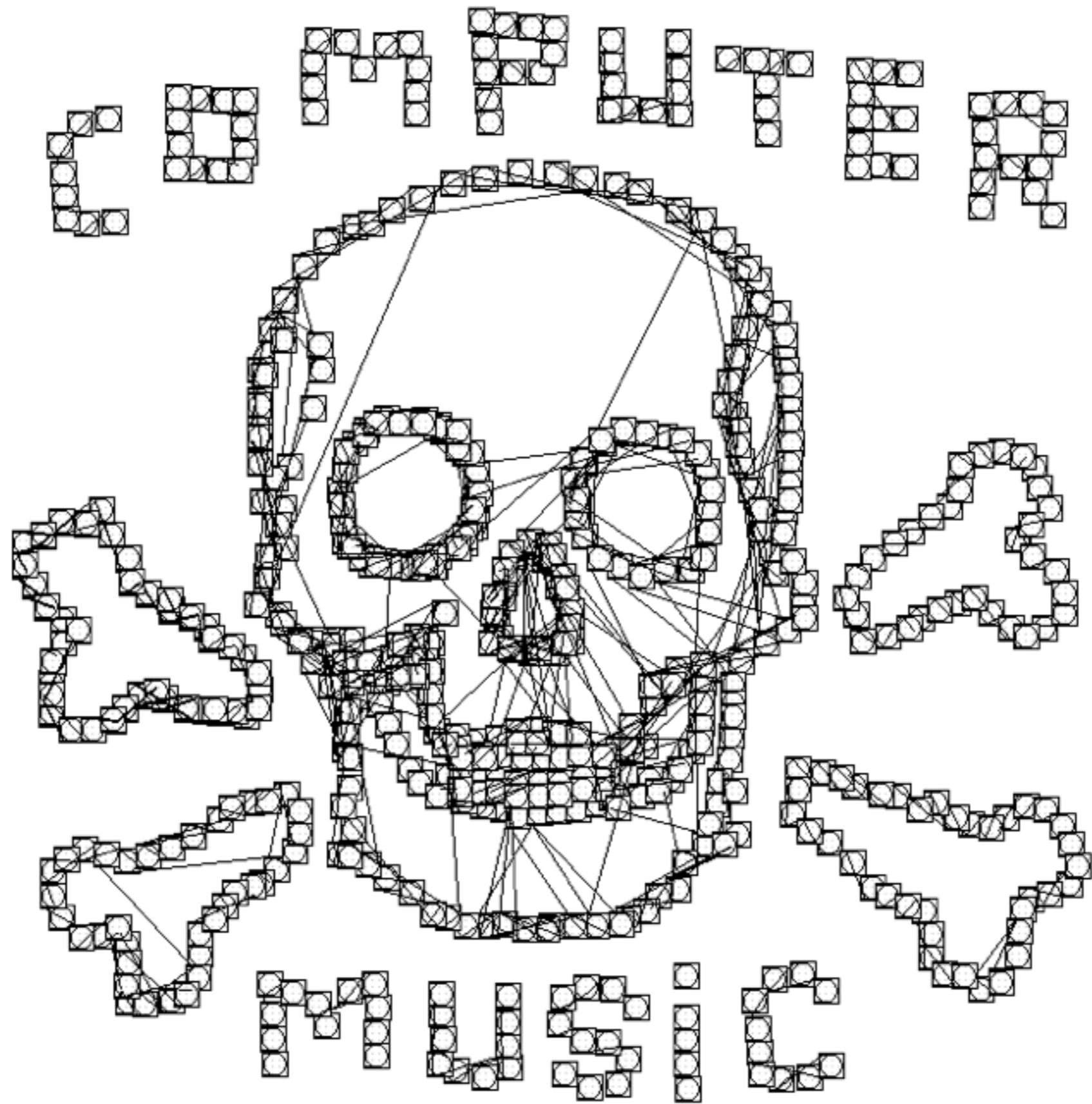


# PdParty

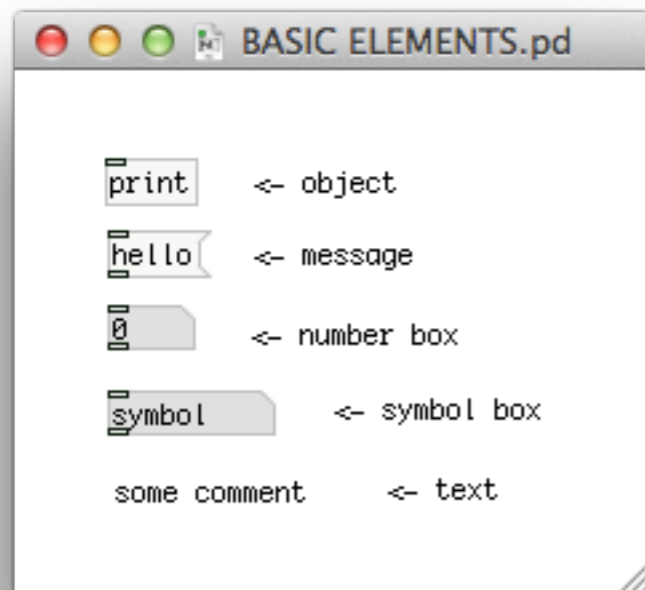
a mobile platform for computer music on iOS using Pure Data

