

DrWebusic : 실시간 인터랙티브 Web 오디오 시스템

홍 의 식

Institute for Computer Music and Technology

DrWebusic : Real-time Interactive Audio System on the Web

Euy Shick Hong

03113 Institute for Computer Music and Technology, 54, Jong-ro 63-gil, Jongno-gu, Seoul, Korea

[요 약]

과학 기술의 발전은 멀티미디어 콘텐츠와 영상, 음악 등을 활용한 실시간 인터랙션 공연을 가능케 하였다. 이러한 멀티미디어 공연은 탈장르 융복합 예술로 관객에게 다양한 볼거리를 제공하고 관객이 작품에 참여하여 실시간 인터랙션을 통해 공연을 이끌어가기도 한다. 또한 미디어와 영상, 음악을 실시간으로 제어하기 위해 컴퓨터와 모바일 기반의 설치형 애플리케이션을 사용한다. 하지만 설치형 애플리케이션은 접근성과 호환성의 효율이 좋지 않아 공연 시스템 활용에 제약이 있다.

본 논문에서는 이를 해결하기 위해 접근성이 좋은 Web 기술을 활용하여 인터랙티브 멀티미디어 시스템을 개발하였다. HTML5의 API와 오디오 라이브러리 DrSax.js를 사용해 오디오시스템을 구현하였고 실시간 인터랙션을 위해 비동기 방식인 Node.js로 서버를 구축하였다. 또한 실시간 인터랙션을 통한 데이터 제어를 위해 모바일이 제공하는 센서를 사용하였다. 이러한 특징을 통해 구현된 Web 오디오 애플리케이션의 성능을 분석하고 본 연구의 효율성 알아보하고자한다.

[Abstract]

The development of science and technology has enabled real-time interactive performances using multimedia contents, video and music. These multimedia performances provide various kinds of spectacle to the audience with the genre fusion arts, and the audience participate in the works and lead the performances by realtime interaction.

It also uses computers and mobile-based, standalone applications to control media, video, and music in real time. However, installable applications are not efficient in accessibility and compatibility, which limits the use of performance systems. In this paper, we developed an interactive multimedia system using web technology with high accessibility. The audio system was implemented using HTML5 API and audio library DrSax.js, and the server was built with Node.js, an asynchronous method for real-time interaction. We also used mobile sensors to control data through real-time interaction. We analyze the performance of web audio application implemented through these features and investigate the efficiency of this study.

색인어 : DrWebusic, Web 애플리케이션, Web Audio API, 멀티미디어 공연, DrSax.js

Key word : DrWebusic, Web Application, Web Audio API, Multimedia Performance, DrSax.js

<http://dx.doi.org/10.9728/dcs.2019.20.9.1755>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 15 July 2019; **Revised** 02 August 2019

Accepted 15 September 2019

***Corresponding Author; Euy shick Hong**

Tel:

E-mail: antaresax@gmail.com

I. 서 론

인터랙티브 멀티미디어 공연(interactive multimedia performance)은 미디어 콘텐츠(media contents), 음악, 영상 간의 실시간 인터랙션(interaction)을 통해 서로를 제어하고 소통한다. 다양한 장르가 융복합되어 관객에게 다양한 경험을 제공하고 관객이 공연에 참여하여 퍼포먼스를 연출하기도 한다. 이러한 멀티미디어 시스템은 실시간으로 음악, 영상 등을 제어하기 위해 설치형 애플리케이션(application)을 사용하여 구현된다[1]-[10]. 하지만 설치형 애플리케이션으로 구현된 시스템은 컴퓨터의 휴대성과 이동성, 접근성에 대한 제약이 있어 공연 재연 및 시스템 활용에 어려움이 있다. 이러한 문제를 해결하기 위해 본 논문에서는 Web 기술을 활용한 멀티미디어 공연 시스템을 구현 하고자한다.

