Play the light of Monet : 웹 기반의 실시간 인터랙티브 오디오-비주얼 네트워킹 시스템 연구

*홍 의 식, *김 준
*동국대학교 영상대학원 멀티미디어학과

Play the light of Monet : Interactive web audio/visual networking system

*Euy Shick Hong, *Jun Kim
*Multimedia department MARTE LAB.
University of Dongguk

● 논문지 구분 : 한국 공학·예술학회 (분야 : 예술공학)

● 키워드 : Web Audio API, Javascript, sound synthesis ,networking

● 교신 저자 : 홍 의 식
주소 : 서울 중구 필동 동국대학교 영상대학원 멀티미디어학과 컴퓨터음악전공
Tel : 010 -8353 -2784

● E-mail : antaresax@dongguk.edu, music@dongguk.edu

Play the light of monet : 웹 기반의 실시간 인터랙티브 오디오-비주얼 네트워킹 시스템 연구

Play the light of monet : Interactive web audio/visual networking system

Abstract

Web technologies have expeditiously developed by programming developer and web company such as some Web applications for audio processing, synthesizing and node.js server platform by JavaScript. Especially, JavaScript has a possibility of incredible and flexible programming language with interactive factors of dynamic action on the Web. This project aims to find interactive networking collaborate system through web technologies with JavaScript in real-time that to control musical expression and various multimedia factors by Web-based musical controller in mobile or PC through node.js server for artistic interactive works.

I. INTRODUCTION

In last few years, web technologies have developed by programming developer and web company such as a number of Web-Audio API[1] [2] [9] [11] applications and node.js of web-server platform by JavaScript. Potential capabilities with HTML5, CSS, JavaScript [8] of the Web Audio API can express various artistic interactive works with multimedia factors for audio processing, synthesizing, visualization and controlling media on the web applications. In this paper, this project aims to find interactive networking collaboration system through web programming technologies with JavaScript that to control musical expression and various multimedia factors. Also, each player can talk about playing and controlling to main audio/visual installation values by web-based API controller at each different place such as mobile SNS message chat. Beside we describe regarding Web audio/visual systems and real-time server technologies with node.js in JavaScript. With advanced web technology, we have researched various interactive networking system and real-time collaboration works that we enable to control web-browser with computer or smartphone and tablet PC in another place such as mobile-SNS system, so this project has suggested following systems.

---

1) www.w3.org/TR/webaudio

- to connect at web-browser with a portable device or computer through WI-FI or data easily in each different place
- to build a web server with node.js in JavaScript-based for controlling audio/visual/text chat on real-time communication.
- to develop a Web-based UI controller for playing.

This project aims to connect web applications easily and to play the music with another player in real-time. In addition, created web applications can easily control that real-time audio processing, visualization, sound effects, and media installation work with artistic emotions. We present interactive web audio/visual networking systems[2] through node.js technologies in JavaScript for collaboration playing[3]. In this paper, we have sought to find interactive networking system through web-based programming technologies for artistic expression and various multimedia works.

II. RELATED WORKS

There are some web technologies in JavaScript in the web browser. Gibberish.js is a strongly optimized audio processing library and provide a complete system for musical Synthesizer instrument that can be used in PC and smartphone. Also, Interface.js enables control of wireless synthesis applications to trough a server application by web interfaces that works with mouse, touch, and gesture [15]. Other libraries present different approaches. For example, JavaScript library expanded WAAX [14] [10] was created to enable audio programming based with Web Audio API augmenting useful features that WAAX has offered a music framework for web-based application and presented to enable and support the development of various web-based works: fast and less coding and Modular and extensible. Such as works, Audiolib.js [6] was A powerful audio processing tools library with JavaScript for web-based frameworks. It enables to set some sound effects and provides to build the entire audio system comfortable for both users for musical expression. Tuna.js[1]is also An audio effects library for web audio. It will be using it to create its sound effects and can easily connect another effect by Web Audio API node. Also, WebGL[12] is developed for visual on the web. WebGL is a cross-platform for a 3D graphics API based on OpenGL ES 2.0, exposed through the web Canvas elements. Another example of extree.js[3] is JavaScript 3d library that aim of the project is to create an easy using, lightweight, 3D library.also
The library offers <svg>, <canvas>, CSS3D.

---

2) https://github.com/drsax/monet
3) https://youtu.be/uZLOY8onwz4

III. PLAY THE LIGHT OF MONET

Monet is an impressionist painter born in France. He rapidly used to paint a light of nature in outside by extempore emotion. Because the light of view rapidly disappears. Extempore factors of Monet are similar the improvisation of jazz. JAZZ band has various instrument player and they have affected each other to communicate. So we plan to make that interactive networking system to control audio and visual by Web Audio API controller.

