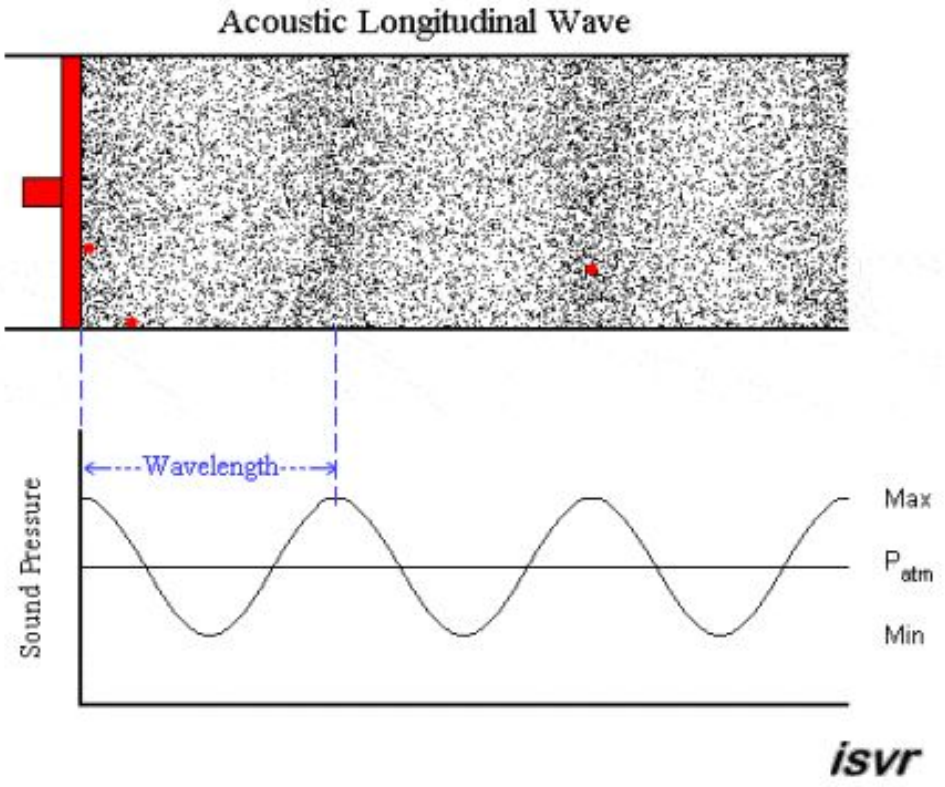
Live Coding and how we got here

Mark Santolucito

Before we start:

Please download SonicPi: https://sonic-pi.net/

How does a .wav file become sound?



Getting Into the Machine

ACOUSTIC SOUND

→ Transducer (microphone)

→ Preamplifier (adds gain to increase signal-to-noise ratio)

→ Lowpass anti-aliasing filter (to preclude digital artifacts)

→ ADC (sample clock)

→ **DIGITAL SOUND** (sound as discrete amplitudes)

