Software for vector synthesis and performance

Douglas Nunn
Cambridge School of Creative Industries
Anglia Ruskin University
douglas.nunn@anglia.ac.uk

~~Sound and Audio Engineering Research Group~~
~~Sound Engineering Research Group~~
~~Sound And Game Engineering Research Group~~
~~Intelligent Systems, Software Engineering and Gaming Research Group~~
StoryLab Research Institute

~~Department of Maths, Physics and Electronics~~
~~Department of Maths and Technology~~
~~Department of Digital Sciences and Technology~~
~~Department of Design and Technology~~
~~Department of Computing and Technology~~
~~School of Computing and Information Science~~
Cambridge School of Creative Industries

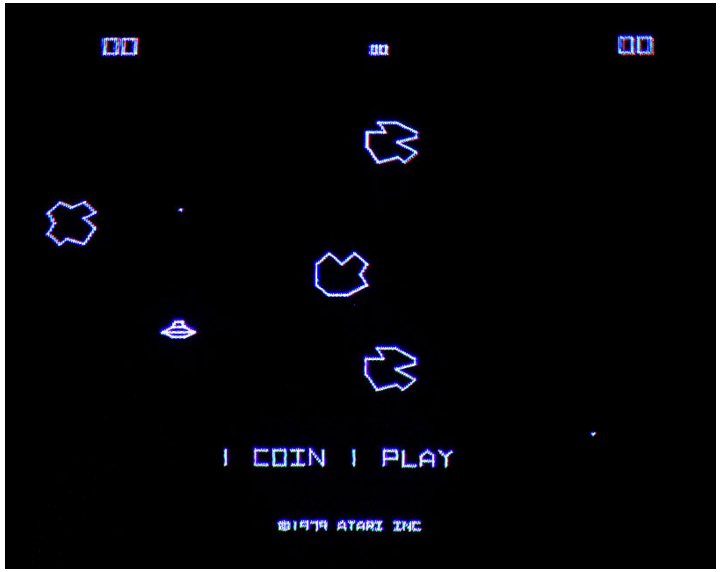
~~School of Applied Sciences~~
~~Faculty of Science and Technology~~
~~Faculty of Science and Engineering~~
Faculty of Arts, Humanities and Social Sciences

~~Anglia Polytechnic University~~
Anglia Ruskin University

~~Bryant 125~~
~~Mellish Clark 122~~
Compass House 307

~~d.j.e.nunn@anglia.ac.uk~~
douglas.nunn@anglia.ac.uk

Why vector graphics?



- Raster graphics dominate computer graphics, including games/video/animation/art
- But vector graphics offers...
  - 'infinite' resolution (no pixellation)
  - line-based aesthetic (like pen and ink)
  - impermanent display (disappearing ink? CRT phosphor persistence)

永

- Other factors
  - reusing/repurposing obsolescent hardware
  - reimplementing historic devices (Rutt-Etra, Scanimate)
  - not using the mainstream approach (e.g. Adobe CS, Resolume)
- Hypothesis – vector displays became obsolescent when computers could not fully exploit them. Now they can!

/* here the font abruptly switches from an emulation of the original 1967 Hershey vector font (still used by plotters and engravers) */

&@~%

/* to Calibri Light – TrueType fonts use Bézier curves as well as lines */

&@~%

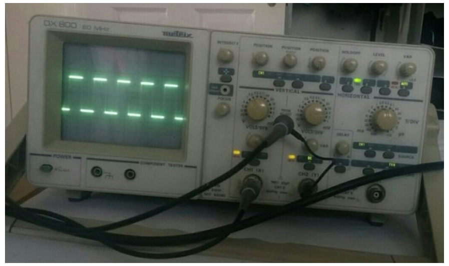
Why use audio tools for vector synthesis?

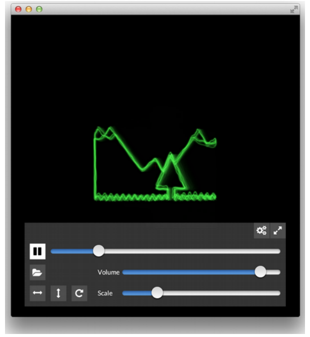
- Can use the same tools to generate audio and video
- Same demand for complex, timed control of multiple parameters
- Many vector synthesis algorithms/effects correspond directly to common audio processes

Vectors	Audio
Draw a circle	Generate cosine and sine waves
Turn to ellipse	Change gain
Randomise position	Add random DC offsets
Repeated linear growth in size	Apply upward sawtooth LFO to gain
Repeated linear brightness fade	Apply downward sawtooth LFO to third channel
Show several growing fading ellipses	Multiplex between several such sources

RaindropsKeepFallingOnMyHamegHM204

Display devices

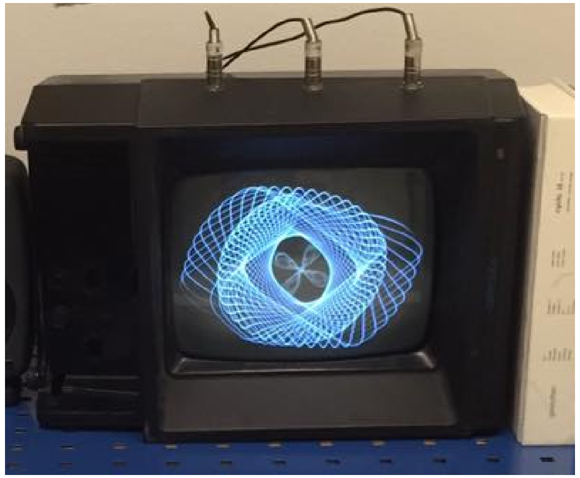




	analogue o’scope	arcade monitor	Vectrex console	CRT TV	o’scope emulator	stroke-to-raster	laser projector
cost	$	$$$	$$	$	free	$$$	$$$$
availability	very common	rare	occasional	very common	free	very rare	easy
modding	-	possible	easy	hard, more $	-	-	-
display size	5” graticule	20”	9”	varies	-	-	huge, coloured
portability	✓	×××	✓	×	✓✓✓	✓	✓