# What is Pure Data?

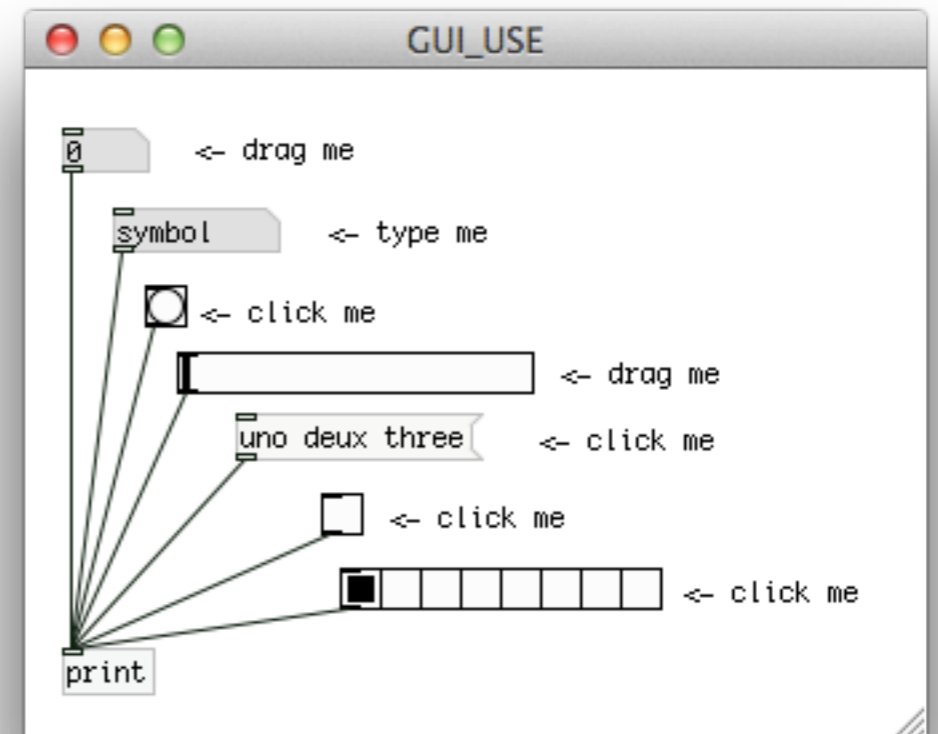




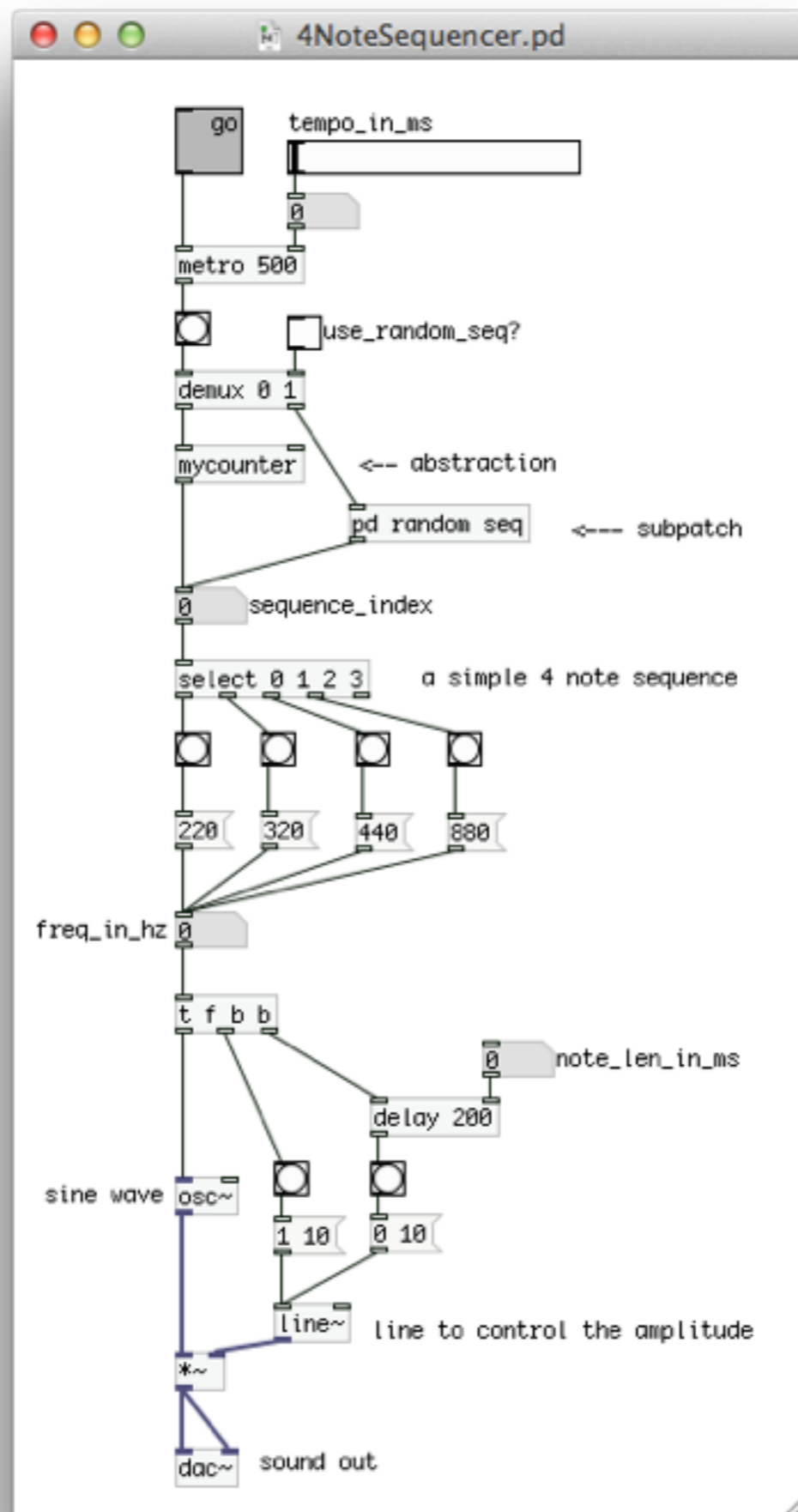
