Contact Info

Name: Jeevan Farias
Email: jeevanfarias@gmail.com
Forum: jvntf

Website: jvn.tf
Code samples: github.com/jvntf

P5.JS.Sound Enhancements -

Abstract Classes, Sound Design, and Composition

Project Abstract

Processing and p5.js are accessible tools that can be easy to understand for the novice programmer and powerful enough for the experienced. With the intention of helping to diversify the potential users and uses of Processing / p5, this proposal outlines a series of enhancements to p5.js.sound, including new effects, presets, and modules for algorithmic composition. These developments will be useful and interesting to electronic musicians interested in visual art, visual artists interested in electronic music, sound artists, and any other potential user of Processing. These developments are inline with the vision of the tool, as they will add features without distracting from its definition as a “flexible software sketchbook”.

Project Description

As a “flexible software sketchbook,” Processing makes code more accessible to artists and designers by giving them a sandbox for quick experimentation with a gentle learning curve. Simply put, coding for visual art in Processing and p5.js intuitively translates the experience of drawing into syntax. I am proposing a series of modifications and additions to the p5.js.sound library, to give artists an equivalent experience when dealing with audio.

This project will be divided into three sections of research and programming, separated by the evaluation periods built into the GSoC Timeline.

The first four weeks will go towards development of the Sound library code base. I will develop a new p5 object, called p5.Effect to setup a frame for future additions to the library. With consent from the community, I think it would make sense to use Effect as a superclass of Filter, and the new equalizer and compression objects that I would like to create. These are two of the most heavily used processing effects

in music production because they quickly add richness and natural-ness to, inherently artificial sounding, computer generated audio; they will will be useful for advanced sound design in p5.js.

A logical companion to a sound library, is an array of presets, which offer users the option of various platforms on which to start building. During the second four weeks, I will develop these presets for EQ, Compression, and Reverb in addition to the Oscillators (will be referred to as Instruments). These presets will be documented well documented with explanations and sound samples. They will load default parameters, which can be changed or specified by the user. Instruments may be particularly challenging, as they need to play notes easily. With consult of the work that has already been done, linked below in *Related Work*, I am interested in a exploring a few different options, including a scheduling system for the playing of notes, or the use of Envelopes to power this process.

The final four weeks will be dedicated to creating a series of modules for algorithmic composition. These functions will accept parameters, (ex. Key, note-range, note-length, rate of evolution...etc), to create uniquely evolving sound. They will be based on mathematical models (references below), and will generate pitches and rhythms to be played by Instruments.

This project is designed to be helpful to users of p5, who come from all disciplines. P5.js is a tool that should be helpful to artists across media, and these enhancements will be useful both by experienced musicians, who want to control their sound in subtle ways while exploring the sketching of visuals, and visual artists, who lack experience in sound design yet want an audio production toolset in their work. P5.js is a fast and flexible tool, and these proposed developments will help yield more sophisticated sound output without complicating its interface and experience.

Development Process

Community Bonding Period May 4th - 30th

Discussion

- Discussion with mentor and community forum regarding proposal / potential uses of enhanced sound library.

P5.js.sound

- Work through source code written by Jason Sigal and documentation. This will be supplemented by experimentation with the existing components of the library.

Investigation Period | May 30th - June 4th

This time will be dedicated to understanding the existing p5.js.sound code, the Web Audio API, and concepts / pseudocode both for audio effects and algorithmic composition. I will consult the following resources in my research:

Web Audio API

- I will consult the documentation and begin to get an idea of how nodes can be combined to produce the effects listed above.
- Construct diagrams of how Web Audio handles routing.

Audio Effects / Algorithmic Composition

- I will consult the source code of RTcMix language to understand how audio effects and instrument simulators can be built from scratch.
- I will also consult various VST's and Audio Units in addition to written overviews of EQ, Compression, and sound design for instruments
- The work of Paul Bourke will be helpful in preparing to write small modules for algorithmic composition. He describes a variety of procedures and explains the theory in terms of of the mathematical function and pseudocode.

First Coding Period | June 5th - June 30th

Additions + Enhancements to p5.js.Sound library

Deliverables: p5.Effect as super class, p5.Filter restructured as a subclass. Additional p5.Eq, p5.Compress and p5.Pan objects to serve as a basic features that inherit from an abstract class and have a consistent API

Week 1 | June 5th - June 11th

- p5.Effect abstract class
    - Restructure the Sound library such that p5.Filter inherits from p5.Effect
    - The Effect API should offer some basic and general functionality, that can be further complicated by new effects

Week 2 | June 12th - June 18th

- Implement Compression as a subset of p5.Effect
    - Build from Web Audio DynamicsCompressorNode
    - Important for narrowing amplitude range of outputted sound; warmer, richer outputs
- Implement spatial-panning as a subset of p5.Effect
    - p5.js.sound allows for left/right panning of a sound file
    - Web Audio SpatialPannerNode can enable 3D positioning of an audio source
    - Very useful feature for artists using p5 to power their performance or installation pieces, as well as for complex applets