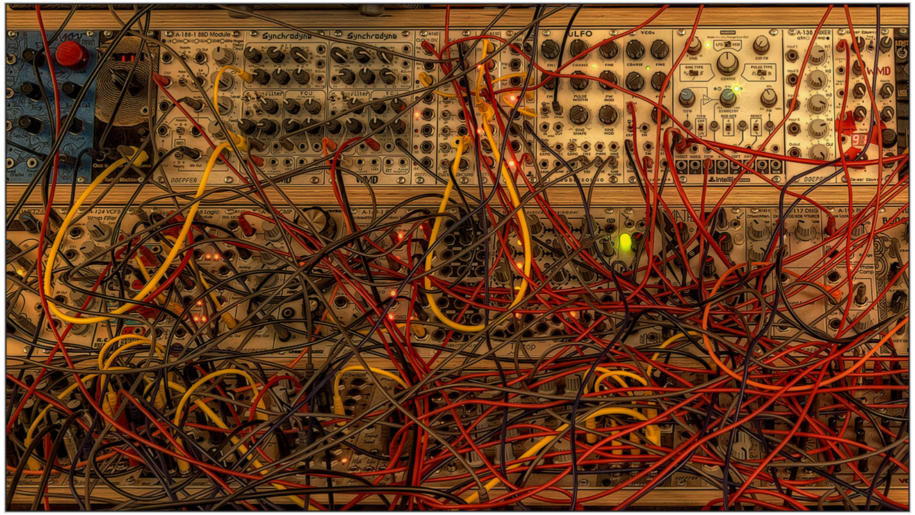
Vectrex games console

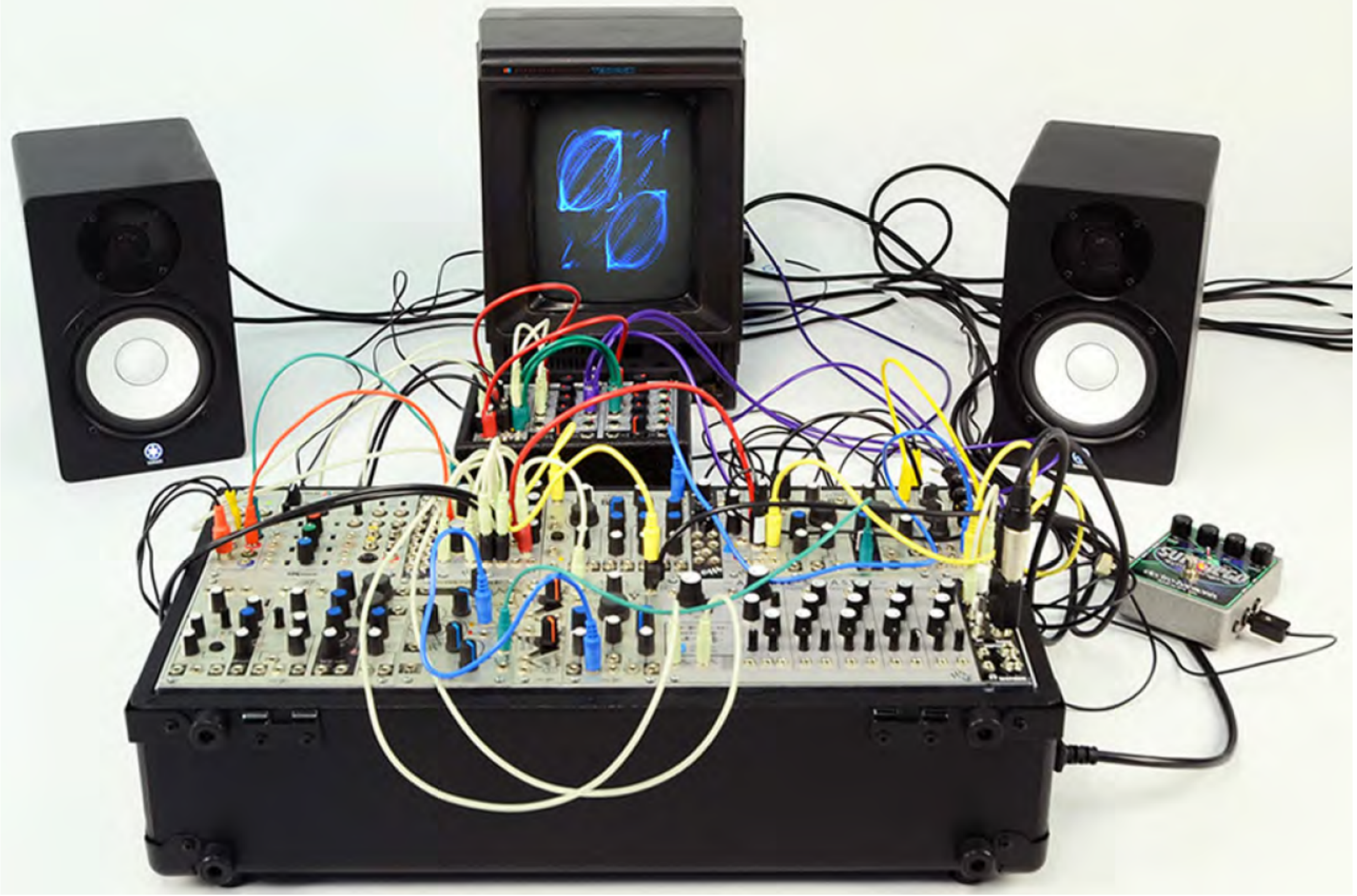
- Monochrome CRT made by Milton Bradley 1982-1984
- Modification (Duff) adds external inputs
- Output captured with HD camera
- “Spot-killer” circuit cuts the beam when the vertical signal has low amplitude or low speed
- This can be defeated by multiplexing an invisible high-frequency signal or the “Holzer-Konopaska” mod



Vector synthesis in hardware

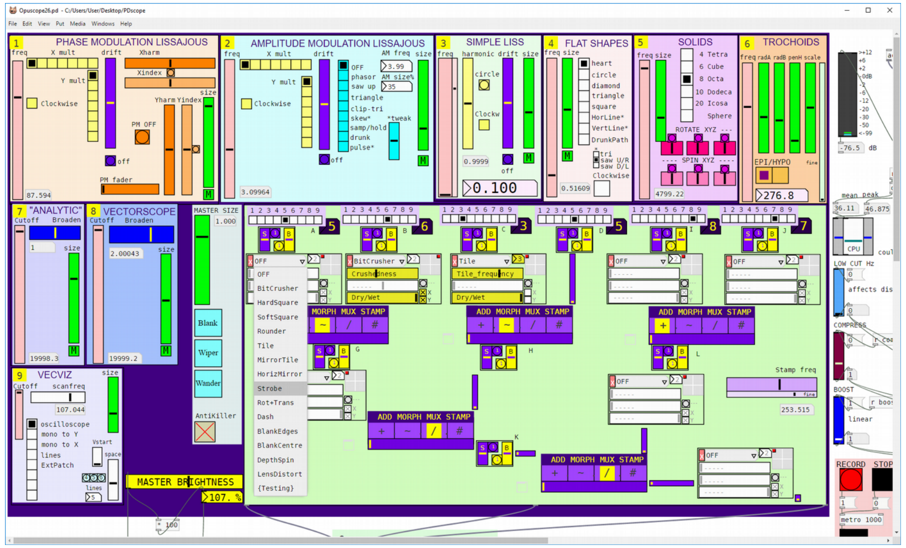
- Often modular analogue synths are used
- Pros
  - Flexible/reconfigurable signal path
  - Lots of physical controls
- Cons
  - Can’t save patches
  - Controls hidden by ‘Kabelsalat’



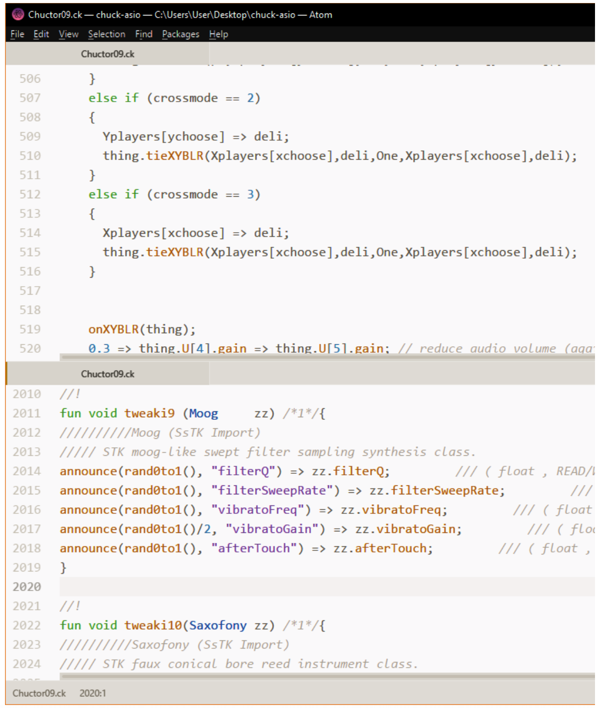


Software

- Pros
  - Effects not possible in hardware
  - Ability to save setup
  - Allows non-real-time rendering
- Cons
  - Fewer physical controls
  - More crashes



Purr Data

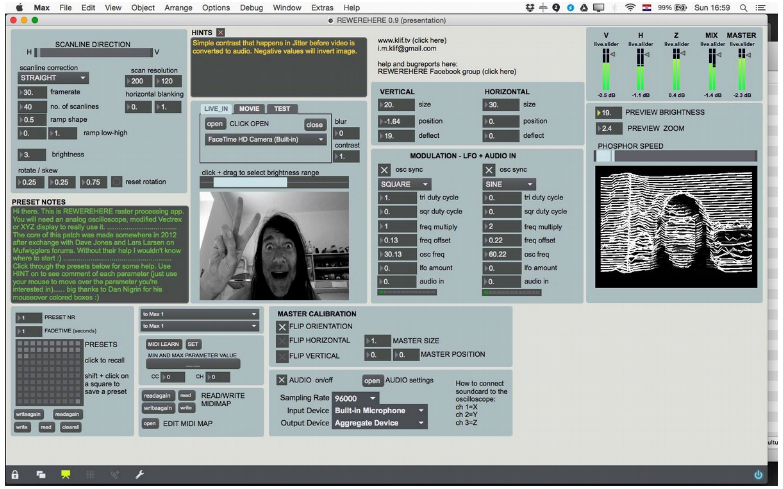

Chuctor09.ck — chuck-asio — C:\Users\User\Desktop\chuck-asio — Atom
File Edit View Selection Find Packages Help
Chuctor09.ck

```
    }
    else if (crossmode == 2)
    {
      Yplayers[ychoose] => deli;
      thing.tieXYBLR(Xplayers[xchoose],deli,One,Xplayers[xchoose],deli);
    }
    else if (crossmode == 3)
    {
      Xplayers[xchoose] => deli;
      thing.tieXYBLR(Xplayers[xchoose],deli,One,Xplayers[xchoose],deli);
    }


    onXYBLR(thing);
    0.3 => thing.U[4].gain => thing.U[5].gain; // reduce audio volume (aga
```

```
//!
fun void tweaki9 (Moog     zz) /*1*/{
//////////Moog (SsTK Import)
///// STK moog-like swept filter sampling synthesis class.
announce(rand0to1(), "filterQ") => zz.filterQ;          /// ( float , READ/
announce(rand0to1(), "filterSweepRate") => zz.filterSweepRate;         ///
announce(rand0to1(), "vibratoFreq") => zz.vibratoFreq;         /// ( float
announce(rand0to1()/2, "vibratoGain") => zz.vibratoGain;         /// ( flo
announce(rand0to1(), "afterTouch") => zz.afterTouch;         /// ( float ,
}

//!
fun void tweaki10(Saxofony zz) /*1*/{
//////////Saxofony (SsTK Import)
///// STK faux conical bore reed instrument class.
```

