Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

Розрахунково-графічна робота

з дисципліни “Методи синтезу віртуальної реальності”  
на тему “Spatial audio”

Виконав студент групи ТР-31мп  
Турулько Олександр

Київ 2024

**Постановка задачі**

Основні завданні

* повторно використати код з практичного завдання №2;
* для тих, хто має сертифікат з курсу FPV дронів: реалізувати обертання джерела звуку навколо геометричного центру ділянки поверхні по колу в часі (поверхня при цьому залишається нерухомою, а джерело звуку рухається). Відтворити улюблену пісню у форматі mp3/ogg, при цьому просторове положення джерела звуку контролюється користувачем;
* візуалізувати положення джерела звуку за допомогою сфери;
* додати звуковий фільтр (за допомогою інтерфейсу BiquadFilterNode) відповідно до варіанту (**26 варіант — Шелфовий фільтр високих частот**). Додайте елемент з прапорцем, який вмикає або вимикає фільтр. Налаштуйте параметри фільтра на свій смак.

**Теоретичні матеріали**

Аудіо в Інтернеті до цього часу було досить примітивним і потребувало використання плагінів, таких як Flash та QuickTime. Впровадження аудіоелемента в HTML5 стало значним кроком вперед, оскільки дозволяє відтворювати базове потокове аудіо. Однак він не достатньо потужний для складніших аудіододатків. Для складних веб-ігор або інтерактивних програм потрібно інше рішення. Метою цієї специфікації є впровадження можливостей сучасних ігрових звукових движків та завдань мікшування, обробки та фільтрації, які присутні в сучасних настільних програмах для створення звуку.

WebAudio API – це потужний засіб для обробки звуку, що надається HTML5. Основні компоненти WebAudio API включають:

* **AudioContext**: Контейнер для створення та обробки звукових сигналів.
* **AudioNodes**: Блоки для обробки звуку, такі як джерела, фільтри, компресори тощо. Вони з'єднуються між собою для створення звукового графа.
* **PannerNode**: Використовується для управління просторовим розташуванням звуку.

Просторове аудіо створює враження розташування звукових джерел в тривимірному просторі. Це досягається за допомогою технології WebAudio API, яка дозволяє програмно управляти аудіо контентом на веб-сторінках. За допомогою WebAudio API можна створювати ефекти, маніпулювати звуковими сигналами і відтворювати просторовий звук.

#### **Обертання Джерела Звуку**

Для обертання джерела звуку навколо геометричного центру ділянки поверхні, використовується PannerNode. Джерело звуку рухається по колу з часом, при цьому його позиція оновлюється за допомогою математичних обчислень (синусоїдальних та косинусоїдальних функцій для X та Y координат). Це дозволяє створити враження, що звук рухається навколо користувача.

#### **Фільтри**

WebAudio API також надає можливість застосування різних фільтрів до звукового сигналу. Кожен варіант завдання передбачає використання конкретного типу фільтра:

* **Фільтр низьких частот** (Low-pass filter) – пропускає низькочастотні компоненти звуку і відсікає високочастотні. Він реалізує стандартний резонансний низькочастотний фільтр другого порядку зі спадом 12 дБ/октаву.
* **Фільтр високих частот** (High-pass filter) – пропускає високочастотні компоненти і відсікає низькочастотні. Це резонансний високочастотний фільтр другого порядку зі спадом 12 дБ/октаву.
* **Смуговий фільтр** (Band-pass filter) – пропускає лише частоти в певному діапазоні і відсікає частоти поза ним. Він реалізує смуговий фільтр другого порядку.
* **Шелфовий фільтр низьких частот** (Low-shelf filter) – пропускає всі частоти, додаючи підсилення або послаблення нижніх частот. Це фільтр низького рівня другого порядку.
* **Шелфовий фільтр високих частот** (High-shelf filter) – пропускає всі частоти, додаючи підсилення до високих частот. Це фільтр високої полиці другого порядку.
* **Піковий фільтр** (Peaking filter) – пропускає всі частоти, додаючи підсилення або послаблення до певного діапазону частот.
* **Режекторний фільтр** (Notch filter) – відсікає вузький діапазон частот і пропускає частоти поза ним.

