Web Audio API & d3js

Gunharth Randolf
Helene Wechselberger

contents

- objectives
- definitions
- html5 audio und the audio web API
- audio transformations
- visualisation
- sources

4 main objectives

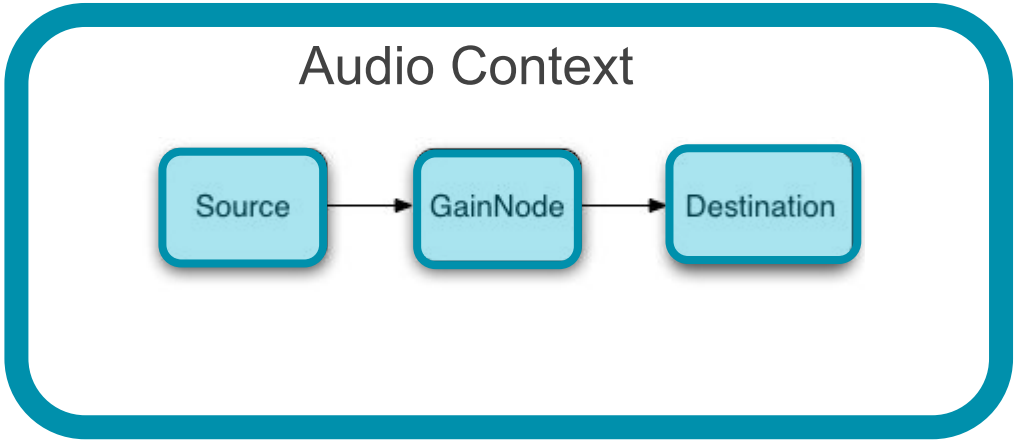
- what ways are there to **create html5 audio elements**?
- on creation, how can we pass them directly into an **audio context** to work with?
- how can we **read out detailed information** about the audio (e.g. frequency)?
- how well does **svg** perform when rendering a visualisation of this information

html5 audio und the web audio API

- Html5 <audio>
  - Native in-browser-support
  - Limited audio processing

- Web Audio API
  - JavaScript built
  - First used 2011/12 in beta stage
  - 2019: still Editor’s Draft

Web Audio API - Audio Context



Definitions

Variables

```
let audioURL; // audio url for html5 audio elements

const audioSource; // source node data of the audio context
```

New audio element

```
let audio = document.createElement( 'audio');

audio.src = audioURL;

$('#' + id).append(audio);
```

New audio context

```
const AudioContext = window.AudioContext || window.webkitAudioContext;

const audioCtx = new AudioContext(); // the defined audio context
```

Load Audio Source ...

… from an existing HTML5 audio element

```
<audio controls src="piano.wav">
```

→ audioURL already set
→ use createMediaElementSource to load audio into an Audio Context:

```
let audioElement = document.querySelector( 'audio');

let audioSource = audioCtx.createMediaElementSource(audioElement);
```

Load Audio Source ...

… by uploading a file through the browser

```
<input type="file" accept="audio/*"

onchange="loadAudioFile(this.files[0]);">
```

Using the JavaScript FileReader:

```
function loadAudioFile(file) {
    let reader = new FileReader();
    reader.onloadend = async function () {
        let arrayBuffer = this.result;
        audioURL = URL.createObjectURL( await new Blob([arrayBuffer]));
        audioSource = await audioCtx.decodeAudioData(arrayBuffer);
    }
    reader.readAsArrayBuffer(file);
```

Load Audio Source ...

... by loading an audio file with Ajax

Using async await with fetch

```
async function loadAudioWithAjax() { // JS Promise
   let arrayBuffer = await (await fetch('guni.ogg')).arrayBuffer();
   audioURL = URL.createObjectURL( new Blob([arrayBuffer]));
   audioSource = await audioCtx.decodeAudioData(arrayBuffer);
}
```

Load Audio Source ...

