



# ADCx GATHER

[TEACHING PEOPLE HOW TO]  
*BUILD AI-ENHANCED AUDIO PLUGINS*

**MATTHEW YEE-KING**

# OBJECTIVES FOR THIS TALK

- Describe the process of writing an audio dev book
- Demonstrate pedagogical audio dev software and explain the thinking behind it
- Present examples of audio technology research that feed into the examples



Google Scholar

**Matthew Yee-King** [FOLLOW](#) [GET MY OWN PROFILE](#)

Full Professor of Computer Science, [Goldsmiths College](#), University of London  
 Verified email at gold.ac.uk - [Homepage](#)  
 Computer Music Sound Synthesis Autonomic Agents Education Technology Computing Education

TITLE	CITED BY	YEAR
<a href="#">In a silent way: Communication between ai and improvising musicians beyond sound</a> J McCormack, T Gifford, P Hutchings, MT Llano Rodríguez, M Yee-King, ... Proceedings of the 2019 chi conference on human factors in computing systems ...	84	2019
<a href="#">Automatic programming of VST sound synthesizers using deep networks and other techniques</a> MJ Yee-King, L Fedden, M d'Inverno IEEE Transactions on Emerging Topics in Computational Intelligence 2 (2 ...	71	2018
<a href="#">Design considerations for real-time collaboration with creative artificial intelligence</a> J McCormack, P Hutchings, T Gifford, M Yee-King, MT Llano, M D'Inverno Organised Sound 25 (1), 41-52	65	2020
<a href="#">Explainable computational creativity</a>	55	2022

Cited by [VIEW ALL](#)

	All	Since 2019
Citations	779	556
h-index	15	11
i10-index	24	13

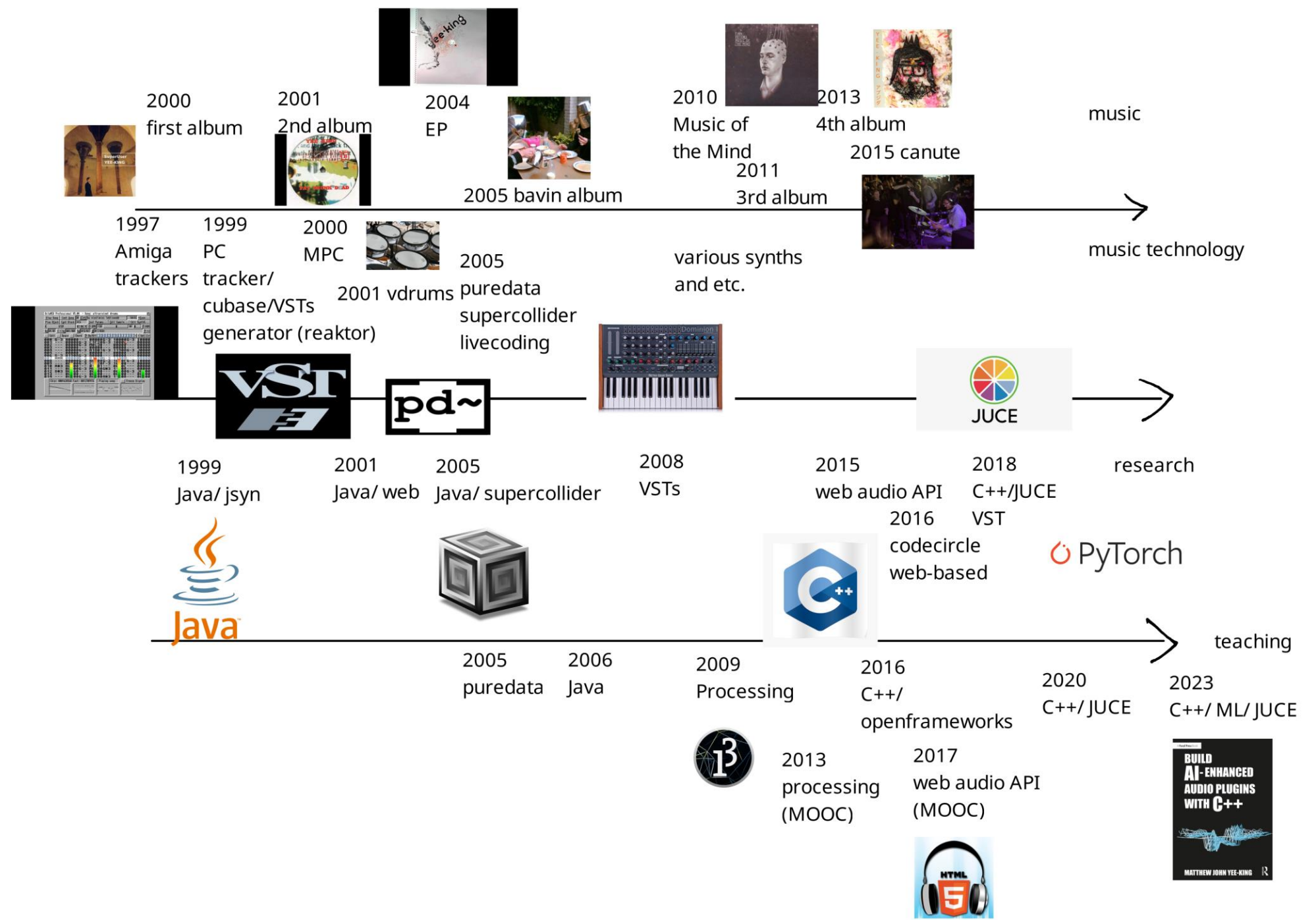
Year	Citations
2017	~10
2018	~15
2019	~25
2020	~25
2021	~45
2022	~45
2023	~110
2024	~100

Secret sauce: three frames

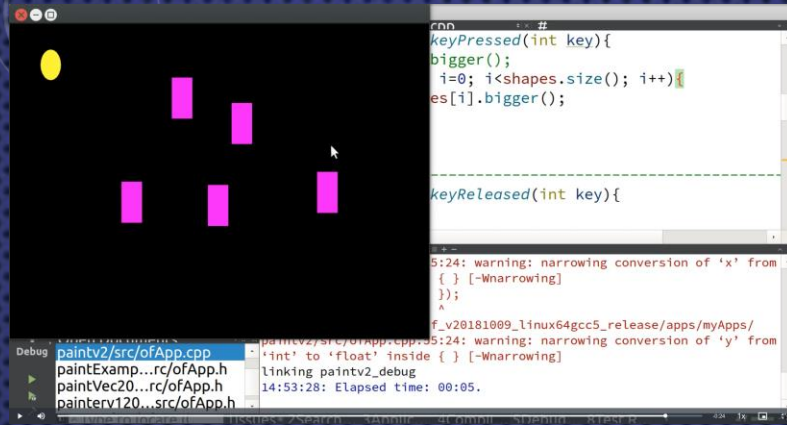
WHO AM I?