**Аспекти реалізації**

#### 

Реалізовано використання шельфового фільтра високих частот для обробки звукового сигналу в аудіосистемі з використанням WebAudio API. Шельфовий фільтр високих частот дозволяє підсилювати високі частоти вище заданої частоти зрізу, що додає яскравості і чіткості звуку.

В подальшому цей функціонал дозволить покращити якість відтворення аудіо у просторовій аудіосистемі, надаючи можливість регулювати високу частотну характеристику звуку залежно від потреб користувача.

Першим кроком для імплементації аудіо системи є створення HTML-елементу <audio>, який містить інформацію про джерело аудіо-доріжки, в моєму випадку це файл song.mp3. Цей елемент керування дозволяє зупиняти та продовжувати відтворення аудіо-доріжки. Після цього, за допомогою JavaScript, а саме WebAudio API, було створено об'єкт аудіо контексту (AudioContext), для якого було створено та під'єднано три основних об'єкти:

* Джерело звуку (MediaElementSource)
* Об'єкт обробки просторового аудіо (Panner)
* Звуковий фільтр (BiquadFilter)

#### **Реалізація фільтра високих частот**

Для реалізації шельфового фільтра високих частот було обрано тип фільтра highshelf. Наступним кроком було встановлення параметрів обраного фільтра. Загалом, BiquadFilter має три параметри для налаштування:

* frequency (частота зрізу)
* Q (ширина смуги)
* gain (підсилення)

Для шельфового фільтра високих частот:

* frequency - гранична частота, нижче якої частоти послаблюються
* Q - контролює, наскільки піковим буде відгук на частоті зрізу. Велике значення робить відповідь більш піковою
* gain - використовується для підсилення високих частот вище частоти зрізу

Джерело звуку, обробка якого в просторі здійснюється об'єктом класу Panner, зображено у WebGL контексті у вигляді сфери, що може переміщуватись, змінюючи параметри об'єкта Panner, щоб при прослуховуванні аудіо зі стерео звуком був ефект переміщення джерела звуку відповідно до місцезнаходження сфери в системі координат.

**Інструкція користувача**Для налаштування просторового зображення циліндроїди використовуються 4 слайдери, якими можна налаштувати наступні параметри:

* Збіжність
* Відстань між очима
* Кут огляду
* Відстань ближнього відрізання.

Плеєр відтворення аудіозапису та чекбокс для увімкнення/вимкнення звукового фільтру.