ChucK

ReWereHere (Max/MSP)



Simple contrast that happens in Jitter before video is converted to audio. Negative values will invert image.

www.klif.tv (click here)
i.m.klif@gmail.com

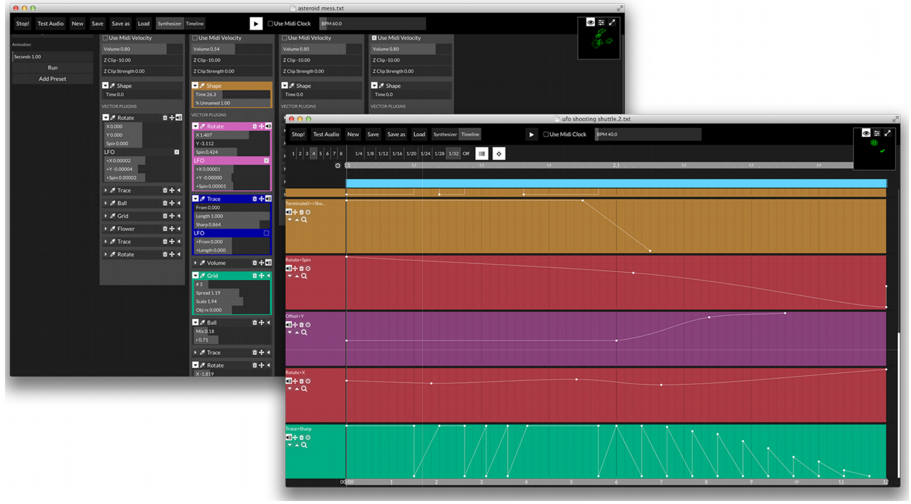
help and bugreports here:
REWEREHERE Facebook group (click here)

PRESET NOTES

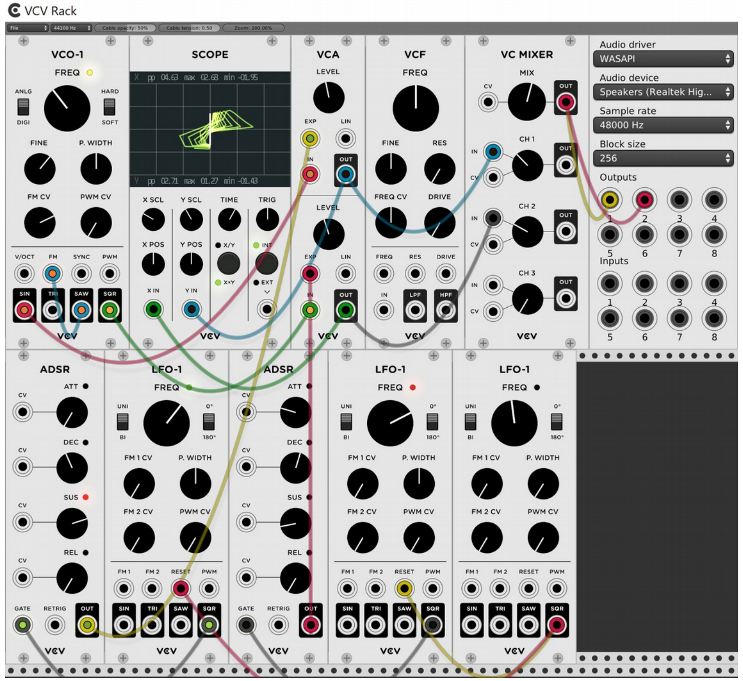
Hi there. This is REWEREHERE raster processing app. You will need an analog oscilloscope, modified Vectrex or XYZ display to really use it. .... The core of this patch was made somewhere in 2012 after exchange with Dave Jones and Lars Larsen on Mufwigglers forums. Without their help I wouldn't know where to start :) .... Click through the presets below for some help. Use HINT on to see comment of each parameter (just use your mouse to move over the parameter you're interested in)...... big thanks to Dan Nigrin for his mouseover colored boxes :)

How to connect soundcard to the oscilloscope:
ch 1=X
ch 2=Y
ch 3=Z

OsciStudio

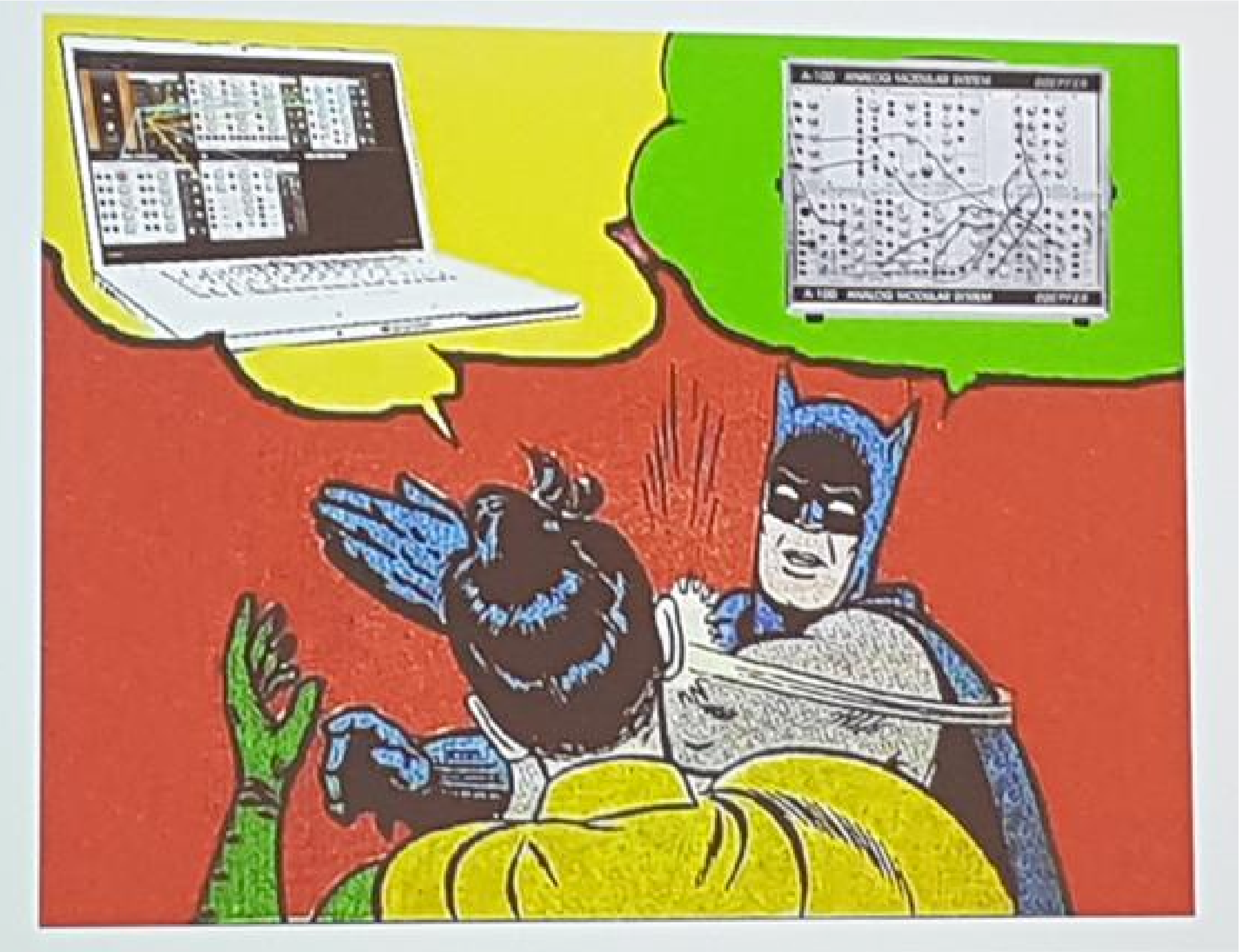


VCV Rack



Hardware or software?