# Music, Technology, Research and Teaching Timeline

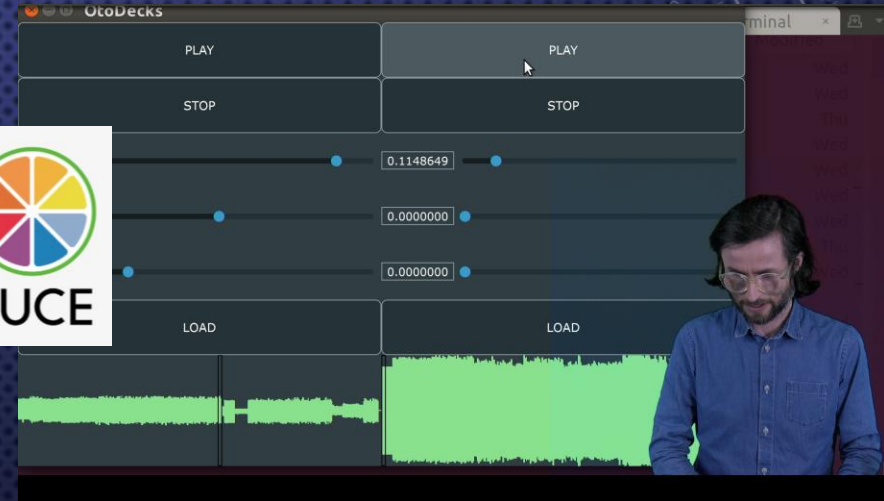


WHY WRITE A BOOK?



2015-2018: Teaching C++/ OpenFrameworks

2020: Developed a new C++ course



2023: Developed a new AI/ Music course

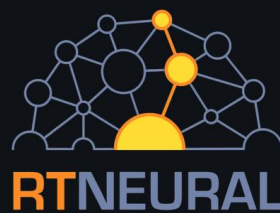


**POLITECNICO**  
MILANO 1863

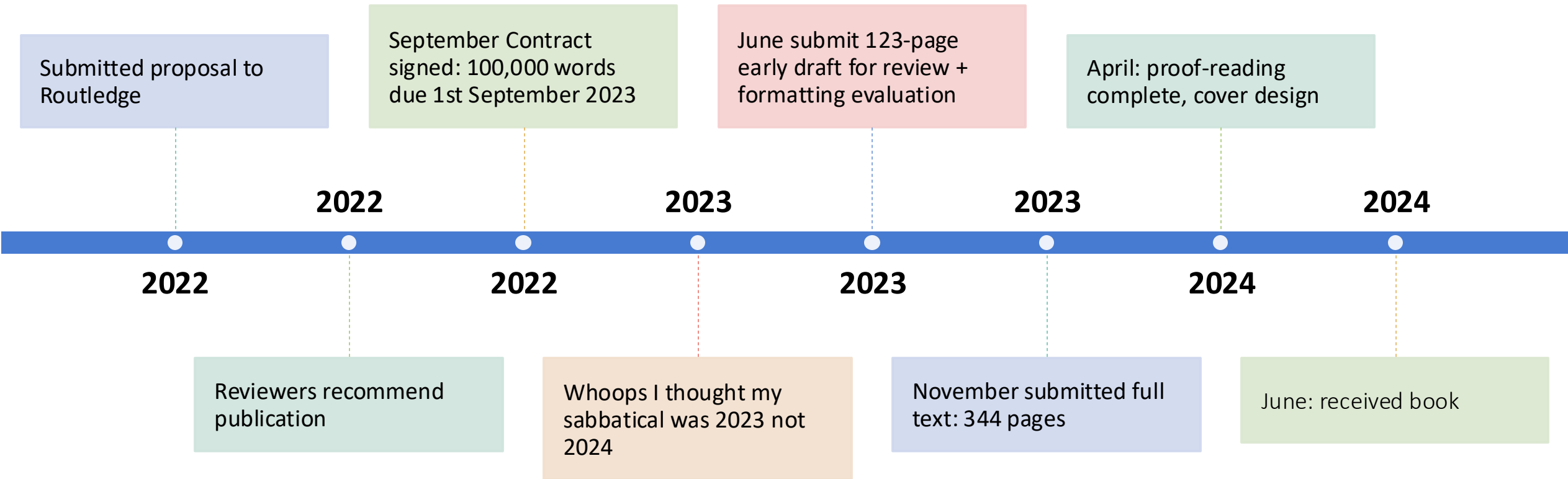
DIPARTIMENTO DI ELETTRONICA  
INFORMAZIONE E BIOINGEGNERIA

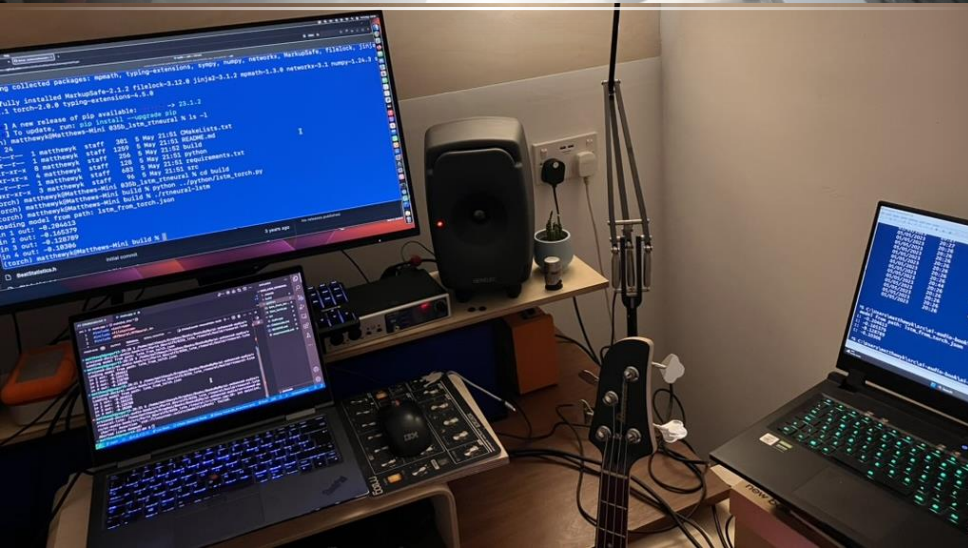


PyTorch



# TIME FOR WRITING THE BOOK





# DEVELOPING THE EXAMPLES IN THE BOOK

- Aimed at students, developers (audio as well as machine learning)
- Cross platform, flexible workflow
  - CMake, JUCE, libtorch, C++
- Range of DSP and machine learning techniques
- Three examples: **amp simulator**, **improviser** and **metacontroller**

DEV  
WORKFLOW:  
CLI

```
├── CMakeLists.txt
├── README.md
└── src
    ├── PluginEditor.cpp
    ├── PluginEditor.h
    ├── PluginProcessor.cpp
    └── PluginProcessor.h
```

