Ciaran McKay

Computational Sound: COMS 3430

Blog Post: Crafting Natural Sounds with WebAudio API

In this assignment, recreating the serenity of natural environments holds a unique charm. My latest endeavor involved simulating two distinct yet harmoniously intertwined natural sounds - a babbling brook and gentle rainfall. This project, part of an explorative assignment, pushed my boundaries in utilizing the WebAudio API, a powerful tool for processing and synthesizing audio in web applications.

The Goal

The objective was twofold: first, to generate the continuous, soothing sound of a babbling brook, and second, to overlay this with the intermittent, softer sounds of rainfall. Capturing the essence of these sounds required a blend of creativity and experimentation with different sounds, especially considering the complexity of natural sounds.

Breakdown

Babbling Brook

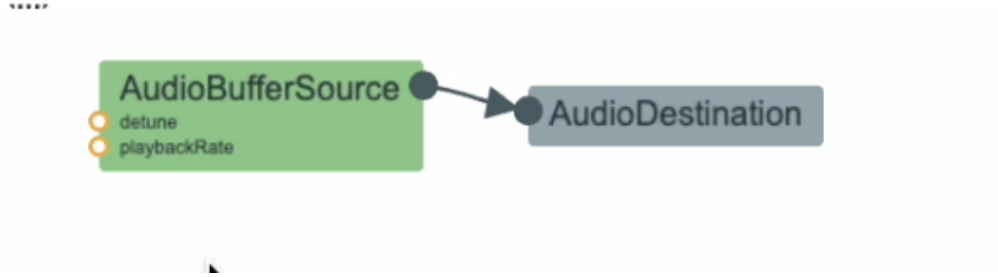
For the brook, I used brown noise generation. Brown noise, or Brownian noise, has a strong emphasis on lower frequencies, providing a deep, rich texture reminiscent of flowing water. The code snippet below illustrates the initial setup:

```
var bufferSize = 10 * audioCtx.sampleRate,
    noiseBuffer = audioCtx.createBuffer(1, bufferSize, audioCtx.sampleRate),
    output = noiseBuffer.getChannelData(0);
var lastOut = 0.0;


for (var i = 0; i < bufferSize; i++) {
    var brown = Math.random() * 2 - 1;
    output[i] = (lastOut + (0.02 * brown)) / 1.02;
    lastOut = output[i];
    output[i] *= 3.5; }
```

This code generates a buffer filled with brown noise, achieving the brook's continuous flow sound.

Diagram from WebAudio visualizer:

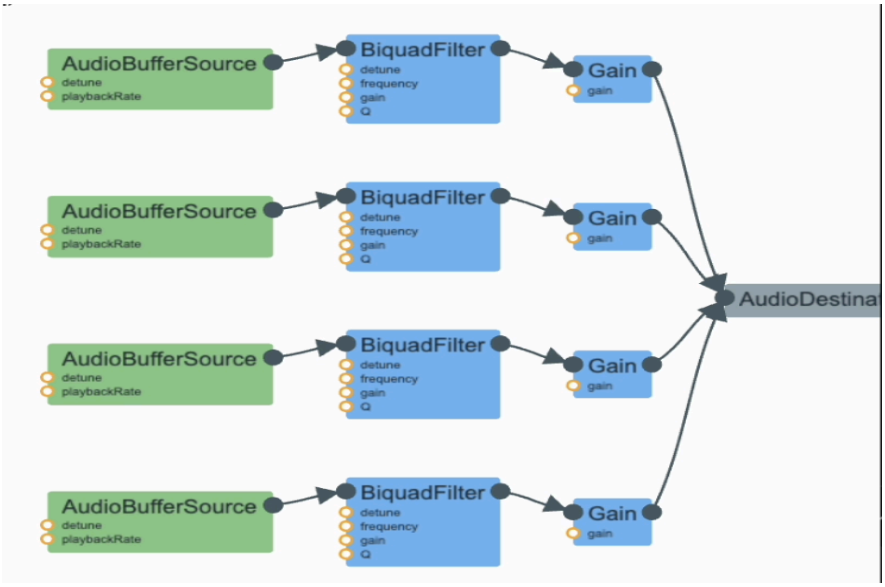


Rainfall

The rainfall effect – inspired from the Farnell book – was created using filtered white noise bursts to simulate raindrops. Each "drop" is a brief burst of noise passed through a bandpass filter, shaping the sound to mimic the impact of a raindrop. This approach allowed for a realistic variance in the raindrop sounds. The key part of the code for generating raindrops is shown below:

```
function createRaindrop() {
    var bufferSize = audioCtx.sampleRate * 0.2;
    var noiseBuffer = audioCtx.createBuffer(1, bufferSize, audioCtx.sampleRate);
    var output = noiseBuffer.getChannelData(0);
    for (var i = 0; i < bufferSize; i++) {
        output[i] = Math.random() * 2 - 1;  }}
```

Diagram from WebAudio visualizer:



The synthesis techniques employed for this project included noise generation and filtering. For the brook, brown noise was the foundation, due to its low-frequency profile. For the rainfall, white noise through a bandpass filter created varied raindrop sounds. These choices were driven by the need to simulate natural sound characteristics accurately.

Overlaying the rain on the brook required careful consideration of volume and density to maintain a balance that reflects a natural setting. Experimenting with different filter frequencies, noise types, and gain levels was crucial in achieving a result that felt both organic and immersive.

This project was a good learning experience in sound synthesis and the capabilities of the WebAudio API. It challenged me to think critically about how to decompose complex natural sounds into simpler, synthesizable components. The choice of synthesis techniques was guided by the inherent qualities of the sounds I aimed to replicate, leading to a deeper understanding of sound properties and how they can be manipulated to recreate the essence of natural environments digitally.

Through this exploration, I not only honed my technical skills but also developed a greater appreciation for the intricacies of sound design. The process was a reminder of the creative possibilities that lie in the intersection of technology and art, encouraging me to continue exploring what more can be achieved.