



Norwegian University of
Science and Technology



MCT4048: Audio Programming

The Fundamentals: Playing Sounds

Anna Xambó

Department of Music, NTNU

5 February 2019

Survey



<https://goo.gl/C1gkae>

Start setting up...



`https://github.com/axambo/audio-programming-workshop/`
Go to: code / d1 / 00-setting-up / checklist.md

Warm-up Activity



Mind map exercise: What is Web Audio? Pros and Cons?

Pros



- Easy access.
- Broad distribution.
- Social features.
- Interactivity / audiovisual / media-rich content.
- Internet-based.
- Cross-platform.
- Fast development.
- Multiple online resources available.

Cons



- Computational performance is limited (code execution is slower than compiled programming languages (Java, C++)).
- Consider writing a native application if your program needs to execute computationally intensive algorithms.
- Web browsers designed to favor user experience, single-thread computations (AudioWorklet is a workaround).
- Dependent on Internet connectivity.
- Ephemerality: sustainability, maintenance and variability of platforms (form factor, input method, computational power...).

This Week: The Fundamentals (40% Individual Work)



- Syllabus: <https://uio.instructure.com/courses/17406>
- Assignment 1 (Total grade: 10%): Presentation WAC paper (individual) – day 3 (February 7, 2019) or 4 (February 8, 2019)
- Assignment 2 (Total grade: 20%): Presentation mini-project 1 (individual) – days 2 (February 6, 2019) (5%), 3 (February 7, 2019) (5%), 4 (February 8, 2019) (10%)
- Assignment 3 (Total grade: 10%): Written blog post about the mini-project 1 – February 11, 2019

Program: Day 1 – 5 February, 2019



- 9.15-10.00: Setting up computers with the tools for the tutorial
- 10.00-12.30: Tutorial: Playing sounds
- 12.30-16.00: Mini-project 1 development (1/4)

Learning Outcomes



- Understand the pros and cons of using Web Audio for audio programming.
- Get familiar with a toolset of web technologies to start developing programs for the web based on audio.
- Be able to find suitable information from the Web Audio API and related webpages / projects and adapt it to own needs.

Setting Up...



Follow the instructions from the “00-setting-up” folder.

DOM



The Document Object Model, usually referred to as the DOM, is an essential part of making websites interactive. It is an interface that allows a programming language to manipulate the content, structure, and style of a website. JavaScript is the client-side scripting language that connects to the DOM in an internet browser.

<https://www.digitalocean.com/community/tutorials/introduction-to-the-dom>

Tutorial



`https://github.com/axambo/audio-programming-workshop/tree/master/code/d1`

Connecting audio nodes



- LittleBits: <https://www.youtube.com/watch?v=4th8p0jSK9E>
- PureData: <https://www.rebeltech.org/2016/04/07/pure-data-patch-introduction/>
- WebAudio: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Basic_concepts_behind_Web_Audio_API

The Web Audio API



- The Web Audio API involves handling audio operations inside an audio context, and has been designed to allow modular routing.
- Basic audio operations are performed with audio nodes, which are linked together to form an audio routing graph.

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

OscillatorNode



The OscillatorNode interface represents a periodic waveform, such as a sine or triangle wave. It is an AudioNode audio-processing module that causes a given frequency of wave to be created.

<https://developer.mozilla.org/en-US/docs/Web/API/OscillatorNode>

GainNode



The GainNode interface represents a change in volume. It is an AudioNode audio-processing module that causes a given gain to be applied to the input data before its propagation to the output.

<https://developer.mozilla.org/en-US/docs/Web/API/GainNode>

AudioBufferSourceNode



The `AudioBufferSourceNode` represents an audio source consisting of in-memory audio data, stored in an `AudioBuffer`. It's especially useful for playing back audio which has particularly stringent timing accuracy requirements, such as for sounds that must match a specific rhythm and can be kept in memory rather than being played from disk or the network.

<https://developer.mozilla.org/en-US/docs/Web/API/AudioBufferSourceNode>

Mini-project development



You are expected to create a mini-project that should be doable within a week. The overall aim is to get familiar with web audio. Here are different approaches that you can take:

- Develop an idea based on what we are seeing in class.
- Adapt an existing code to your needs and document what are the changes.
- Other?

Working style



- Individual work but in shared rooms. You are encourage to share and discuss with your peers.
- One-to-one talks via Zoom or personally with the instructor to catch up.
- There will be 4 time slots during the week to work on the project. It is okay to change the topic over the course of the week. Keep a research journal.

Relevant Links



- Syllabus: `https://uio.instructure.com/courses/17406/pages/syllabus`
- GitHub slides & code: `https://github.com/axambo/audio-programming-workshop`