

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»

Кафедра цифрових технологій в енергетиці

Візуалізація графічної та геометричної інформації

Розрахунково графічна робота

Виконав:

Студент 5-го курсу, ІАТЕ

групи ТР-21мп

Потапов Денис Вікторович

Перевірив: Демчишин А.А.

Київ-2023

Теоретичні положення

Web Audio API - це вбудоване в HTML5 програмний інтерфейс, який дозволяє розробникам створювати і маніпулювати аудіоданими веб-додатків. Він надає потужний набір інструментів для синтезу, обробки та маніпуляції аудіо на клієнтському боці.

Web Audio API дозволяє створювати та контролювати аудіо контексти, які включають в себе вхідні та вихідні аудіо вузли. Вхідні вузли приймають аудіо джерела, такі як мікрофон або аудіофайли, а вихідні вузли відтворюють аудіо на пристрої.

За допомогою Web Audio API можна застосовувати різноманітні обробки до аудіо, такі як еквалайзери, гучність, затримки та інші ефекти. Він також надає можливість синтезу аудіо з допомогою вбудованих осциляторів та генераторів шуму, що відкриває безліч можливостей для створення музики та звукових ефектів без використання зовнішніх аудіофайлів.

Web Audio API може бути використаний для реалізації різних застосувань, таких як веб-синтезатори, секвенсори, мікшерні пульті, ігри з звуком та багато іншого. Цей API надає гнучкі можливості для роботи з аудіо на веб-сторінках і дозволяє створювати багатограні звукові ефекти та інтерактивні досвіди для користувачів.

В завданні розрахунково графічної роботи для кожного варіанту потрібно реалізувати відповідний тип фільтру, розглянемо детальніше кожен з них:

1. Фільтр низьких частот (Low-pass filter): Фільтр низьких частот пропускає сигнали нижче певної частоти (називається "частотою відрізання") і пригнічує сигнали вищих частот. Він широко використовується для зменшення шуму або високочастотних спотворень в аудіо сигналах. Фільтр низьких частот використовується, наприклад, для створення ефекту "живої" звукової системи або для виділення низькочастотних басів.

2. Фільтр високих частот (High-pass filter): Фільтр високих частот пропускає сигнали вище певної частоти і пригнічує сигнали нижчих частот. Він використовується для видалення низькочастотного шуму або ефекту "дзижчання" з аудіо сигналів. Фільтр високих частот також може бути використаний для виділення вокальних частот або інших вищих частотних компонентів звуку.
3. Смуговий фільтр (Band-pass filter): Смуговий фільтр пропускає сигнали певному діапазоні частот і пригнічує сигнали за межами цього діапазону. Він використовується для виділення певних діапазонів частот або для пригнічення непотрібних частотних компонентів. Наприклад, смуговий фільтр може бути використаний для виділення голосу зі звукового запису або для видалення шумів в певному частотному діапазоні.
4. Шелфовий фільтр низьких частот (Low-shelf filter): Шелфовий фільтр низьких частот пригнічує або підсилює сигнали нижче певної частоти, але не впливає на сигнали вищих частот. Цей фільтр часто використовується для контролю низьких частот, зміни гучності басів або для створення ефекту "теплішого" звучання.
5. Шелфовий фільтр високих частот (High-shelf filter): Шелфовий фільтр високих частот пригнічує або підсилює сигнали вище певної частоти, але не впливає на сигнали нижчих частот. Цей фільтр може бути використаний для контролю високих частот, зміни гучності високочастотних компонентів або для створення ефекту "яскравого" звучання.
6. Піковий фільтр (Peaking filter): Піковий фільтр дозволяє підсилювати або пригнічувати сигнали в певному діапазоні частот. Він має центральну частоту і регульований діапазон підсилення або пригнічення. Цей фільтр часто використовується для коригування еквайзера, зміни гучності певних діапазонів частот або для створення специфічних звукових ефектів.

