Tuning-in on the Web: Developing a Framework for ensemble interaction in the browser

Tyler Ptak
Purdue University
tylerptak@gmail.com

Dr. Ian Hattwick
Massachusetts Institute of Technology
hattwick@mit.edu

Introduction

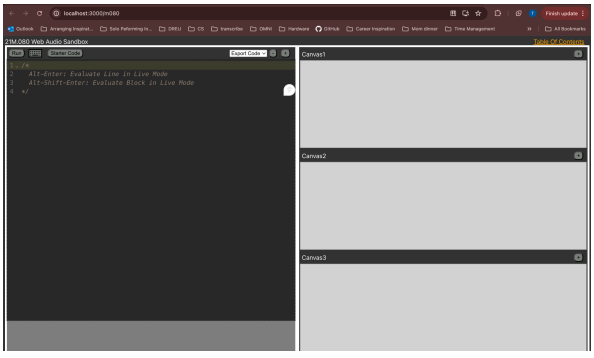
Music technology has always reflected the technology of the era it was born in. The analog era created synths and amplifiers. The digital era created software instruments and DSP. Today, the internet era is creating web audio. Most people's experience with audio on the web comes from streaming services like spotify, though computer musicians and sound designers take it further. Over the past 20 years, developers have worked hard to create tools for audio processing in browsers. These tools are useful for creating web based instruments and games. For Dr. Ian Hattwick, these tools offer endless possibilities. My project this summer was called *Tuning-in on the Web: Developing a Framework for ensemble interaction in the browser*. We used web audio to create collaborative musical instruments hosted on the web. Web audio is a very new field though we believe that it has the potential to create unique and exciting musical experiences.

Motivation

Computer music is the art of creating music using software tools and for Dr. Ian Hattwick, it's a lifelong passion. Dr. Hattwick is a professor of music technology at MIT and the leader of the faMLE laptop ensemble. The laptop ensemble is a group of computer musicians that perform at MIT events and concerts. In his experience, he noticed that creating this type of music can be isolating, especially with web audio tools. It's often hard to have a performance where two performers can organically play off one another and have a conversation back and forth. We want to make more interactive experiences so people can get the same feeling one gets when playing acoustic instruments together.

Previous Work

We built this project on top of the website for 21M.080, MIT's introductory music technology course. Last summer Dr.



Hattwick and a team created the website - a web audio sandbox. The website lets students write code and run it directly in the website. It also includes a list of examples which teach students how to use Tone.js and Dr. Hattwick homemade features. The interface includes a code box and three canvases for GUIs. Some of Dr. Hattwick's features include an oscilloscope, a spectroscope, a midi keyboard, and stock GUI elements which can control musical elements.

For inspiration, we also used web audio weekly, a journal which posts web audio projects from around the world. These projects range from instruments, effects, or full musical performances (Lowis, 2013).

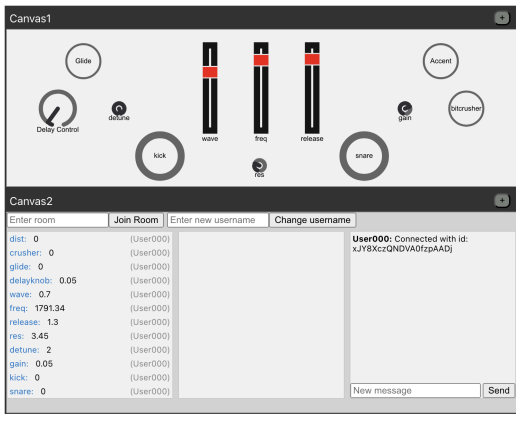
Implementation

To do this, we decided to use web audio tools to create our own software instruments and then add collaborative software. The web audio API is built off of javascript and was introduced in 2013 to provide a powerful system for controlling audio on the web. It allows users to create an audio context where they can chain different audio nodes together for different effects. The modular nature of this design makes it possible for users to create complex audio effects. The chains start with an audio input, have a series of effects, and then end with an output. Many effects are already defined in the API like delay, filters, compressors, gain stages, and distortions. You can chain these building blocks together to create new and more complex effects (Web audio, 2013). There's also a web midi API that handles functionality for processing and triggering MIDI messages.