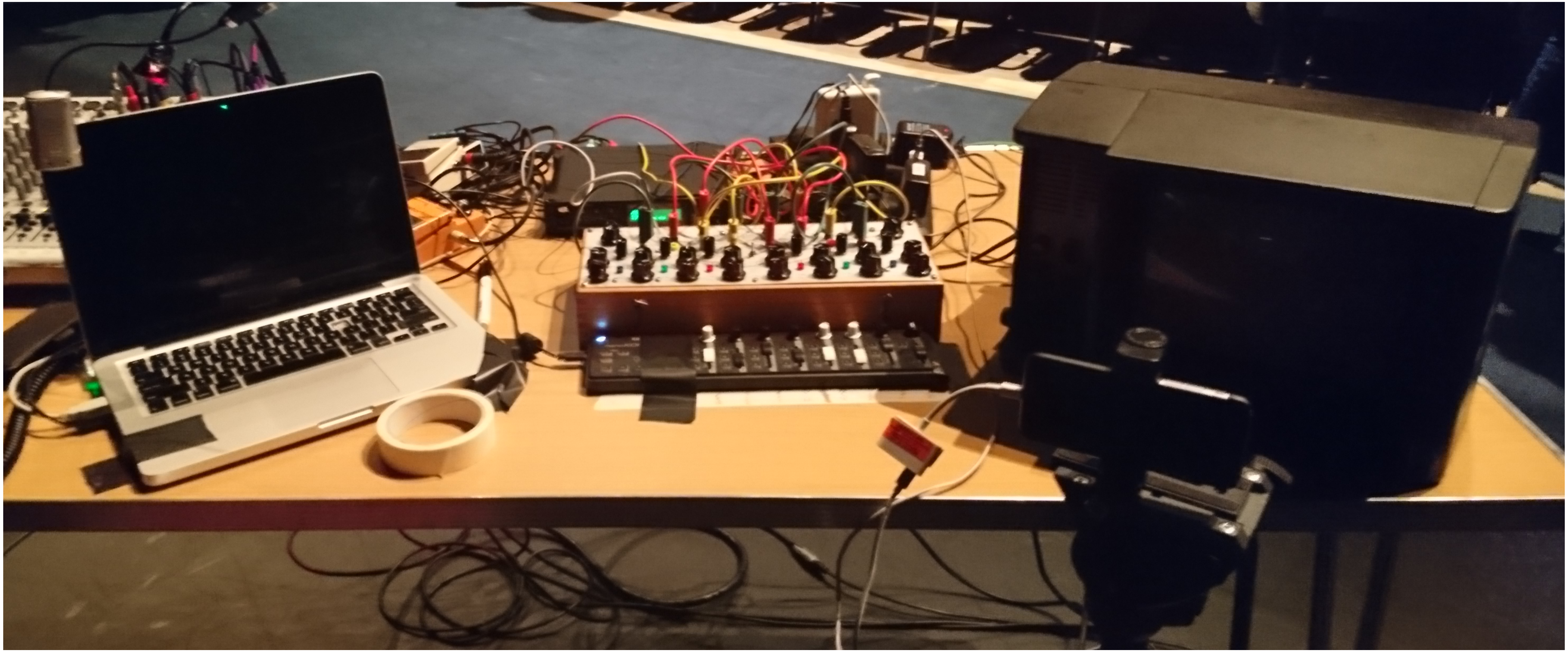
```
  (usespotkiller)
{
  puts("\nVectrex AntiSpotKiller deployed.");
  _dispX => askX => Xclip => dac.chan(1); // rotate
  _dispY =>         Yclip => dac.chan(0);
  _dispB => askZ => Bclip => dac.chan(2); // gr
```



No decision needs to be made!

Both are powerful, and both have pros and cons.

Nothing stops us using hardware AND software!



Derek Holzer’s performance setup, Bath Spa, March 2018

Like other audiovisual art,
vector graphics uses different degrees of crossmodality

- Visuals accompany audio but are independent
  or
- Events are both audio and visual, but not directly related
  or
- Visuals depict audio (e.g. waveform/spectrum display)
  or
- Visuals are identical to audio (e.g. Jerobeam Fenderson ‘Oscilloscope Music’)

Possible software for performance/synthesis

- Requirements – multichannel, real-time
- Preferences – free, open-source
- Graphical (dataflow) programming
  - **Purr Data** (based on Pure Data extended)
  - Max/MSP
  - VCV Rack – modular synth emulation
  - TouchDesigner – designed for video
- Text-based programming
  - **ChucK** – quick audio prototyping
  - Processing – designed for graphics
  - Faust – low-level DSP

Text generation in Purr Data

- First method - crude results and patch was unwieldy
  - Ringing is clearly seen, and blanking had not been implemented
  - Demonstrated pitch control

DoReMi

- Second method - converted nine fonts from ILDA to WAV using LaserBoy software, then played as samples
  - (has multiplexing artefacts)

OscilloscopeFonts

- Looks better on Vectrex, with pitch mapped to height
  - (still has multiplexing artefacts)

DingDongMerrilyOnHigh

- One alternative is XYScope for Processing (Ted Davis)

Video vectorisation – ***not*** real-time

- Aim – convert raster video to vectors (using free software)
- Process
  - Downsample video to 500*400@16
  - Detect edges in each colour plane, convert to monochrome, threshold
  - Trace edges to DXF vectors (PoTrace)
  - Convert DXF files to WAV (LaserBoy)
  - Remove half of each loop (C)
  - Optimise frames to reduce vertex count (LaserBoy)
  - Extend to 6000 vertices per frame (C) (96000 Hz / 16 fps)
- Tested with game footage, music notation and football
- Results – principle works, but ***lots*** of optimisation needed
  - For notation, a better approach might use InScore

GrandTheftAutoViceCity

ChopinNocturne

Semifinal

Real-time vectorisation can be done in XYScope!

Synthesis algorithms

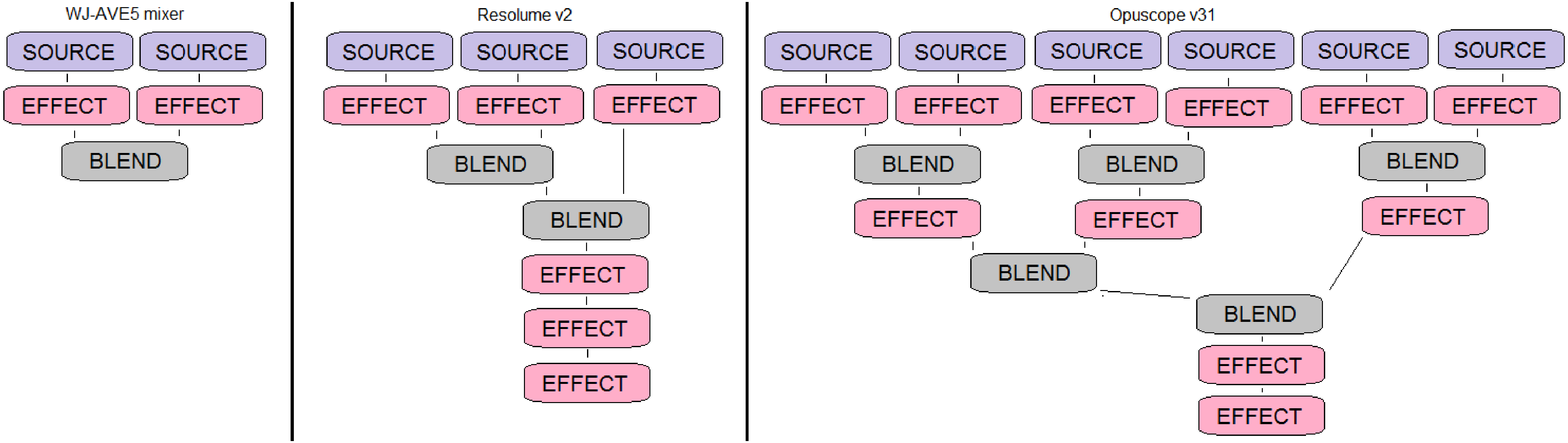
- FM/AM
- Waveshaping
- Sampled vector playback / granular synthesis
- Simulation of chaotic systems
- Rutt-Etra scan processing
    - Rewerehere (Max), Vector synthesis library (Pure Data, but not in Purr Data)
- Audio visualisation