7. Режекторний фільтр (Notch filter): Режекторний фільтр, також відомий як фільтр глибокого загасання або фільтр рівнів, пригнічує сигнали в певному діапазоні частот. Він підсилює або пригнічує сигнали поблизу центральної частоти, тоді як сигнали за межами цього діапазону пропускаються майже без змін. Режекторні фільтри використовуються для пригнічення вузьких діапазонів частот, таких як певні непотрібні шуми або перешкоди, не впливаючи на інші частоти.

Хід роботи

В код з попередніх робіт було інтегровано об'єкт аудіоконтексту, що дозволяє отримати доступ до Web Audio API. Далі на html інтерфейс було додано елемент «audio», що дозволяє програвати обрану музику та елемент «checkbox» для керування фільтром (рис. 1).

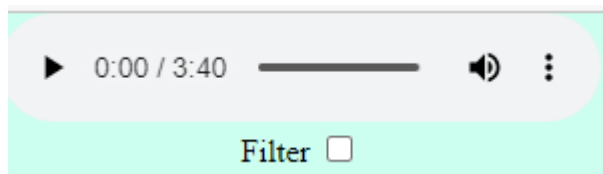


Рисунок 1 – аудіопрогравач та керування фільтром.

Далі в об'єкт «MediaElementSource» було передано елемент «audio» відповідно створений раніше. Наступним кроком було створення «panner» з Web Audio API, він дозволяє проводити просторову роботу зі звуком. Відповідно до контексту даної роботи він використовується для зміни позиції джерела звуку в результаті маніпулювання з матеріальним інтерфейсом (обертання смартфона). Використовуючи інтерфейс BiquadFilterNode було створено об'єкт фільтру і налаштовано (рис. 2)

```

function setupAudio() {

    if (audio) {
        audio.addEventListener('play', () => {
            if (!ctx) {
                box = document.getElementById('boxFilter');
                audio = document.getElementById('audio');
                ctx = new AudioContext();
                source = ctx.createMediaElementSource(audio);
                panner = ctx.createPanner();
                filter = ctx.createBiquadFilter();

                source.connect(panner);
                panner.connect(filter);
                filter.connect(ctx.destination);

                filter.type = 'bandpass';
                filter.frequency.value = 80;
                filter.Q.value = 0.1;

                ctx.resume();
            }
        });

        audio.addEventListener('pause', () => {
            console.log('pause');
            ctx.resume();
        });
    }
}

```

Рисунок 2 – смуговий фільтр та налаштування.

Відповідно до завдання джерело звуку повинно бути реалізовано в вигляді сфери, для цього було створено рівняння для координат сфери і побудовано відповідну поверхню (рис. 3)

```

const sh_x = (r, u, v) => r * Math.sin(u) * Math.cos(v);
const sh_y = (r, u, v) => r * Math.sin(u) * Math.sin(v);
const sh_z = (r, u) => r * Math.cos(u);

function createCordSphere(r, u, v) {
    return { x: sh_x(r, u, v), y: sh_y(r, u, v), z: sh_z(r, u) };
}

function createMusicSurface(R) {
    const i = 0.1;
    let lon = -Math.PI;
    let lat = -Math.PI * 0.5;

    let ArrVer = [];
    while (lon < Math.PI) {
        while (lat < Math.PI * 0.5) {
            let fc = createCordSphere(R, lon, lat);
            let sc = createCordSphere(R, lon + i, lat);
            let tc = createCordSphere(R, lon, lat + i);
            let frc = createCordSphere(R, lon + i, lat + i);
            ArrVer.push(fc.x, fc.y, fc.z);
            ArrVer.push(sc.x, sc.y, sc.z);
            ArrVer.push(tc.x, tc.y, tc.z);
            ArrVer.push(tc.x, tc.y, tc.z);
            ArrVer.push(frc.x, frc.y, frc.z);
            ArrVer.push(sc.x, sc.y, sc.z);
            lat += i;
        }
        lat = -Math.PI * 0.5;
        lon += i;
    }
    return ArrVer;
}

```

Рисунок 3 – створення графічного джерела звуку.

