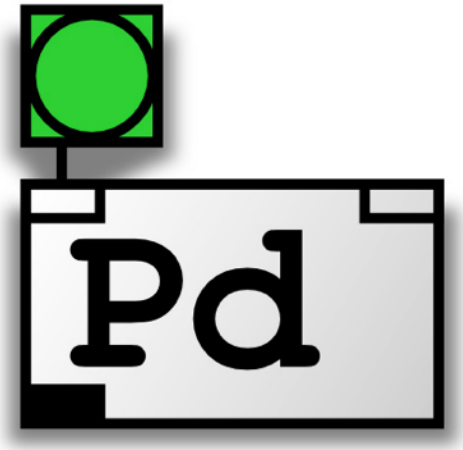
DT2300
Introduction to Pure Data

Emma Frid

KTH CSC

Overview



- Background
- Installing Pd & basic config
- Pd Basics : Getting started
- Pd for Audio

Background

Pure Data (or Pd) is a real-time graphical programming environment for audio, video, and graphical processing.

Written by Miller S. Puckette (previous co-developed the well known and similarly structured software Max/Msp).

Open Source! As opposed to Max/Msp.

Visual metaphor from analogue synthesizer patches.

Installing Pure Data

Download **Pd Extended** from:
https://puredata.info/downloads

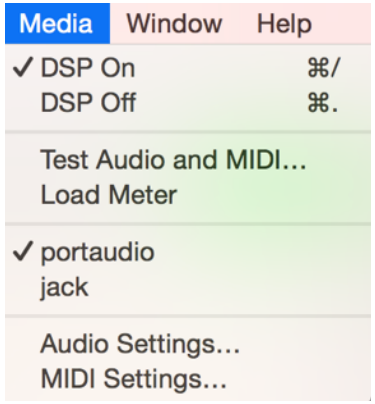
Miller Puckette's version of Pure Data is called **Pd-Vanilla**. It has just the basic minimum set of functionality.

Detailed instructions:
http://en.flossmanuals.net/pure-data/installing/windows/
http://en.flossmanuals.net/pure-data/installing/osx/

Basic Configuration

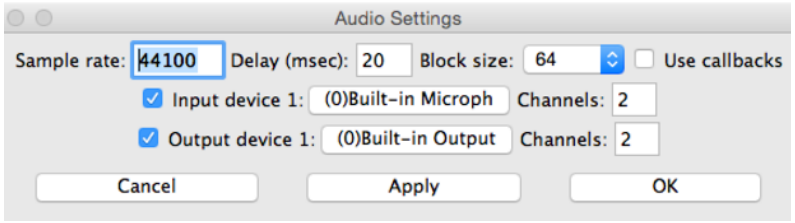
1. Audio Drivers

OSX : Media - portaudio/jack
Windows : Media - ASIO (via portaudio)
Linux : Media - OSS/ALSA/jack

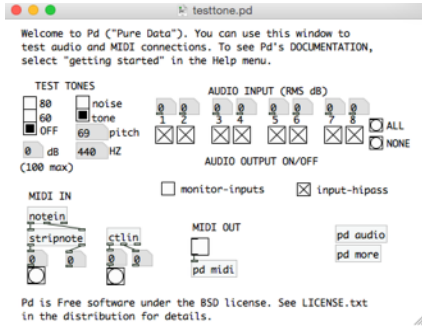


2. Audio Settings

OSX : Pd-extended - Preferences - Audio Settings
Linux & Windows : Media - Audio Settings



3. Test Audio and MIDI…



DO NOT FORGET TO TURN DSP ON!

Introducing Pure Data

Pure Data files are called “patches”

Programming with Pure Data - interaction that is much closer to the experience of manipulating things in the physical world

The most basic unit of functionality is a box, and the program is formed by connecting these boxes together into diagrams

Diagrams represent the flow of data but is also performing the operations mapped out in the diagram

Introducing Pure Data

Encapsulation : sub patches

```
[pd subpatcherName]
```

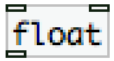
sub patches can have inlets and outlets!

```
[inlet] [outlet]
[inlet~] [outlet~]
```

Abstractions : save a patch with a name such as "abstraction1.pd" and then invoke it as "abstraction1" in an object box:

```
[abstraction1]
```