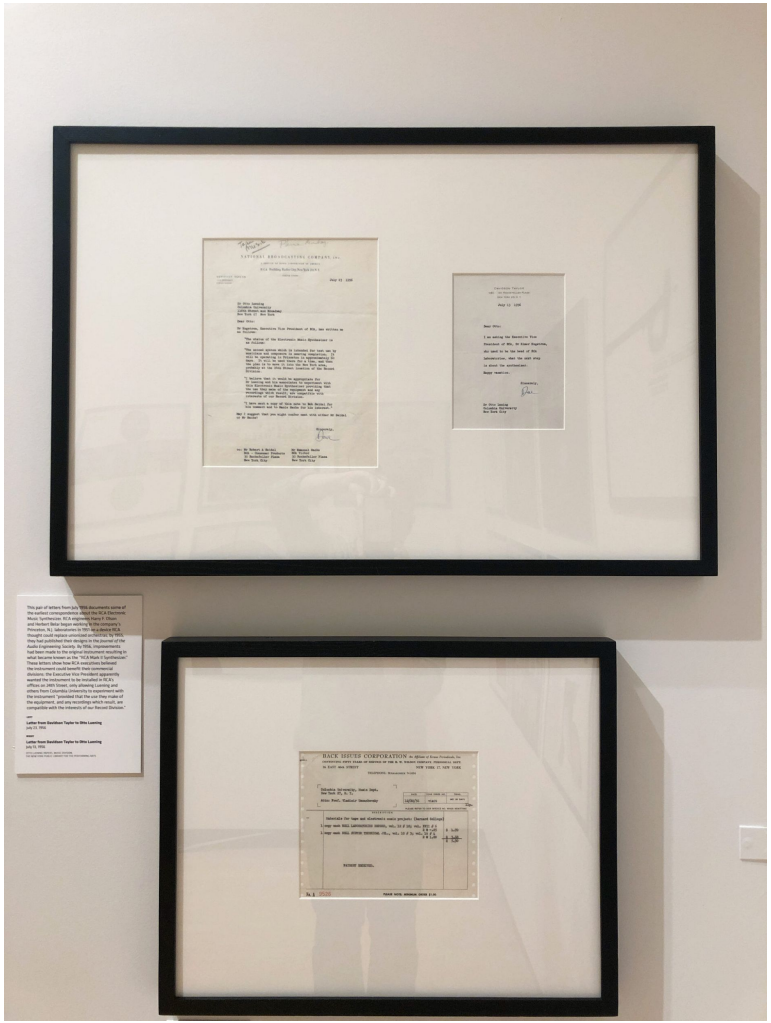
- Once digital, filtering, processing and synthesizing can be (simple) mathematical operations

Digital Audio:
A sequence of discrete amplitude levels over a quantized/discrete amount of time.

a overview of computer music from
1956 to the present
with a special focus on the
Barnard/Columbia/Princeton/Bell Labs collective


Played by
I.B.M. 7090
Computer and
Digital to Sound
Transducer

https://openai.com/blog/jukebox/



Sounding Circuits: Audible Histories

https://www.youtube.com/watch?v=8WI9mQiQuSg

Lejaren Hiller (1924-94)

U. of Illinois, Urbana–Champaign

Chemist → Composer

Taught or worked with lots of composers and others: Cage, Tenney, Moog, etc

First music composed with a computer: Illiac Suite (1957)

Wrote Experimental Music: composition with an electronic computer (1959)

Hard coded rules + Monte Carlo

Max Mathews (1926-2011)

Worked @ Bell Labs, IRCAM, CCRMA

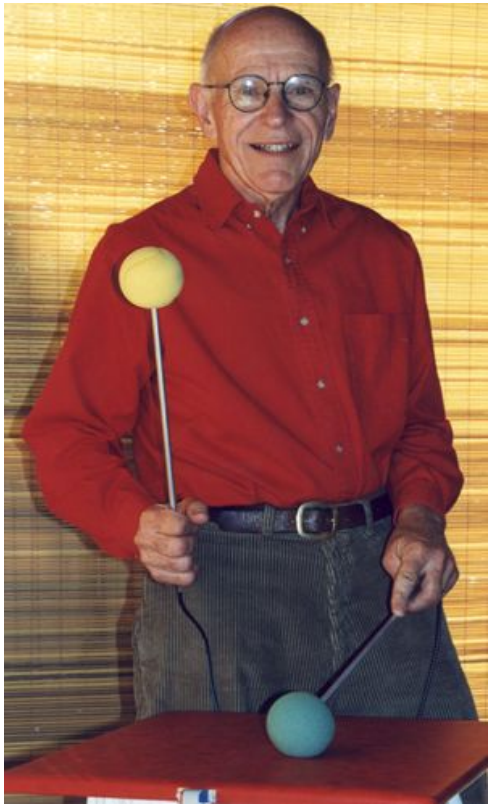
MUSIC N Languages

GROOVE

IBM's Bicycle built for two

Very interested in computer music in realtime

"What now is the musical challenge of the future? I believe it is the limits in our understanding of the human brain; and specifically knowing what sound waves, sound patterns, timbres and sequences that humans recognize as beautiful and meaningful music – and why!" -Mathews, Introduction to Boulange's Audio Programming Book