2 directories, 6 files

SET UP

GEN

PROJECT

```
cmake_minimum_required(VERSION 3.15)
set(CMAKE_CXX_STANDARD 17)
project(JUCE_PLUGIN_WITH_TORCH VERSION 0.0.1)

### some libtorch stuff
https://forum.juce.com/t/using-libtorch-with-juce/4
# list(APPEND CMAKE_PREFIX_PATH "<my_path_to>/libtorc
set(CMAKE_PREFIX_PATH "~/src/sw/libtorch-250/") # loc
#set(CMAKE_PREFIX_PATH "C:/Users/matthewyk/src/sw/lib

find_package(Torch REQUIRED)
set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} ${TORCH_CXX_F
# and libtorch stuff

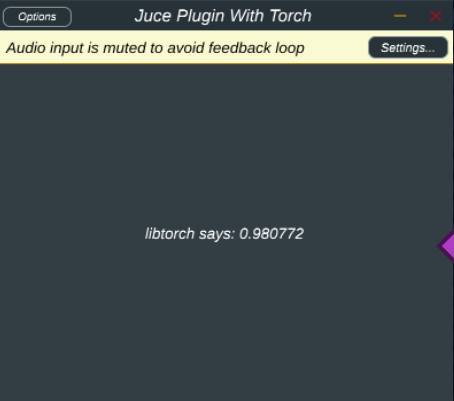
add_subdirectory(../../JUICE ./JUICE)
```

```
matthew@TeaForTwo:20:55:00$ cmake -B build .
-- Found Torch: /home/matthew/src/sw/libtorch-250/lib/libto
-- Found PkgConfig: /usr/bin/pkg-config (found version "1.8
-- Checking for module 'alsa'
--   Found alsa, version 1.2.11
-- Checking for module 'freetype2'
--   Found freetype2, version 26.1.20
-- Checking for module 'libcurl'
--   Found libcurl, version 8.5.0
-- Checking for modules 'webkit2gtk-4.0;gtk+-x11-3.0'
--   Package 'webkit2gtk-4.0', required by 'virtual:world',
--   Configuring juceaide
-- Building juceaide
```

BUILD

```
[ 97%] Building CXX object CMakeFiles/JucePluginWithTorch_Standalone.dir/home/m
matthew@TeaForTwo:20:56:37$
[ 97%] Building CXX object CMakeFiles/JucePluginWithTorch_Standalone.dir/home/m
[ 97%] Linking CXX executable "JucePluginWithTorch_artefacts/Standalone/Juce Pl
gin With Torch"
[ 97%] Built target JucePluginWithTorch_Standalone
[100%] Linking CXX shared module "JucePluginWithTorch_artefacts/VST3/Juce Plugi
With Torch.vst3/Contents/x86_64-linux/Juce Plugin With Torch.so"
-- Installing: /home/matthew/.vst3/Juce Plugin With Torch.vst3
-- Installing: /home/matthew/.vst3/Juce Plugin With Torch.vst3/Contents
-- Installing: /home/matthew/.vst3/Juce Plugin With Torch.vst3/Contents/x86_64-
linux
-- Installing: /home/matthew/.vst3/Juce Plugin With Torch.vst3/Contents/x86_64-
linux/Juce Plugin With Torch.so
[100%] Built target JucePluginWithTorch_VST3
matthew@TeaForTwo:20:56:37$
```

RUN





# DEV WORKFLOW: VSCODE

The screenshot shows the Visual Studio Code interface. The top editor displays the `CMakeLists.txt` file with the following content:

```
1  
2 cmake_minimum_required(VERSION 3.15)  
3 set(CMAKE_CXX_STANDARD 17)  
4 project(JUCE_PLUGIN_WITH_TORCH_VERSION 0.0)
```

The terminal window below shows the output of CMake:

```
[cmake] -- Found alsa, version 1.2.11  
[cmake] -- Checking for module 'freetype2'  
[cmake] -- Found freetype2, version 26.1.20  
[cmake] -- Checking for module 'libcurl'  
[cmake] -- Found libcurl, version 8.5.0  
[cmake] -- Checking for modules 'webkit2gtk-4.0;gtk+-x11-3.0'  
[cmake] -- Package 'webkit2gtk-4.0',  
required by 'virtual:world', not found  
[cmake] -- Configuring juceaide  
[cmake] -- Building juceaide
```

The Explorer sidebar on the right shows the project structure:

- 008\_LIBTORCH\_AND\_JUCE
  - build
  - src
    - PluginEditor.cpp
    - PluginEditor.h
    - PluginProcessor.cpp
    - PluginProcessor.h
  - CMakeLists.txt
  - README.md

The status bar at the bottom shows the current mode as `[Debug]`, the build kit as `[GCC 13.2.0 x86_64-l...]`, and the build button (gear icon) is highlighted with a purple arrow.

MODE

BUILD KIT

BUILD

RUN

The screenshot shows the `Juce Plugin With Torch` options dialog box. It has a title bar with `Options` and `Juce Plugin With Torch`. A yellow warning bar at the top says `Audio input is muted to avoid feedback loop` with a `Settings...` button. The main area is dark grey and contains the text `libtorch says: 0.980772`.

# PEDAGOGICAL CHALLENGE: LOTS OF LAYERS

CLI  
GUI  
AUDIO  
ML  
IDE  
PROJECT  
PYTHON  
C++

COGNITIVE LOAD

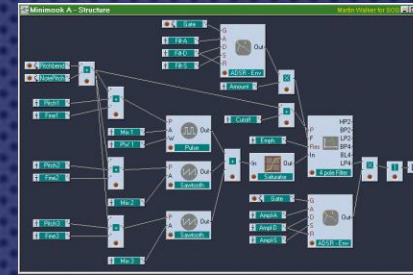


# META-CONTROLLER INSPIRATION 1

## INSPIRATIONAL RESEARCH EXAMPLE: EVOSYNTH

[HTTP://WWW.YEEKING.NET/EVOSYNTH](http://www.yeeeking.net/evosynth)

