Web Audio Basics

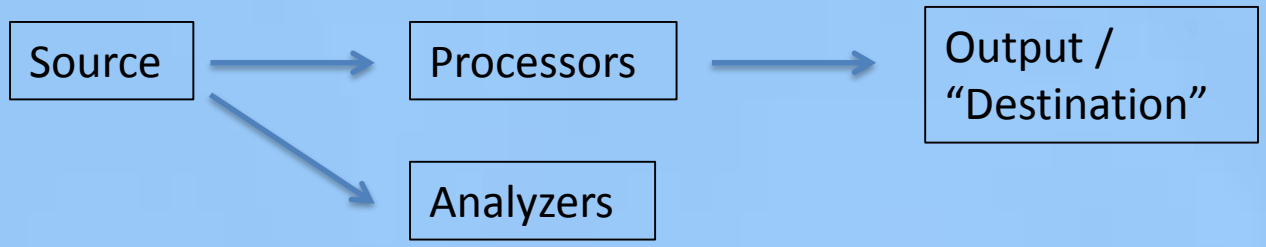
Scott Hawley

Assoc. Prof. of Physics

Belmont University

Web Audio Generally

- Web Audio (WA) is “a high-level JavaScript API for processing and synthesizing audio”
- WA is a “graph-based” programming API for connecting different “nodes”: sources, processors, analyzers, output.
  - Not unlike MAX/MSP



- Except you (generally) can’t program it graphically, instead you write JavaScript to control it. JavaScript is plain text and everything runs client-side so…
- Key Point #1: No way to make your code proprietary
  - Good news is: Lots of free tutorials on the web!

Web Audio History

- Web Audio (WA) was introduced via separate forks of “WebKit” (Chrome, Safari, Firefox)
- Although the flavors have essentially merged, and included in the HTML5 standard, there is still some variability*.
- Key Point #2: Always try out multiple browsers
  - Could be the code you’re trying to use won’t work on your current browser, but might in another browser

*Compatibility summary: http://caniuse.com/#feat=audio-api

Web Audio History p.2

- Also, the API has evolved over time, so some "free" code/demos won't work anymore without (minor) changes.
  - Common changes from old to new:
    - "nodeOn()" -> "start()", "nodeOff()" -> "stop()"
- Key Point #3: Open the Browser's JavaScript/Developer Console to see what the errors are, i.e. to see why it's not working, so you can fix it.
- Handy Tip: Firefox offers a graphical Web Audio editor tool which presents a visual representation (graph) of the node tree. ☺

Features / Demos

- Designed for real-time streaming:
  - Oscillators, Filters, Delay, Compression, Convolution, FFT effects, Panning, Binaural (HRTF) audio
- Great for coding up web demos of things
- Demos:
  - Chromium Demo Page
  - Hawley's Demos
    - DumbDAW, Convolution, Compressor, 3D Player, Quincke Tube
  - http://webaudioapi.com/samples/

Limitations / Weirdness

- WA's AnalyzerNode supports a *maximum* FFT size of 2048.
- Safari on iOS requires user interaction (i.e., screen press) before any audio is enabled

```
var alreadyPlaying = false;
window.addEventListener('touchstart', function() {
        if (!alreadyPlaying) {
                alreadyPlaying = true;
                //start up whatever else you need here
        }
}
```

- Remember Key Point #2: Always try out multiple browsers

```
if ((window.AudioContext === undefined) && (window.webkitAudioContext === undefined)) {
        alert('The Web Audio API is not supported in your browser!');
}
```

References & More

- 2nd Web Audio (world) Conference coming up in early April at Georgia Tech! http://webaudio.gatech.edu/
- Offical W3C WA Page: https://webaudio.github.io/web-audio-api/
- Getting Started:
  - "Getting Started" (2011, some code needs updating): http://www.html5rocks.com/en/tutorials/webaudio/intro/
  - Load/Play user content: http://hedges.belmont.edu/~shawley/webaudio/loadplay.html

Other: Python Audio Analysis App

- "SHAART" Acoustic Tools, open source in Python http://hedges.belmont.edu/~shawley/SHAART/
- General toolkit written for Physics for Audio Engineering class.

Feature list keeps growing:

- Reverberation Time
- Spectrogram
- "Inverse Spectrogram"
- Waterfall Plots
- Convolution
- Equation to Audio File
- Room Modes
- Sabine Equation