Інструкція користувача

Після запуску програми користувач бачить перед собою інтерфейс, що зображено на рисунку 4

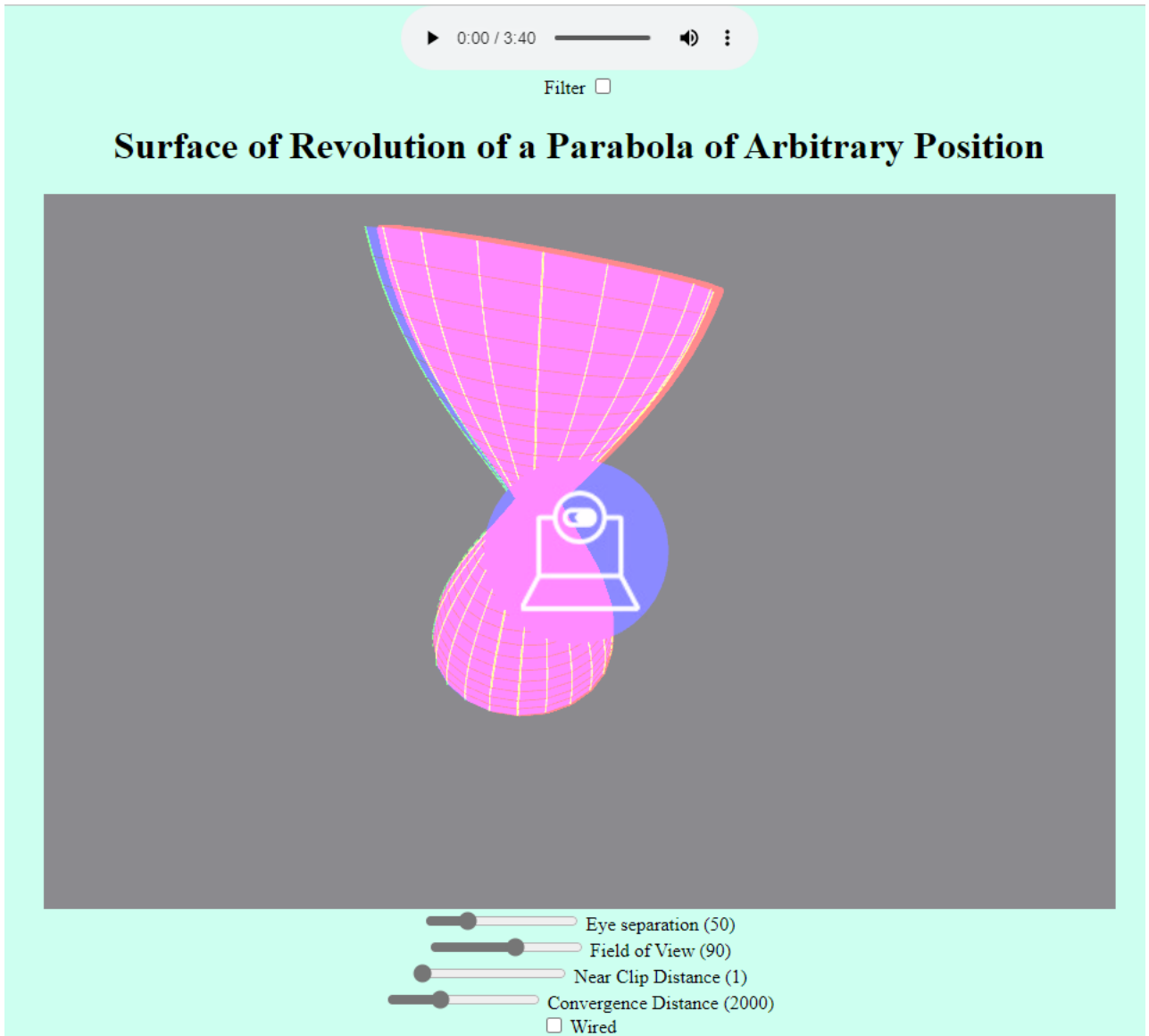


Рисунок 4 – інтерфейс.