# Graphical



Object Oriented



Data Flow



**MAX/MSP? TOO MAINSTREAM...**



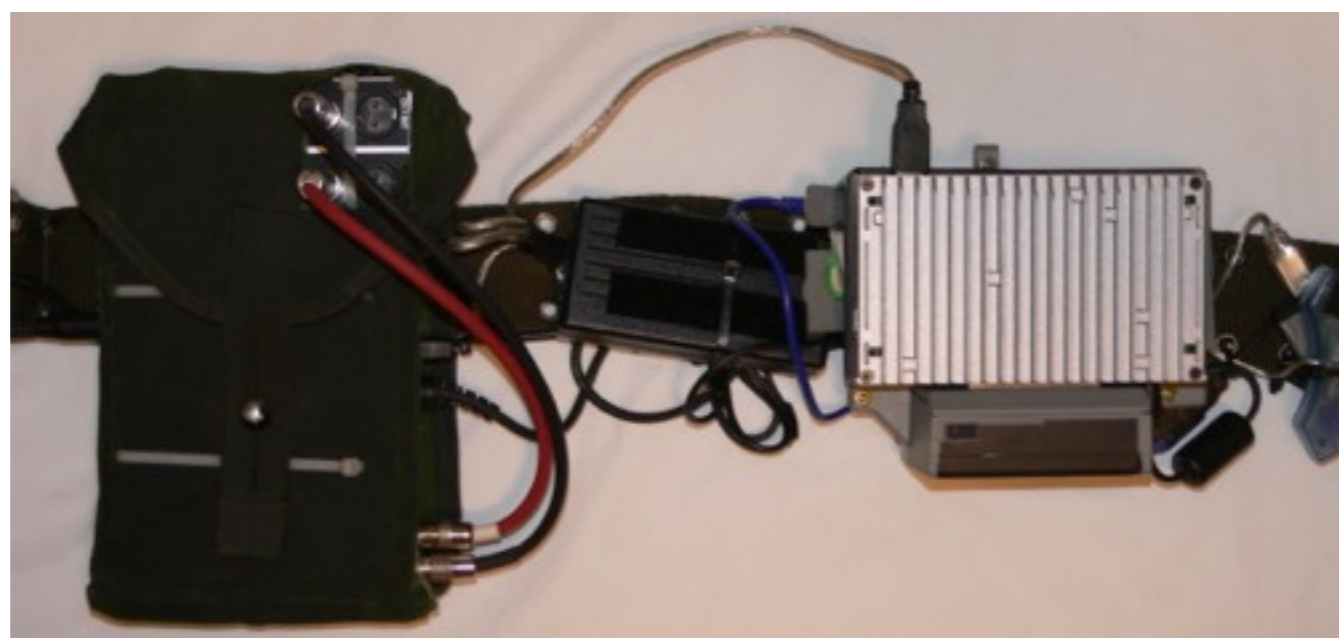
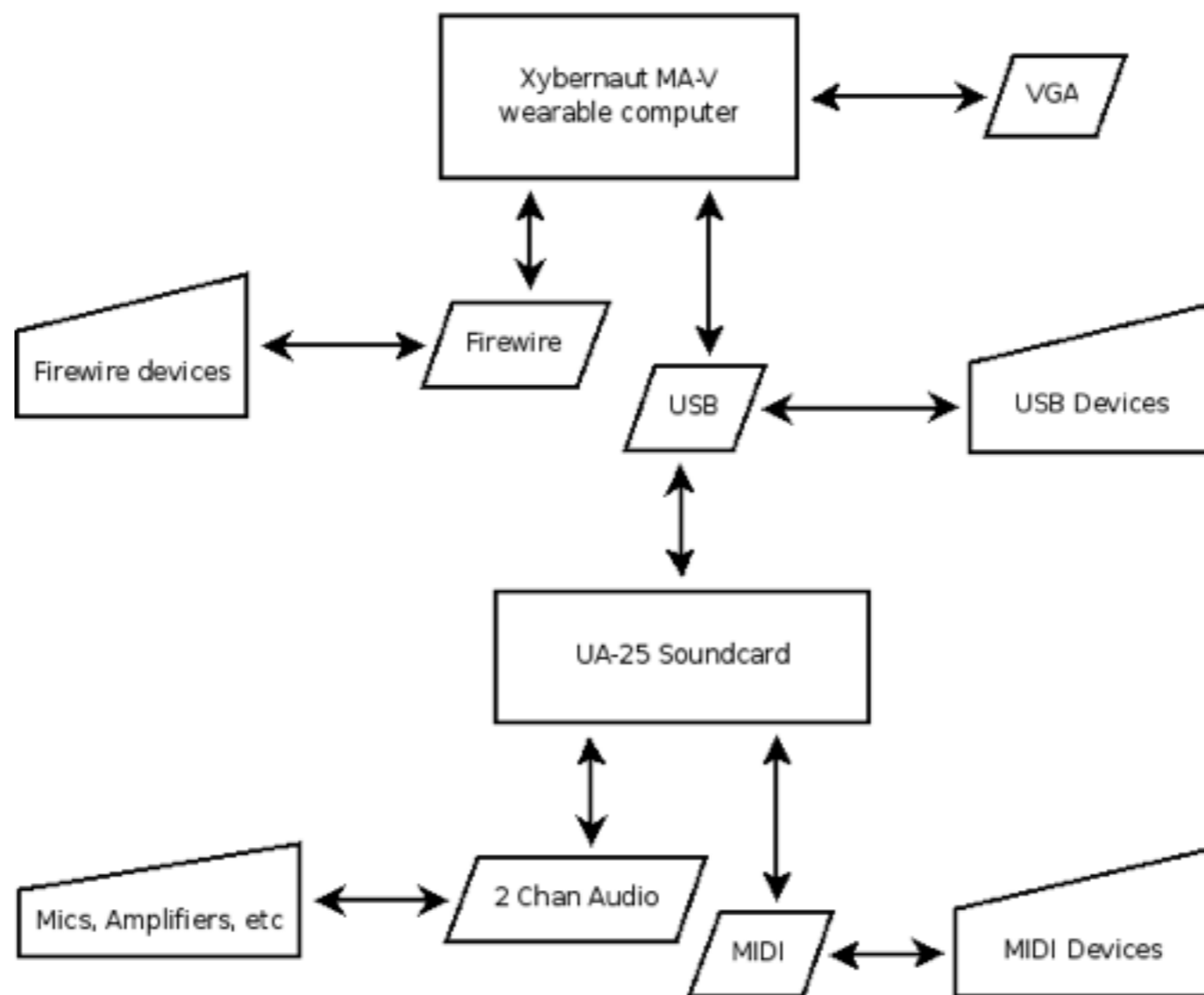
**USE PUREDATA INSTEAD.**

