Politecnico di Milano

Music and Acoustic Engineering

Advanced coding tools and methodologies

Development of a subtractive synthesizer

Olga Besedova 992692

Andres Bertazzi 102855

Natasa Popovic 993401

Date: September, 2022

Contents

1 Introduction

The goal of the project was the development and implementation of a subtractive synthesizer for web browsers using Javascript, Tone.js web audio framework, as well as Nexus API for creating the graphical user interface. The final result is a user-playable instrument where the sound is fully customizable by the user, and the user is able to interact with the instrument in three different ways: by interacting with the screen keyboard, with the computer's keyboard, or with an external midi keyboard.

2 Synthesizer - Logic and main components

2.1 Subtractive synthesis

Subtractive synthesis is one of the main and most popular types of sound synthesis, in which partials of an audio signal are attenuated by a filter to alter the timbre of the sound. This type of synthesis starts from a waveform where the type of the waveform can be chosen by the user (sine, triangle sawtooth, and so on) and its harmonic content that it has so that through a filter it is possible to synthesize different musical timbres. In its most basic form, subtractive synthesis requires 3 core elements: oscillator, filter, and envelope.

- Oscillator generates the waveform of the user's choice, which will be decisive in the final sound of the subtractive synthesis. Among the different options, particularly, we can find sine, sawtooth, square, and triangle, where each of them has different harmonic properties. Actual shape of the sound generated can be seen on the Oscilloscope.

- Filter is the main component, as this is where the actual subtracting is happening. In this application, three types of filters are available: high-pass, low-pass, and band-pass, where the user can select the type, as well as control Cutoff and Gain parameters.

- Envelope generator determines what type of sound the synthesizer is generating, as it can vary from percussion sounds to something similar to the sound of the violin. For the envelope generator, attack, sustain, decay, and release (ADSR) are taken into account. These parameters can be varied in order to adapt the sound to fit the user's preferences. The user has the possibility to modify the values of the parameters attack, decay, sustain and release.

In order to give more possibilities, and obtain more interesting results, two different channels were implemented in the synth, so the user has the possibility to choose to play one single channel or both, superimposing the two outputs. The user can control the volume of the individual channels, but also the mix of the two.

2.2 Effects

Besides being able to customize the parameters of the oscillator, filter, and ADSR, the user has been given a chance to apply various effects to the sound. Effects that were chosen to be implemented in the project are: Chorus, Vibrato, Tremolo, Phaser, Reverb, and Ping-pong delay.

- Chorus - this effect is by far one of the most popular phase modulation effects. It is meant to simulate the subtle pitch and timing differences that occur when multiple musicians play the same note but vary slightly in pitch and timing. We can describe the sound obtained by using this effect as a doubling effect that adds thickness so the sound sounds "larger" than it would on its own. A chorus achieves subtle changes in pitch by modulating the delay time of the waveform with a low-frequency oscillator (LFO). The timing differences cause a constantly changing phase shift between the original signal and the duplicated signal. The user is able to control the LFO, Delay Time, and Depth parameters of this effect.

- Vibrato - it is a modulation effect that varies pitch. As the pitch cyclically goes higher and lower in response to the modulation, a sense of movement and rhythm are created, and the variations in pitch can also make the sound seem stronger and richer. The user is able to control the Frequency and Depth parameters of this effect.

- Tremolo - although often confused with vibrato, this modulation effect creates movement and rhythm by varying the amplitude of the signal. Tremolo is particularly useful for creating percussive shuddering or stuttering, as well as gentle or intense pulsating effects.

- Phaser - this effect combines the original signal with a copy that is slightly out of phase with the original: this means that the amplitudes of the two signals reach their highest and lowest points at slightly different times. The user is able to control Frequency, Base frequency, and Octave parameters.

- Reverb - this effect adds reverberation: rapid, modified repetitions blended with the original sound that gives an impression of ambiance. User can customize the Decay parameter and therefore vary the impression of the sound.

- Ping-Pong Delay - this effect is a stereo feedback delay where the delay bounces back and forth between the left and right channels. The user is able to adapt both Delay and Feedback parameters.

