

Reactive Audio

Brian Dunn
Hashrocket
918-282-2577
brian@hashrocket.com

ABSTRACT

React has revolutionized the way client side developers think about their applications. It makes heavy use of the principles of functional reactive programming. How can these design concepts aid in the development of live music applications?

With React, events effect state, and the library implements side effects that keep the presentation synchronized. This allows for a simplicity and determinism that restores joy to the process of building complex interactions. We will look at how this powerful idea can be applied to real time music synthesis, rendering music as a side effect of state change.

Timing is everything when rendering music. We will look at some different approaches to achieving precise timing in a reactive programming model. These approaches include push-based, where changes to state precipitate re-rendering, and pull-based, where state is constantly rendered and changes to state are reflected on the next render. The pros and cons of these approaches will be examined.

In terms of the WebAudio API, all of these changes find their way to automating the values of an AudioParam sooner or later. We will discover some potential improvements to this class that could simplify and clarify implementations in this problem domain.

This will be a practical approach. There will be examples in ClojureScript and (hopefully) Elm. Live code execution and glorious rhythmic beeping will keep things fascinating.

BIO

Brian is a senior developer at Hashrocket obsessed with writing browser based music creation tools. He has been using the Web Audio API since may of 2014. Brian has several open source projects available on github, including the ClojureScript apps seqseq and slamplr, and the Ember based seq25. His web audio interests include poly rhythms, functional programming, and designing interfaces that enable happy accidents.

WEBLINKS

Bio on the Hashrocket website: <http://hashrocket.com/team/brian-dunn>

Github profile: <https://github.com/briandunn>

ClojureScript loop based sequencer: <https://github.com/briandunn/seqseq>

ClojureScript audio sample manipulation and playback: <https://github.com/briandunn/slamplr>

Seq25: <http://seq25.com>



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

Web Audio Conference WAC-2016, April 4–6, 2016, Atlanta, USA.

© 2016 Copyright held by the owner/author(s).