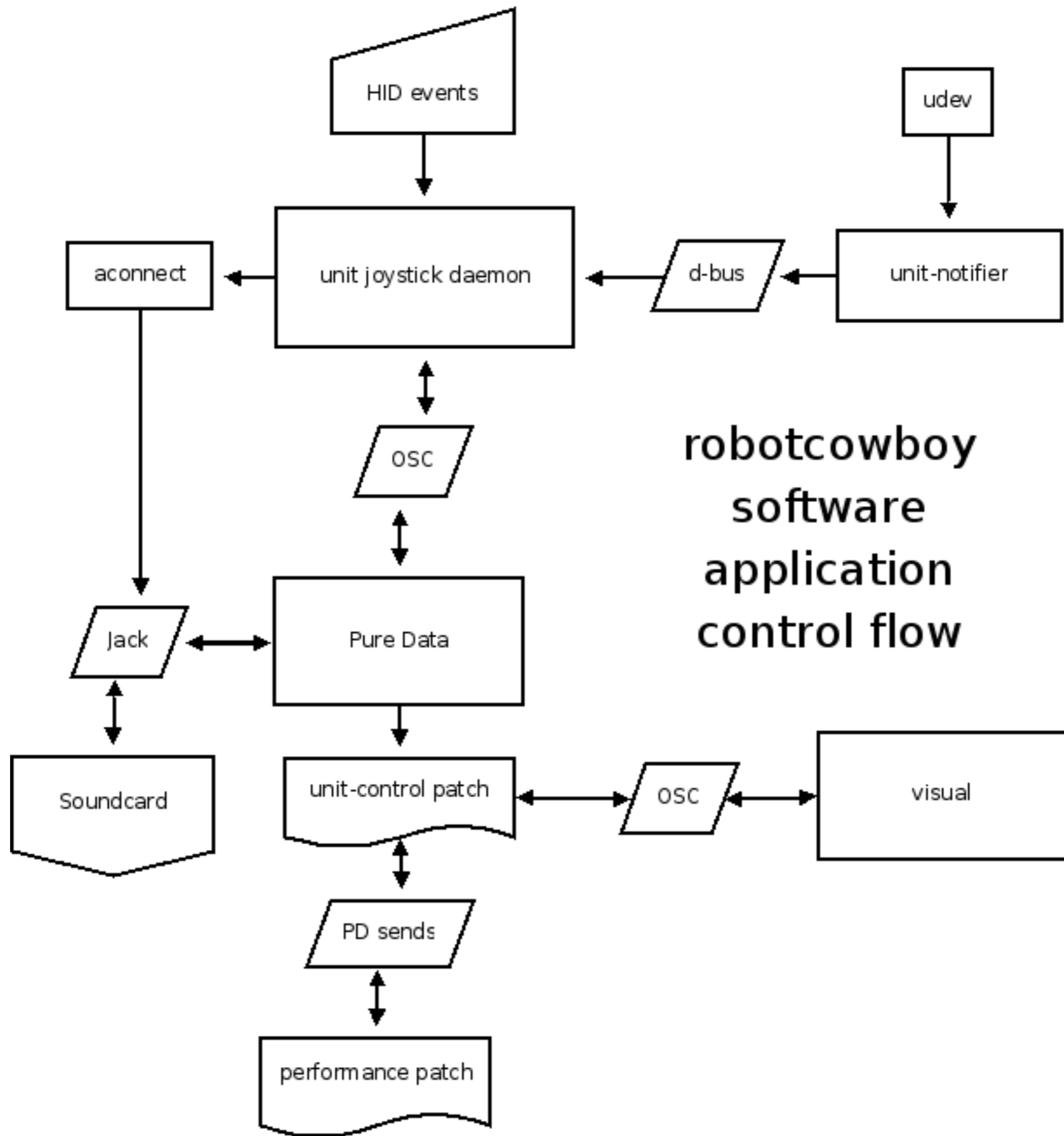
Background





# robotcowboy hardware signal flow



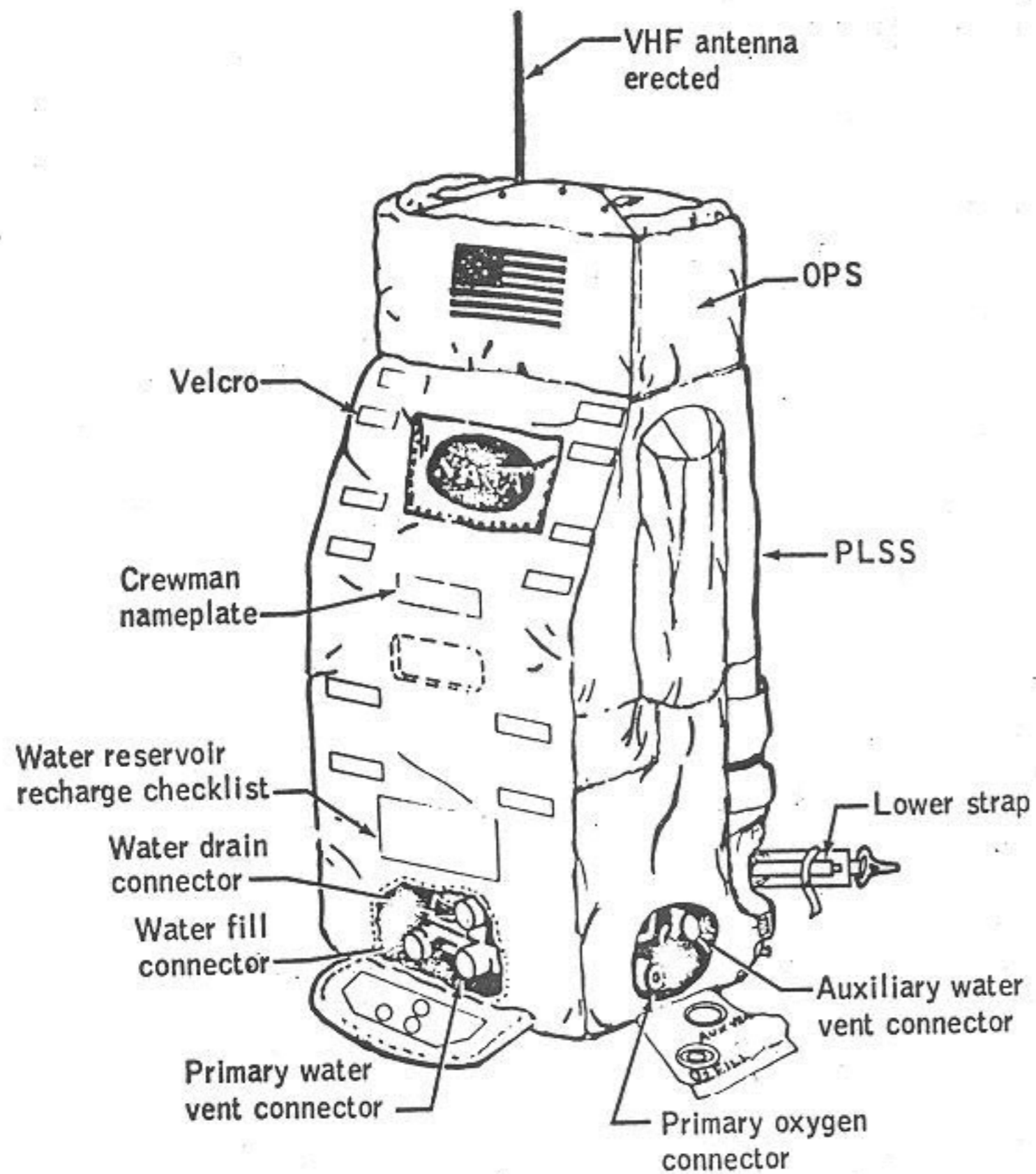




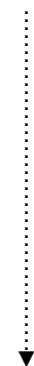
robotcowboy

# Onward to Mars





Portable *Life*  
Support System



Portable *Audio*  
Support System

What's PdParty?

## **Native iOS App with:**

Pure Data interpreter (libpd)

