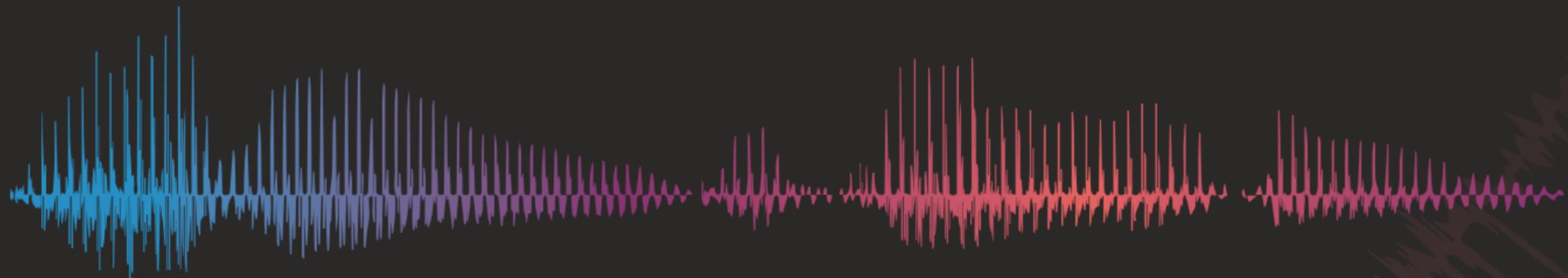


ADC<sup>21</sup>



# EXPLORING JAVASCRIPT HACKS FOR MODERN ELECTRONIC MUSIC PRODUCTION

XYZZY





# ABOUT ME

Xyzy ( ski-zee )

Stack: C + Web + Javascript + Python / Hylang 

Gear: Electribe, Guitar

DAWs: Pure Data, Reaper, Renoise

Genres: Techno, Electronic, Post Punk

# GOAL

Make music in javascript that sounds as good as the daw

Result - [Bitrhythm](#)

80%

*Literate Programming ftw!*

Docs + Code + Notes

# BITRHYTHM OVERVIEW

1. Early Attempts
2. How DAWs work
3. Mapping DAWs to Javascript

## DEMO 1 + TUTORIAL

### HACKS AND DAW PRIMITIVES

1. Blank Canvas + Code Mirror
2. Music Loop
3. Dials and Samples
4. Patterns
5. State
6. Knobs and Guards
7. Observers
8. Misc

## DEMO 2

# WHERE DOES THIS FIT IN ?

1. Defining music coding genres
2. Hard Coding
3. Conclusion
4. Version Control

# WRAPPING UP

1. Limitations
2. Future
3. Support
4. Questions

# ASSUMPTION

I will assume some familiarity with DAW workflows and working knowledge of javascript.

# BITRHYTHM OVERVIEW

# 1. EARLY ATTEMPTS

Python + portmidi

Javascript + osc + renoise

Algorave scripts

## 2. HOW DO ABLETON/FL STUDIO/LOGIC MAKE MUSIC ?

Mimics hardware - Recording

A Track is a combination of

- Virtual Instrument / Samples and Notes
- Software Effects, Automation and Mastering

### 3. MAPPING DAW TO WEB AUDIO

Tone JS provides instruments and effects.  
Javascript, webmidi provide notes and automation.  
Blackhole and Reaper for the final steps

Everything is rendered into the audiocontext, which renders to the sound card

FILE EDIT ADD PATTERNS VIEW OPTIONS TOOLS HELP

check this out.flp Close

Browser - All

- Current project
- Recent files
- Plugin database
- Plugin presets
- Channel presets
- Mixer presets
- Scores
- Backup
- Clipboard files
- Demo projects
- Envelopes
- IL shared data
- Impulses
- Misc
- Music
- My projects
- Packs
- Project bones
- Recorded
- Rendered
- Sliced audio
- Soundfonts
- Speech
- Splice
- splice\_folder
- Templates

Channel rack

uk\_snare\_b

- 10 CHAMPAGNE\_DRIP\_snare\_one\_shot\_live\_01
- 11 dp\_snr\_pleasedome
- 12 Hyper DoubleBarrel Clap
- 13 Hyper HouseParty Clap 02
- 14 HouseGen Clap 02
- 15
- 15

VMS\_Kit1\_Vocals\_142\_C#\_minor\_10 (Master)

Envelope

Filter

Root note: C5

Knob

0.2

Attack time

Mixer - Master

- (none)
- Maximus
- Fruity parametric EQ 2
- Fruity parametric EQ 2
- Fruity Limiter
- Fruity Limiter
- Edison
- Slot 7
- Slot 8
- Slot 9
- Slot 10

Clap 2 Clap 3 Karra Karra Dry Karra Vocoders Karra Buss Just Drums Met Drums Dry Drums NY Drums Spire Bass Another Tyme Ensemble Trumpet Brassius Maximus Dsc EX Rise

Equalizer

(none)

Out 1 - Out 2

Demo 1

Audio Issues ?

