Declarative Web Audio API

Alexander Inkin
Tinkoff
Nizhny Novgorod, Russia
alexander@inkin.ru

ABSTRACT

In this talk, we will explore differences between imperative and declarative approach to Web Audio API. We will see what it takes to turn Web Audio API declarative with dependency injection pattern also known as inversion of control and what benefits it brings. We will create basic proof of concept using plain JavaScript and compare it to more advanced code based on Angular [1] framework. All these techniques will help us write reusable structures and program music with Web Audio API. On top of that, we will look at Web MIDI API and create playable synthesizer using same declarative approach so we can perform music live. We will also touch on how using this knowledge and a little bit of web assembly [2] we can create a Guitar Hero-like and Karaoke game which can be played directly in the browser, even on modern smartphones.

1. DECLARATIVE APPROACH

Many native JavaScript APIs are imperative in their nature, including Web Audio API. We can clearly see the difference when we think about graphics. Sure, there are advantages to using low-level imperative Canvas API to draw in many cases. However, it is much easier to work with declarative SVG when you need to have reusable graphics. Web Audio API is essentially a graph of nodes that process the signal. If we could turn this graph into visual XML-like tree, it would be easier to understand, manipulate and reuse. We can achieve this with native Web Components with a few lines of code that we will explore.

2. DEPENDENCY INJECTION

To be able to create connected XML graph of audio nodes, deeper nodes need to have ability to receive node standing before them and provide themselves to the nodes going after. We can accomplish this through mechanism of dependency injection. This way each node becomes “pluggable” which makes structures easy to reuse.

3. PROGRAMMING MUSIC

There are frameworks utilizing dependency injection and declarative approach, namely Angular. We will see how we can write a small wrapper library to turn Web Audio API declarative and consume it in a framework application [3]. Programming music works a lot like old player piano. We need to create a playback mechanism and feed it instructions at a certain pace. A stream of instructions can also be passed to all parts of the program through dependency injection. We will create a simple loop of several tracks playing in sync to demonstrate the approach.

4. PERFORMING MUSIC

Following discussed architecture, it is easy to adapt code to live performance. Instead of a stream of instructions, we will just listen to stream of MIDI events. Angular uses RxJS Observable model to process events [4]. Web MIDI API is based largely on traditional DOM events and we can adapt it to the same model [5]. This would allow us to work with MIDI easily and even supplement it with emulation in non-supporting browsers or different environments, such as touch screen displays. Which would come in handy in our final goal – browser based music game.

5. FINAL APPLICATION

With all this in place, we can create a game with two modes – playing and singing. While pitch detection can be done with Web Audio API, for such task web assembly is better-suited [2]. We will see how with other native browser APIs, such as intersection observer, our goal can be achieved with relatively little code and almost no math. Declarative approach and other proposed patterns would help us create maintainable and transparent code, which is easy to complement with interchangeable instruments, transposition and other features.

6. REFERENCES

[1] Google. *Angular* https://angular.io.

[2] The aubio team. *aubio* https://aubio.org

[3] Alexander Inkin. Writing Retrowave in Angular *Newline.co* (March 19, 2020). https://www.newline.co/@waterplea/writing-retrowave-in-angular--20856fee.

[4] RxJS https://rxjs-dev.firebaseapp.com

[5] Alexander Inkin. Jam on your MIDI keyboard in Angular *Newline.co* (May 3, 2020). https://www.newline.co/@waterplea/jam-on-your-midi-keyboard-in-angular--6c16e1e0

Web Audio Conference WAC-2021, July 5–7, 2021, Barcelona, Spain.