Audio visualisation

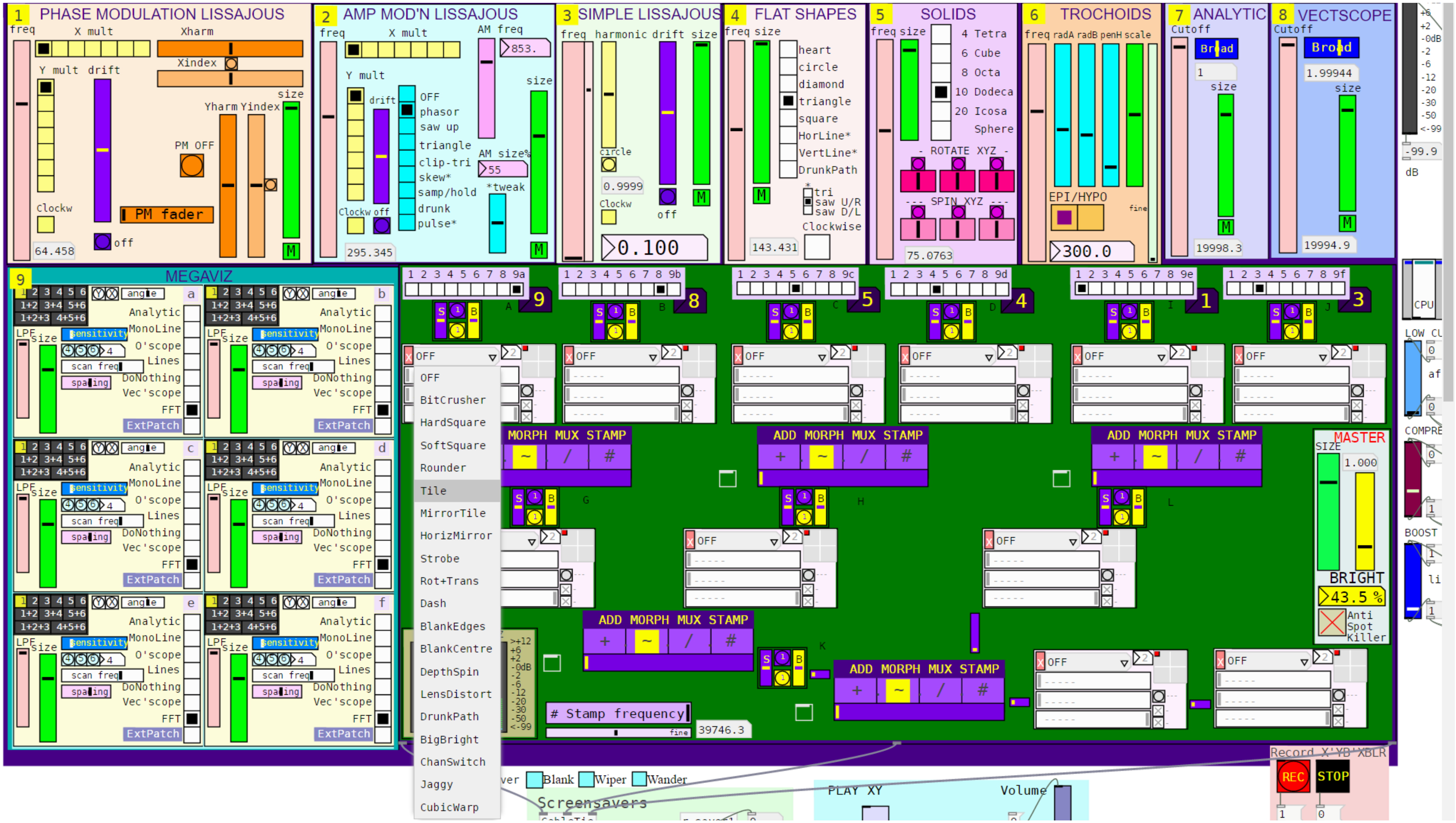
- Intended for VJ-style scenarios
- Usually benefits from compression and low-pass and high-pass filters
- Waveform display (like oscilloscope)
- Audio vectorscope (X=L+R, Y=L–R), usually used to show stereo spread
- Spectral display
- Display of “analytic” signal using the “Hilbert transform”
  - The analytic signal is a complex signal with no negative frequencies, e.g. $\cos(\omega t) \rightarrow e^{i\omega t}$
  - The Hilbert transform is non-causal so cannot be used in real time
  - The “fake” Hilbert transform is causal and generates an approximation to $e^{i(\omega t + \theta(\omega))}$
- Indirect methods – e.g. pitch tracking

Performance patch design

- Model on conventional signal flows



- SOURCES – PM/AM/simple Lissajous, 2D/3D shapes, Trochoids (“Spirograph”), Audio vectorscope, Analytic signal, Audio visualisation
- EFFECTS – bitcrusher, square↔circle, tile/mirror, strobe, rotate/translate, dash, blank edges/centre, depth spin, lens distortion, etc.


1 PHASE MODULATION LISSAJOUS
freq
X mult
Xharm
Xindex
Y mult
drift
size
Yharm
Yindex
PM OFF
Clockw
PM fader
off
M
64.458
2 AMP MOD'N LISSAJOUS
freq
X mult
AM freq
853.
Y mult
drift
OFF
phasor
saw up
triangle
clip-tri
skew*
samp/hold
drunk
pulse*
AM size%
55
*tweak
size
Clockw
off
M
295.345
3 SIMPLE LISSAJOUS
freq
harmonic
drift
size
circle
0.9999
Clockw
off
M
0.100
4 FLAT SHAPES
freq
size
heart
circle
diamond
triangle
square
HorLine*
VertLine*
DrunkPath
*tri
saw U/R
saw D/L
Clockwise
M
143.431
5 SOLIDS
freq
size
4 Tetra
6 Cube
8 Octa
10 Dodeca
20 Icosa
Sphere
- ROTATE XYZ -
--- SPIN XYZ ---
75.0763
6 TROCHOIDS
freq
radA
radB
penH
scale
EPI/HYPO
fine
300.0
7 ANALYTIC
Cutoff
Broad
1
size
M
19998.3
8 VECTSCOPE
Cutoff
Broad
1.99944
size
M
19994.9
+6
+2
-0dB
-2
-6
-12
-20
-30
-50
<-99
-99.9
dB
CPU
LOW CUT
af
COMPRESS
BOOST
li
9 MEGAVIZ
1 2 3 4 5 6
1+2 3+4 5+6
1+2+3 4+5+6
angle
a
b
c
d
e
f
Analytic
MonoLine
O'scope
Lines
DoNothing
Vec'scope
FFT
ExtPatch
LPF
size
sensitivity
4 5 6 4
scan freq
spacing
1 2 3 4 5 6 7 8 9a
1 2 3 4 5 6 7 8 9b
1 2 3 4 5 6 7 8 9c
1 2 3 4 5 6 7 8 9d
1 2 3 4 5 6 7 8 9e
1 2 3 4 5 6 7 8 9f
9
8
5
4
1
3
A
B
C
D
I
J
G
H
L
K
S
1
B
OFF
2
OFF
BitCrusher
HardSquare
SoftSquare
Rounder
Tile
MirrorTile
HorizMirror
Strobe
Rot+Trans
Dash
BlankEdges
BlankCentre
DepthSpin
LensDistort
DrunkPath
BigBright
ChanSwitch
Jaggy
CubicWarp
ADD MORPH MUX STAMP
+ ~ / #
MASTER
SIZE
1.000
BRIGHT
43.5 %
Anti Spot Killer
>+12
+6
+2
-0dB
-2
-6
-12
-20
-30
-50
<-99
# Stamp frequency
fine
39746.3
ver
Blank
Wiper
Wander
Screensavers
PLAY XY
Volume
Record X'YB'XBLR
REC
STOP
1
0