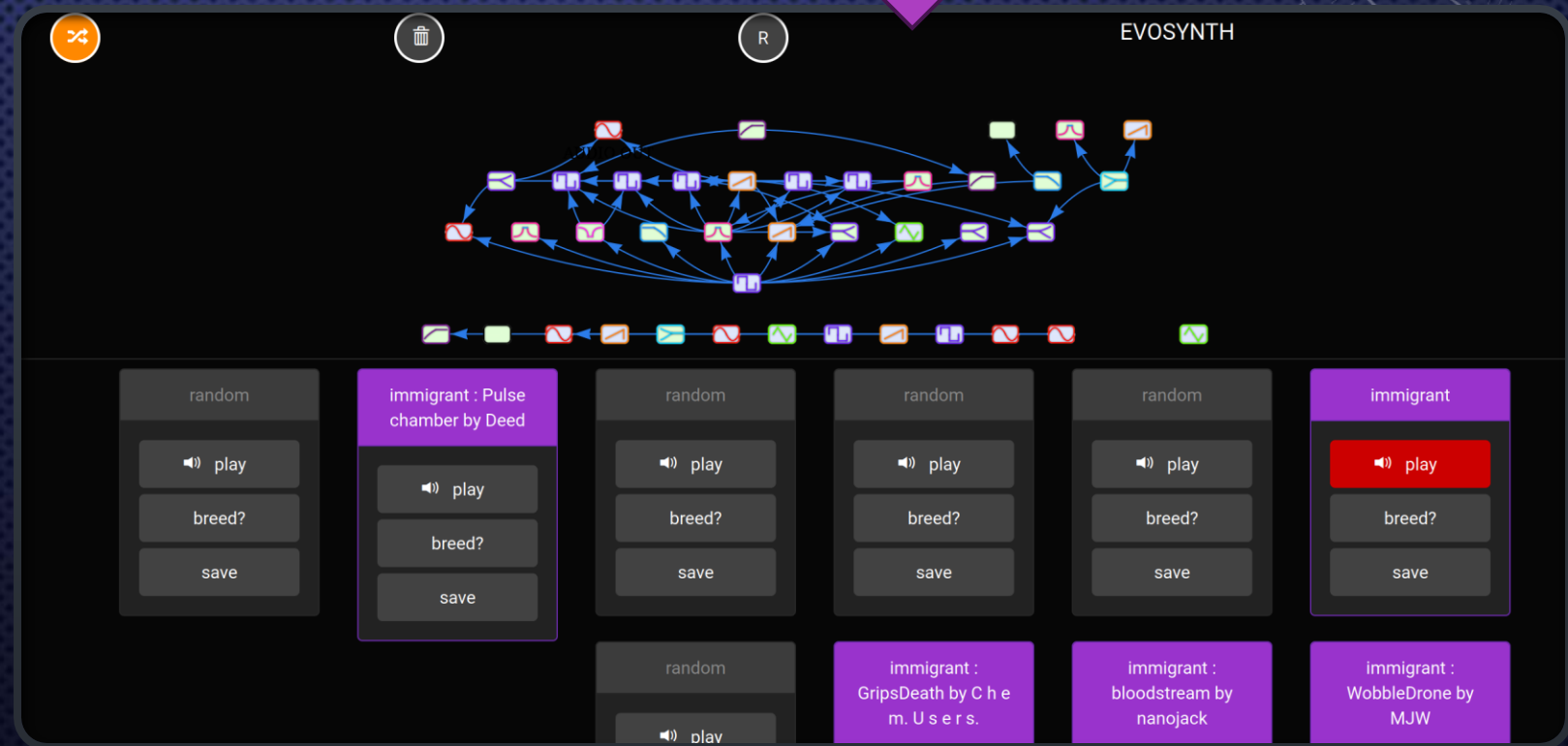
ALSO: FIEBRINK'S WEKINATOR



Inspired by working with generator (now reaktor)



- Genetic algorithm powered 'modular synth'
- **2000** with Java and JSyn
- Java applet + native audio library == complex install
- **2015** Web Audio API rebuild
- Musicians can use it!



EVOSYNTH

random

immigrant : Pulse chamber by Deed

random

random

random

immigrant

play

breed?

save

play

breed?

save

play

breed?

save

play

breed?

save

play

breed?

save

play

breed?

save

random

immigrant : GripsDeath by C h e m. U s e r s.

immigrant : bloodstream by nanojack

immigrant : WobbleDrone by MJW

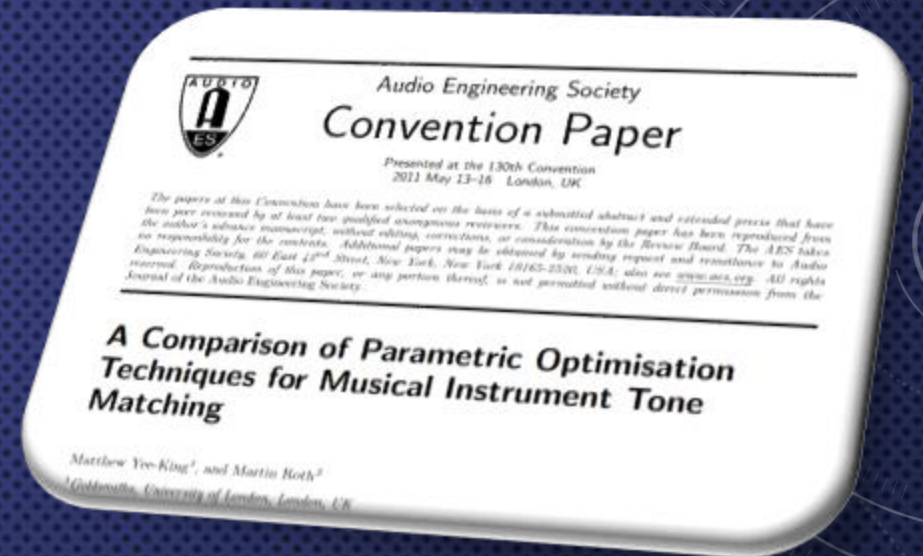
play

# META CONTROLLER INSPIRATION 2 SYNTHBOT SYNTH PROGRAMMER

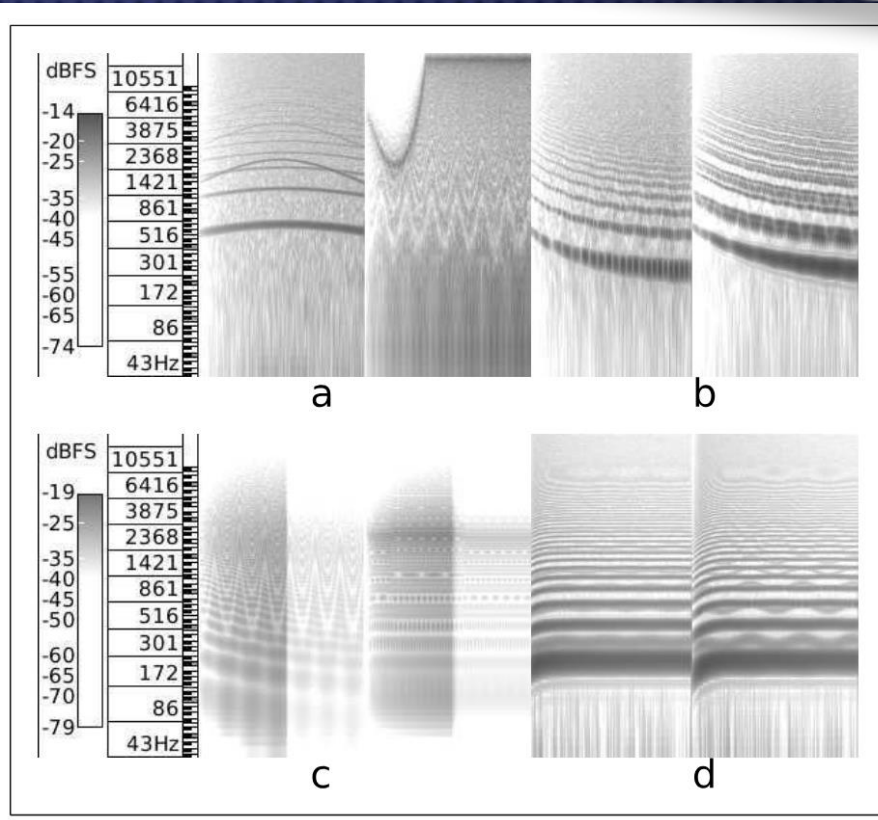
- 2007: Java VST host + Java algorithm
- 2017: Python/ C++ vst host + Python algorithm (JUICE!)
- Research software – not for musicians
- BUT VST...