Basic Elements

Objects 

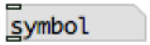
rectangular - object name and default value

Messages 

indentation on the right side
passes data which is stored inside of them when clicked

Numbers 

atoms

Symbols 

atoms

Comments comment

Basics Functionality

Play or Edit Mode: Patcher/canvas locked/unlocked
CMD E (Mac), CTRL E (Windows)

Output can be printed in the Terminal

Pd blocks have inlets on the topside and outlets on the bottom

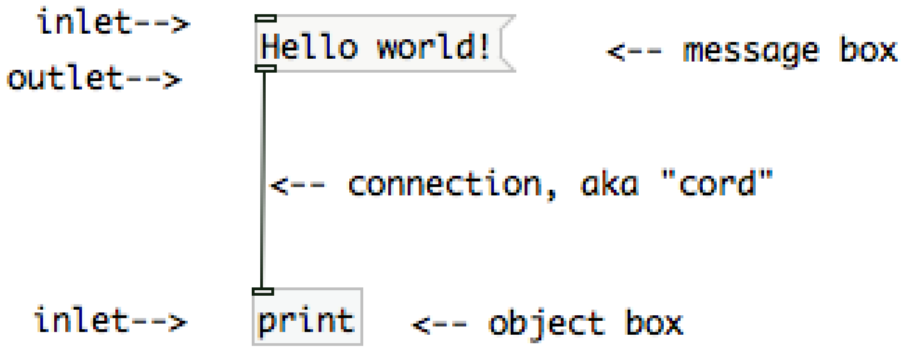
To create a connection between two blocks, drag a line from the outlet of one block to the inlet of another block

Use shortcuts! (different for Windows and Mac)

Hello World

In Pd, programming is done with boxes which are connected together. The boxes have "inlets" and "outlets", where they are connected.

Click on the box with "Hello world!" in it:



Getting Started

Examples and descriptions can be found in the Pd Help Browser

Help/Pd Help Browser

You can get help with anything by right-clicking a box and select “help”

Common Objects

`[bang]` does stuff!

`[tgl]` toggle 1 or 0

`[trigger]` sequence messages in right-to-left order

`[metro]` set bangs periodically

`[select]` bangs when received specific number

`[pack]` packs lists

`[send][receive]`

`[maxlib/scale]` scaling input/output ranges

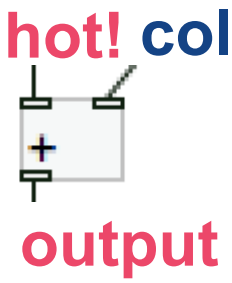
Pd Essentials: Hot and Cold Inlets (1)

The right inlet will typically set an argument to a function. Only input to the active inlet will create an output from the outlet(s).

the left-most inlet is “hot” : it will output something whenever it receives data

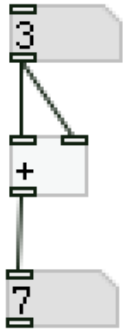
all other inlets are generally “cold”: they just store data

when the object receives input on the “hot” inlet, the object will read the stored data from all inlets and do stuff

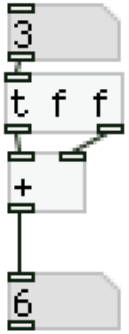


Pd Essentials: Hot and Cold Inlets (2)

Problems can arise when a single outlet is connected (either directly or through arbitrarily long chains of message passing) to different inlets of a single object:



?!

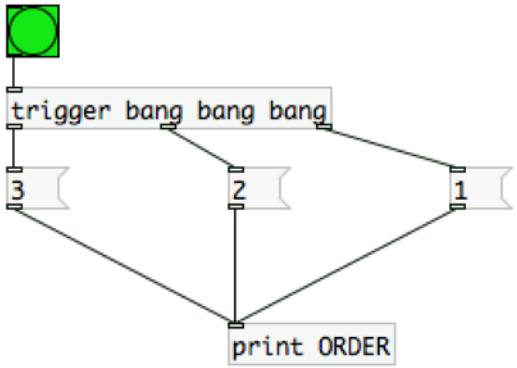


:)

Pd Essentials: Right to Left Order

Patches are read from right to left, top to bottom.
Objects output from right to left.