Production methodology with Purr Data

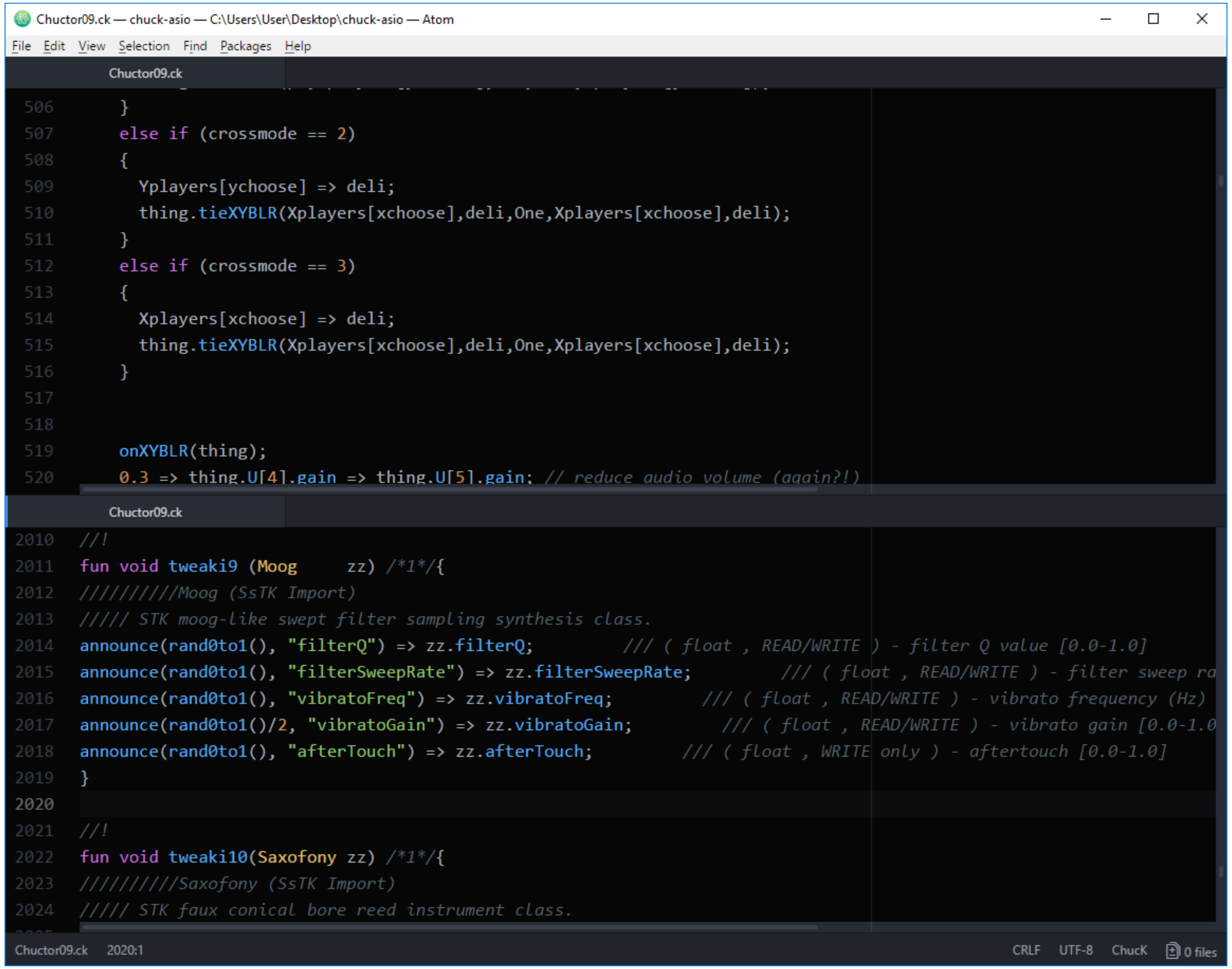
EggboyLightFade

- Recording of performance
  - Audio from second computer - Eggboy album ‘thirteenpointeight’ and Kuba ‘Animalia’
  - PD records the output vectors and audio
  - Vectrex and PD audio output captured by HD camera (Canon XA10)
  - Synchronisation of recorded video with original audio using Audacity/FFMPEG
  - Can replay vectors in order to record video
- Appraisal
  - Purr Data allows a graphical interface but programming is less straightforward
  - Would benefit from external (i.e. MIDI) controller and/or keyboard controls
  - Audio clicks due to refreshing GUI

ChucK experiments

- Text-based language, closest to C
- Developed by Ge Wang (Princeton)
- Active, but sporadic, development
- MiniAudicle IDE not used (doesn't support ASIO)
  - ATOM editor template available

- Experiments
  - Spectral display
  - Synthesis ToolKit ("Monsters")
  - Chaotic oscillators
  - Joystick control


Chuctor09.ck — chuck-asio — C:\Users\User\Desktop\chuck-asio — Atom
File Edit View Selection Find Packages Help
Chuctor09.ck

```
    }
    else if (crossmode == 2)
    {
      Yplayers[ychoose] => deli;
      thing.tieXYBLR(Xplayers[xchoose],deli,One,Xplayers[xchoose],deli);
    }
    else if (crossmode == 3)
    {
      Xplayers[xchoose] => deli;
      thing.tieXYBLR(Xplayers[xchoose],deli,One,Xplayers[xchoose],deli);
    }


    onXYBLR(thing);
    0.3 => thing.U[4].gain => thing.U[5].gain; // reduce audio volume (again?!)
```

```
//!
fun void tweaki9 (Moog     zz) /*1*/{
//////////Moog (SsTK Import)
///// STK moog-like swept filter sampling synthesis class.
announce(rand0to1(), "filterQ") => zz.filterQ;         /// ( float , READ/WRITE ) - filter Q value [0.0-1.0]
announce(rand0to1(), "filterSweepRate") => zz.filterSweepRate;         /// ( float , READ/WRITE ) - filter sweep ra
announce(rand0to1(), "vibratoFreq") => zz.vibratoFreq;         /// ( float , READ/WRITE ) - vibrato frequency (Hz)
announce(rand0to1()/2, "vibratoGain") => zz.vibratoGain;         /// ( float , READ/WRITE ) - vibrato gain [0.0-1.0
announce(rand0to1(), "afterTouch") => zz.afterTouch;         /// ( float , WRITE only ) - aftertouch [0.0-1.0]
}

//!
fun void tweaki10(Saxofony zz) /*1*/{
//////////Saxofony (SsTK Import)
///// STK faux conical bore reed instrument class.
```

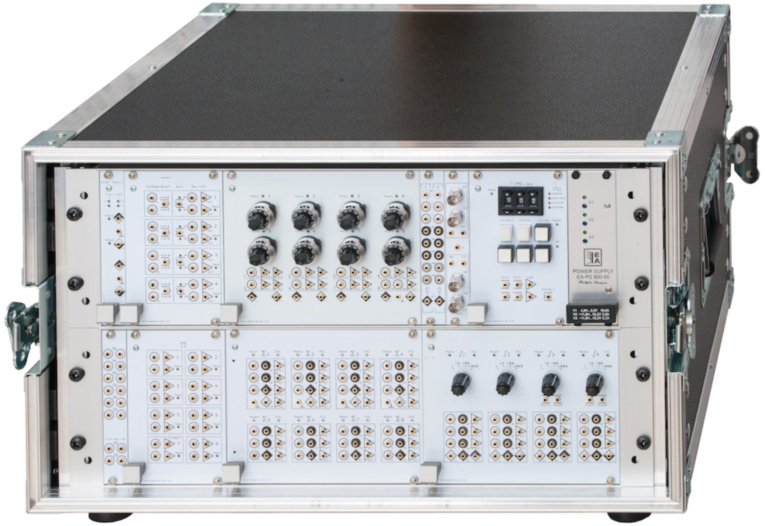
Synthesis ToolKit (Monsters)

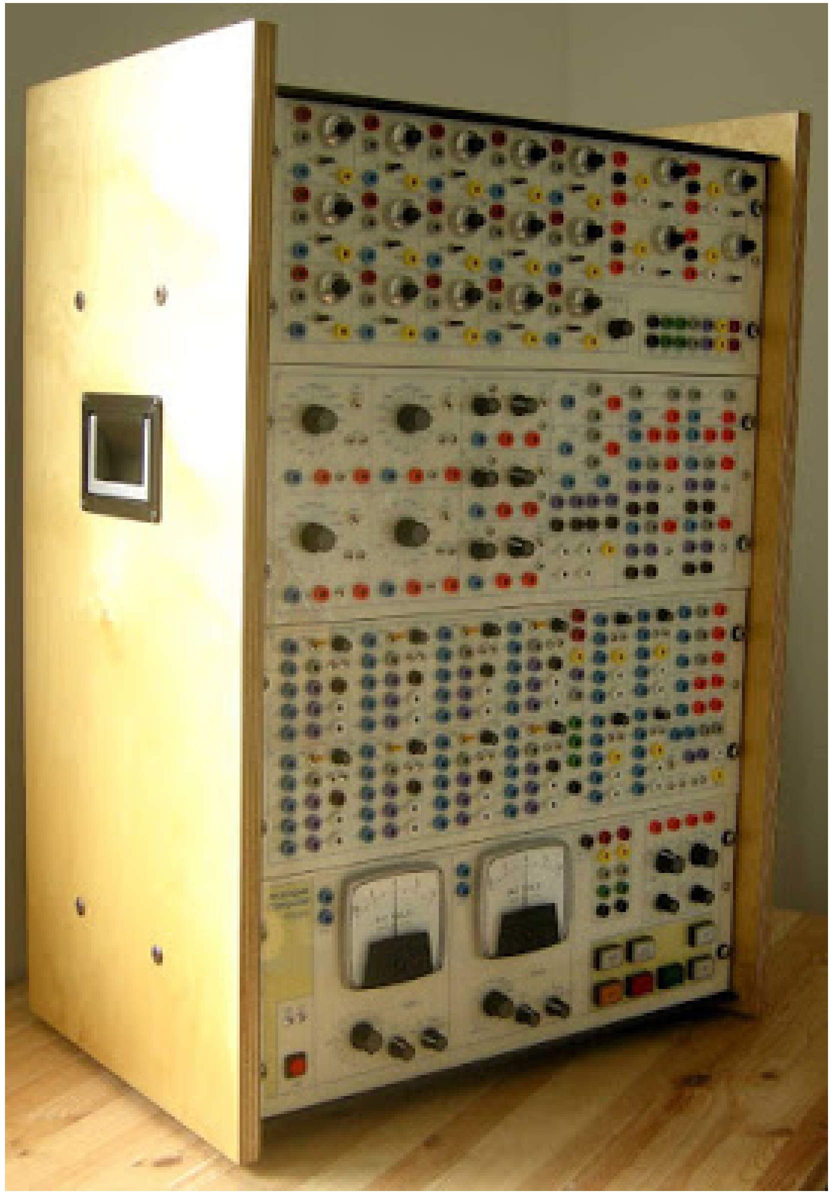
- CCRMA Synthesis ToolKit (Stanford)
  - Audio synthesis library implemented in ChucK (and other languages)
- Patch randomly picks X and Y instruments with random parameters
  - Either different instruments on X and Y, or
  - Same instrument on X and Y, but different parameters, or
  - Same instrument on X and Y, with same parameters, but Y delayed by ¼ cycle
- Range of random parameters adjusted (many are inaudible/invisible)

