

# Procedural Soundscape

Claire Goeckner-Wald & Amy Xiong

## 1 Introduction

A short overview of your project idea, why it is interesting and what you would like to learn from it.

Example citation: [1]

## 2 Detailed Description and Features

### 2.1 Description

Precise description of features

### 2.2 Technical requirements

Audio requirements (and probably also requirements necessary to run the files)

### 2.3 Technical challenges

### 2.4 Technological Review

Discuss relevant github repos and projects and articles

Wheelibin's synaesthesia (drum kit and synths). We based our chords.js and rhythms.js off of synaesthesia. Tambien's Jazz.Computer but they had a very strange music set-up with interpolation and mediators between synths. We attempted to replicate by playing multiple synths simultaneously but the resulting 'piano' was very tinny and sounded like Scottish bagpipes. We got mostly drums synths and pads from Yotamm.

### 2.5 Licensing

## 3 Implementation

### 3.1 Technical Design

Technical design and reasons for choosing this design

We've decided to use tone.js instead of audiosynth.js by Keith William Horwood. We also chose tone.js over SuperCollider, Flocking.js, PureData, because of web integration ease. Flocking also had web integration but a smaller online community and seemed less robust and less abstracted. Audiosynth.js did not have looping technologies. We briefly considered integrating with audiosynth.js but it proved to be too difficult.

We looked at Karplus-Strong String Synthesis but determined it was only relevant if we used audiosynth.js.

## 3.2 Implementation Issues

Implementation issues, note any particular technical or audio difficulties and work-arounds

We looked into playing and modifying audiofiles (MP3s or WAVs) but the integration with Tone.js seemed too complicated [3]. This threw us off track for several weeks as we searched for good synths to use (we found none). Claire had switched to DuckDuckGo during the project, which has a much less 'intelligent' search algorithm, which would not return the repo "tonejs-instruments" from a search for "tone js instruments" [1]. In `tonejs-instruments`, they use publicly available WAVs from about a dozen instruments, including saxophone and piano (but no drums). Google, however, did, so eventually it was found. We relied heavily on this repo. <https://github.com/Tonejs/Tone.js/issues/290> references why we can't use .wav files in Tone.js (simply not supported, it seems).

We discussed moving instruments synths into instruments.js, out of main.js. Moving roots/scales/chord progressions/rhythms/min & max on octaves in chords.js

Licensing a no-licensed repository (wheelibin's synaesthesia) [4]. Choose-a-license, hosted and run by GitHub, was very helpful [2].

## 4 Analysis & Conclusion

### 4.1 Original Goals

How close did you come to achieving original goals?

### 4.2 Regrets

What would you do differently if you knew at the start what you know now.

### 4.3 Next step

What would be your next steps if you were to continue working on the project.

## 5 Code

### 5.1 File

Description:

Verbatim code.

## 6 Contact

Example contact.

## 7 Appendix

Include resources here.

- Item example.

## References

- [1] Nicholas P. Brosowsky. tonejs-instruments, 2018.
- [2] Choose A License. No license, 2018.
- [3] Srgy Surkv. Add to tone.js real instruments for music applications, 2018.
- [4] Jon Wheeler. synaesthesia, 2018.