Live Coding and ConTEXt

18th International ConTEXt Meeting

2024

Riviera Taylor

Introduction

- Live coding with Tidal Cycles
- Open Sound Control (OSC): the protocol at work behind the scenes
- Live Coding a ConTEXt Document

What Is Tidal Cycles?

- Live Coding software for generating algorithmic audio
- Domain Specific Language written in Haskell
- Developed about 10-15 years ago by Alex McLean

```
d1 $ n (run 8)
  # sound "bd"
```

- Patterns, “any function of time”, feature extensively in the software (McLean in mylarmelodies, 2024)

Live Coding Audio With Tidal Cycles

Tidal In The Browser

1. strudel.cc (Tidal Cycles in JavaScript)
2. estuary.mcmaster.ca (Collaborative live coding)
3. flok.cc (More collaborative live coding)

Open Sound Control (OSC)

Designed by Matt Wright and Adrian Freed at the turn of the century

> Open SoundControl is an open, efficient, transport-independent, message-based protocol developed for communication among computers, sound synthesizers, and other multimedia devices
>
> (Wright and Freed, 1997)

- OSC Libraries available in many languages

The Composition of OSC Messages

```
Message: /dirt/play ,sssfsfsfsfsiss
    Header
    String: _id_
    String: 1
    String: cps
    Float: 0.666667
    String: cycle
    Float: 581.25
    String: delta
    Float: 0.1875
    String: n
    Float: 2
    String: orbit
    Int32: 0
    String: s
    String: bd
```

Repurposing OSC

- Writing ConTEXt documents via OSC communication foregrounds:
  - Tidal's OSC client capabilities and limitations
  - OSC's ability to send typed data over a network in real time
- The client and server must closely reflect one another, for example:
  - Addresses must match up
  - Arguments need to be unpacked in the correct order

Implementing \blackrules (Step 1)

```
let inheritsSetupblackrules = [("width", Just $ VF 1),
                               ("height", Just $ VF 1),
                               ("depth", Just $ VF 0),
                               ("distance", Just $ VF 0),
                               ("n", Just $ VI 3),
                               ("alternative", Just $ VS "a"),
                               ("style", Just $ VS "STYLE COMMAND"),
                               ("color", Just $ VS "black"),
                               ("variety", Just $ VS "yes"),
                                 -- alias for ("type", Just $ VS "yes")
                               ("mp", Just $ VS "NAME")
                               ]

let blackrulesMessage = [OSC "/blackrules" $ ArgList inheritsSetupblackrules]
```

Implementing `\blackrules` (Step 1)

(OSC) Type	Value constructor
String	VS
Integer	VI
Float	VF
Boolean	VB
Binary “blob”	VX

Implementing \blackrules (Step 2)

```
:{
    let width = pF "width"
        height = pF "height"
        depth = pF "depth"
        distance = pF "distance"
        n = pI "n"
        alternative = pS "alternative"
        style = pS "style"
        color = pS "color"
        variety = pS "variety"
        mp = pS "mp"
:}
```

Implementing \blackrules (Step 3)

```
let setupblackrulesMap = [(contextTarget, setupblackrulesMessage)]

setupblackrulesStream <- startStream (defaultConfig) setupblackrulesMap

blackrules = streamReplace setupblackrulesStream 1
```

Implementing `\framed`

corner	**backgroundcorner**	frameradius	depth	backgrounddepth
topframe	leftframe	region	frameoffset	background
backgroundcolor	extras	foregroundcolor	**offset**	**height**
align	autostrut	autowidth	top	blank
empty	roffset	boffset	anchoring	yanchor
framecorner	radius	backgroundradius	framedepth	framecolor
bottomframe	rightframe	rulethickness	frame	**backgroundoffset**
component	foregroundstyle	setups	**width**	minheight
strut	location	lines	bottom	profile
loffset	toffset	orientation	xanchor	linedirection

Several options accept multifarious data

Conclusion

- Three-step overview of OSC in Tidal Cycles
- Tidal Cycles is for making music
- Perhaps a ConTEXt-specific live coding protocol is needed
  - Would be designed around luametaTEX data types
  - And implemented in Lua

Bibliography

mylarmelodies (2024) ‘How to make music for free, with code: Why We Bleep podcast with ALGORAVE’, [Online]. Available at https://www.youtube.com/watch?v=NUgJgCvX4Y4 (Accessed 9 August 2024).

Wright, M. and Freed, A. (1997) ‘Open SoundControl: A New Protocol for Communicating with Sound Synthesizers’.