Список використаної літератури

1. Зандстра М. PHP. Обьекты, шаблоны и методики программирования / М. Зандстра – М.: Вильямс, 2010 – 480 с.
2. Торп Э. Критерий Келли в блек-джеке, спортивных тотализаторах и на фондовой бирже [Електронний ресурс] / Э. Торп – 1998. – Режим доступу: http://sportbet.w6.ru/programm_knigi/krit_kelli.html
3. Тесла Ю. Прогнозирование футбольных матчей [Електронний ресурс] / Ю. Тесла – 2012. Режим доступу: http://stavkiprognozy.ru/kapperu/matematicheskie-algoritmy-sportivnyx-stavok-stati.html

КОРНОУХОВ В.В., РЕЗНІКОВ С.А.

ДОСЛІДЖЕННЯ ІНСТРУМЕНТУ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ WEB AUDIO API

KORNOUHOV V.V., REZNIKOV S.A.

THE RESEARCH OF WEB AUDIO API INSTRUMENT FOR THE SOFTWARE DEVELOPMENT

The introduction of the audio element in HTML5 is very important, allowing for basic streaming audio playback. But, it is not powerful enough to handle more complex audio applications. For sophisticated web-based games or interactive applications, another solution is required. It is a goal of this specification to include the capabilities found in modern game audio engines as well as some of the mixing, processing, and filtering tasks that are found in modern desktop audio production applications.

For example, the player researched one of the most promising tools in web development. Web Audio API contains a lot of functionality, which is increasingly used by developers. Though of course also has its drawbacks, like there is no universal audio format of good quality that could be used a web application with no hesitations about it.

Keywords:webaudioapi, audio, web, sound, javascript.

Вступ

Інтернет реалізація аудіо була досить примітивною до цього моменту, в основному використовувалися плагіни, такі як Flash та QuickTime. Мета даної специфікації – використовувати можливості, які були реалізовані у сучасних ігрових аудіо двигунах та сучасних десктопних аудіо-підтримуючих додатках.

Це API було розроблено з урахуваннях найрізноманітніших випадків використання аудіо у веб-технологіях. В ідеалі API має підтримувати будь-яку подію пов’язану зі звуком, яка може бути обґрунтовано реалізована оптимізованим C++ двигуном контрольованим з допомогою JavaScript, та відкритим у браузері. Проте, сучасне програмне забезпечення для аудіо може мати дуже розширені можливості, деякі з яких було б важко, або неможливо створити за допомогою цієї системи. Logic Audio від Apple є одним з таких додатків, який має підтримку зовнішніх контролерів MIDI, довільних плагінів звукових ефектів і синтезаторів, високо оптимізованого direct-to-disk зчитування та запису і т.д.. Тим не менш, запропонована система буде цілком здатна

підтримувати широкий спектр досить складних ігор та інтерактивних програм, музичних в тому числі. І це може бути досить гарним доповненням до більш просунутих графічних функцій, що пропонуються WebGL. API розраховане на вдосконалення.

Постановка завдання

На основі інструменту Web Audio API дослідити можливості технології на прикладі програвача.

Основна частина

Web Audio API має багато атрибутів (baseLatency, currentTime, destination, listener, onstatechange, sampleRate, state). Наприклад атрибут baseLatency (тип double) – це фактична затримка обробки AudioContext, а саме – кількість секунд процесу затримки понесені AudioContext у обробці аудіо через граф. Наприклад, якщо аудіо контекст працює на частоті 44,1 кГц і AudioDestinationNode реалізовує подвійну буферизацію внутрішньо і може обробляти і виводити звук кожні 128 семплових кадрів, то затримка обробки буде приблизно *(1)*.

Одним з основоположних понять при роботі з Web Audio Api є саме аудіо контекст.

```
var context = new AudioContext();
```

У одного документа може бути тільки один контекст. Цього цілком достатньо для всього спектра завдань, що вирішується Web Audio API. Наявність одного аудіо контексту дозволяє будувати скільки завгодно складні аудіо графи з необмеженою кількістю джерел і одержувачів звукового сигналу. Практично всі методи і конструктори для створення аудіо модулів є методами аудіо контексту.