SynthesisToolkitMonsters

Chaotic oscillators

- Sample-rate integration is tricky in PD
- Inspired by analogue computers
  - Analog Paradigm (German manufacturer)
  - Analog Ontology (home-built)
- Fourteen chaotic oscillators made in ChucK



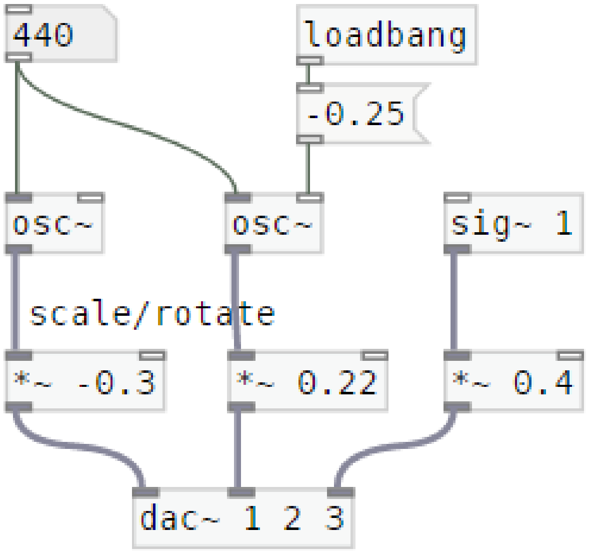


ChuckChaosCaptioned!

Graphical or text-based?

Purr Data

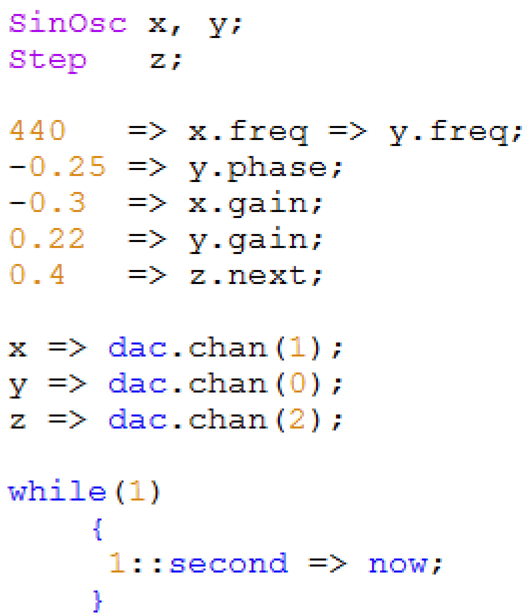
- one of several variants of Pure Data
- very flexible GUI
- unconventional programming paradigm
- good for patching, poor for sequencing
- better for real-time use



Both can have audio glitches when the graphics are updated.

ChucK

- lack of GUI (on PC)
- more flexible
- sequencing is easier
- less actively developed, small user base
- better for offline rendering



```
SinOsc x, y;
Step   z;

440   => x.freq => y.freq;
-0.25 => y.phase;
-0.3  => x.gain;
0.22  => y.gain;
0.4   => z.next;

x => dac.chan(1);
y => dac.chan(0);
z => dac.chan(2);

while(1)
    {
     1::second => now;
    }
```

Use both? PD/ChucK can communicate via MIDI/OSC

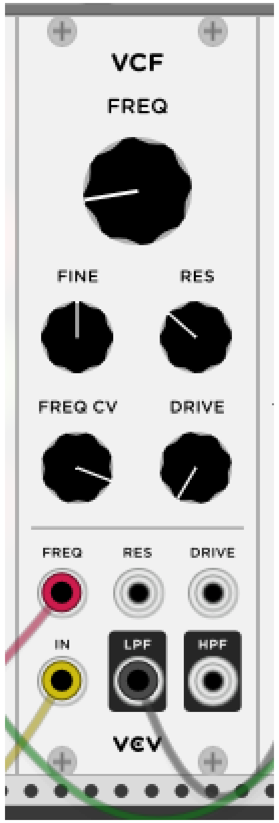
Other software to consider?

- OsciStudio (talk on Thursday)
- Processing plus XYScope library (talk on Thursday)
- Max/MSP plus ReWereHere patch (talk on Friday)
- Axoloti Patcher (talk on Saturday)

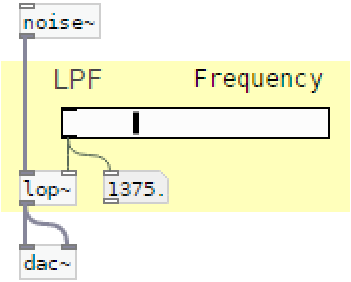
- VCV Rack
- TouchDesigner
- Faust
- Other audio programming environments e.g. Csound, Supercollider
- Other audio tools e.g. Ableton Live
- Non-audio programming environments e.g. Matlab, Octave

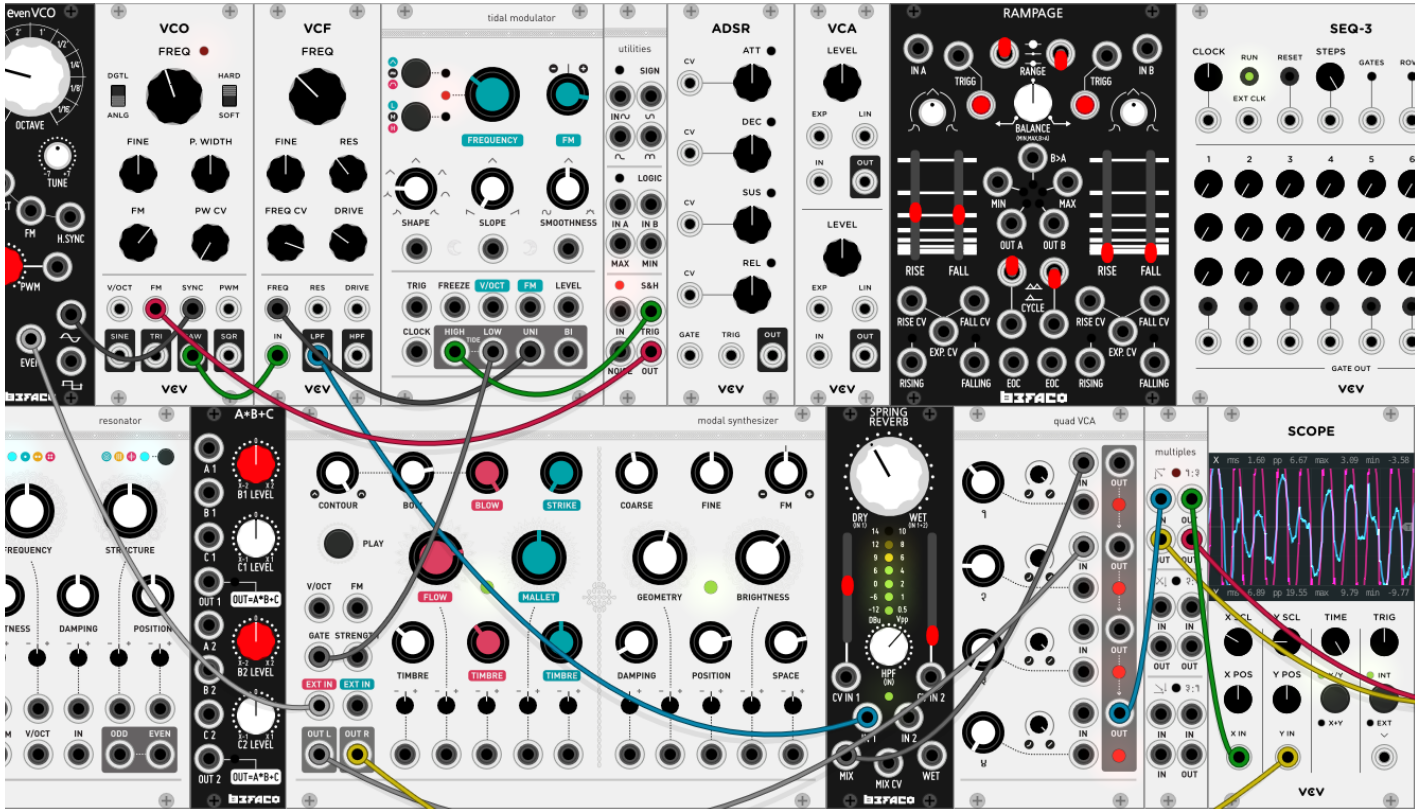
VCV Rack

- Open-source virtual modular synth (vcvrack.com)
- Program is free, modules free or paid for
- Powerful but processor-intensive
- Wastes screen space?