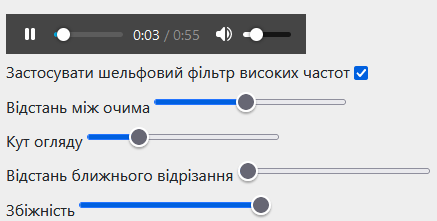
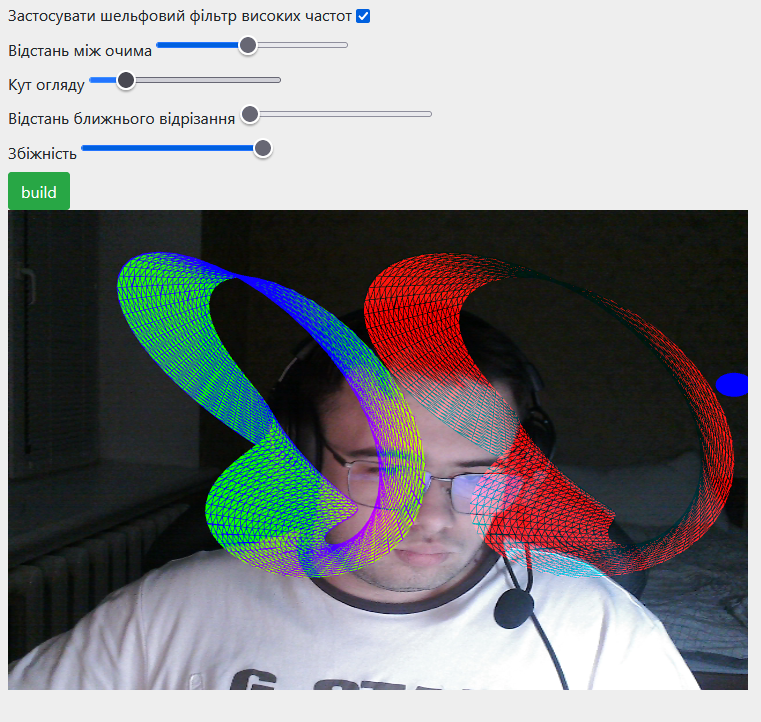
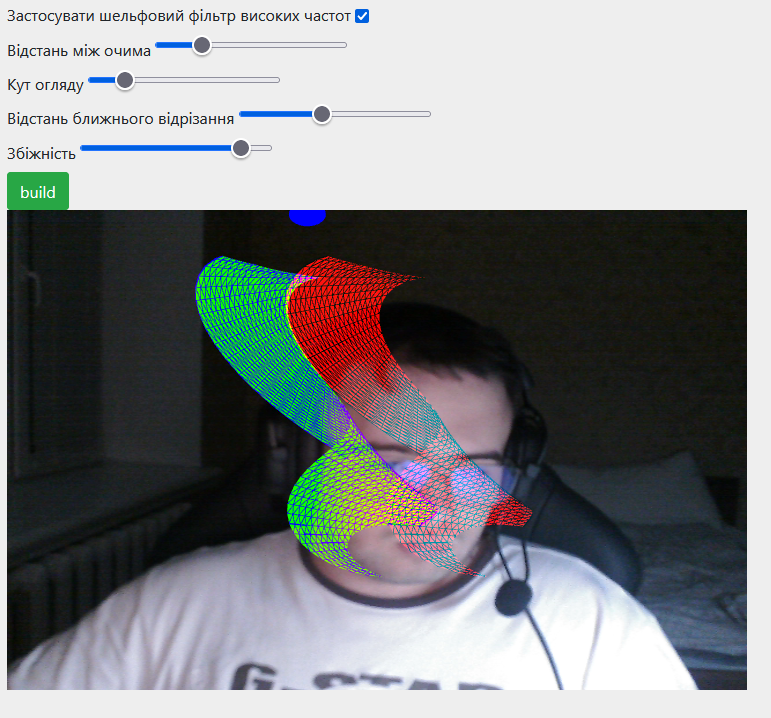


Рисунок 1 - Панелі керування

Приклад зміни даних параметрів:



Рисунки 2 – 3 — Приклад зміни параметрів 3-D відображення

**Код програми:**

let context,

audio,

source,

biquadFilter,

panner;

function setupAudio() {

audio = document.getElementById('audio');

audio.volume = 0.2;

audio.addEventListener('play', () => {

if (!context) {

context = new AudioContext();

source = context.createMediaElementSource(audio);

panner = context.createPanner();

biquadFilter = context.createBiquadFilter();

source.connect(panner);

panner.connect(biquadFilter);

biquadFilter.connect(context.destination);

biquadFilter.type = 'highshelf';

biquadFilter.gain.value = 25;

biquadFilter.frequency.value = 100;

context.resume();

}

});

audio.addEventListener('pause', () => {

context.resume();

});

}

function initAudio() {

setupAudio();

let peakingEnabled = document.getElementById('filterCheckbox');

peakingEnabled.value = true;

peakingEnabled.value = document.getElementById('filterCheckbox').value;

peakingEnabled.addEventListener('change', function() {

if (peakingEnabled.checked) {

panner.disconnect();

panner.connect(biquadFilter);

biquadFilter.connect(context.destination);

} else {

panner.disconnect();

panner.connect(context.destination);

}

});

audio.play();

}