# Flocking.js: JavaScript Audio for Artists

Colin Clark

OCAD University Toronto,

Canada

cclark@ocadu.ca

## Abstract

*Flocking* is a web-based framework for audio synthesis, composition, and performance [1]. Written entirely in JavaScript, Flocking applications can run on a variety of platforms including traditional desktop operating systems as well as mobile platforms. Flocking is also used by the author to create sound installations and compositions on embedded platforms such as the popular Raspberry Pi computer using Node.js. This paper will position Flocking within the emerging context of web audio tools and techniques; several recent pieces composed with it will be demonstrated.

Unlike traditional music frameworks, Flocking is based on a declarative approach that represents instruments and compositions as data rather than code. In Flocking, synths and scores alike are composed from collections of unit generators that are specified as trees of JavaScript Object Notation (JSON) documents [2]. Flocking's declarative approach represents a form of *metaprogramming* that can support the creation of programs that are able to understand and transform the structure and content of an instrument or musical algorithm. For example, the author, supported by students funded by the Google Summer of Code program, is working on graphical tools for creating and editing instruments built with Flocking. The goal is to enable musicians who are more comfortable with visual flow environments such as Max/MSP to collaborate directly with developers who prefer traditional text-based computer music programming.

***Example 1****: A synth definition for a an FM instrument, consisting of a tree of unit generators represented in JSON format:*

{

synthDef: {

id: "carrier"

ugen: "flock.ugen.sin",

mul: 0.3,

freq: {

ugen: "flock.ugen.value",

value: 400,

add: {

id: "modulator",

ugen: "flock.ugen.sin",

mul: 150,

freq: 124

}

}

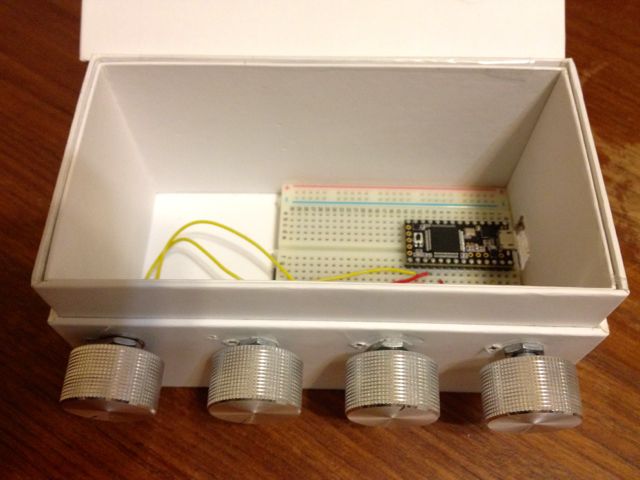
}

}

Flocking provides a very flexible means for reusing and adapting instruments and compositions. A simple but powerful object merging algorithm allows musicians to overlay their own customizations onto an existing instrument without having to directly change its internals. This helps to promote the sharing of modules amongst a body of compositions and across communities of digital instrument designers.

Recently, support has been added for Open Sound Control [3] specifically to handle hardware controllers. An example of a simple OSC controller, dubbed the *Flock Box*, will be demonstrated. The Flock Box consists of four potentiometers connected to a Teensy embedded controller [4] that issues OSC value changes each time a pot is moved.

***Image 1****: The* ***Flock Box****, a home-built Open Sound Control controller that can be used with Flocking:*



A Node.js server listens for OSC serial port messages and forwards them to Flocking. The OSC data sent by the Flock Box and other OSC sources can be easily mapped to arbitrary inputs within an instrument's unit generator tree using a declarative *input map* specification.

***Example 2****: A declarative input map, which binds instrument inputs (keys) to incoming OSC message addresses (values):*

inputMap: {

"carrier.freq.value": "pots.0",

"modulator.freq": "pots.1",

"modulator.mul": "pots.2"

}

Flocking has proved to be highly germane to the author's own musical practice, supporting an integral conversation between the composition process and the emerging technical capabilities of the system. It has promoted a compositional approach that embraces constraints and an aesthetic of making the most out of small, composable building blocks. Flocking is a viable musical tool for composers and sound artists working in a variety of media and styles, and is available on Github for use and modification under a liberal open source license [5].

## References

[1] http://flockingjs.org

[2] Crockford, Douglas. *Introducing JSON*. Web. http://www.json.org/ Accessed February 21, 2014.

[3] Wright, Matt. *The Open Sound Control 1.0 Specification*. http://opensoundcontrol.org/spec-1\_0 Web. Accessed February 21, 2014.

[4] Stoffregen, Paul and Robin Coon. *Teensy USB Development Board*. Web. https://www.pjrc.com/teensy/ Accessed February 21, 2014.

[5] <https://github.com/colinbdclark/Flocking>

## Bio

Colin Clark is a composer, video artist, and software developer living in Toronto. He is the creator of Flocking, a JavaScript framework for web-based creative audio coding. Colin is currently the Lead Software Architect at OCAD University’s Inclusive Design Research Centre and is a recognized leader in the field of inclusive design and software architecture. His music has been performed by Arraymusic, the neither/nor collective, the Draperies, and his own ensembles, Lions and Fleischmop. Colin’s soundtracks for experimental films by Izabella Pruska- Oldenhof and R. Bruce Elder have been shown at film festivals and cinematheques internationally. He is currently working towards a MFA in Interdisciplinary Art, Media, and Design at OCAD University.