... by using the MediaStream Recording API

```
async function recordFromMicrophone() {
    let chunks = [];
    let stream = await navigator.mediaDevices.getUserMedia({ audio: true, video: false });
    let mediaRecorder = new MediaRecorder(stream, { mimeType: "audio/webm" }); // opus
    mediaRecorder.start();
    mediaRecorder.ondataavailable = function (e) { chunks.push(e.data); };
    mediaRecorder.onstop = function () {
        let blob = new Blob(chunks, { type: mediaRecorder.mimeType });
        audioURL = URL.createObjectURL(blob); // audioSource from audioURL
    }
}
```

DEMO TIME I

Transform audio

... !!! experiments !!! ...

```
let ctx = new OfflineAudioContext();
```

Audio Nodes used:

- Convolver
- WaveShaper
- BiquadFilter
- Gain
- DynamicsCompressor
- Oscillators
- Google Jungle library

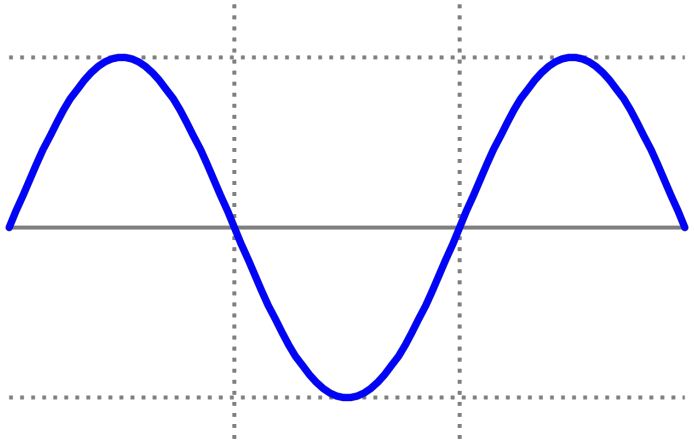
DEMO TIME II

Visualise audio with d3js

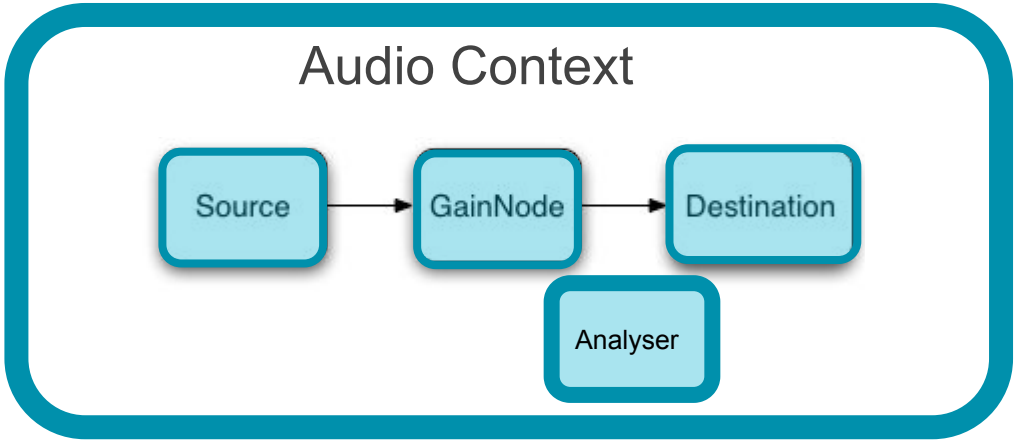
Bar chart == Frequency

?

Line chart == Waveform



Web Audio API - Audio Context



Audio Analyzer Node

```
analyser = audioCtx.createAnalyser();

frequencyData = new Uint8Array(analyser.frequencyBinCount); // 0 to 256

waveformData = new Float32Array(analyser.fftSize); // -1 to 1

audioSrc.connect(analyser);

audioSrc.connect(audioCtx.destination);
```

```
console.log:

Uint8Array(256) [163, 209, 222, 226, 234, 248, 242, 226, 118, …]

Float32Array(256) [0.21924972534179688, 0.22220782935619354, 0.19966179132461548, 0.18443599343299866,

0.19523921608924866, 0.2031596302986145, 0.17519770562648773, 0.12178484350442886, …]
```

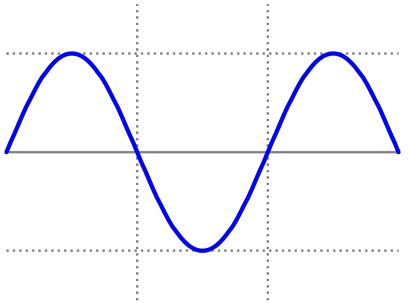
d3js

```
<svg id="svg">
    <g class="frequency"></g>
    <g class="waveform"><path></path></g>
</svg>
```

```
let frequencyGroup = d3.select('.frequency');
let waveformGroup = d3.select('.waveform');
```

```
frequencyGroup.selectAll('rect')
    .data(frequencyData)
    .attr('y', function (d) {
        return svgPathHeight - d;
    })
    .attr('height', function (d) {
        return d;
    })
    .attr('fill', function (d) {
        return 'rgb(' + d + ', 40, 50)';
    });
```

```
waveformGroup.select('path')
    .datum(waveformData)
    .attr('d', waveLine);
```