Open Sound Control server (UDP)

CoreMIDI (usb/wifi)

WebDAV server for managing patches

support for RjDj & DroidParty scenes

multitouch, accelerometer, compass, & heading events

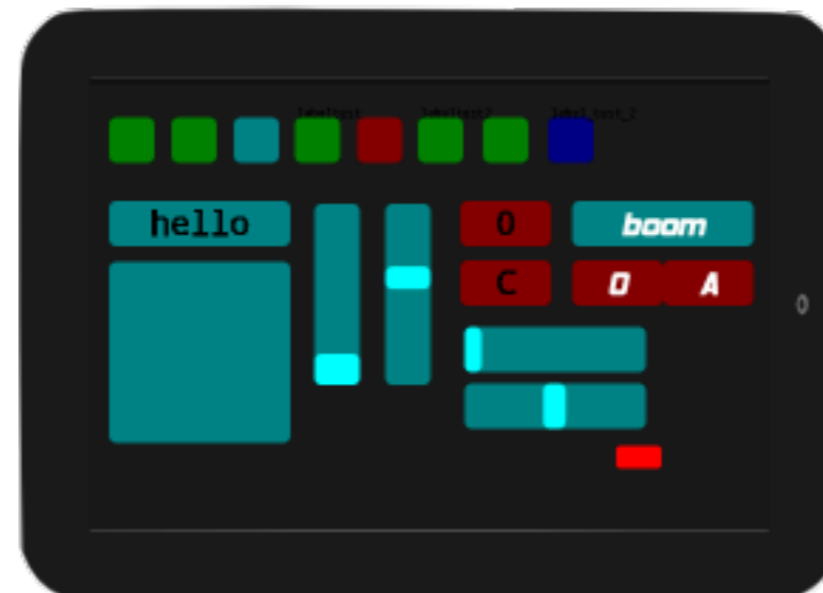
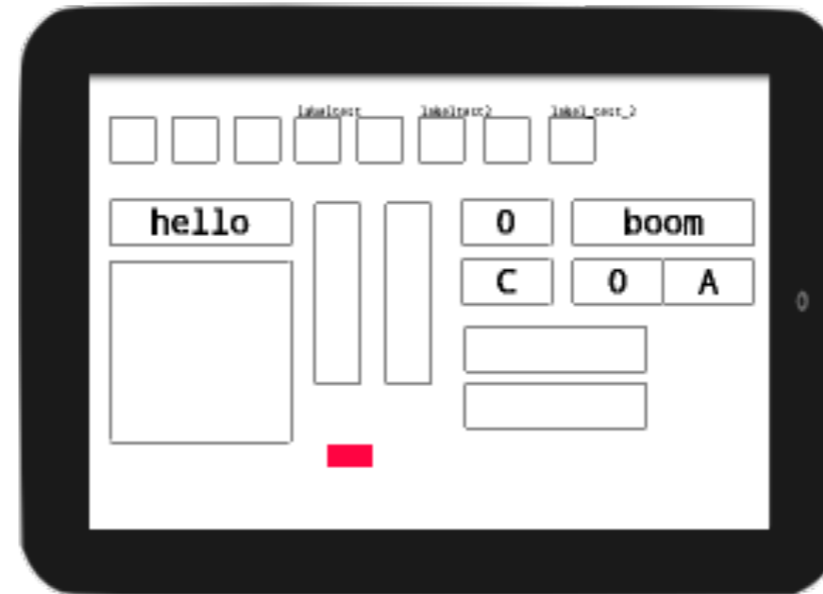
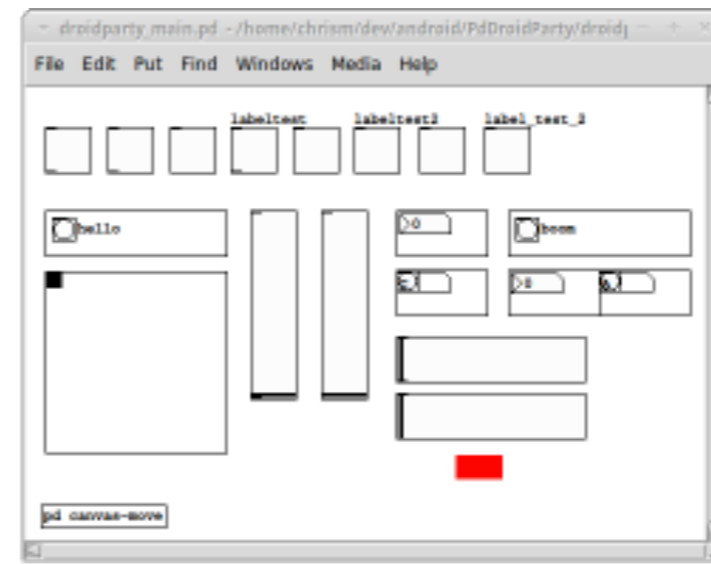