3.1 installation structure flow

With web technology, we have considered an interactive networking system by using novel portable devices such as smartphones and tablets. That device can easily access web-browser at each different place through WI-FI or 3G-4G data. Moreover, the accessed player can control audio and visual in web-browser and communicate text chat on portable devices in real-time. The main screen is designed on the Chrome web-browser that can show on TV or Beam project screen. Also, music is played on the main web-browser. Figure 1 is shown networking system flow

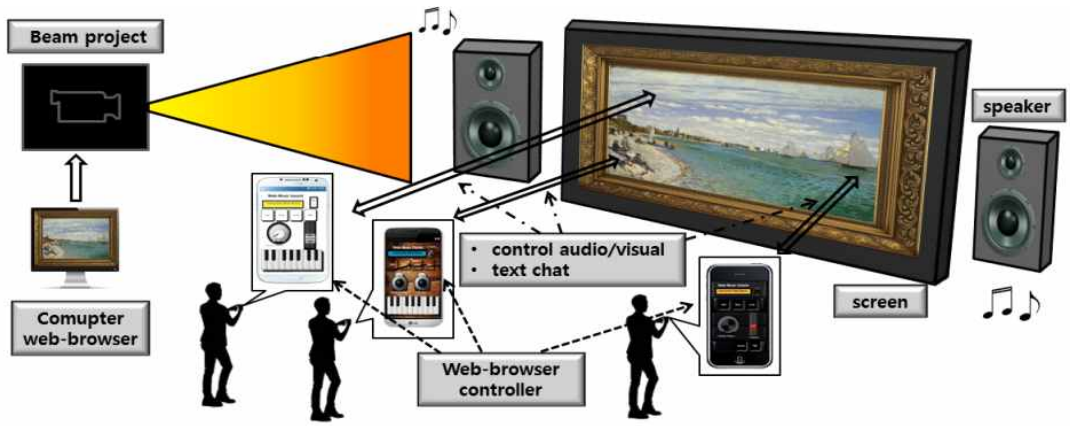

Figure 1: Play the light of Monet : concept a Interactive networking ensemble system

3.2 main screen

We have designed main visual on the web browser like a frame in an art gallery. Pictures(ship, clouds, and waves) in the frame is naturally moving. developed web-based controllers can control brightness, gray, position of four pictures on the main browser in realtime.

3.3 controller design

We have designed four controllers of each different style on the mobile-web such as real instruments. Each controller has various features. However, generally, all interfaces have similar functions that sound on/off, text chatting, sound/visual processing, volume control, sample sound with buttons, dials, slider, piano keyboard. Main text monitor and on/off switch was set alike.

Figure 2: Web audio instrument controller of various types.

IV. INTERACTIVE NETWORK SYSTEM

Generally, in web networking, most of a client requests some data to server and server return requested data to the client. but almost server system is not in real time. so, we have used socket.io module in node.js with JavaScript. It can communicate in real-time. Also, we can program server with various data mapping through JavaScript

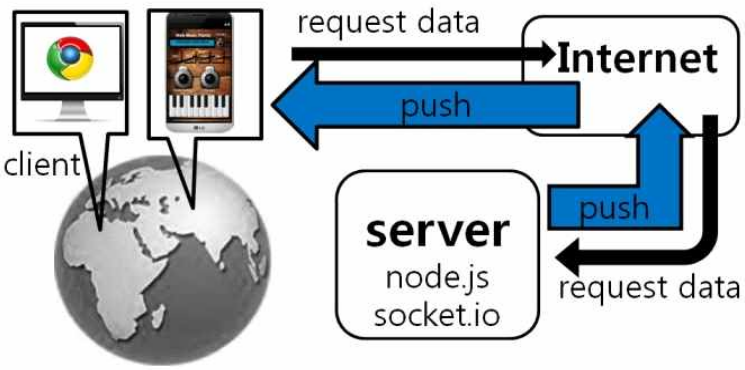

Figure 3: Interactive networking system flow.