<https://github.com/fedden/RenderMan>

<https://github.com/yeeking/deepvstprogrammer>



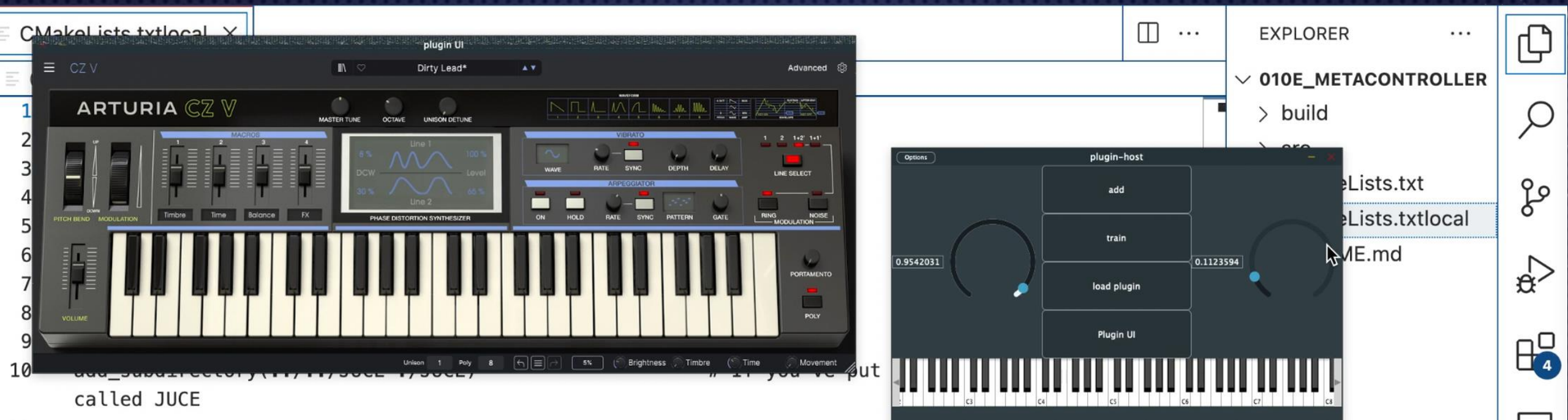
Worst  
match



Best  
match

# META CONTROLLER DEMO

- Plugin that hosts and controls other plugins
- Based on my synthesis parameter work + Fiebrink's 'Wekinator'
- Complete libtorch training and inference workflow in C++
- Fun example (does tend to destroy plugins though :))



called JUCE

# IMPROVISER RESEARCH INSPIRATION

- Supercollider
- VST
- Spot the difference...