Mathews playing his Radio Baton

Music N

Music I, 1957

Music IV 1963

Music V 1966 (Written in Fortran, code is still out there I think - Xtra™ credit to anyone who gets it running)

Early (first?) example of open source

Descendants include CSound, MaxMSP, SuperCollider

Vladimir Ussachevsky (Columbia Prof)

With Otto Luening (Barnard faculty), founded the Columbia-Princeton Electronic Music Center at Columbia in 1959

Specified ADSR envelope in 1965

Wendy Carlos

(M.A., 1965 from Columbia)

Student of Ussachevsky

Convinced Robert Moog to add touch sensitivity to keyboard

Switched on Bach, 1968, won 3 grammys

A Clockwork Orange (1971)
The Shining (1980)
Tron (1982)

Laurie Anderson

B.A. Barnard, 1969
(Started at Mills College and probably met Pauline Oliveros)
M.F.A. in Architecture 1972 from Columbia

“Voice of authority”

first artist-in-residence at NASA

O Superman, 1981

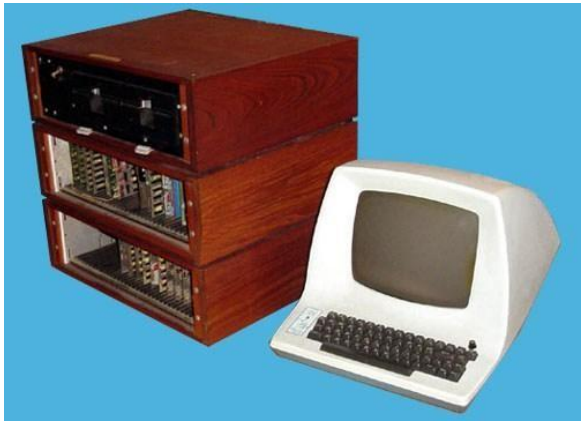
1977- : Lack of good, affordable support for musical purposes leads to:

- purpose-built, proprietary computer instruments

- add-on systems to the emerging PC systems starting with the Apple][

Synclavier I & II (1977 - 1990s)

From Hybrid systems to Mixed Digital Engineering → purpose-built digital hardware for DSP that controlled analog gear. Laid the foundation for the digital synth revolution to come.



Synclavier and CMI established creds of digital synthesis and DSP devices. Also helped establish proprietary ROM software / hardware for DSP and other music-specific hardware as the norm. (As opposed to analog gear which is essentially transparent.)

Fairlight

Competition to the Synclavier, the Fairlight CMI: Computer Music Instrument (1979) -- originally supposed to be analog modeling, it became a sort of resynthesized sampler. Also later included a Realtime Composer Interface for pattern-based event sequencer -- anticipating MIDI sequencers of the later 80s and 90s.

These systems didn't really last (cost prohibitive) but have some descendants including outboard DSP processors in studios.

Towards Democracy: The personal Computer

Apple][released late 1977

Key feature was expandability with at least 7 expansion cards: multiple companies built audio cards for the system.

The MIDI Revolution: 1983-?