4.1 node.js server

With the development of web technologies, two-way response system has developed between the web server and client. The system can control interactive media works on the web in real-time. WebSocket API [13] can communicate two-way. However, occasionally it does not work all browser. So we use Node.js[4] server with Socket.IO [7] server [5] module that works regardless of various web browser.
Socket.IO is a module that can communicate two-way on web API application on JavaScript such as text chat system on mobile in real time, rapidly.

```
1 var express = require('express');
2 var app     = express();
3 var http    = require('http').Server(app);
4 var io      = require('socket.io')(http);
5 var path    = require('path');
6 app.use(express.static(
  path.join(__dirname,"public")));
7 var port = process.env.PORT || 8358;
8 http.listen(port, function(){
9 console.log("Dr.Hong server on!"); });
10 var objects = {};
11 var count=1;
12 io.on('connection', function(socket){
13 var name = count++;
14 console.log('user connected: ',name);
15 io.emit('change name',name);
16 socket.on('1', function(str){
17 console.log(str);
18 io.emit('1',str);
19 });
```

Listing 1: Using Socket.io and Node.js

4.2 socket.io

For accessing to web API at various place, we have connected to the node.js server. In Listing 1, we have presented Socket.IO at line 4. From line 12 to 19, it presents to communicate with the node.js server and web client. At line 16, server received data from a web browser and return immediately by line 19. It enables interactive networking in real-time between server and client. Socket.io is a module for an interactive network in real-time on web-browser.

V. PROGAMMING

5.1 sound generation

In this project, we have designed two sound part : edited sound file and audio synthesis on the Web Audio API.

```
1 audio4 = new Audio();
2 audio4.src = "drum.mp3";
3 audio4.playbackRate = 1;
4 audio4.volume=0.5;
5 audioContext = new AudioContext();
6 amp = audioContext.createGain();
7 amp2 = audioContext.createGain();
8 amp3 = audioContext.createGain();
9 lowpass =audioContext.createBiquadFilter();
```

Listing 2: Sound units

From line 1 to 4, the code loads 'drum.mp3' sound file and sets a playing speed and volume. Edited four sound files and audio synthesis was controlled by web-controller in real-time. From line 5, it shows a coding regarding sound processing and sound synthesis.

5.2 visual composition

2D and 3D visual technologies in web-browser were developed with JavaScript. For this project, we have divided four part that cloud, a wave of sea, beach, and ships. Each part was controlled it owns color of brightness or position on the canvas by a web-interface controller. From line 1, it is the main image file. On the main image file, two cloud image file is located with visualizations. From line 6, some ships are moving by a sea of wave and sound processing. Line 8, it shows the main frame of a picture.

```
1 <img id="img" src="monet.png"    style="width:1430px height:800px;">
2 <img id="img1" src="monet1.png" style="width:1178px;height:330px;">
3 <img id="cloud" src="cloud.png" style="width:678px; height:130px;">
4 <img id="img3" src="monet3.png" style="width:1128px;height:336px;">
5 <img id="img4" src="monet4.png" style="width:1478px;height:257px;">
6 <img id="boat" src="boat.png"    style="width:160px;  height:80px;">
7 <img id="boat1" src="boat1.png" style="width:150px;  height:70px;">
8 <img id="frame" src="frame.png" style="width:145px; height:732px;" >
```

Listing 3: UI code of controller

These pictures were controlled during sound processing by interface player in real-time.

5.3 web-based controller

We have designed web audio controllers of each different type on the mobile. Listing 4 shows From line 1 to 4, the code sets various button, dial, slider, switch to control a

sound processing and visual. These web audio instrument controllers have functions that sound file on/off, sound processing, visual controlling, text chatting, volume control, sample sound with buttons, dials, slider, piano keyboard.

```
1 <button class="button1"type="button"id="mute" style="position:absolute;left:230px;
                                                    top:140px"value="0" max= "1" >mute<|button>
2 <webaudio-knob id="monet2"src="cont/img/red.png" style="position:absolute;left:40px;top:250px"
                              value="0" max="2000" step="1" diameter="150" sprites="30"><|webaudio-knob>
3 <webaudio-slider id="monet22"  style="position:absolute;left:250px;top:230px"
     src="s1.png"   knobsrc="snn.png"  value="50" min="0" max="200" step="1"width="50"height="170"
  basewidth="40"baseheight="170"knobwidth="40" knobheight="40"ditchLength="100"> <|webaudio-slider>
4 <webaudio-switch id="switch3"  style="position:absolute;left:260px;top:50px"
                         src="cont/img/onoff.png" value="0" height="80"  width="80"><|webaudio-switch>
```

Listing 4: Main visual design

VI. PERFORMANCE

This project has suggested for Interactive networking system for collaborative works in real-time such as musical ensembles: Jazz quartet. each player can communicate through other player's playing about audio/visual in realtime at same space or different place.