Week 3 | June 19th - June 25th

- Implement Equalization as a subset of p5.Effect
    - 8 band EQ system, combining bandpass filters to create a smooth EQ curve
    - Options for hi/lo-cut
    - Use biquadFilter to string together bandPass filters along the spectrum, interpolate the sound spectrum
    - Potential: include a p5.js sketch example that creates a graphical equalizer for a loaded soundfile, for testing and calculation of EQ parameters

Week 4 | June 26th - 29th

- Comments and Reference Documentation

Second Coding Period | June 30th - July 28th

Presets for p5.js.sound

Deliverables: p5.Instrument abstract class, presets for various effects and instruments. Presets can be stored as a JSON files (similar to Tone.js) or as constructible variables within p5.js classes

Week 1 | June 30th - July 6th

- Equalization and Compression Presets
    - Develop library of pre-configured EQs, organized by instrument-sound (meaning real-world sounds such as a strings, bass, drums, voice..etc), and effect
- Develop reverb presets (based on architectural spaces, i.e. *basement, bathroom, cathedral, drum room*)
    - These are described conceptually in the Web Audio API
    - Reference: commercial reverb plugins
- These presets are a matter of testing and tweaking / reapplying the specifications of Tone.JS and wad.js to work with p5
- TDB: how to save these presets? JSON vs as variables in the library

Week 2 | July 7th - July 13th

- Timing
    - Develop p5.js.Sound's ability to play sound in time
    - This will be crucial to accurately playing algorithmic compositions
    - (It is possible that this will take more time, in which case I would push the instrument design into the Third Coding Period, and forego some work on Algorithmic Composition)

Week 3 | July 14th - July 20th

- Develop Instrument (abstract class?) for synthesized sound objects
    - Setup control of p5.js objects such as a oscillator, envelope, gain, distortion, reverb, and filters
    - Interface through which the user can specify an instrument by name, and call the construction of the necessary objects to build this sound
    - Would like to investigate a schedule-based system
        - Play notes by specifying: pitch X, duration Y, start time Z.
    - Could also try using envelopes and a concrete measure of tiime from the start of the sketch (when the page is fully loaded).

Week 4 | July 21st - July 27th

- Instrument presets
    - Design basic set of instruments based on oscillators
        - Bass, synthesizer, and ambient sound instruments
        - Translation from note format to frequency
    - Compare storage as .json files or as a variable information in the instrument.js file

Third Coding Period | July 28th - August 21st

Algorithmic Composition

Deliverables: A series of examples of algorithmic composition in p5.js. These algorithms will develop pitches to be sent through an Instrument object. Documentation with explanations, sound, and code

Week 1 | July 28th - August 3rd

- Translational composition
    - Load files and convert into pitches / rhythms, to be played by an Instrument
    - Will focus on translation from image to sound

Week 2 | August 4th - August 10th

- Mathematical composition
    - Use functions of 3D geometry to produce pitches / rhythms
    - Reference to the documentation of Paul Bourke as linked above
- Interest:
    - Markov chains, fractals, strange attractors

Week 3 | August 11th - August 17th

- Evolutionary Composition
    - Genetic algorithms

Week 4 | August 18th - August 21st

- Final documentation and submission

Risk: As stated above, the Timing / Instrument section of the project is expected to be the most challenging portion. In the event that the second coding period's tasks take more time than allotted, I think it will be fine to abandon some of the algorithmic composition developments. These will be more in the realm of development of examples and explanations, and thus are less important than the successful creation of a timing system and usable Instruments

---

Related Work

p5.Synth work by b2renger
https://github.com/processing/p5.js-sound/issues/47
Wad
https://github.com/rserota/wad
Tone.js:
https://github.com/Tonejs/Tone.js
RTcMix
http://rtcmix.org

More about you

I am a musician, programmer, and designer who works across both digital and non-digital media. I am interested in social and environmental justice, free information, digital security, combinations of digital technology and traditional craftsmanship, music that makes you dance, and objects that are beautiful and functional. Playing and listening to music has been a central part of my life since the age of three. This part of my life fuels my explorations into other creative and experimental endeavors, having taught me about collaboration, rule-following, rule-breaking, enjoyment, and dedication.

I have experimented with Processing and p5.js, using it as a playground for visuals when I first began to program. Diving into Processing documentation was my first time getting excited about computer science and getting completely lost in examples, reference, and trial-and-error.

I am currently working on a related computer music project in a course with Professor Brad Garton at the Columbia Computer Music Center. In studying the RTcMix language and its use with MaxMSP and Unity, I have learned a lot about the concepts of digital synthesis and processing, which is pertinent to the tasks outlined in the Second Coding Period. I will consult the work done in my course along with resources provided by my professor to aid in the development process.

I have not contributed to any open source projects, but my online portfolio of art and design work is on my website at jvn.tf and code samples are posted at github.com/jvntf. I have additional sample of advanced work in C and C++ that I can forward via email (cannot be posted online because they are course assignments).