original RjDj app



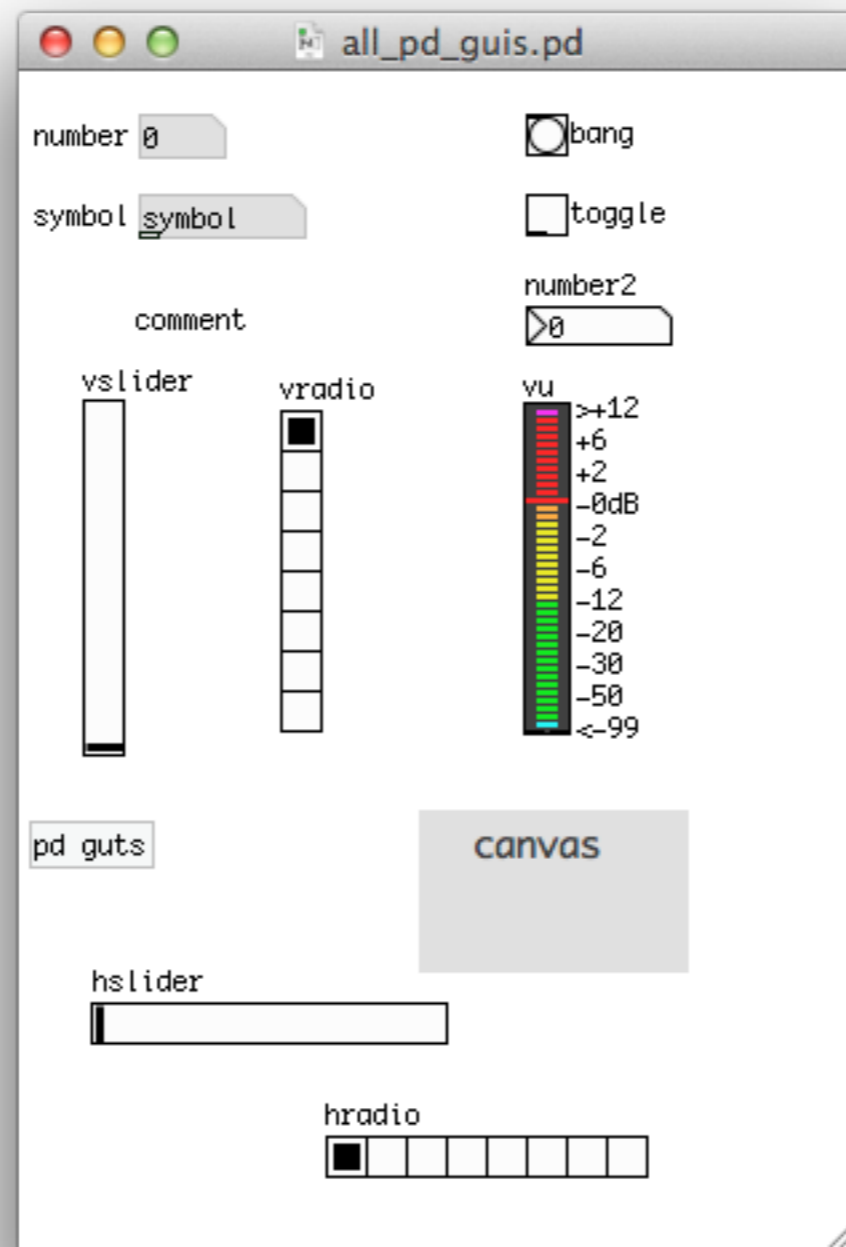
Android RjDj scene player



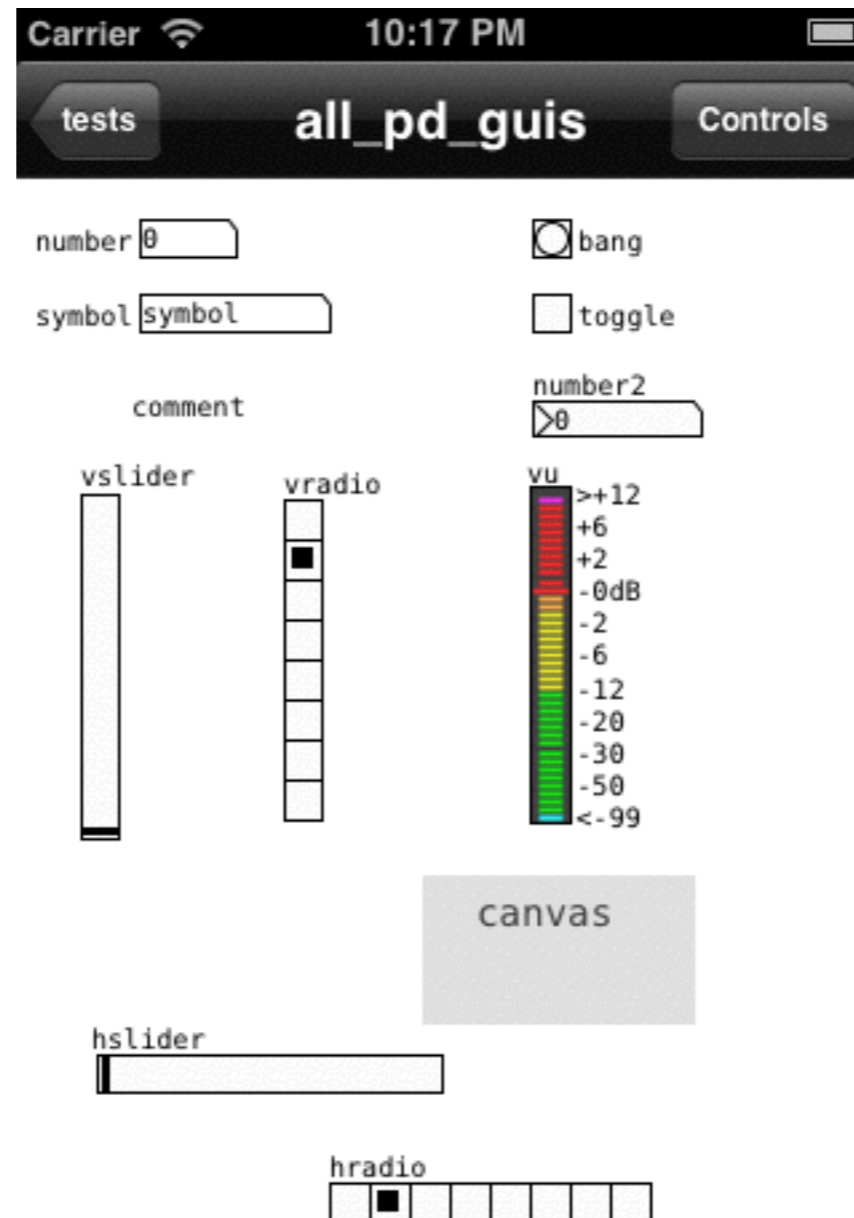


DroidParty for Android

This patch in Pure Data ...



becomes this on iPhone ...



number 0

symbol symbol

comment

vslider



vradio



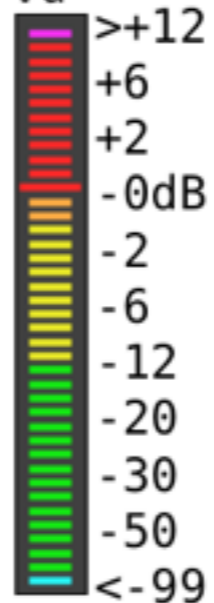
bang

toggle

number2

0

vuv



canvas

hslider



hradio



... and this

on iPad

How does it work?

# Patch like normal, but ...

The screenshot shows a Pure Data patch window titled "SENDS\_AND\_RECEIVES". It contains two examples of message routing. The first example shows two message boxes containing "0" connected by a vertical line. A double-headed arrow labeled "change" points to a second configuration where the top "0" box is connected to a "send myNumber" object, which is connected to a "receive myNumber" object, which is then connected to a bottom "0" box. The second example shows an audio signal path starting with an "osc~ 0.025" object, branching to a "send~ mySig" object, which connects to a "receive~ mySig" object, which then connects to an "env~ 2048" object, and finally to a "94.32" object. A similar path exists on the left side of the patch.

any GUI connected using send/receives (virtual patch cords) are emulated on the device

# Receive device event from special receivers

```
PdParty events.pd  
  
r #touch  
|  
unpack s f f f eventType id x y  
  
r #accelerate  
|  
unpack f f f x y z  
  
r #locate  
|  
unpack f f f f f f f s lat lon alt speed course hacc vacc timestamp  
  
r #heading  
|  
unpack f f s degrees accur timestamp  
  
r #osc-in receive data over OSC  
  
s #osc-out send data over OSC  
  
s #pdparty internal messages to set locate/heading accur, etc
```

Send/receive OSC data