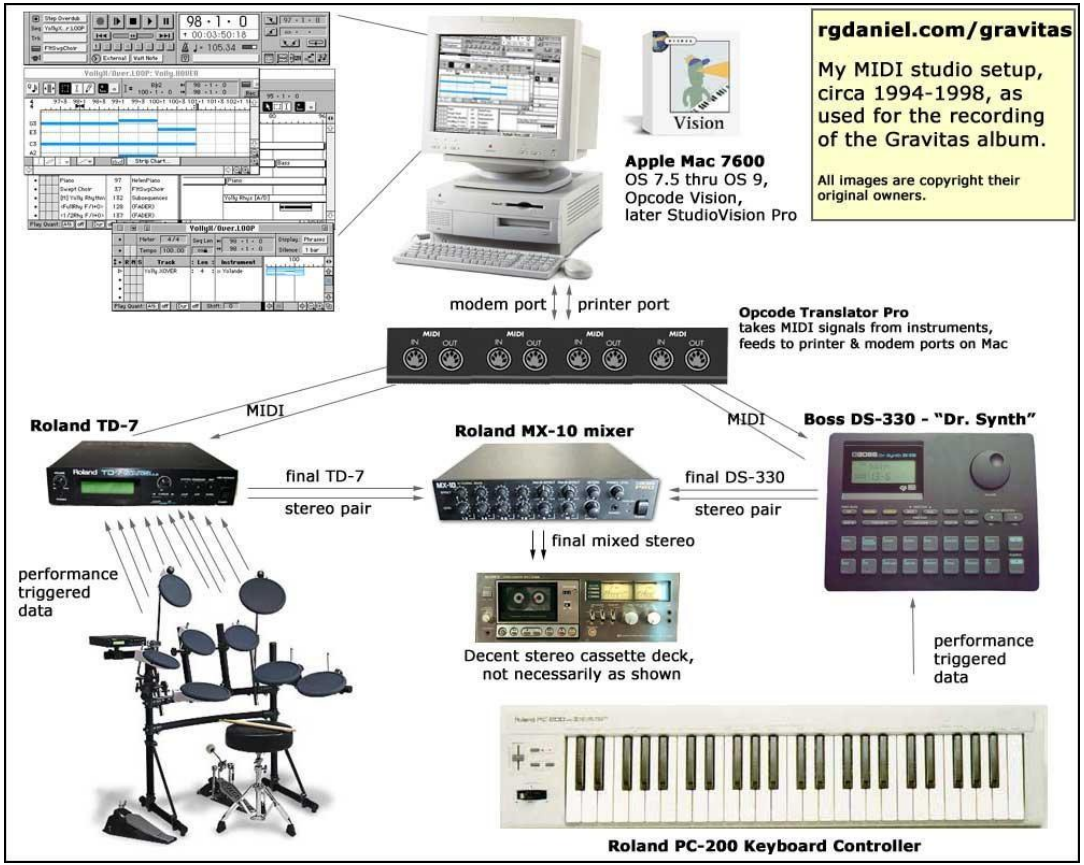
MIDI (1983)

Revolutionary

Standard inter-application, inter-instrument communication protocol

Integer based (0-127)

Data includes, note on/off, volume, control changes, after-touch, program changes, more



The 1990s

Computers get Smaller, faster, and cheaper (According to Moore's Law)
Audio I/O moves from cards to external boxes (for laptops)

MIDI sequencers blend with audio sequencers to become DAWs

Digital recording (DAWs) common in studios by the late 1990s

OSC Communication protocol early 2000s

MIDI communication with computers changes to USB on the computer side

Currently...

1950s:

Prohibitively expensive, large, clunky, required system expertise

Systems only available in large, academic or corporate settings

Virtually no software (you wrote your own) and available only to a select few

Study focused on mathematical foundations and early synthesis techniques

201X:

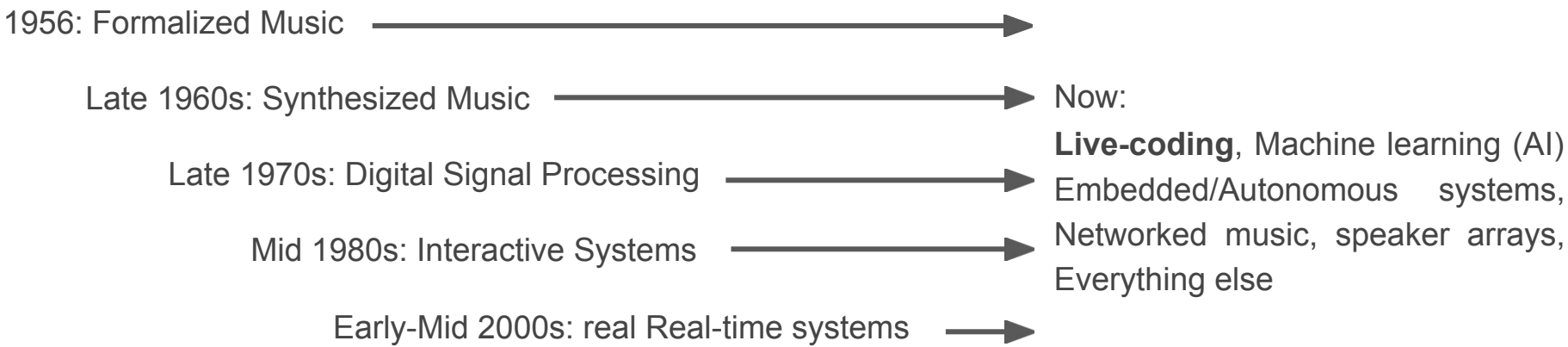
Extremely portable (e.g phones) and cheap (e.g. microcontrollers)

Systems widely available

More than enough open-source software

Broad range of study and applications: stick around for the rest of the class to find out

A rough timeline of practices



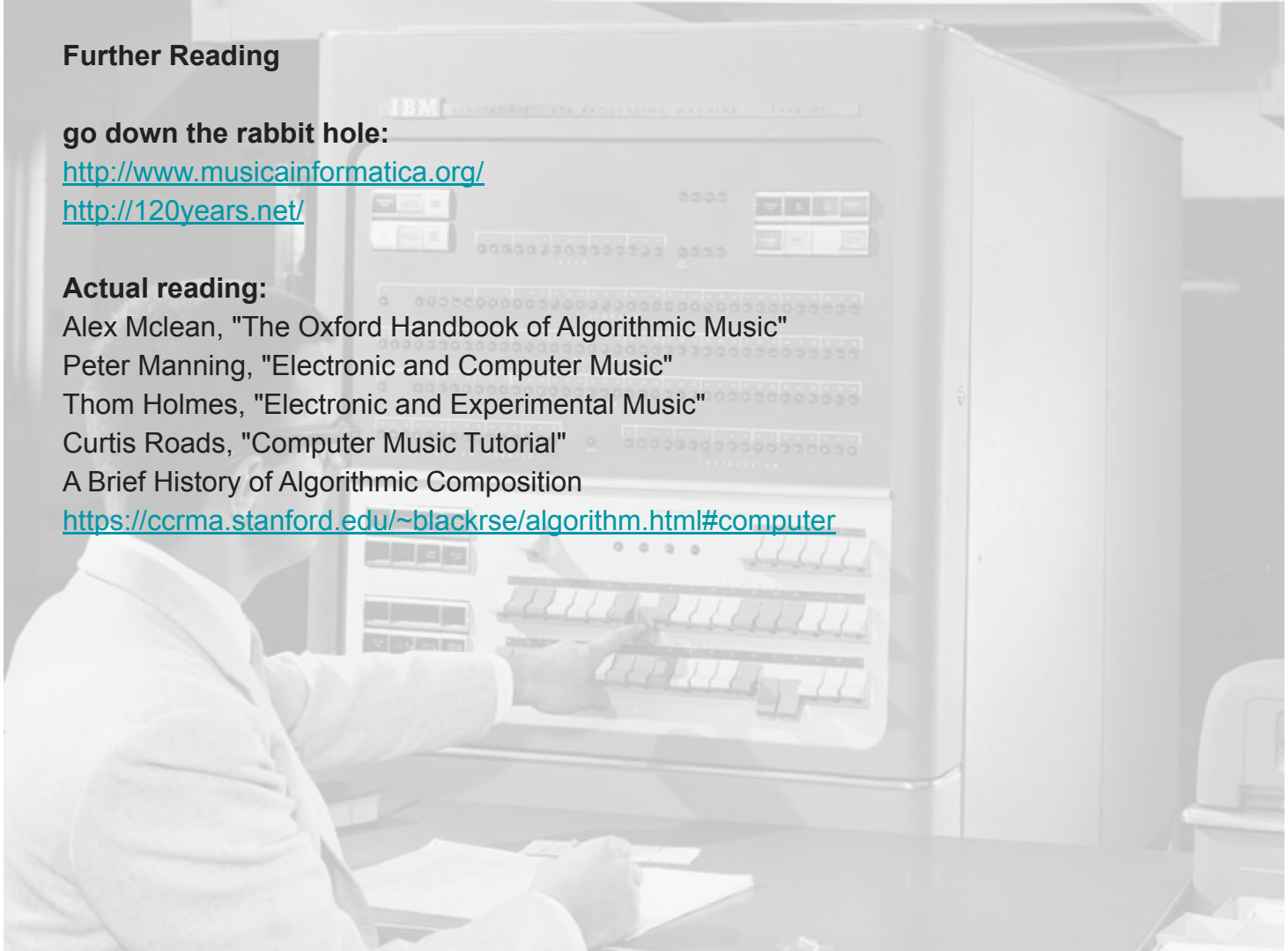
Q. Why these dates?

