



Norwegian University of
Science and Technology



MCT4048: Audio Programming

The Fundamentals: Dealing with Time

Anna Xambó

Department of Music, NTNU

6 February 2019

Start setting up...



Download:

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

Go to: `code/d2/00-setting-up/checklist.md`

Warm-up Activity



A round of one sentence each: Tell to the group something that you did learn yesterday during the course.

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

Schedule of presentations



- Alphabetical order?
- February 7, 2019: Ashane, Eigil, Eirik, Guy, Jonas, Jørgen
- February 8, 2019: Karolina, Mari, Sam, Sepehr, Shreejay

Program: Day 2 – 5 February, 2019



- 9.15-10.00: Setting up computers with the tools for the tutorial
- 10.00-12.30: Tutorial: Dealing with time
- 12.30-13.00: Lunch break
- 13.00-15.00: Mini-project 1 development (2/4)
- 15.00-16.00: Speedy presentations mini-project 1 (1/3)

Learning Outcomes



- Get a sense of how to deal with time when using Web Audio and JavaScript.
- Be familiar with scheduling events in Web Audio.
- Be able to create an independent project relating concepts and building up from previous knowledge.

setTimeout(), setInterval()



The browser's provides two built-in methods to deal with time: `setTimeout()` and `setInterval()`. They use the same thread as the rest of the DOM.

Timing Web Audio Events



- Challenge! JavaScript is asynchronous. Events cued to be executed as soon as possible, thus time precision is difficult.
- Solution: Using the internal clock of Web Audio. The web audio clock operates on a separate thread than the rest of the DOM.

`audioContext.currentTime`

BaseAudioContext.currentTime



The `currentTime` read-only property of the `BaseAudioContext` interface returns a double representing an ever-increasing hardware timestamp in seconds that can be used for scheduling audio playback, visualizing timelines, etc. It starts at 0.

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

Changing Audio Parameters Over Time



- The Web Audio API includes a collection of methods for scheduling changes in audio parameter values at present or in the future:

```
setValueAtTime(arg1,arg2)
exponentialRampToValueAtTime(arg1,arg2)
linearRampToValueAtTime(arg1,arg2)
setTargetAtTime(arg1,arg2,arg3)
setValueCurveAtTime(arg1,arg2,arg3)
```

setValueAtTime



The `setValueAtTime` method allows to create an abrupt change of an audio parameter at a future period in time. The first argument is the value the parameter will be changed to, and the second argument is the time that it will take to change to that value.

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

exponentialRampToValueAtTime



The `exponentialRampToValueAtTime` method allows to create a gradual change of the parameter value, and follows a gradual exponential curve.

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

linearRampToValueAtTime



The `linearRampToValueAtTime` method also allows to create a gradual change of the parameter value, but follows a gradual linear curve.

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

setTargetAtTime()



The `setTargetAtTime` method schedules the start of a gradual change that is useful for decay or release portions of ADSR envelopes.

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

setValueCurveAtTime()



The `setValueAtTime` method schedules the parameter's value to change following a curve defined by a list of values. The curve is a linear interpolation between the sequence of values defined in an array of floating-point values.

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

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. Feel free to build up everyday, or change if not convinced.
- 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 OK 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`