supercollider as a synthesis tool to create an audio visual performance

about myself

audiovisual performance

realtime
playing the computer with no timeline
DIY controllers and lights
audiovisual synthesis

designing and engineering scenography

intergrating instruments into scenography
lights
conceptual aproach

DIY to engineering

design pcbs
3d modeling for digital fabrication techniques
coding sound and visual tools

past projects

Quadr

past projects

AntiVolume v0.1

past projects

AntiVolume v1

past projects

AntiVolume v2

past projects

AntiVolume v3

current project

AntiVolume v4

timeline

about myself

10 years old

scratching/turntablism

13 years old

controllerism

midi controller and ableton
my parents are musicians
aproaching the computer as an instrument

16 years old

DIY controllers

sketching custom built controllers

monodeck by robert henke http://roberthenke.com/technology/monodeck.html

midibox (pre Arduino) (fail) http://www.midibox.org/dokuwiki/doku.php?id=user_projects

monome & arduinome (fail)

tried learning supercollider for the first time (fail)

max/msp processing arduino

beginning of multitouch: lemur & first iphone

20 years old

studies

moved to Montreal

studies in digital music at UdeM

Montreal community and festivals

mutek.org

going in depth

going more into visuals

tried learning supercollider for the second time (fail)

24 years old

post-studies

lots of technical work for other artists
Herman Kolgen, Adam Basanta, Nicolas Bernier…
First artist residency
small concerts with friends in Montreal
opening up to the Montreal art scene

26 years old - to now

full-time artist

AntiVolume v0.1 to v4

residencies

playing Mutek Montreal

first international concerts

france - uk - usa - southkorea - mexico - spain - switzerland - croatia

about supercollider

released in 1996 by James McCartney
real-time audio synthesis and algorithmic composition
supercollider 3

what can you do with it?

live coding
building instruments
sound installations
algorithmic composition
& more

similar software

max/msp

reaktor

pure data

why supercollider

expressive

unlimited

stable

powerfull in large scale complex project why did you take this workshop?

it’s very technical
but remember
it’s about sculpting signals

it’s actually only logic and middle school math

- addition
- subtraction
- multiplication
- division
- sinus
- modulo

realtime synthesis

concepts in this workshop

modular synthesis

lego for sound

freedom from traditional commercial synthesis paradigms

resurgence of analog modular synthesis

code as language

writing quick and expressive sketches
writing a more complex project

audio visual synchronisation

pre audio reactive visuals
using OSC to share control signals between sound and visual software

audio visual synchronisation

Touch Designer

also modular
shares similarities with audio synthesis tools

some concepts about sound

audio frequency range

20-20000hz

audio amplitude

in decibels -96DB to 0DB

sine waves and spectrum

basics of supercollider

installation

getting started

download the cheatsheet!

github.com/lucasParis/bragaWorkshop

getting started

important keyboard shortcuts

execute line - cmd + enter
stop sound - cmd + .

getting started

Server / Interpreter

boot server - cmd + b

getting started

making sound

```
{ SinOsc.ar(200) }.play
```

getting started

arguments

```
{ SinOsc.ar(freq:200, mul:0.1) }.play
```

getting started

arguments

simple math
mul and add

getting started

ugens

are the building blocks for making sound

getting started

beware of sonic explosions!!

when first testing a sound don’t wear your headphones completely
always have your hands ready for cmd + . to stop sound

exercice 1

your first sounds

SinOsc WhiteNoise LFSaw

getting started

exercice 1.1

generating tones

getting started

exercice 1.2

modulating tones

using ( ) to execute multiple lines

```
(
a = {
	Limiter.ar(
		SineShaper.ar(
			LFSaw.ar(-10).range(0,1).pow(100).linlin(0,1,0,200) + SinOsc.ar(40)*LFSaw.ar(40.3).range(-5,10) + ClipNoise.ar(0.3),
			LFSaw.ar(-00.1).range(0.2,5)
		)
	)!2;
}.play;
)
```

getting started

control rate and audio rate

.kr

.ar

exercice 1.3

controlling with MouseX MouseY

modifying the example

getting started

ugens

finding more ugens

in the cheatsheet

using the documentation

getting started

lists

exercice 2

using lists

generating tones with lists and SinOsc

some extra notions for later

arguments and variables

some extra notions for later

types

string
int
float
ugen
bus

some extra notions for later

triggers and envelopes

some extra notions for later

sequencing with patterns

about synthesis

evolution in learning digital music

software built-in preset and samples
custom made presets and samples
custom made tools

about synthesis

building blocks of synthesis

oscilators

filters

modifiers

about synthesis

techniques of synthesis

frequency modulation

ring modulation

additive synthesis

how to synthesize popular sounds

kick

snare

hihats

pads

synth sequences

noise

controlling with MIDI and OSC

Lemur
Open Stage Control

using linlin linexp lincurve

AFTERNOON

AFTERNOON

AFTERNOON

inner workings of AntiVolume

workflow concepts

“what comes first? the visuals or the sound?”

back and forth between sound and visuals the art of fine tuning of controls

lutherie

more philosophy

learning code, understanding technical systems
finding your voice with unconventional austere tools in the preset style age
investing time in yourself with complicated but rewarding tools
learning and personal growth
being technical shouldn’t mean killing creativity
struggling with a creativity/tool building balance
importance of sketches simple and light way to create in the present
bottom up/top down approach

more advanced supercollider

afternoon project

making a synthesis instrument for performance sketch some sounds

use midi or OSC to create controls

think about what and how you want to control