forcing a specific execution order can be done with the “trigger” object



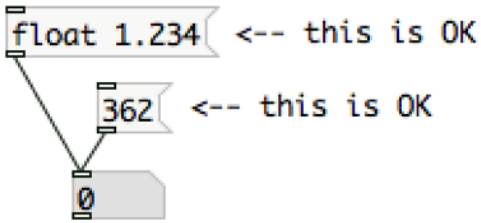
this patch will output :

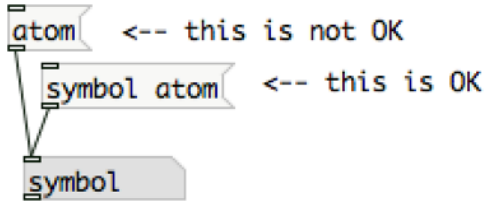
```
ORDER 1
ORDER 2
ORDER 3
```

Pd Essentials: Atomic messages

The “symbol” message is crucial when you are sending symbol

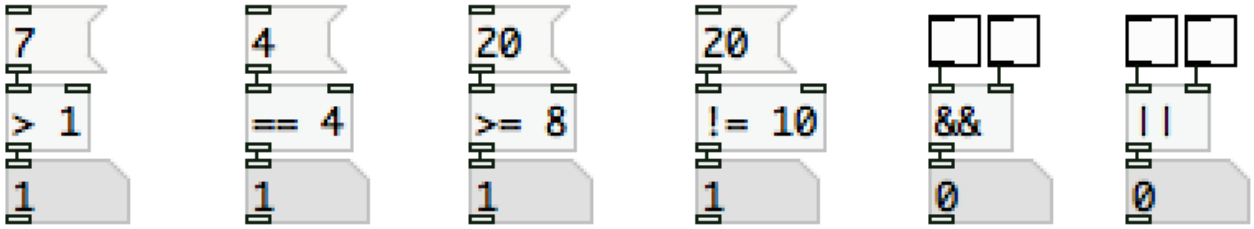
the “float” message is not necessary when you are sending floats





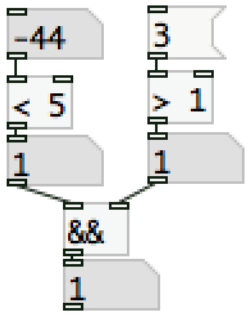
Comparing Numbers (1)

Logic operators



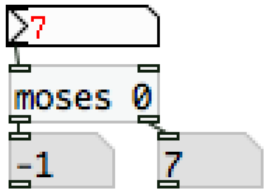
Comparing incoming data flow with static numbers : do something when true (i.e. 1)

Multiple Comparisons



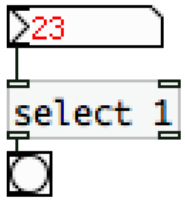
Comparing Numbers (2)

`[moses]` - splits a range of numbers



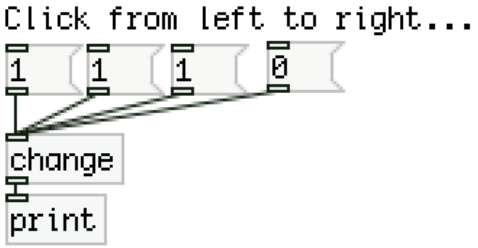
values below 0 will be outputted to the left
values equal to or larger than 0 will be outputted to the right

`[select]` - bangs when number is received

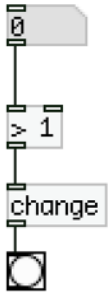


Comparing Numbers (3)

`[change]` - eliminates redundancy in a number stream



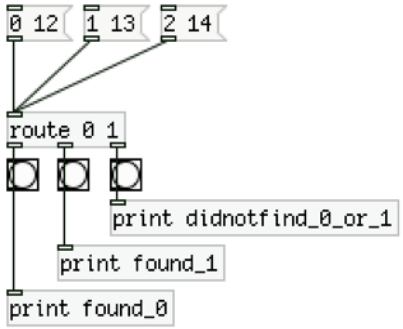
output is only allowed when value is changed