Supercollider + Access

Virus



Live-coded Supercollider



VST plugin

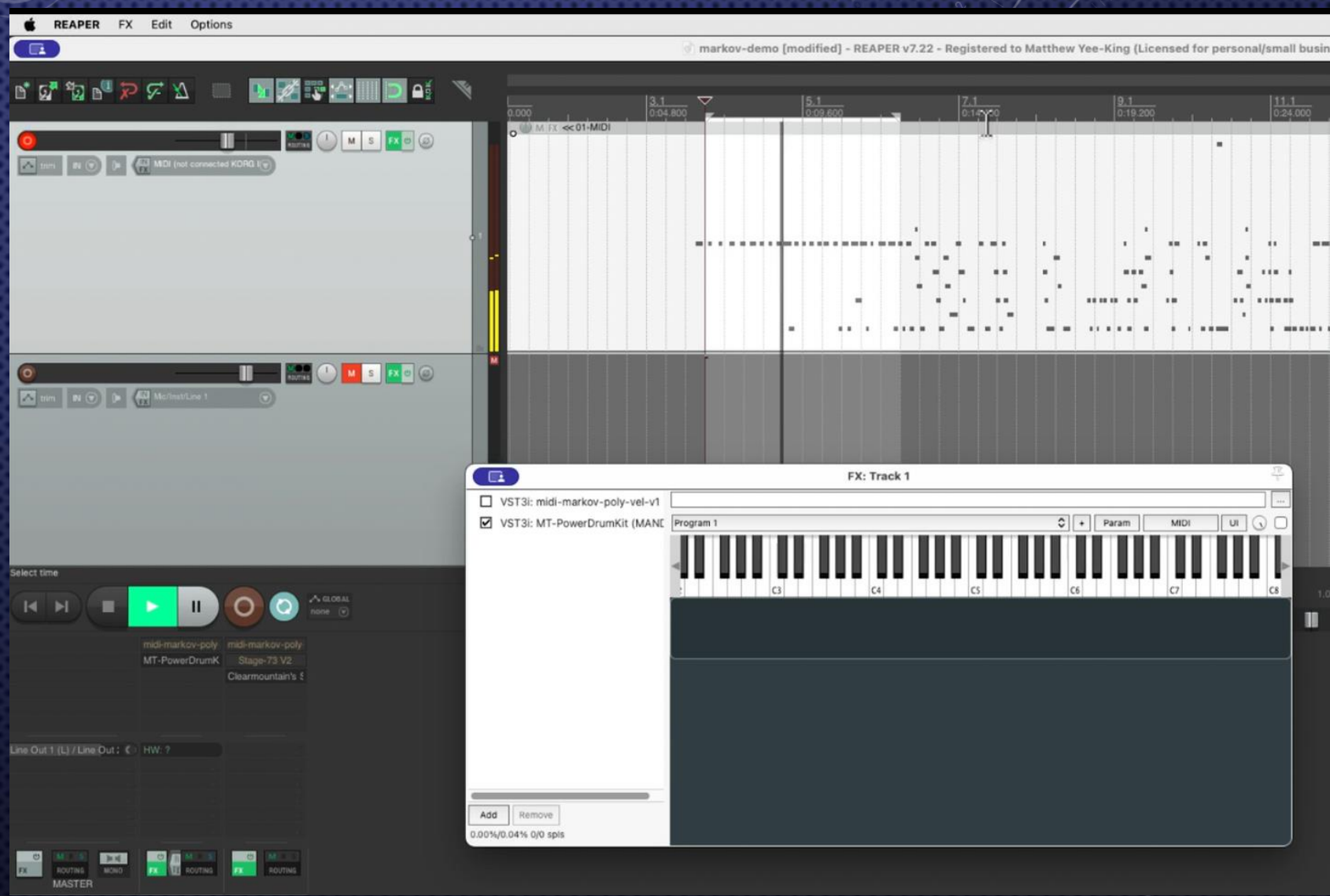


VST plugin



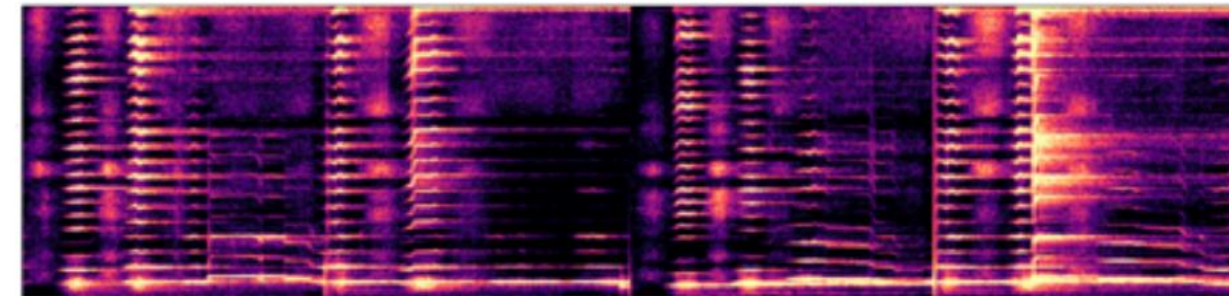
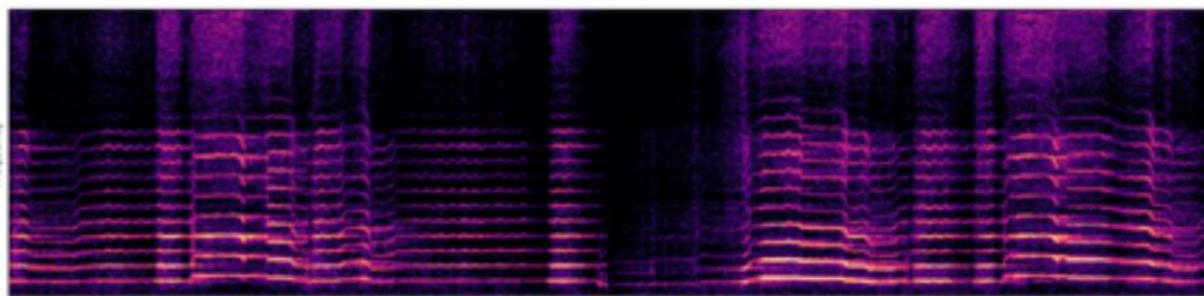
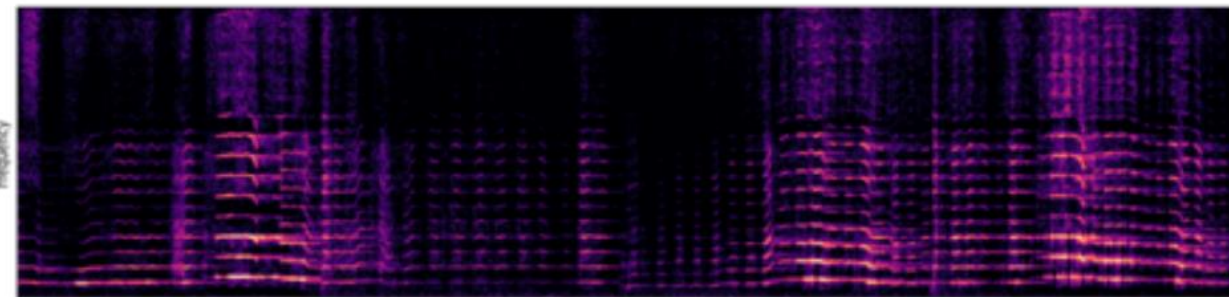
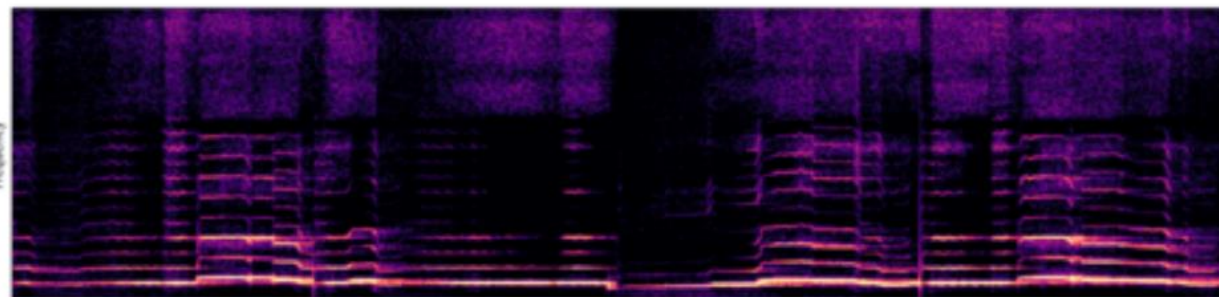
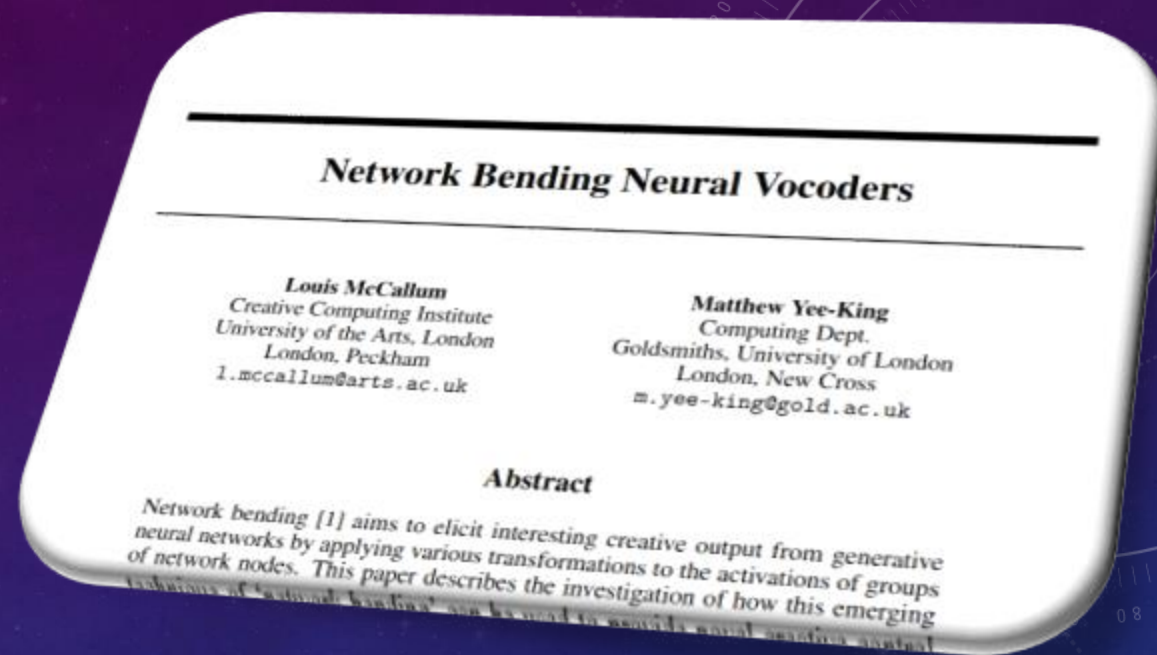
# IMPROVISER DEMO