2008년, Web 표준으로 제정된 HTML5¹⁾는 다양한 API(Application Programming Interface)²⁾[11]를 제공하면서 Web 브라우저(browser)에서 플러그인(plugin) 없이 영상, 음악 등의 멀티미디어 콘텐츠 구현을 가능하게 하였다[12]-[15]. 또한 Web 브라우저의 인터랙션을 제어하는 자바스크립트(JavaScript)³⁾의 발전과 비동기방식의 Node.js⁴⁾가 등장하면서 실시간 인터랙션 기술 활용이 용이해졌다.

Web 애플리케이션은 프로그램 설치 없이 PC와 모바일의 브라우저를 통해 사용이 가능하고 업데이트와 호환성의 이점이 있다. 특히 모바일의 경우 Web 애플리케이션의 접근이 좋고 마이크, 카메라, 센서(sensor) 등을 내장하고 있어 활용성이 좋다.

본 연구에서는 이러한 Web 기술을 활용하여 실시간 인터랙티브 오디오 시스템에 사용가능한 Web 애플리케이션 DrWebusic 개발을 목적으로 한다. ICACS 2017에서 발표된 오디오 라이브러리 DrSax.js[16]를 사용하여 오디오 시스템을 구축하고 모바일 기술을 활용하여 실시간 인터랙션 시스템을 구현한다[17]. 모바일은 실시간 인터랙션을 통해 신디사이저를 제어하고 이펙터를 통해 사운드를 프로세싱 한다. 또한 공연에 필요한 유틸리티 기능들을 컴포넌트로 제공한다.

본 논문에서 논의된 연구는 다음과 같이 두 가지로 구분한다. 첫째, Web 기반의 오디오 라이브러리와 Web 기술을 활용하여 실시간 인터랙티브 공연에 필요한 Web 오디오 애플리케이션 구현을 한다. 둘째, 실시간 인터랙션을 위해 모바일 기반의 Web 애플리케이션을 구현하고 실시간 인터랙션을 위한 모바일 활용방법을 고찰한다.

1) HTML의 5버전으로 월드 와이드 웹 (World Wide Web)에서 사용하는 마크업 언어.

2) 프로그래밍에 필요한 기능을 제공하는 인터페이스.

3) Web 브라우저에서 사용하는 객체 기반의 스크립트 프로그래밍 언어.

4) 2009년 Ryan Dahl이 개발한 자바스크립트 기반의 서버사이드 언어.

이를 위해 본 연구에 최적화된 Web 오디오 애플리케이션을 구현한다. 그리고 구현된 애플리케이션의 성능 및 특징을 고찰하고 향후 연구방향을 제시한다.

II. 오디오 시스템 아키텍처

2-1 오디오 시스템 디자인 및 프로세스 구조

본 논문에서 개발한 DrWebusic은 실시간 인터랙티브 공연을 위해 제작된 Web 오디오 애플리케이션이다. 오디오 라이브러리 DrSax.js를 사용하여 오디오 시스템을 구현하였고 Node.js 로 서버를 구축하여 실시간 인터랙션 기술을 사용하였다.

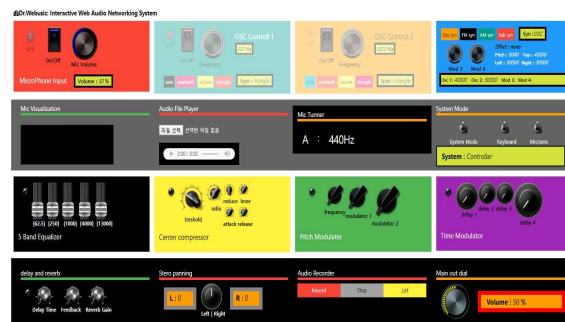


그림 1. 메인 DrWebusic 애플리케이션
Fig. 1. Main DrWebusic Application

DrWebusic은 메인 애플리케이션과 컨트롤러 애플리케이션으로 구성된다. 메인 DrWebusic 애플리케이션은 사운드 인풋과 신디사이저, 사운드 이펙터로 구성되어 있고 튜너, 녹음기, 오디오 플레이어 등의 유틸리티 기능을 제공한다. 그림 1은 메인 DrWebusic UI(User Interface)를 보여주고 있다.