Figure 4: Controlling with web audio instruments

Also, instrument interface of player is a portable smartphone or tablet PC, of course, a desktop is possible. so the player can easily access to the web browser through WI-FI or 3G, 4G data.

6.1 audio processing

On the web-interface controller, sound on/off was controlled by 'start/stop' buttons. Also, a sound can be muted by 'mute' button. Sound button controls sound processing and visualization on the main picture simultaneously.

6.1.1 edited and sample sound

This interface can control the main melody that is flute and organ, Moreover, sample sound such as waves sound of the sea. 'start' button can load all sound in same time and 'stop' button is to stop all playing sound and to reset sound time. However, 'mute' button is muted it owns sound of an interface. Also, 'sound' button can play sample sounds such as waves and gulls.

6.1.2 sound processing

- Timbre : the loaded sound file was processed with a low-pass filter. 'Q' value and 'cutoff' value of the filter was set '0.4', '20'. the frequency of filter value was set from 0 to 10000Hz. It was controlled by dial of controller in real-time.

- Volume : volume was controlled by dial or slider. gently It needs for balancing each sound playing from 0.0 to 1.0.

- Keyboard : web-music pianist and web-music bassist only have a keyboard for playing a melody with sound synthesizer but a range of keyboard about one octave, so it can't possible to play dynamic. web-music pianist was set from c3-d5 and web-music bassist was set from c3-d5. It enables to play additional on main sound.

6.2 visual controlling

Designed four controllers of each different style can control by a dial, slider, and button in real-time.

- Clouds : we can change a color of brightness and gray by instrument interface that web-music soloist: dial of timbre and volume. Brightness value was set from 0 to 200, and a gray value was set from 0 to 1 by mapped data. Gray data is changed a little by the volume control and changed view is only black and white. However, a range of brightness value is wide that it is effective various sound processing.
- Sea : generally, as a brightness of sea is changed by the sunshine, timbre dial of

web-music pianist can control the brightness of sea on the main picture of Monet. Brightness value was set from 0 to 200. Gray value of sea is changed by volume dial from 0 to 1.

- Beach : brightness and gray of beach color was changed by timbre and volume dial of web-music bassist that range is from 0 to 200, and from 0 to 1.
- Ships : web-music drummer's dial and volume can control the brightness and gay : from 0 to 200 and from 0 to 1. Also, 'sound' button can change a position of the ship that to repeat on the sea.

6.3 text chatting

Players can talk with connected another player in different place in real-time by 'Talk' button.Send texts to display the main picture and each instrument interface. This chatting system helps to monitor about playing and visualization.

VII. CONCLUSIONS

Developing technologies of the Web Audio API have some potential capabilities with HTML5, CSS, JavaScript and various web libraries for audio processing and synthesizing. Especially, JavaScript has possibility of incredible and flexible programming language such as interactive factors of dynamic action in HTML. We have approached interactive web-audio/visual networking system that to made a web audio controller on mobile or tablet computer and to control an networking data with a web server through node.js. In this paper, we have tried to find interactive networking system through web-based technologies with JavaScript for artistic expression and various multimedia works.

References

[1] An audio effects library for web audio. https://github.com/Theodeus/tuna.

[2] C. rogers, web audio api-w3c working draft. https://www.w3.org/TR/webaudio.

[3] Javascript 3d library. https://github.com/mrdoob/three.js/.

[4] Node.js. https://nodejs.org.

[5] npm -node.js realtime framework server. https://www.npmjs.com/package/socket.io.

[6] A audio tools library for javascript. https://github.com/jussi-kalliokoski/audiolib.js.

[7] socket.io. http://socket.io.

[8] tutorial of javascript,html, css. https://www.developphp.com/.

[9] Web audio api - mozillawiki. http://wiki.mozilla.org.

[10] Web audio api extension. https://github.com/hoch/waax.

[11] Web audio tutorials -middle ear.
http://middleearmedia.com/category/web-audiotutorials/.

[12] Webgl - opengl es 2.0 for the web. https://www.khronos.org/webgl/.

[13] The websocket api. https://www.w3.org/TR/2011/WD-websockets-

[14] H. Choi and J. Berger. Waax: Web audio api extension. in Proceedings of the 13th Conference on New Interfaces for Musical Expression(NIME-13), Daejeon, Korea.

[15] C. Roberts, G. Wakefield, and M. Wright. The web browser as synthesizer and interface. in Proceedings of the 13th Conference on New Interfaces for Musical Expression(NIME-13), Daejeon, Korea,, 2013.