Побудова графів. Отже, у будь-якій задуманій схемі може бути один або кілька джерел і одержувачів звукового сигналу, а також модулі для роботи зі звуком. Схема може бути з прямими і зворотними зв'язками, кожен модуль може мати як завгодно багато входів/виходів. Задаємо абстрактну схему, щоб розібратися, як вона будується за допомогою коду (рис 1.).

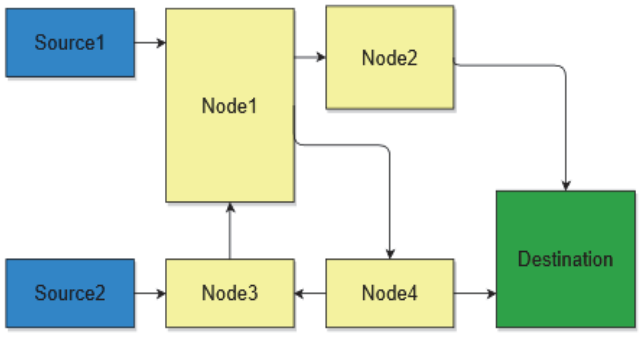

Рис. 1. Схема обробки аудіо

Творці Web Audio API зробили побудову будь-яких графів (схем) витонченим і простим для розуміння. У кожного модуля є метод .connect (...), який приймає один параметр, власне говорить про те, до чого потрібно під'єднатися.

Ось все, що потрібно написати для побудови вищезгаданої схеми:

```
source1.connect(node1);
```

```
source2.connect(node3);
node1.connect(node4);
node1.connect(node2);
node2.connect(destination);
node3.connect(node1);
node4.connect(destination);
node4.connect(node3);
```

Передзавантаження аудіо та відтворення. Приклад роботи з Web Audio API, де джерелом звукового сигналу є буфер, створений з аудіо файлу, передзавантаження за допомогою XMLHttpRequest (AJAX), а одержувачем є системний звуковий вихід.

```
// створюємо аудіо контекст
var context = new window.AudioContext(); //
// переменные для буфера, источника и получателя
var buffer, source, destination;
// функція для підгрузки файлу в буферvar loadSoundFile =
function(url) {
  // робимо XMLHttpRequest (AJAX) на серверvar xhr = new
XMLHttpRequest();xhr.open('GET', url, true);
  xhr.responseType = 'arraybuffer'; xhr.onload = function(e) {
   // декодуємо бінарну відповідь
context.decodeAudioData(this.response,function(decodedArrayBuffer) {
  // отримуємо декодований буфер buffer = decodedArrayBuffer;
  }, function(e) {console.log('Error decoding file', e);    }); };
  xhr.send();}// функція для початку відтворення var play =
function(){// створюємо джерело
source = context.createBufferSource();// підключаємо буфер до
джерела source.buffer = buffer;
 // стандартний отримувач звуку destination = context.destination;
 / підключаємо джерело до отримувача  source.connect(destination);
  // відтворюємо source.start(0);}
 // функція зупинки відтворювання var stop =
function(){source.stop(0);}loadSoundFile('example.mp3');
```

Висновки

На прикладі програвача досліджено один з найперспективніших інструментів у веб-розробці. Web Audio API містить багато функціоналу, який все більше застосовується розробниками. Хоча звичайно й має свої недоліки, наприклад, такі як відсутність універсального аудіо формату, який можна не замислюючись використовувати в web додатках, але все це буде поступово вирішуватися.

Література

1. Web Audio API – новые возможности генерации, обработки и объемного распределения звука в браузере. [Електронний ресурс] // http://html5.by. – 2013. – Режим доступу до ресурсу: http://html5.by/blog/audio/.

2. Web Audio API Editor's Draft [Електронний ресурс] // Audio Working Group. – 2016. – Режим доступу до ресурсу: http://webaudio.github.io/web-audio-api/#introduction.