# Command Line Algorithmic Music System (CLAMS)

M. Edward (Ed) Borasky

2023-03-25

# Me and Forth

## Me - retired scientific applications programmer

- I got paid to write
    - Assembly
    - Fortran
    - Awk/sed/grep/*nix shell
    - Perl
    - R

## I learned for hobby projects

- Lisp
- Forth
- Ruby

## I never learned

- COBOL
- APL
- SNOBOL4
- Smalltalk
- Algol/Pascal/C/C++/Java/C#/Objective C/D/Rust/Go
- PHP
- Python
- JavaScript

# Forth

- 1980s
  - learned Forth via HESForth on Commodore 64
- mid-late 1990s
  - primary Forth engine was HP100LX Palmtop PC
    - 16-bit 80186
  - wrote some articles for FORTH Dimensions
  - wrote some trading system software
  - used mostly hforth and Tom Almy's Forth compiler

# Why I stopped writing Forth

- by 1999 I had faster machines that ran my hobby code in Perl
- I wasn't using Forth at work
- I was learning Linux and R at work
- Too much mental context switching kills productivity!

(Two sunspot cycles pass . . . )

# He's baaack! And he brought . . . CLAMS!

# CLAMS goal

- algorithmic music composition and performance . . .
- live . . .
- in real time . . .
- on a Raspberry Pi Pico!
  - eventually other boards that use the RP2040 microcontroller
  - and maybe other Arm Cortex boards

# Software inspirations - HMSL

- Hierarchical Music Specification Language (HMSL)
- long history and experience with experimental composers
- currently being actively enhanced!
  - https://github.com/philburk/hmsl.git
- based on `pforth`
  - https://github.com/philburk/pforth.git

# Software inspirations - FORMULA

- FORMULA - FORth MUsic LAnguage
- designed for improvisation in real time
- contained a real-time operating system
- ran on inexpensive hardware
- David P. Anderson and Kuivila (1989), D. P. Anderson and Kiuvila (1991)
- Python successor: Numula https://github.com/davidpanderson/Numula/wiki

## Software inspirations - Live coding

- *Live Coding: A User's Manual* (Blackwell et al. 2022)
- The TOPLAP Manifesto
  - "Show Us Your Screens"
  - https://toplap.org/wiki/ManifestoDraft
- GitHub Awesome Live Coding
  - https://github.com/toplap/awesome-livecoding

# CLAMS architecture

- a domain-specific language implemented in Forth
- conceptually, ChucK (https://chuck.cs.princeton.edu/), (Salazar et al. 2014) semantics with Forth syntax
  - low-level words: digital synthesis and microcontroller audio
  - mid-level words: construct signal flow graph
  - high-level words: algorithmic composition and live performance

# Forth base: zeptoforth

# Highly optimized!

- subroutine-threaded
- allows inline expansion of words
- many primitives in assembly

## Close to the metal

- words for nearly all the RP2040 hardware
- has an RP2040 assembler
- can compile to RAM or flash
- real-time operating system capabilites

## Current limitations

- no USB serial support - requires UART connection
- does not support wireless yet

## zeptoforth on GitHub

- repository: https://github.com/tabemann/zeptoforth
- wiki: https://github.com/tabemann/zeptoforth/wiki

# Road map

# About the RP2040 processor

- Dual-core Arm Cortex-M0+ processor, flexible clock running up to 133 MHz
- 264kB on-chip SRAM
- 2 × UART, 2 × SPI controllers, 2 × I2C controllers, 16 × PWM channels
- 1 × USB 1.1 controller and PHY, with host and device support
- 8 × Programmable I/O (PIO) state machines for custom peripheral support

# About the Raspberry Pi Pico board

- 2 MB flash
- four versions
  - Raspberry Pi Pico: $4US, surface mount
  - Raspberry Pi Pico H: $5US, male headers and a debug connector
  - Raspberry Pi Pico W: $6US, surface mount, 2.4 GHz wireless
  - Raspberry Pi Pico WH: $9US, male headers, 2.4 GHz wireless
- the wireless ones tend to be hard to get . . . usually bundled in starter kits

# Target audio hardware

- Pimoroni Pico Audio Pack
  - ~ $16US plus shipping
  - requires an expander board for UART connection (~ $9US)
- Waveshare Pico Audio Expansion Module
  - ~ $20US plus shipping
  - includes speakers
- Waveshare Overall Evaluation Board
  - ~ $50US plus shipping
  - includes 3.5 inch resistive touch screen
  - includes serial-to-USB adapter
  - includes microSD card adapter

# Proof of concept (v0.2.5)

- direct digital synthesis / sine wave of any frequency
- maybe two oscillators and frequency modulation
- maybe “triangle” / “sawtooth” / “pulse” waves
- maybe a filter and envelope generator and low-frequency oscillator
- target date: 2023-03-17

# First release (v0.5.0) - Software

- will re-scope project after proof of concept!
  - need to assess audio performance constraints
- all synthesis / audio I/O words
- goal is all synthesis algorithms in Csound 7
  - https://flossmanual.csound.com/
- target date: 2023-05-12

# Second release (v0.7.5)

- signal flow graph interpreter
- target date: 2023-06-16

# Full release (v0.9.0+)

- live performance user interface
- possibly port to other audio microcontrollers
- target date: 2023-07-14

# CLAMS on the web

- GitHub: https://github.com/AlgoCompSynth/CLAMS
- blog: https://www.algocompsynth.com/#category:CLAMS/
- this presentation: https://github.com/AlgoCompSynth/CLAMS/blob/main/presentations/CLAMS-intro.pdf

# References

Anderson, D. P., and R. Kiuvila. 1991. "Formula: a programming language for expressive computer music." *IEEE Computer* 24 (7): 12–21. https://doi.org/10.1109/2.84829.

Anderson, David P., and Ron Kuivila. 1989. "Continuous Abstractions for Discrete Event Languages." *Computer Music Journal* 13 (3): 11–23. http://www.jstor.org/stable/3680007.

Blackwell, A. F., E. Cocker, G. Cox, A. McLean, and T. Magnusson. 2022. *Live Coding: A User's Manual*. Software Studies. MIT Press.

Salazar, S., A. Kapur, G. Wang, and P. Cook. 2014. *Programming for Musicians and Digital Artists: Creating Music with ChucK*. Manning.