DrWebusic 컨트롤러는 모바일에서 구현되는 애플리케이션으로 마이크 인풋 값과 모바일의 센서 값을 맵핑하여 메인 DrWebusic과 실시간 인터랙션 한다. DrWebusic의 오디오 시스템을 구성하는 사운드 컴포넌트는 오디오 기능을 제공하는 최소 단위로 정의한다. 다음절에서는 DrWebusic을 구성하고 있는 사운드 컴포넌트 종류와 특징에 대해 살펴보겠다.

2-2 사운드 컴포넌트 구성

1) 메인 DrWebusic

메인 DrWebusic은 공연에 필요한 15개의 컴포넌트로 구성된다. 각 컴포넌트를 기능에 따라 Sound Input, Synthesis, Effector로 분류하고 UI를 구현하였다. Sound Input은 다음과 같이 4개의 컴포넌트 Mic Input, Tuner, Recorder, Audio Player로 사운드 데이터를 출력하는 기능

을 수행한다. Mic Input 컴포넌트는 마이크를 통해 사운드 소스를 입력받는다. 입력받은 사운드는 이펙터를 통해 프로세싱 된다. Tuner 컴포넌트는 입력받은 사운드의 피치를 분석하여 주파수 값을 제공한다. Recorder 컴포넌트는 입력받은 사운드를 녹음하고 파일로 제공한다. Audio Player 컴포넌트는 오디오 파일을 재생한다. Synthesis는 4개의 신디사이저 컴포넌트 Basic Oscillator, Frequency Modulation, Amplitude Modulation, Subtractive Modulation로 구성된다. Effector는 7개의 이펙터 컴포넌트 EQ, Compressor, Pitch Modulation, Time Modulation, Delay, Reverb, Panner로 구성된다.

2) DrWebusic 컨트롤러

DrWebusic 컨트롤러는 메인 DrWebusic의 사운드를 실시간으로 제어하기 위해 구현하였다. 데이터 제어를 위해 모바일의 마이크와 모션 센서를 활용하였다. 모바일은 접근성의 이점이 있어 인터랙션 인터페이스로 활용도가 높다. 그림 2는 DrWebusic 컨트롤러 UI를 나타내고 있다.



그림 2. 모바일 기반의 DrWebusic 컨트롤러

Fig. 2. DrWebusic controller on mobile

DrWebusic 컨트롤러의 컴포넌트 구성은 Sound Input, Effector controller로 구성된다. 먼저 Sound Input는 모바일의 마이크를 통해 사운드를 입력 받는다. 입력된 사운드의 주파수 값은 메인 DrWebusic의 신디사이저 컴포넌트에 데이터를 전달한다. Effector controller는 모바일의 탑재된 모션 센서를 활용하여 메인 DrWebusic의 이펙터 컴포넌트 제어를 위해 개발되었다. 제어되는 이펙터 컴포넌트는 Pitch Modulation, Time Modulation, Delay, Reverb이다.

2-3 모바일 컨트롤러(실시간 인터랙션) 활용한 사운드 컴포넌트 제어 및 활용방법

DrWebusic 컨트롤러는 모바일의 모션 센서를 활용하였다. 센서는 X, Y, Z 축의 움직임 값을 입력받으며 입력 범위는 0 ~ 360도 이다. 이 데이터는 맵핑(mapping)을 통해 신디사이저의 주파수 값과 사운드 이펙터 값을 제어한다. 다음은 모바일의 움직임을 통한 DrWebusic 컨트롤러 활용법에 대해 알아보도록 하겠다.