A. Combination of 1) Aesthetics and 2) Technical Limitations

Why should I care about any of this?

Incredible number of areas of study (and jobs…) at the intersection of technology (CS) and music/audio:

web based audio applications
recommendation systems
copyright violation detection
video call noise removal
spatialization
embedded systems (low-level interfaces, driver design),
audio hardware (DSP, I/O, etc.)
Music performance interfaces (still big)
Music production (editing, mixing, mastering, delivery)
Music therapy



Further Reading

go down the rabbit hole:
http://www.musicainformatica.org/
http://120years.net/

Actual reading:
Alex Mclean, "The Oxford Handbook of Algorithmic Music"
Peter Manning, "Electronic and Computer Music"
Thom Holmes, "Electronic and Experimental Music"
Curtis Roads, "Computer Music Tutorial"
A Brief History of Algorithmic Composition
https://ccrma.stanford.edu/~blackrse/algorithm.html#computer

Ethos of Live coding

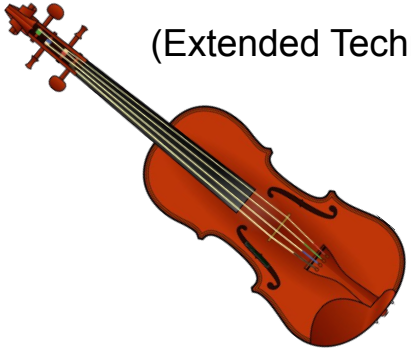
The act of programming becomes a critical part of the end product.

“Art of re-programming; changing your mind about a process once established”

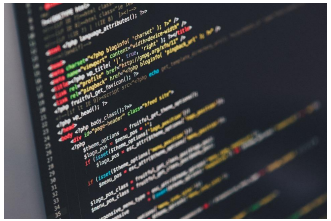
- Nick Collins

“Live coding is about embracing failure”

Traditional Instruments as Code



(Extended Technique)

