Web Audio Visualizer Part III

Creating a Web Audio Node

We going to go ahead and walk you through the process of creating an Audio Effect Node and connecting it to your sound source. You might also want to look back to part I of this exercise for a review of Web Audio concepts.

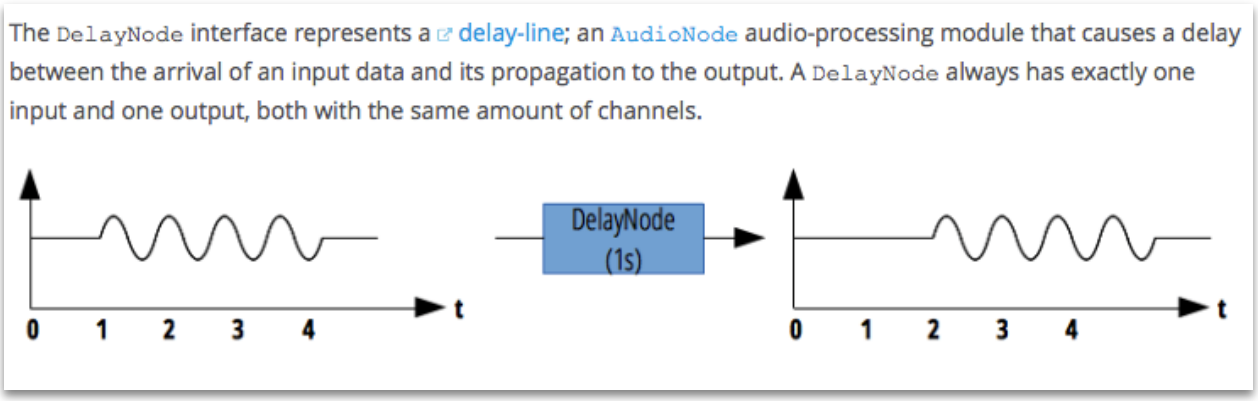
The best docs for the Web Audio API are:

http://webaudio.github.io/web-audio-api/

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

There are also many examples and tutorials - ask the google!

Today we'll look at creating a sound effect filter called a `DelayNode`, which delays the incoming sound signal by a set amount.



Delay lines can be used for many digital effects - see here:
https://ccrma.stanford.edu/realsimple/Delay/Delay_lines.html

Docs for DelayNode:

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

http://webaudio.github.io/web-audio-api/#the-delaynode-interface

http://blog.chrislowis.co.uk/2014/07/23/dub-delay-web-audio-api.html

A) Add the following *context variables*:

```
var delayAmount = 0.5;
var delayNode;
```

B) Add the following to `createWebAudioContextWithAnalyserNode()`

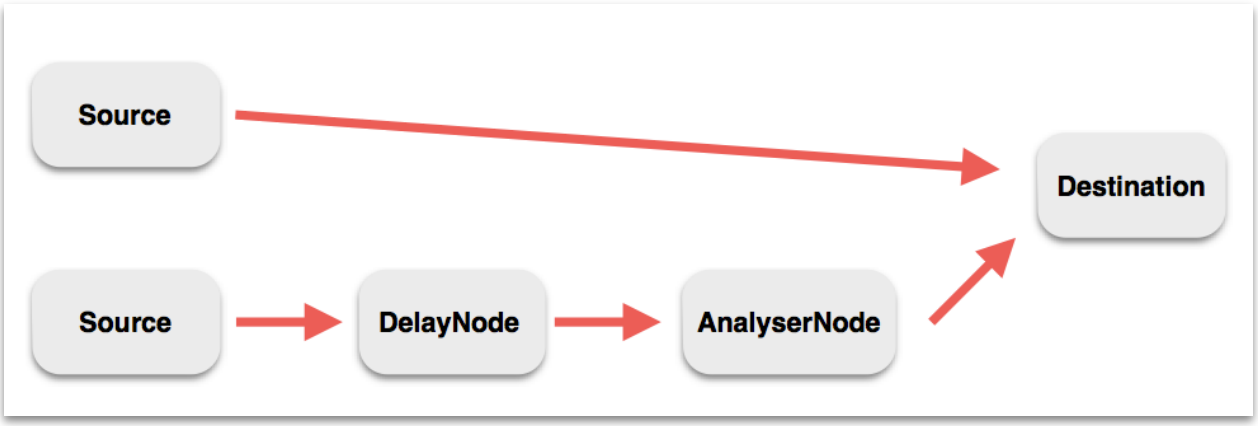
```
// create DelayNode instance
delayNode = audioCtx.createDelay();
delayNode.delayTime.value = delayAmount;
```