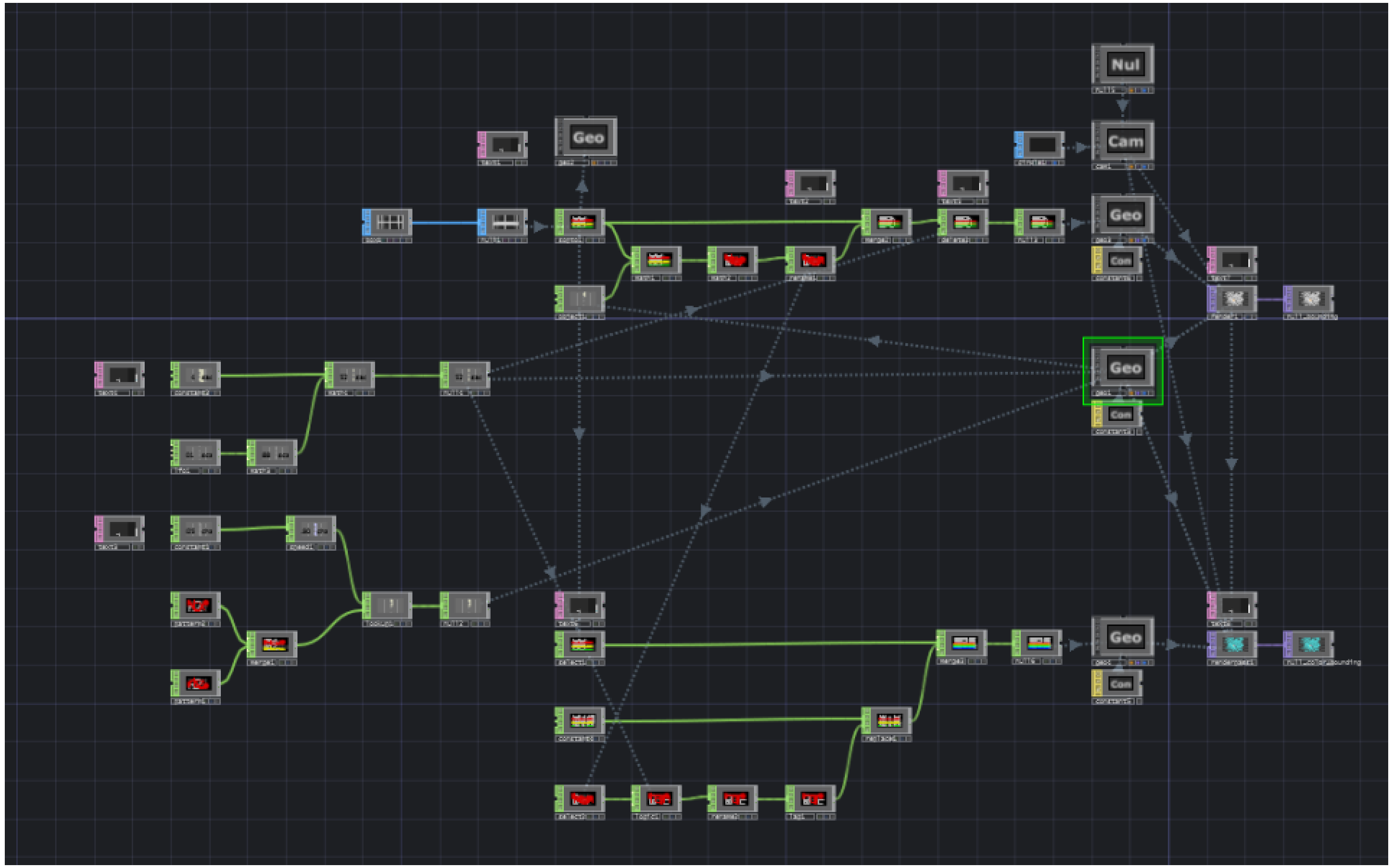
vs.





TouchDesigner

- Patching environment for video (derivative.ca)
- Free for non-commercial use
- Graphical, but can also use Python



Faust

- Functional programming language for DSP, (see faust.grame.fr)
- Very terse code, which can be:
  - Compiled into C/C++
  - Run in PD, ChucK, SuperCollider etc.
  - Made into an external (Max/MSP, PD, SuperCollider, Csound) or a VST plug-in
- Includes audio libraries, physical models and GUI

```
// Simple Organ
import("stdfaust.lib");

midigate  = button ("gate");                               // MIDI keyon-keyoff
midifreq  = hslider("freq[unit:Hz]", 440, 20, 20000, 1);   // MIDI keyon key
midigain  = hslider("gain", 0.5, 0, 10, 0.01);             // MIDI keyon velocity

process = voice(midigate, midigain, midifreq) * hslider("volume", 0, 0, 1, 0.01);

// Implementation

phasor(f)   = f/ma.SR : (+,1.0:fmod) ~ _ ;
osc(f)      = phasor(f) * 6.28318530718 : sin;

timbre(freq)= osc(freq) + 0.5*osc(2.0*freq) + 0.25*osc(3.0*freq);

envelop(gate, gain) = gate * gain : smooth(0.9995)
                with { smooth(c) = * (1-c) : + ~ * (c) ; } ;

voice(gate, gain, freq) = envelop(gate, gain) * timbre(freq);
```

Software comparison

L	cost	Win	Mac	Linux	designed for	interface	notes
PD	free	✓	✓	✓	audio	graphical	several distributions
VS library for PD	free	✓	✓	✓	vectors	graphical	
Max/MSP	$100/year	✓	✓	✗	audio	graphical	
ReWereHere for Max/MSP	free	✓	✓	✗	vectors	graphical	
Axoloti Patcher	free	✗	✗	✗	vectors	graphical	no brightness control, needs Axoloti hardware
TouchDesigner	free (non-commercial)	✓	✓	✗	video	graphical	
ChucK	free	✓	✓	✓	audio	text	
Processing	free	✓	✓	✓	graphics	text	
XYscope for Processing	free	✓	✓	✓	vectors	text	
Faust	free	✓	✓	✓	audio	text	
VCV Rack	free (some paid)	✓	✓	✓	audio	GUI	
Oscistudio	€34	✓	✓	✗	vectors	GUI	
LaserBoy	free	✓	✓	✓	lasers	text	unusual interface

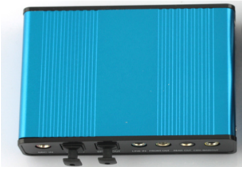
Remaining issues

- Results very dependent on characteristics of display device
  - Bandwidth
  - CRT persistence
  - Spot killer (on Vectrex)
  - Graticule (on oscilloscope)

- Capturing the display raises issues
  - Frame rate
  - Blocking outside light
  - Colour balancing
  - Screen curvature

Software still needs hardware

- The programs are mostly mouse-controlled, but MIDI controllers are very useful
- Audio interface with at least 3 channels (X, Y, brightness), DC coupling, and ideally a high sample rate



Modded Cmedia

Most MOTU interfaces

Echo AF4

- HD camera / SD camera / Smartphone
- Raster display device

Conclusions

- Audio tools are an efficient way to generate vector graphics
- The display device imposes restrictions on the methodologies
- Both PD and ChucK are suitable software platforms, each with clear pros/cons
  - choice depends on the application scenario
  - both can be used simultaneously (e.g. synthesis in ChucK, GUI in PD)
  - both benefit from physical controllers
- Real-time performance is easier in PD
  - But graphical programming is awkward for complex tasks
- Non-real-time rendering is easier in ChucK
  - Lack of GUI is the biggest disadvantage
- Other software deserves further evaluation

References and resources

- Holzer D., Vector Synthesis – an investigation into sound-modulated light – http://www.econtact.ca/19_2/holzer_vectorsynthesis.html
- Vectrex modification – http://users.sussex.ac.uk/~ad207/adweb/assets/vectrexminijackinputmod2014.pdf
- PD Vector Synthesis library – https://github.com/macumbista/vectorsynthesis
- Rewerehere – https://www.facebook.com/groups/REWEREHERE
- Video Circuits – https://www.facebook.com/groups/VIDEOCIRCUITS
- DC-coupled audio interfaces – http://www.expert-sleepers.co.uk/siwacompatibility.html
- CMedia soundcard modification – http://www.whence.com/soundcard-dc-dac
- Oscilloscope emulator – https://github.com/kritzikratzi/Oscilloscope
- Pure Data – https://puredata.info
- ChucK – http://chuck.cs.princeton.edu
- LaserBoy – http://laserboy.org
- PoTrace – http://potrace.sourceforge.net