

Cyclic Gibbering

Charles Roberts

Interactive Media & Game Development Program
Department of Computer Science
Worcester Polytechnic Institute
charlie@charlie-roberts.com

ABSTRACT

This live coding performance uses Gibber, a browser-based environment for live coding. It is the first performance to feature a browser-based implementation of the TidalCycles mini-notation, enabling new rhythmic and sequencing possibilities in Gibber.

1. PERFORMANCE NOTES

The community around TidalCycles [2] has nurtured it into one of the world's most successful live coding languages, used internationally to both perform and teach. I've been attracted to the expressiveness of its mini-notation for describing rhythmic patterns, and often wished that I could bring the same rhythmic expressiveness to sequencing in Gibber, a browser-based live coding system I design and develop [3]. With the help of Mariana Pachon-Puentes, I recently developed a browser-based implementation of the TidalCycles mini-notation, and incorporated it into Gibber, adding experimental annotations to help reveal the state and progression of running musical patterns as shown in Fig. 1.

This performance will be the first public performance to use the TidalCycles mini-notation in Gibber, and will feature sounds synthesized in the browser mixed with samples drawn from the Freesound online repository [1].

2. PERFORMER BIO

I am an Assistant Professor in the Department of Computer Science at the Worcester Polytechnic Institute, with an affiliation in the Interactive Media and Game Development program. I research human-centered computing in digital arts practice, and am the lead designer and developer of Gibber, an open-source, creative coding environment for the browser. I have given live coding performance using Gibber throughout North America, Europe, and Asia.

3. REFERENCES

- [1] V. Akkermans, F. Font Corbera, J. Funollet, B. De Jong, G. Roma Trepas, S. Togias, and X. Serra. Freesound 2: An improved platform for sharing audio clips. In *Klapuri A, Leider C, editors. ISMIR 2011*:

```
verb = Bus2('spaceverb')
delay = Bus2('delay.1/6')
Tidal.cps = 140/120/2
drums = EDrums().connect( verb, .05 ).connect( delay, .1 )

drums.tidal(`
< [kd kd*2]
[ sd kd ]
~
< [ kd ~ sd kd ]
[ kd sd ]
[ kd sd sd ]>
< ch ch*2 >
< cp cp ~ cp*2 ~ kd >
`)

bass = Monosynth('bass')
bass.note.tidal( `[<0 7 -7>] 3 ~ 4 ~ [<7 0>]*2` )

pad = PolySynth('bleep').connect( verb, 1 )
pad.chord.tidal( `[0,2,4,5]` )
```

Figure 1: Multiple annotated sequences running concurrently. One kick drum token and two notes in the bass line are highlighted to indicate activity.

Proceedings of the 12th International Society for Music Information Retrieval Conference; 2011 October 24-28; Miami, Florida (USA). Miami: University of Miami; 2011. International Society for Music Information Retrieval (ISMIR), 2011.

- [2] A. McLean and G. Wiggins. Tidal-pattern language for the live coding of music. In *Proceedings of the 7th sound and music computing conference*, 2010.
- [3] C. Roberts, M. Wright, and J. Kuchera-Morin. Music Programming in Gibber. In *Proceedings of the International Computer Music Conference*, pages 50–57, 2015.



Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** owner/author(s).

Web Audio Conference WAC-2019, December 4–6, 2019, Trondheim, Norway.

© 2019 Copyright held by the owner/author(s).