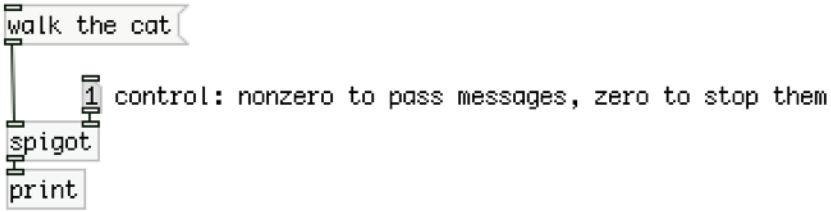
Example: bang once when value exceeds 1

Comparing Numbers (4)

`[route]` - route messages according to a selector (or first element)



`[spigot]` - passes messages from its left inlet to its outlet, as long as a nonzero number is sent to its right inlet



Arithmetic Objects

```
[+ ]
[- ]
[* ]
[/ ]
[pow]
[max]
[min]
```

`[expr]` allows you to write mathematical formulas

Working with sounds

Audio computations in Pd are carried out by "tilde objects" such as `[osc~]`

`[dac~ 1 2 3]` real-time audio output to channel 1,2,3

Make sure DSP is on! 

Common objects

[dac~]

[adc~]

[osc~]

[snapshot~]

[vline~]

[sig~]

[clip~]

[+~] [-~] [*~] [/~]

… see PureData Audio Examples in the help Browser

additive synth : D07.additive.pd

classic synth : J08.classicsynth.pd

Pd Essentials: Digital Audio

Good resource to learn the basics:

http://en.flossmanuals.net/pure-data/introduction/what-is-digital-audio/

Volume control, Mixing, Clipping

Sampling

DC Offset

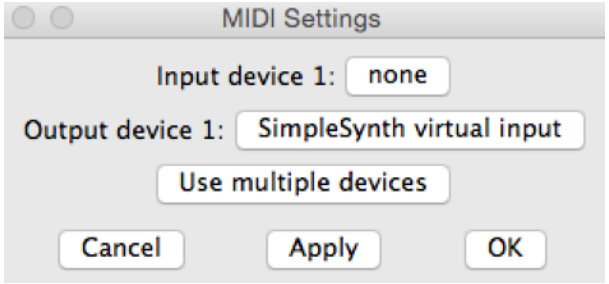
….

MIDI

For Mac:

1. Download **SimpleSynth** from
http://notahat.com/simplesynth/

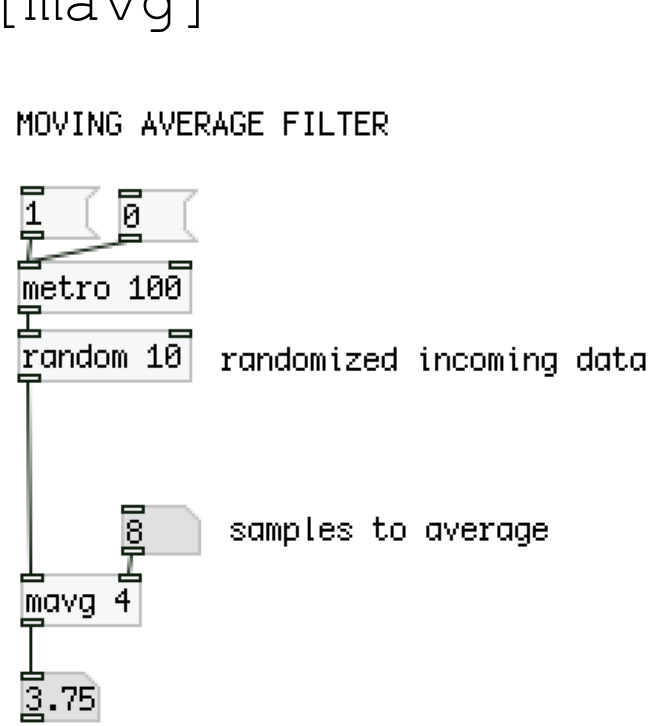
2. Set output to your synth device



3. Test MIDI!

Filtering sensor data

[mavg]



Recources online

Pure Data Portal : http://puredata.info/

Download Pure Data: http://puredata.info/downloads

Programming electronic music in Pd : http://www.pd-tutorial.com/

Good Manuals:

http://en.flossmanuals.net/PureData/

Video Tutorials with Dr. Rafael Hernandez:

https://www.youtube.com/channel/UC-RatzHn1ukuuINLqnbBYeg

Also try Max/Msp! https://cycling74.com/products/max/