The web audio/midi API is powerful though it's also convoluted to use. If you're not a skilled developer, there's a good chance that you'll get too confused and give up. This makes it very hard for musicians who want to use the tech for their own projects. To solve this, Yotam Mann at NYU created Tone.js, a javascript library that packages web audio/midi functions into more intuitive calls. Its goal is to be familiar to both musicians and programmers. It's a great piece of software which functions like a common DAW while also providing blocks to create synths, effects, and complex signal chains (Tone.js, 2014). We used Tone.js as the foundation for our new instruments. To make the instruments collaborative, we used Collab-Hub, an API created by Nick Hwang, Eric Sheffield, and Tony Marasco. Collab-hub is a server that was created as a tool for musical collaboration. It allows multiple users to send data easily so they can collaborate on that data. The server lets you send and receive control data and events to other users on the platform (Hwang, 2021).

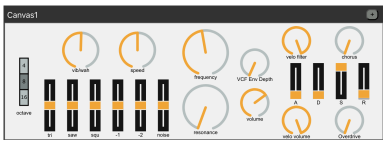
Instruments

Over the course of the summer, we achieved our goal of creating collaborative musical instruments and added more features to the 21M.080 framework. We started by replicating the DatoDuo, a



simple subtractive synth/sequencer. On my four person team, I created the synth, Diego created the sequencer, Artem worked on Collab-Hub, and Javier worked on other features. This project was our sandbox to learn and test our ideas. I learned how to create my own synths and effects, Diego learned how to work with time in Tone, and Artem learned how to use Collab-Hub. Out of all the tasks, the Collab-Hub portion was the hardest. After weeks of work, Artem had found a way to link two copies of the same instrument on different computers. Within each GUI element, we added a field called link which must be the same name on both computers. Behind the scenes, both instruments are connected to Collab-Hub, sending messages in real time back and forth. This keeps the value of each element the same between users. He also created functions allowing us to send custom messages to other users from the code box.

After the DatoDuo, I created two more collaborative instruments. The ESPsynth is a clone of Logic Pro X's ES P, a 6 voice polyphonic synth. The synth includes an LFO which can control vibrato or wah, a filter, an ADSR, chorus, and distortion. The filter is controlled by the frequency, resonance, and Env Depth knobs. The amplitude is controlled by the volume knob. A demo of the instrument can be found here. I also created the audience choir, an instrument that leverages the full power of collabhub. The instrument was inspired by Jacob Collier's audience choir where he splits the crowd into sections and controls them like a choir. The audience choir instrument simulates this with a host and a crowd interface. At a concert, the host would be the performer who starts the audio. Then, audience members could scan a QR code and open their interface. Every time an audience member joins, it creates a new voice on the synth. Then they can control the note that's played by clicking an up or down button. The host could then control the choir just like Jacob Collier does but instead of singing, the crowd is using their phones as an interface. A demo of the instrument can be found here.



New Features

On top of instruments, we also added features to the 21M.080 website. We added a share to link function, which encodes all the text in the code box into a link. Then, you can share the link with others to send projects back and forth. We created an ASCII handler which allows users to set up handlers for keyboard presses. We added a theory module which makes playing chord progressions or scales easier. We added functionality for setting and saving synth presets. We also added a Polyphony class which can make any monophonic synth polyphonic.

Plans for the future

We made a lot of progress this summer though we couldn't implement everything. In the future, we want to continue to create more instruments. We also want to have an export as

webpage function which would let users export their code from the code box into a standalone web page.

Acknowledgements

Thanks to Dr. Ian Hattwick for having me in their lab for the summer. I learned a lot about music technology and its impact on performance. Thanks to DREU and the CRA for accepting me and providing funding for this internship. Thanks to the Massachusetts Institute of Technology for hosting me this summer.

References

Hwang, N. (2021). *Welcome to Collab-Hub*. GitHub.

https://github.com/rioter00/Collab-Hub/blob/master/docs/index.md

Lowis, C. (2013). *Web Audio Weekly. https://www.webaudioweekly.com/*

Tone.js. (2014). https://tonejs.github.io/

Web audio API. MDN Web Docs. (2013).

https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API