- Plugin that learns from MIDI input
- Generates similar MIDI
- Pure C++/ JUCE
- Hierarchical Markov Model
- Agency and creative algorithm design
- Ethical datasets
- Realtime/ interactive training



# AMP EMULATOR RESEARCH INSPIRATION

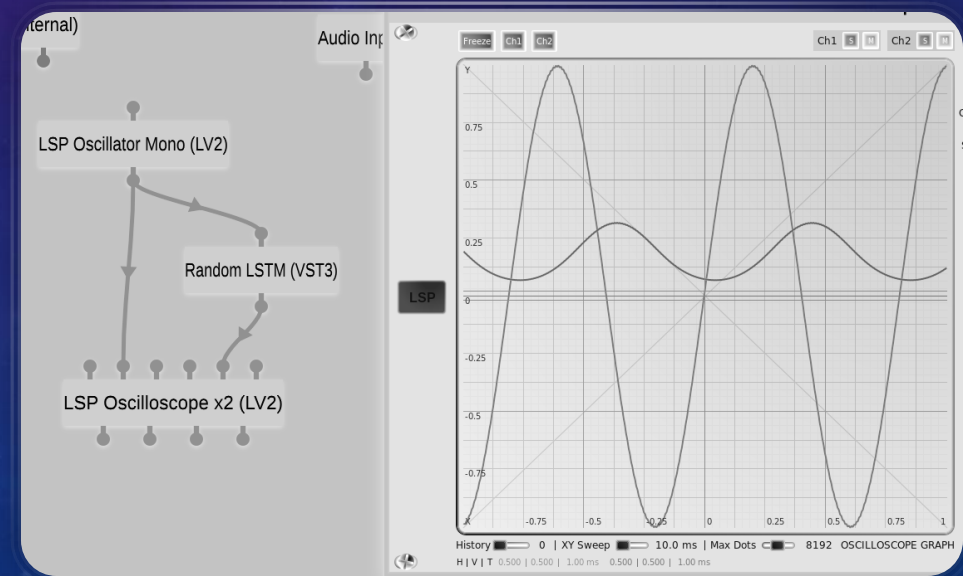
- Neural audio processing and generation
- E.g. 2020 'network bending DDSP' paper
- Then... all the neural FX work such as Steinmetz, Damskägg and Wright, GuitarML etc.





# AMP EMULATOR

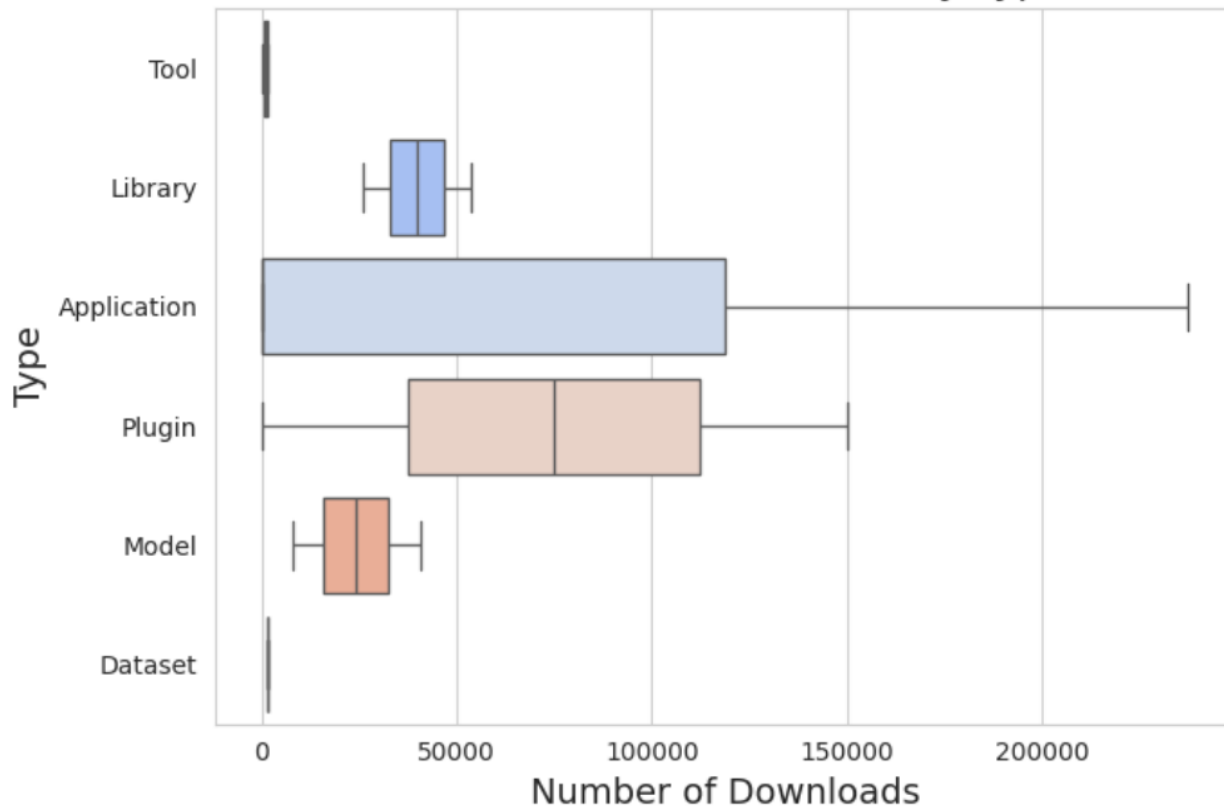
- Lots of guitar players out there – less 'experimental'
- Demonstrate python training workflow, importing trained models into C++
- Dig deeper into training and data
- Performance evaluation
- Ended up writing a mini DSP course!