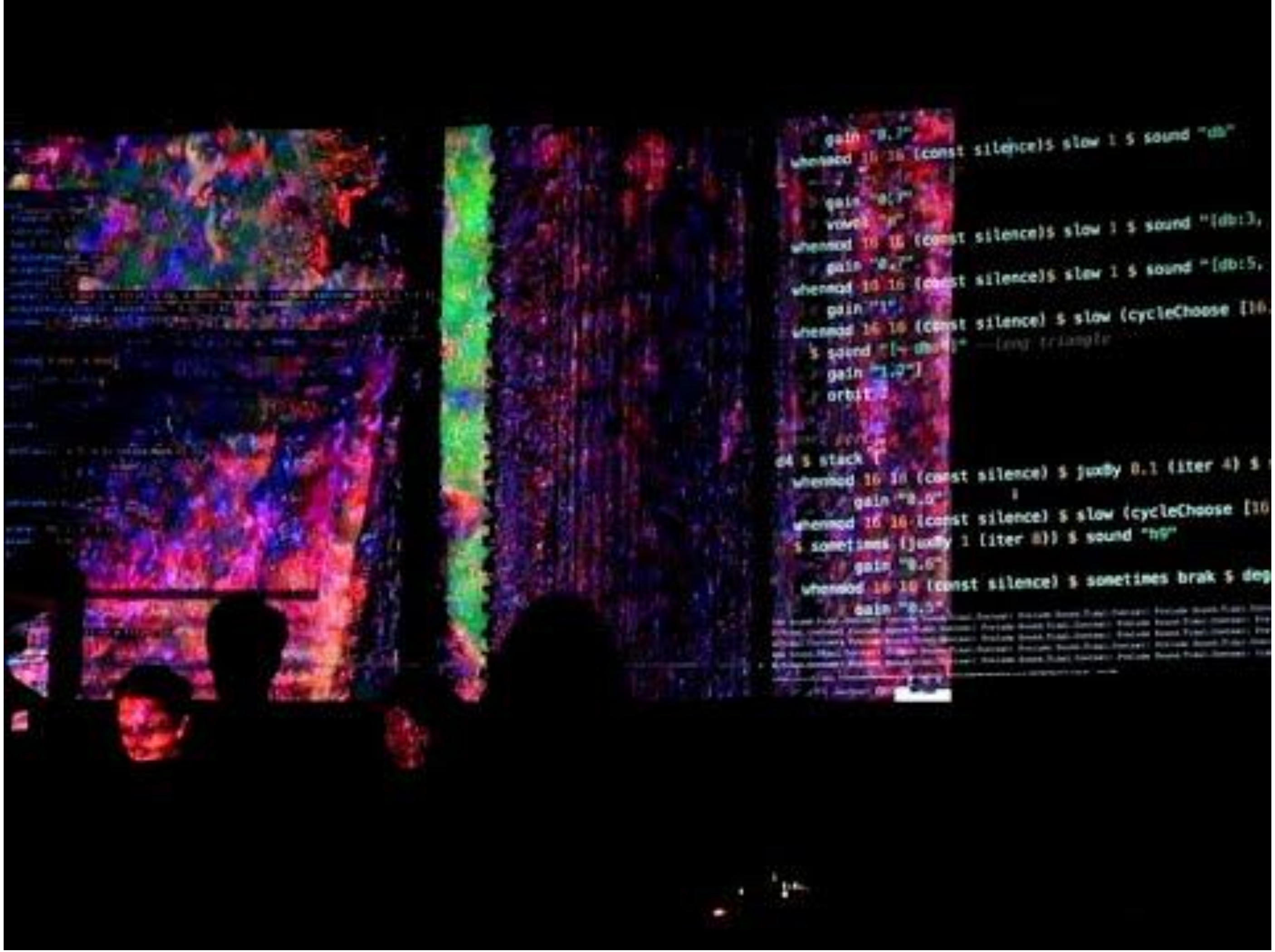


Fewer possible sounds

More possible sounds

Other parameters include:
reconfigurability, natural physical interface, what is “hello world”, ...


LIVE-CODING MUSIC
#TDCMCR


whenmod 16 16 (const silence)$ slow 1 $ sound "db"
whenmod 16 16 (const silence)$ slow 1 $ sound "[db:3,
whenmod 16 16 (const silence)$ slow 1 $ sound "[db:5,
whenmod 16 16 (const silence) $ slow (cycleChoose [16,
d4 $ stack [
whenmod 16 16 (const silence) $ juxBy 0.1 (iter 4) $
whenmod 16 16 (const silence) $ slow (cycleChoose [16
$ sometimes (juxBy 1 (iter 8)) $ sound "h9"
whenmod 16 16 (const silence) $ sometimes brak $ deg

Demo time

SonicPi