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

Сосновенко В.В.

**Перевірив:** Демчишин А.А.

**Київ-2023**

1. **Завдання**

Тема роботи: Звук у просторі. Імлементувати звук у просторі за допомогою WebAudio HTML5 API

Вимоги:

* Перевикористати код з практичної роботи №2.
* Імплементувати обертання джерела звуку навколо геометричного центру поверхні за допомогою матеріального інтерфейсу. Програвати улюблену пісню у форматі mp3/ogg, змінюючи розташування джерела звуку відповідно до введення користувача.
* Візуалізувати джерело звуку у вигляді сфери.
* Додати звуковий фільтр за варіантом. Додати «галочку», яка вмикає ти вимикає фільтр. Задати параметри фільтру за смаком.

1. **Теоретичні відомості**

**Web Audio API** надає потужну та універсальну систему для керування аудіо в Інтернеті, дозволяючи розробникам вибирати аудіоджерела, додавати ефекти до аудіо, створювати аудіо візуалізацію, застосовувати просторові ефекти (наприклад, панорамування) та багато іншого. Одним із ключових аспектів API веб-аудіо є його здатність обробляти аудіо та керувати ним за допомогою модульного підходу, який дозволяє створювати складні конвеєри обробки аудіо.

Серед багатьох об’єктів, доступних у API веб-аудіо, три широко використовувані – це AudioContext, MediaElementSourceNode, PannerNode і BiquadFilterNode.

**AudioContext:**

**AudioContext** представляє граф обробки аудіо та діє як центральний центр для створення та підключення аудіо вузлів. Він служить точкою входу для доступу та керування аудіофункціями, які надає Web Audio API. Розробники можуть отримати доступ до різноманітних методів і властивостей для керування відтворенням аудіо, маршрутизацією та ефектами, створивши AudioContext.

Об’єкт AudioContext, який формує основу конвеєра обробки аудіо, ініціалізується у фрагменті коду як «context = new AudioContext();».

**MediaElementSourceNode:**

**MediaElementSourceNode** використовується для отримання аудіоданих із медіа-елементів HTML, таких як <audio> або <video>. Він являє собою джерело аудіо, яке можна зв’язати з іншими аудіовузлами для додаткової обробки чи маршрутизації. Розробники можуть застосовувати різні звукові ефекти або маніпуляції до вже існуючих медіа-елементів та інтегрувати їх в екосистему Web Audio API за допомогою MediaElementSourceNode.

У коді “source = context.createMediaElementSource(audio);” створює MediaElementSourceNode, де змінна audio посилається на елемент HTML <audio>. Це дає змогу використовувати Web Audio API для обробки аудіоданих із зазначеного медіа-елемента.

**PannerNode**:

**PannerNode** відповідає за просторове позиціонування та панорамування звуку. Змінюючи положення, орієнтацію та швидкість джерела звуку у фіктивному 3D-просторі, він імітує 3D-аудіо. За допомогою цього об’єкта програмісти можуть створювати захоплюючий звуковий ефект, який створює враження глибини та руху, коли звук надходить з певних напрямків.

У коді “panner = context.createPanner();” створює PannerNode, який можна підключити до звукового графіка. PannerNode можна використовувати для керування положенням і рухом аудіоджерела, забезпечуючи динамічне розподілення звуку в просторі.

**BiquadFilterNode**:

**BiquadFilterNode** є об'єктом веб-аудіо API, який представляє біквадратний фільтр другого порядку. Це дозволяє застосовувати різноманітні фільтраційні ефекти до аудіосигналів в реальному часі. Біквадратні фільтри є одними з найбільш загальних і корисних типів фільтрів у цифровій обробці сигналів.

У коді “biquadFilter = context.createBiquadFilter();” створює BiquadFilterNode. Після підключення до аудіографа цей вузол можна використовувати для застосування ефектів фільтрації до аудіосигналу, покращуючи або змінюючи його спектральні характеристики.