*Note all the things you see moving*

Tutorial  
Video  
Code

# HACKS AND DAW PRIMITIVES

# 1. CANVAS + CODE MIRROR

Vj with p5 or webgl

## 2. MUSIC LOOP

```
var delta = 60 * 1000 / tempo / lines * 4;
setInterval(function () {
  always();
  text = getText();
  for (pattern, track_no in patterns) {
    isHit = parse(pattern, track_no);
    editMode ? eval(oldText) : eval(text);
    if (runTransition) { // once
      transition();
    }
  }
}, delta)
```

## 3.DIALS

available as a global array `dials` which can be subscribed to

Can daws provide a simpler menu system which can be configured per project ?

## 4.SAMPLES AS URLS

Sample Browser

# 5. PATTERNS / DANCE TABS

DSL

Absolute Pitch, no problem!

We still use Roman Numbers!

Roma/Gypsy Decimals > Hexadecimals

```
x[2;^C2;+0.01;_0.1] 0 0 . *4 x[0.1] 0 x[1] 0 0 0 x[1] 0
- vol, ^pitch, +delay, _pan

`0 ~ a ~ 10 ~ 1100
`1 ~ b ~ 11 ~ 1101
...
```

## 6. STATE

Use `mem` and simple logic to compose notes relative to other notes

```
if (mem["kick_hit_thrice"]) {  
    pn("crash")  
}
```

# 7. AUTOMATION AS ARRAY ROTATION AND GUARDS

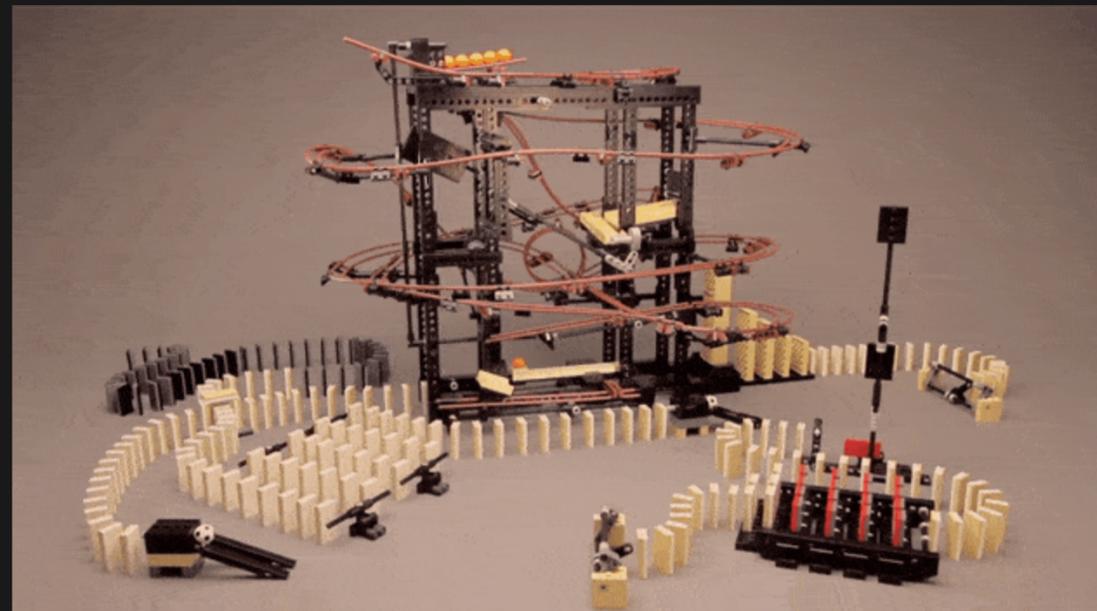
MIDI, mouse, XY  
DSL like turtle graphics

Endless Acid Banger convinced me to model knobs directly

```
k1 = knob({
  ramp : [0.525, 0.8, 0.4, 1,
          0.25, 0.75, 1, 0.25, 0.1
        ],
  direction: false,
  step: 0.01,
  "number": dials[2]["cell"]
});
g = guard([-20,15])
always = function () { // run tweak transition
  stab_filter.frequency.value = g(k1.move() * 10);
}
```

## 8. OBSERVERS + TIMERS

- bang and spigot from pure data
  - sidechain anything
  - constraint propagation
- arpeggios/chords/progressions/counterpoint are just things tied together



# 9. MISC HACKS

Alerts

Pads + Keyboard Mapping

**BONUS**

A basic 303!