Для управління аудіотреком можна використовувати програвач, а також за допомогою прапорця можна увімкнути чи вимкнути звуковий фільтр (рис. 7)

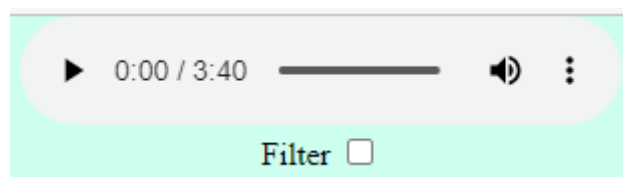


Рисунок 5 – частина інтерфейсу для аудіотреку.

За допомогою мишки користувач може переміщувати фігуру. Такі маніпуляції дозволяють створювати звуковий ефект переміщення звуку. Результати маніпуляцій за допомогою матеріального інтерфейсу продемонстровано на рисунках 6-

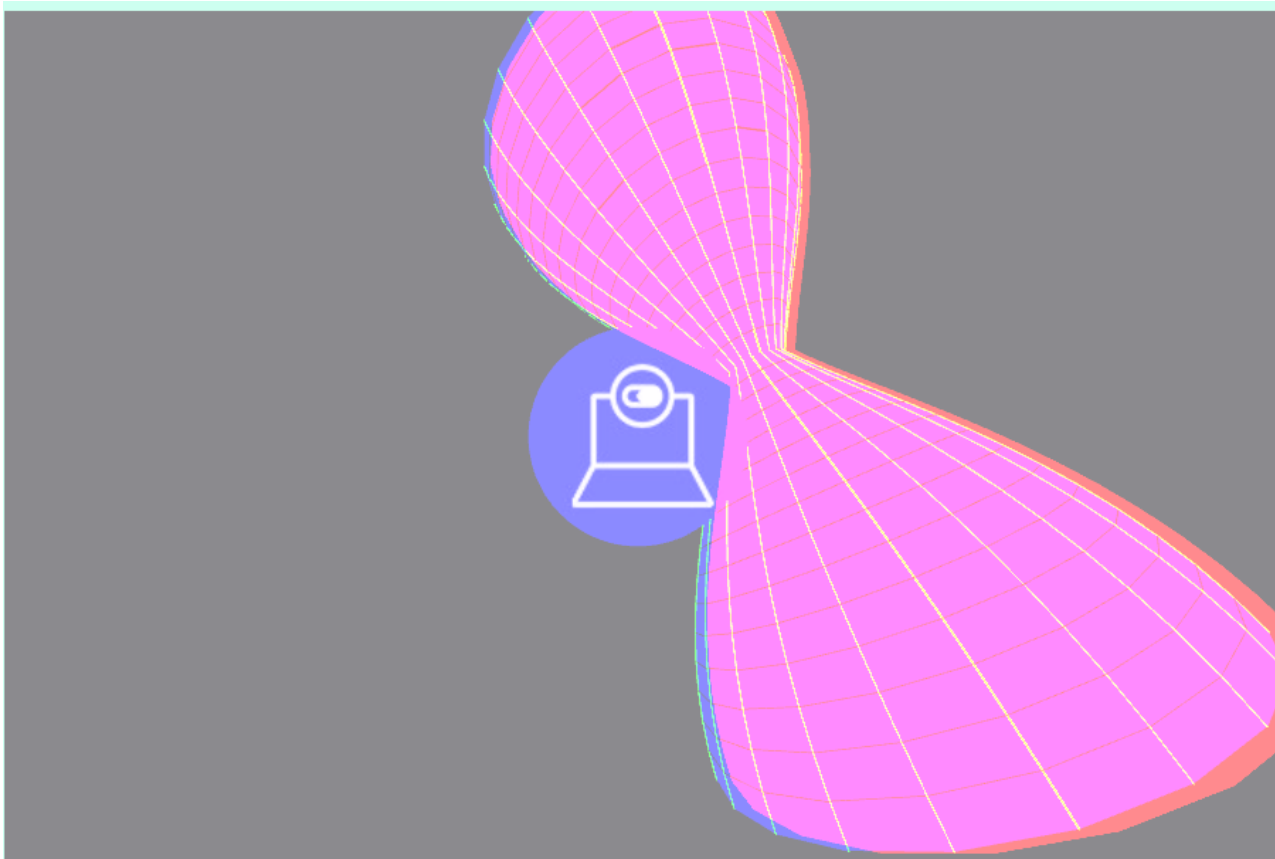


Рисунок 6 – обертання фігури

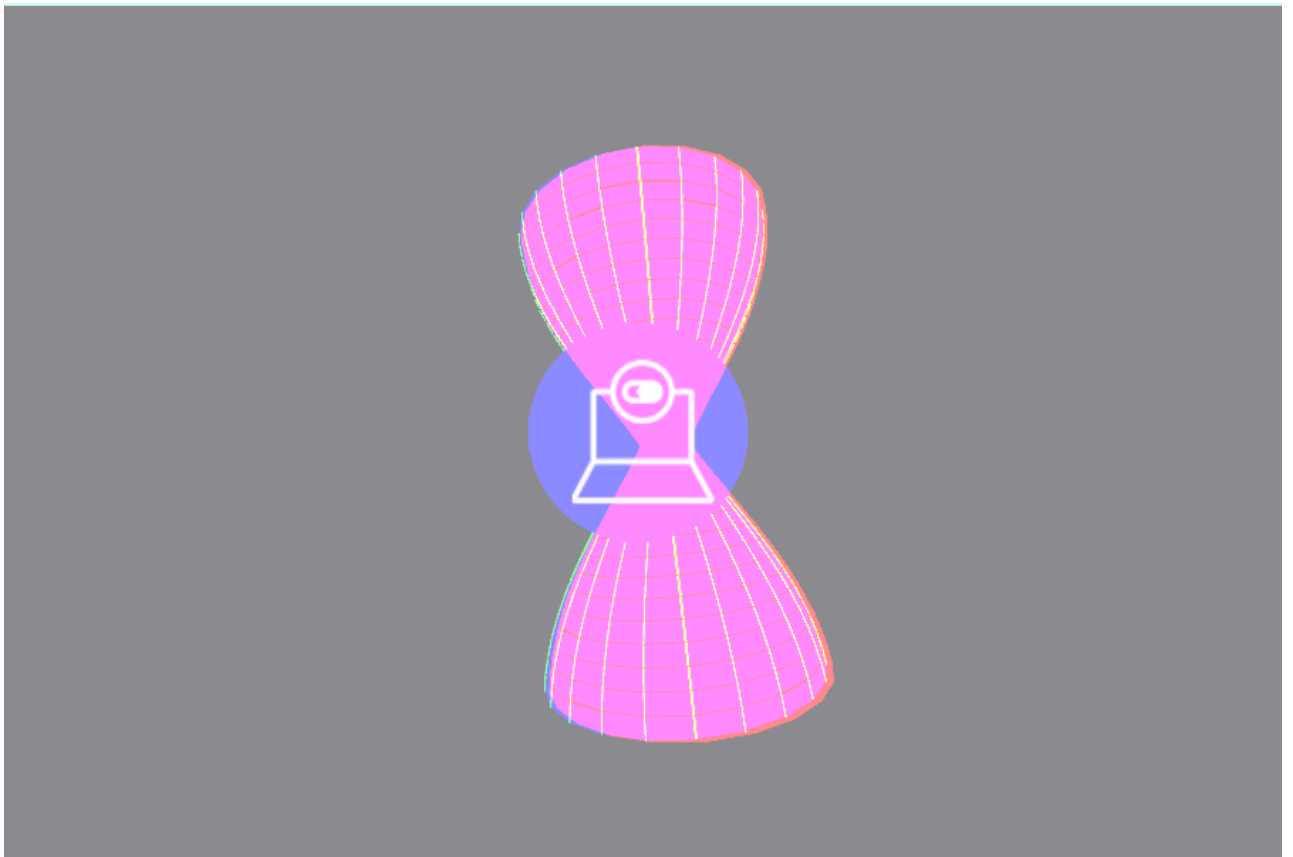


Рисунок 7- обертання фігури