3 Development

3.1 Technologies

The project was built as a web application. It has been developed using HTML, CSS, and Javascript. External Javascript libraries were used: Nexus.js (https://nexus-js.github.io/ui/), Tone.js (https://tonejs.github.io/), WebMidi.js (https://github.com/djipco/webmidi) and a css framework
$(http://thisisdallas.github.com/Simple-Grid/)$.

3.2 Backend

The backend part is based on the Tone.js web audio framework. Tone.MonoSynth and Tone.DuoSynth are used for subtractive synthesis implementation as these Tone.js objects perfectly fit our general idea. For example, MonoSynth is composed of one oscillator, one filter and the envelope, and DuoSynth is a synth composed of two MonoSynths run in parallel. The effects part is implemented as well with the help of Tone.js objects such as Tone.Chorus, Tone.Vibrato, Tone.Phaser and others. Before playing the sound, all selected settings are gathered and saved in the sound_setting class which consists of an array of effects and an array of synth options. Every time the user changes an effect or a synth option, sound settings are updated. Midi and keyboard parts are implemented using WebMidi library and events handling mechanism.

3.3 GUI

Graphical User Interface was created with the help of NexusUI - a collection of HTML5 interfaces and Javascript helper functions to help with building web audio instruments in the browsers. With the support of the NexusUI, implementation and design was made simpler with the help of interface components such as: Dial, Toggle, Slider, Oscilloscope and Spectrogram. Interface was divided in 3 components: one dedicated to Effects (upper part), one dedicated to actual Subtractive Synthesis (middle part) and one dedicated to the keyboard and visual representations of the sound - Oscilloscope and Spectrogram, located in the bottom part of the screen. Special care was taken to organize the interface grid in such way and label the components, so it is the most understandable by the user, as well as user friendly. For the visual style of the synthesizer, simple colors and style were chosen, in order not to make whole experience too overwhelming. Visual style of the interface changes depending of the state: dials, toggles, and effect blocks change their color when turned on.

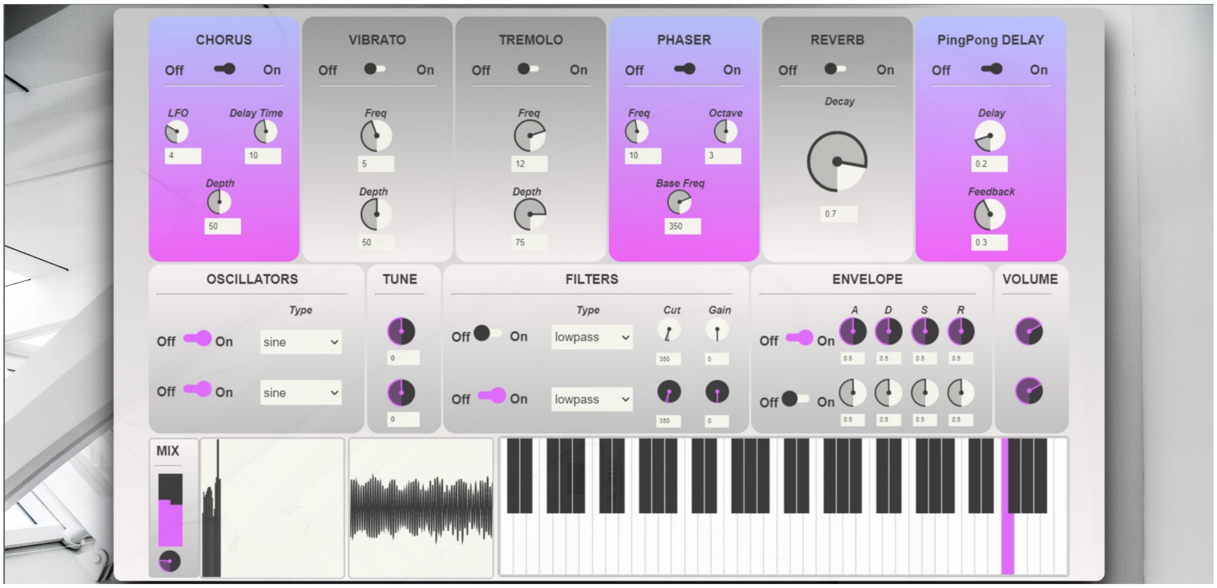

Figure 1: Graphical User Interface