Demo 2  
Video  
Code

**WHERE DOES BITRHYTHM FIT IN THE CONTEXT OF EXISTING  
TOOLS ?**

**WHAT KIND OF MUSIC CAN YOU DO WITH THIS ?**

# MUSIC CODING GENRES

## INTERACTIVE CODING ✓

Share track as a url

## ALGORITHMIC CODING ~

Music as algorithms with experimental patterns

IDM / Ambient / Generative / ML

## LIVE CODING ~

Coding itself as performance

**ADAPTIVE CODING ☒**

Games

**CREATIVE CODING ✓**

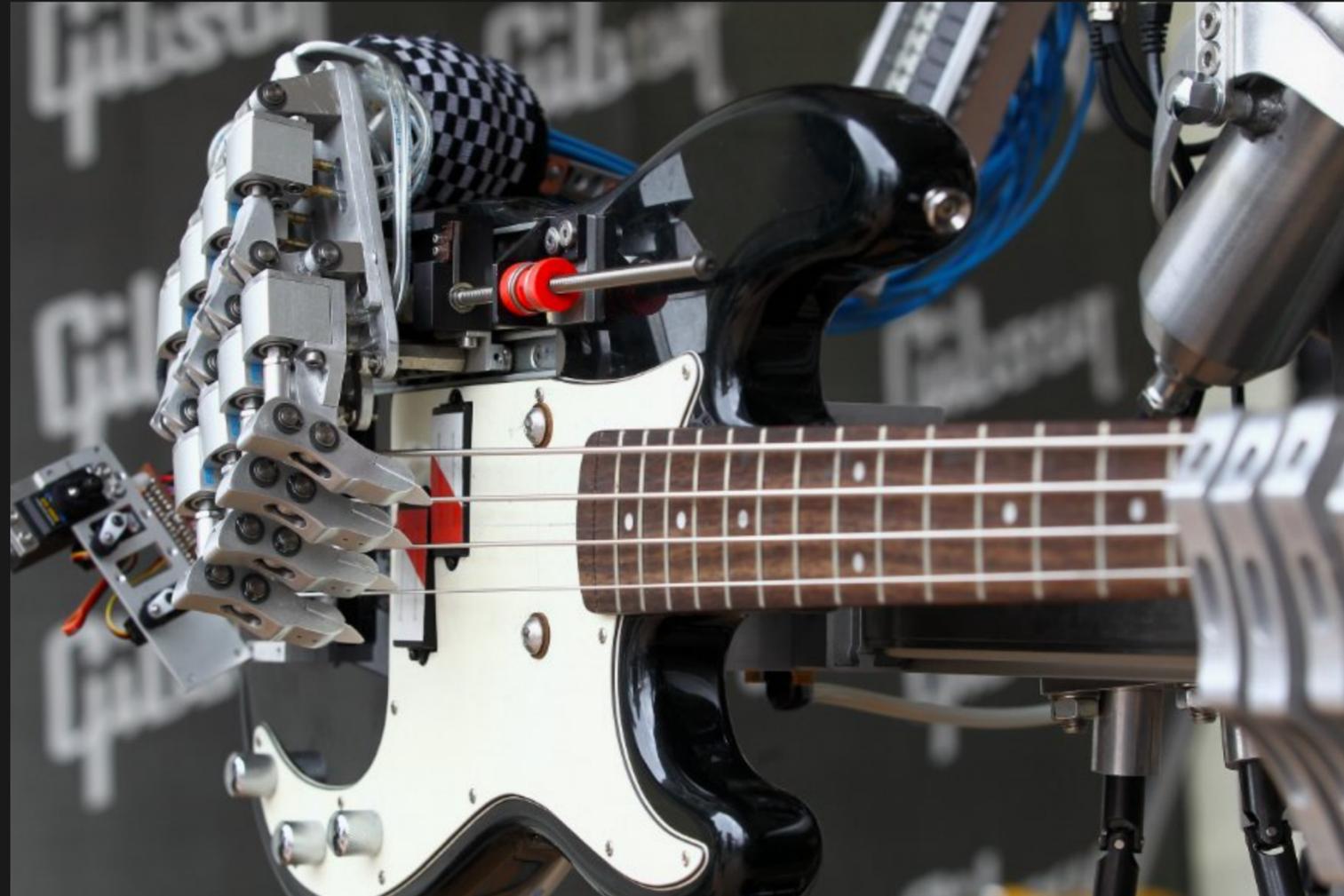
Demoscene

# PHYSICAL CODING ☒

websocket vs osc



# MACHINE CODING ☒



## **HARD CODING ✓**

Music as data with tracker notation or static constraints

No randomness or ML

Chiptune / Tracking / Executable Music

# CONCLUSION

Code is the new music sheet!

*Can a text-driven interface replace and improve over DAW workflows, without sacrificing quality ?*

Yes, but sound design and mastering is better done in UI.

- Copy Paste
- Chunking
- Prototyping

# VERSION CONTROL

see [diff](#)

# WRAPPING UP

# LIMITATIONS

Reverb

Gc + GUI

Timer

Big latency while screen recording

# FUTURE

*90 minute set*

Recording audio, performance and automation  
websocket based collab

# ALTERNATE IMPLEMENTATIONS

C / raylib to avoid latency

Juce based VST ?

Elementary

Csound Web

Gibber

## OTHER PROJECTS AND LINKS!

[Build your own Data Flow Engine](#)

(inspired by Pure Data)

[Twixter](#)

[More Apps + Code + My Music + Social](#)

[Post Issues on Github](#)

Support me on [Gumroad](#)

**HARD CODING - A NEWSLETTER ON MUSIC + CODE**

[Substack](#)

QUESTIONS ?

Credits