Підсумовуючи, Web Audio API це потужний інструмент, який дозволяє розробникам маніпулювати та обробляти звук у веб-додатках. AudioContext діє як основний інтерфейс, тоді як такі об’єкти, як MediaElementSourceNode, PannerNode і BiquadFilterNode, пропонують спеціальні функції для вилучення аудіоданих, розміщення звуку у віртуальному просторі та застосування ефектів цифрової фільтрації. Використовуючи ці об’єкти та можливості Web Audio API, розробники можуть створювати захоплюючі та інтерактивні аудіо в Інтернеті. Web Audio API дозволяє застосовувати різноманітні аудіо-ефекти, фільтри, оброблювати та аналізувати аудіо-сигнали, створювати синтезовані звуки, керувати гучністю та панорамою звуку, виконувати запис та відтворення аудіо, а також створювати складні мультимедійні аудіо-додатки.

1. **Деталі імплементації**

В ході другої лабораторної роботи було імплементовано можливість обертати поверхню під назвою «Двічі косий трохоїд циліндроїд» у стерео вигляді за допомогою програмного сенсора смартфона: обертаючи телефон – відповідно оберталась і фігура. Демонстрацію застосування програми можна обачити на рисунках 3.1-3.2.

A hand holding a cell phone

Description automatically generated with low confidence A picture containing text, screenshot, multimedia, communication device

Description automatically generated

Рис.3.1-3.2 Демонстрація роботи програми, створеної в ході другої лабораторної роботи

За допомогою Web Audio API, а саме документації представленої на сторінці https://webaudio.github.io/web-audio-api/ було імплементовано основну частину завдання розрахунково-графічної роботи.

В ході виконання лабораторної роботи необхідно було спочатку створити об’єкт аудіоконтексту, що дозволяє отримати доступ до Web Audio API.

Для виконання роботи було також обрано аудіо-файл формату mp3 і представлено його на веб-сторінці через HTML-елемент <audio>.

Наступним кроком було створити джерело аудіо передавши аудіо-елемент в конструктор.

Також необхідно було створити об’єкт panner в контексті, для подальшої маніпуляції звуком, зокрема позицією, що буде змінюватися по обертанню телефоном(джерело звуку буде знаходитися на умовній відстані 2 від центру в сторону відповідну повороту телефону в просторі).

Важливою частиною завдання було застосувати фільтр до вихідного звуку. За варіантом було імплементовано смуговий фільтр з параметрами вказаними в розділі 5.

Далі потрібно було поєднати об’єкту, передавши відповідні об’єкти іншим.

Було додано eventListener, що відповдає за зупинку та продовження програвання аудіо-файлу.

Крім цього необхідно було створити поле для увімкнення та вимкнення фільтру, а також додати інший eventListener для перемикання фільтра по перемиканню вище вказаного поля.

Оновлення позиції звуку через переміщення об’єкту panner було імплементовані в основній функції під назвою draw.

1. **Вказівки користувачу**

Користувач може керувати переміщенням умовної cфери, що показує користувачу умовне місцезнаходження джерела звуку. Керування відбувається поворотом смартфону.

Відповідне переміщення можна побачити на рисунках 4.1 та 4.2

A screenshot of a phone

Description automatically generated with low confidence A screenshot of a phone

Description automatically generated with medium confidence

Рис.4.1-4.2 Демонстрація керування переміщенням умовної сфери джерела звуку

При обертанні телефону сфера переміщується навколо фігури. З переміщенням сфера створюється ефект переміщення джерела звуку, який найкраще відчувається в навушниках та аудіо стерео системах.

Окрім іншого на сторінці представлено елементи інтерфейсу для зміни параметрів стерео-зображення, а саме значення eye separation, field of view, near clipping distance та convergence. Дані слайдери можна побачити на рисунку 4.3

A picture containing screenshot, text, line, number

Description automatically generated

Рисунок 4.3 Слайдери для зміни параметрів стерео зображення

На сторінці можна побачити елементи управлінням аудіо-файлом: перемотка, пауза, продовження, керування гучністю, див. рисунок 4.4.

A picture containing text, screenshot, font, white

Description automatically generated

Рисунок 4.4 Елемент управління аудіо-файлом

Було також створено елемент «чекбокс» для увімкнення та вимкнення фільтру, див. рисунок 4.5.

A picture containing text, font, screenshot, white

Description automatically generated

Рисунок 4.5 Чекбокс перемикання стану фільтру

1. **Основні фрагменти коду**

Ініціалізація аудіо

A picture containing text, screenshot, software, font

Description automatically generated

A screen shot of a computer program

Description automatically generated with low confidence

Функція draw

A picture containing text, screenshot, software

Description automatically generatedA picture containing text, screenshot

Description automatically generated