DEMO TIME III

Project Code & Demo

https://gitlab.web.fh-kufstein.ac.at/gunharth/webaudio-d3js

Name	Last commit	Last update
audio	Code cleanup	2 weeks ago
css	Styles and order of samples	4 hours ago
js	comments	6 minutes ago
.eslintrc.json	eslint	1 week ago
README.md	Styles and order of samples	4 hours ago
index.html	Styles and order of samples	4 hours ago
package-lock.json	Code cleanup	2 weeks ago
package.json	Code cleanup	2 weeks ago

Sources

„Web Audio API“, 2018. [Online]. Verfügbar unter: https://www.w3.org/TR/webaudio/. [Zugegriffen: 21-Jän-2019].

M. Buffa, J. Lebrun, J. Kleimola, O. Larkin, und S. Letz GRAME, „Towards an open Web Audio plugin standard“, in *WWW '18 Companion Proceedings of the The Web Conference 2018*, 2018, S. 759–766.

D. Humphrey, C. Brook, und A. MacDonald, „Exposing audio data to the web: an API and prototype“, in *Proceedings of the 19th international conference on World wide web*, 2010, S. 1365–1368.

„Basic concepts behind Web Audio API | MDN“, 2018. [Online]. Verfügbar unter: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Basic_concepts_behind_Web_Audio_API. [Zugegriffen: 21-Jän-2019].

M. Rouse, „What is HTML5?“, 2014. [Online]. Verfügbar unter: https://searchmicroservices.techtarget.com/definition/HTML5. [Zugegriffen: 22-Jän-2019].

P. Shah, „What all you need to know about HTML5“, 2017. [Online]. Verfügbar unter: https://opensourceforu.com/2017/06/introduction-to-html5/. [Zugegriffen: 22-Jän-2019].

B. Smus, *Web Audio API*. Sebastopol, CA 95472: O'Reilly Media, Inc., 2013.

„Introduction to web APIs“. [Online]. Verfügbar unter: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction. [Zugegriffen: 22-Jän-2019].

„HTML5 Audio“. [Online]. Verfügbar unter: https://www.w3schools.com/HTML/html5_audio.asp. [Zugegriffen: 22-Jän-2019].

B. Smus, „Getting Started with Web Audio API - HTML5 Rocks“, *2011*. [Online]. Verfügbar unter: https://www.html5rocks.com/en/tutorials/webaudio/intro/. [Zugegriffen: 21-Jän-2019].

„FileReader | MDN“. [Online]. Verfügbar unter: https://developer.mozilla.org/en-US/docs/Web/API/FileReader. [Zugegriffen: 22-Jän-2019].

„AnalyserNode | MDN“. [Online]. Verfügbar unter: https://developer.mozilla.org/en-US/docs/Web/API/AnalyserNode. [Zugegriffen: 22-Jän-2019].

„Recording audio in HTML5 Archives“. [Online]. Verfügbar unter: https://addpipe.com/blog/category/recording-audio-in-html5/. [Zugegriffen: 22-Jän-2019].

„Web Audio API“, 2018. [Online]. Verfügbar unter: https://webaudio.github.io/web-audio-api/. [Zugegriffen: 21-Jän-2019].

„D3.js - Data-Driven Documents“. [Online]. Verfügbar unter: https://d3js.org/. [Zugegriffen: 22-Jän-2019].

“Blob” , 2018. [Online]. Verfügbar unter: https://developer.mozilla.org/en-US/docs/Web/API/Blob. [Zugegriffen: 02-Jän-2019].

“MediaStreamRecording API” [Online]. Verfügbar unter: https://developer.mozilla.org/en-US/docs/Web/API/MediaStream_Recording_API. [Zugegriffen: 02-Jän-2019].

“ACM Classification”. [Online]. Verfügbar unter: [Zugegriffen: 02-Jän-2019].

Samples: http://mdn.github.io/voice-change-o-matic/, http://webaudioplayground.appspot.com, https://musiclab.chromeexperiments.com/, https://mozdevs.github.io/MediaRecorder-examples/record-live-audio.html