



Norwegian University of
Science and Technology



MCT4048: Audio Programming

The Extensions: AudioWorklets

Anna Xambó

Department of Music, NTNU

15 February 2019

Start setting up...



Download:

`https://github.com/axambo/audio-programming-workshop/`

Go to: `code/d8/00-setting-up/checklist.md`

This Week: The Extensions (40% Group Work)



- **Syllabus:** <https://uio.instructure.com/courses/17406>
- **Assignment 4** (Total grade: 30%): Presentation mini-project 2 (group) – days 6 (February 13, 2019) (10%), 7 (February 14, 2019) (10%), 8 (February 15, 2019) (10%)
- **Assignment 5** (Total grade: 10%): Written blog post about the mini-project 2 (group) – February 22, 2019

Program: Day 8 – 15 February, 2019



- 9.15-9.30: Setting up computers with the tools for the tutorial
- 9.30-10.30: Tutorial: AudioWorklets
- 10.30-12.30: Mini-project 2 development (4/4)
- 12.30-13.00: Lunch break
- 13.00-15.00: Mini-project 2 development (4/4)
- 15.00-16.00: Speedy presentations mini-project 2 (3/3)

Learning Outcomes



- Get a sense of Object-Oriented Programming Concepts in JavaScript.
- Get familiar with the AudioWorklet object to create custom audio processor with JavaScript code.
- Be able to work in a group project relating audio programming concepts and building up from previous knowledge.
- Be aware of best practices in web development in group projects.
- Be able to complete a group project using Web Audio and present it.

Warm-up Activity – Loops: while and for



- The `while` loop: The condition is checked before each iteration.
- The `do...while` loop: The condition is checked after each iteration.
- The `for` loop: The condition is checked before each iteration, additional settings available.
- To make an “infinite” loop, usually the `while(true)` construct is used. Such a loop, just like any other, can be stopped with the `break` directive.

Source: <https://javascript.info/while-for>

ES6 - Classes

- Key elements:
 - Object: A real-time representation of any entity with state (attributes), behavior (how it will act), and identity (unique identifier).
 - Class: A blueprint for creating objects which encapsulates data for the object.
 - Method: A mechanism to facilitate communication between objects.
- Example: A car is an object that has data (model, number of doors, vehicle number) and functionality (accelerate, shift, open doors, turn on headlights).
- Classes can be created using the class keyword in ES6.
- Classes can be included in the code either by declaring them (`class Class_name{}`) or by using class expressions (`var var_name = new Class_name{}`).

Source: https://www.tutorialspoint.com/es6/es6_classes.htm

Web Assembly (Wasm)



- WebAssembly is a standard that defines a binary format and a corresponding assembly-like text format for executables used by web pages.
- The purpose of Wasm is to enable the JavaScript engine of a web browser to execute page scripts nearly as fast as native machine code.

Source: <https://webassembly.org>

A Design Flaw Before AudioWorklets...



- Custom audio processing was supported by `ScriptProcessorNode` in the main thread (and not a separate thread like audio processing in Web Audio API).
- Issues: It happened using event handling (latency!) and code execution in the main thread (glitches from too crowded main thread).

Source: <https://developers.google.com/web/updates/2017/12/audio-worklet>

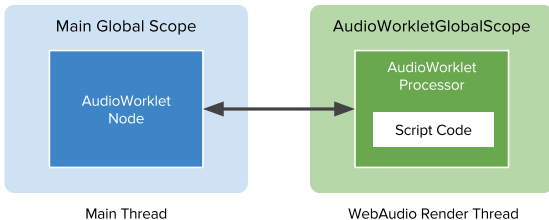
The AudioWorklet object



- The `AudioWorklet` object allows to create a custom audio processor with JavaScript code.
- `AudioWorklet` keeps the user-supplied JavaScript code all within the audio processing thread (zero additional latency and synchronous rendering).
- Live demos: <https://googlechromelabs.github.io/web-audio-samples/audio-worklet/>

Source: <https://developers.google.com/web/updates/2017/12/audio-worklet>

Registration and Instantiation



- Using AudioWorklet consists of two parts: `AudioWorkletProcessor` and `AudioWorkletNode`.
- `AudioWorkletProcessor` represents the actual audio processor written in JavaScript code.
- `AudioWorkletNode` takes care of the connection to and from other `AudioNodes` in the main thread.

Source: <https://developers.google.com/web/updates/2017/12/audio-worklet>

The Future of Web Audio



“Web-based music software is not free from its fair share of skepticism. The web platform is still fighting an uphill battle to compete with native alternatives and in a large way we are pouring considerable effort into solving problems that are already solved: lower latency, less glitches, multithreaded rendering, and so on. We should be willing to accept the fact that the web platform was not designed for the computer music but be optimistic in our belief that we can change the landscape through gradual progress.”

— Hongchan Choi (2018) *AudioWorklet: The Future of Web Audio*. Proceedings of the International Computer Music Conference.

Mini-project development (4/4)



You are expected to create a mini-project in teams that should be doable within a week. The overall aim is to explore a little bit further Web Audio. Here are different approaches that you can take:

- Develop an idea based on what we are seeing in class. Feel free to build up everyday, or change if not convinced (from scratch approach).
- Adapt an existing code to your needs and document what are the changes (remake approach).
- Combine projects from last week (hybrid approach).
- Other?

Working style



- Work with the same team throughout the week, ideally across campuses.
- Make sure to clarify who has developed what part of the code. For example, divide the work into functions and add the author name at the header of each function.
- The instructors in both sites will keep an eye on the groups to catch up.
- There will be 4 time slots during the week to work on the project.
- Keep a research journal.

Assignment 4 Part 3 (10%) - Final Presentation Mini-Project

Each group will have 15 minutes (only questions within this time) to explain the following using slides and live demo:

- **Description:** What is the title of the project and main concept. Overview of the technologies used.
- **Timeline:** Provide an overview of the 3 days that you have been working in the mini-project. What have you been working on?
- **Division of labour:** Explain who has been working in what, how you have documented it, working strategies and technologies used.
- **Live demo:** Try to allocate some time for a live demo.
- **Achievements:** Give a summary of the achievements of this week through the project. Any progress?
- **Challenges:** Give a summary of the challenges that you have been encountering over the week and how you did face them?
- **What is next?** e.g. blog post, code repository, website publishing... more development?

Assignment 5 (10%) - Written Blog Post Mini-Project 2

Deadline: February 22, 2019 at 5pm

- Make sure to include the information requested in your presentation (see previous slide) plus any additional feedback received or follow-up insights.
- Be familiar with the new metadata to be used so that the blog post has author name, date, the right category (“Audio-Programming”), and a thumbnail for the homepage. See the documentation at: <https://github.com/MCT-master/mct-master.github.io/blob/master/README.md>.
- At least you should use a screenshot and an embedded video showcasing the mini-project. It is a plus if you add links to the source code and a live demo link (e.g. directory in a personal webpage).
- Add relevant links in the text to generate traffic.

Final Remarks



- **Post-survey:** please complete by February 22, 2019.
- **Web Audio Conference 4-6 December 2019** at NTNU in Trondheim. Participate (music, artworks, demos, talks...)! We will also need student volunteers!
- Last but not least: **Congratulations and thank you!** God helg!

See you at WAC 2019!

WAC
2019



Web Audio Conference / NTNU, Trondheim
4-6 December 2019

Diversity in Web Audio

#wac2019

<http://ntnu.edu/wac2019>



Relevant Links



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