C) The delay connection code is a little more complicated than usual and requires two "channels" in the sound graph. Comment out your old "node connection" code, and add the following to the bottom of `createWebAudioContextWithAnalyserNode()`:

```
// connect source node directly to speakers so we can hear the
// unaltered source in this channel
sourceNode.connect(audioCtx.destination);

// this channel will play and visualize the delay
sourceNode.connect(delayNode);
delayNode.connect(analyserNode);
analyserNode.connect(audioCtx.destination);

// Explanation:
// the destination (speakers) will play both channels simultaneously
// if we didn't connect both channels to the destination,
// we wouldn't be able to hear the delay effect
```



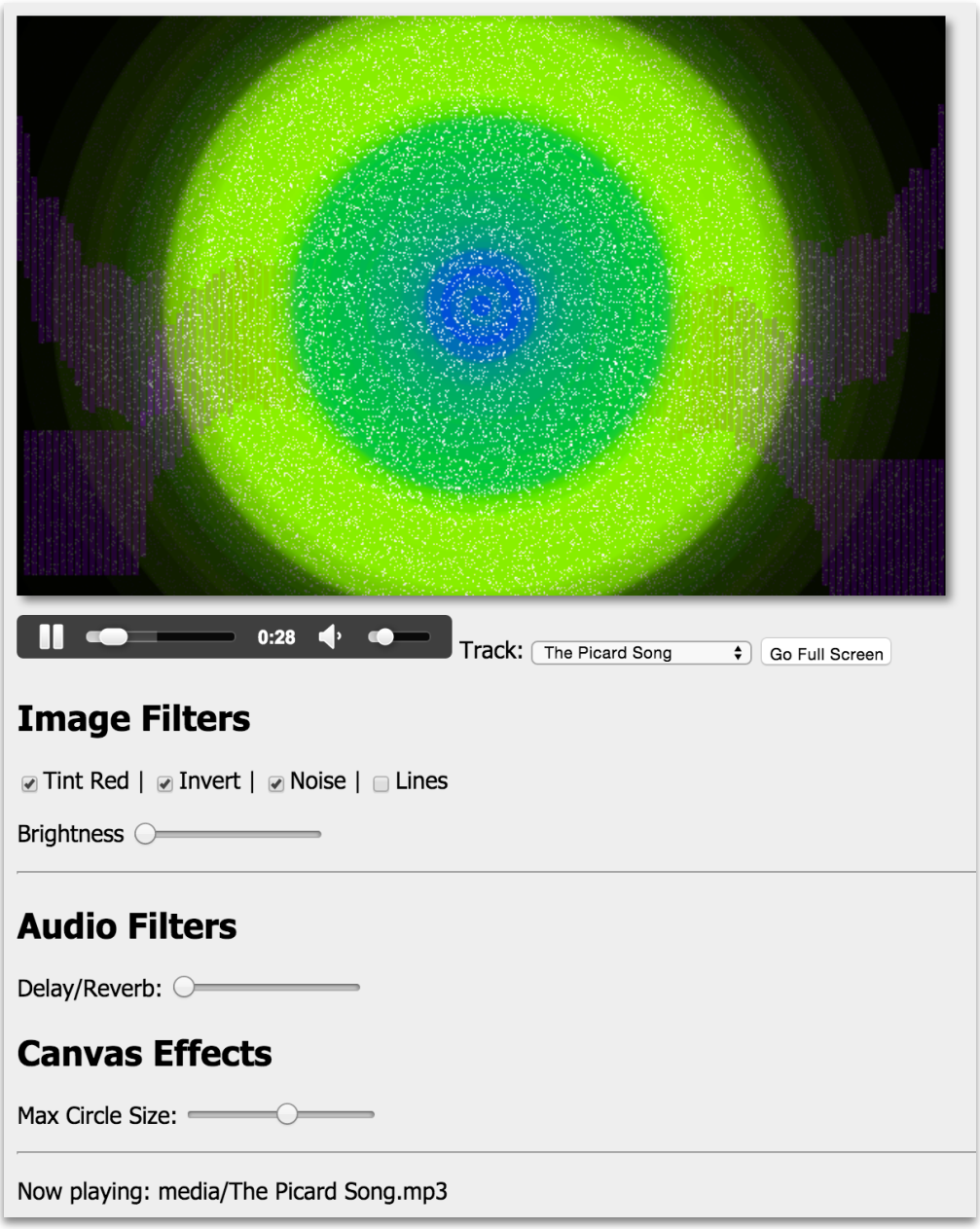
An audio graph for delaying a sound

You should now hear the 0.5 second delay effect. This effect works best with the "Captain Picard Song" track, especially while Picard is speaking.

3) HW and Rubric

- Get the reverb effect working (5 points)

- Add a slider to control the amount of reverb - from 0.0 to 1.0 second `delayAmount` would probably be a good range (5 points)



Hints:

- The HTML for your slider looks like this:

```
<p>
    <label>
        Delay/Reverb:
        <input id="delaySlider" type="range" min="0.0" max="1.0" value="0" step="0.1">
    </label>
</p>
```

- To change the value of the delay audio node after it's been created, you'll need code like this in the `update()` method:

```
delayNode.delayTime.value = delayAmount;
```