others' patches on my website, a code repository, or a LAC2020 website for everyone to have and experiment with. The goal is to stimulate creative programming and get people excited to make with pd-l2ork, an open-sourced tool that everyone has access to. Pd-l2ork is a very versatile audio programming language for creating electronic instruments and other audio processing and is available for everyone to use and experiment with.

3. ACKNOWLEDGMENTS

I would like to give thanks to Ico Bukvic for creating pd-l2ork and the Linux Laptop Orchestra and for helping me learn the program. Also, I thank Eric Lyon and Charles Nichols for introducing me to Pure Data and encouraging me to pursue knowledge. Lastly, I give thanks to my current colleague Tanner Upthegrove for helping me with life and career help.