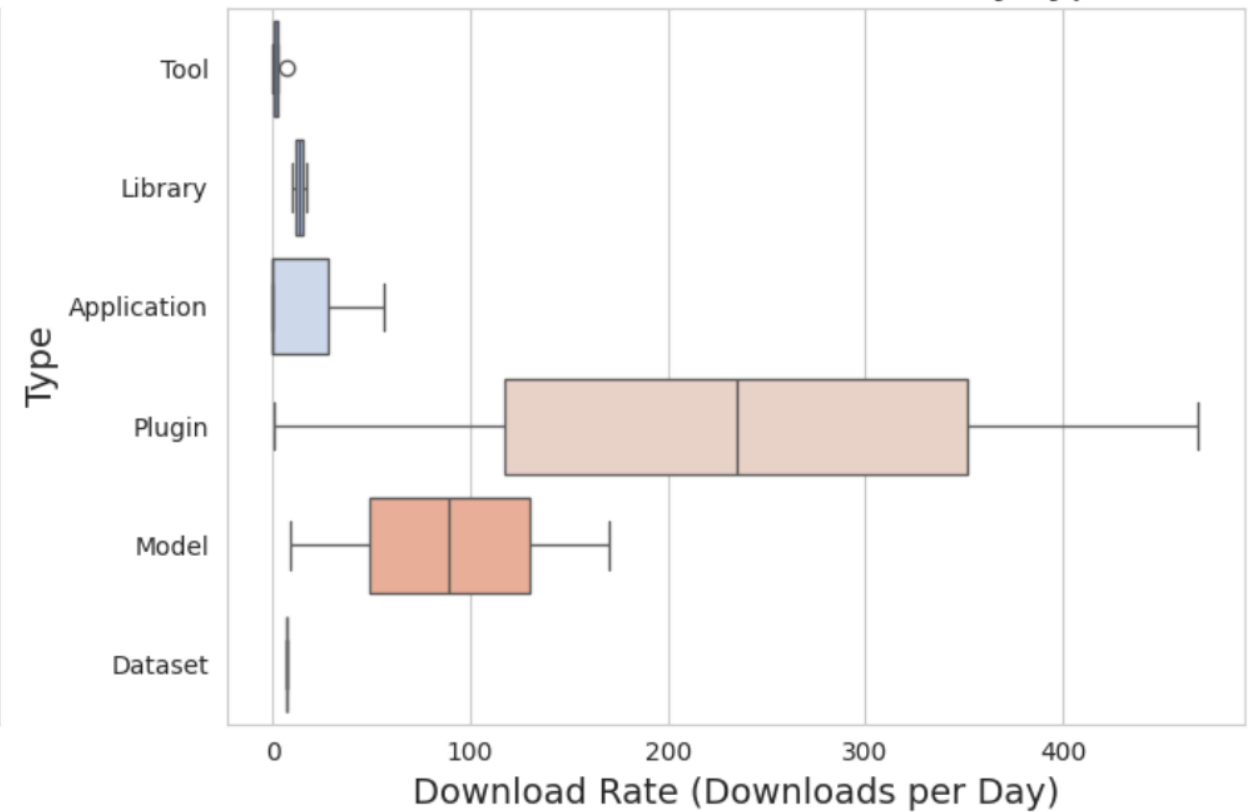
# ARE RESEARCH PLUGINS A GOOD IDEA? MAYBE

- Examining github stats for download rates by type

Distribution of Downloads by Type



Distribution of Download Rate by Type



## RESULT OF ALL THAT TEACHING...

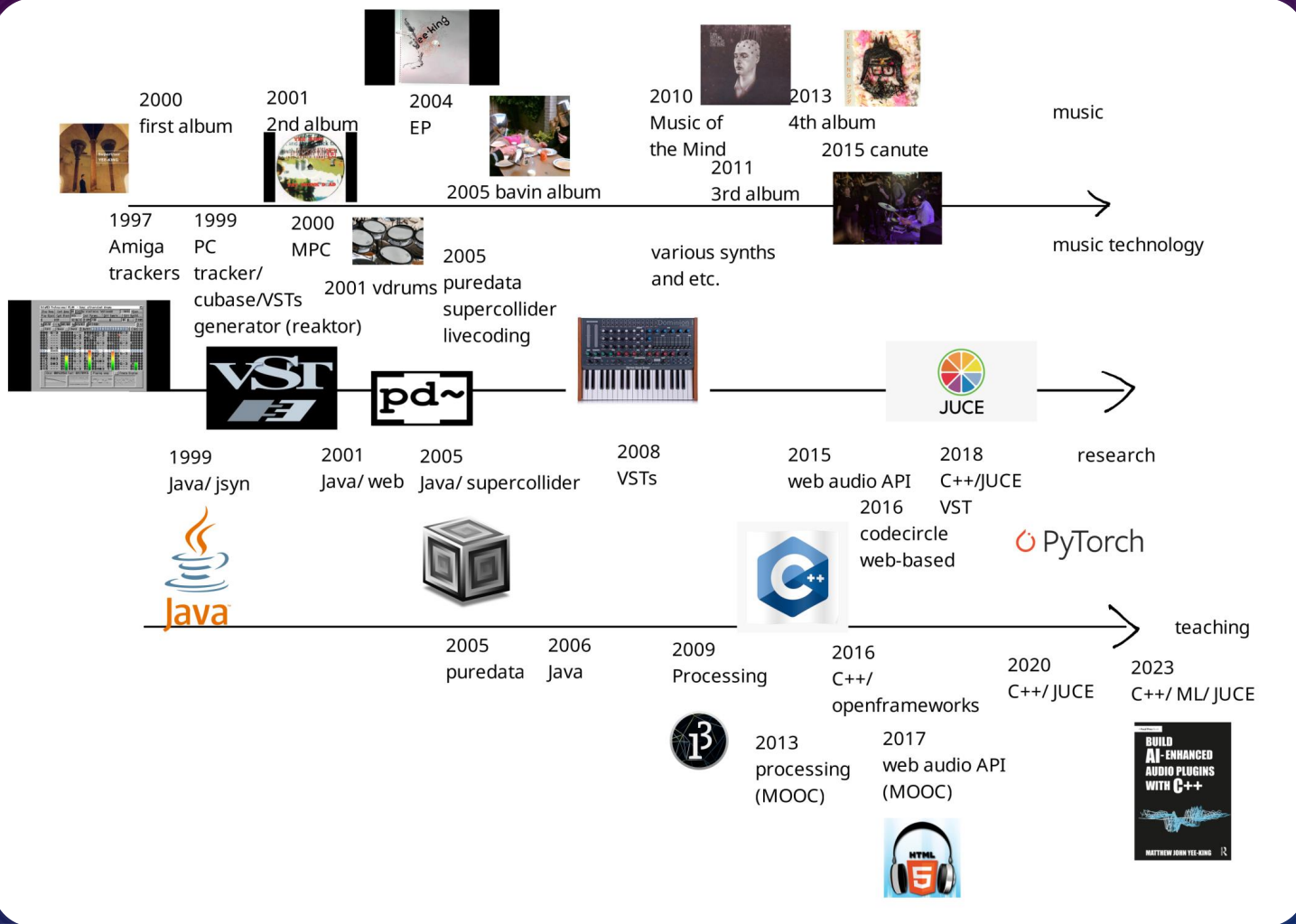
- C++
- CMake
- JUCE
- PyTorch
- 3 examples:
  - Improviser
  - Amp emulator
  - Synth controller

Yee-King, Matthew John.  
*Build AI-Enhanced Audio  
Plugins with C++*. Routledge  
2024.



# OBJECTIVES FOR THIS TALK

- Describe the process of writing an audio dev book
- Demonstrate pedagogical audio dev software and explain the thinking behind it
- Present examples of audio technology research that feed into the examples



# THANKS FOR LISTENING