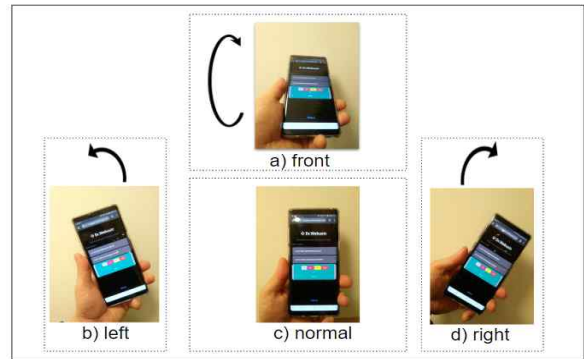


그림 3. DrWebusic 컨트롤러의 센서 제어

Fig. 3. sensor control of DrWebusic controller

DrWebusic 컨트롤러는 메인 DrWebusic과 실시간 인터랙션을 위해 크게 3가지 형태로 데이터를 전송한다.

첫째, 메인 DrWebusic의 마이크 입력과 이펙터 제어 활용법이다. 메인 DrWebusic은 악기 연주나 노래 등의 사운드를 마이크를 통해 입력받는다. DrWebusic 컨트롤러의 움직임을 통한 센서 값은 실시간 인터랙션을 통해 메인 DrWebusic의 여러 이펙터 컴포넌트의 파라미터를 제어하여 사운드를 실시간으로 프로세싱 한다.

둘째, 메인 DrWebusic의 신디사이저 제어 활용법이다. 메인 DrWebusic의 신디사이저 오실레이터는 DrWebusic 컨트롤러의 움직임을 통한 센서 값을 입력받아 주파수 값을 제어한다. 또한 메인 DrWebusic의 이펙터 파라미터를 제어하여 신디사이저 사운드를 실시간으로 프로세싱 한다.

셋째, 메인 DrWebusic의 신디사이저 제어를 위해 DrWebusic 컨트롤러의 마이크 인풋 값을 활용하는 방법이다. DrWebusic 컨트롤러의 마이크를 통한 입력 사운드 주파수 값을 분석하여 메인 DrWebusic 신디사이저의 오실레이터를 제어 한다. 마이크 입력 음량이 일정 값을 넘어야 주파수 값이 전송된다.

III. 인터랙티브 네트워킹 시스템

DrWebusic은 실시간 인터랙션을 위해 비동기 방식의 Node.js를 사용하였다. 그림 4는 DrWebusic의 실시간 인터랙션을 나타내고 있다. DrWebusic 컨트롤러 사용자는 인원수에 구애받지 않고 실시간 인터랙션에 참여가 가능하다. 또한 3G, 4G, 5G, Wi-Fi 네트워킹 환경에서 모바일로 쉽게 접근이 가능하다.

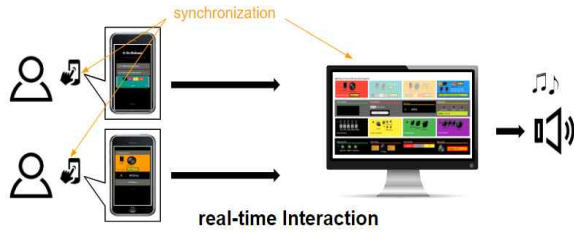


그림 4. DrWebusic의 인터랙션
Fig. 4. DrWebusic interaction

IV. 시스템 구현 및 기술 활용

4-1 오디오 시스템 프로그래밍

DrWebusic은 DrSax.js를 사용하여 오디오 컴포넌트를 구현하였다. 총 15개의 컴포넌트를 구현하였고 그림 5는 메인 DrWebusic에 구현된 4종류의 신디사이저인 오실레이터, FM 신디사이저, AM 신디사이저, subtractive 신디사이저를 나타내고 있다.

```
var osc = new DSX.Osc ({type: "sine", freq: 1700});
var fm = new DSX.FM ({carrier_type: "sine",
  carrier: 500, mod_type: "sine", modfreq: 700,
  depth: 1800, gain: 0.0,
});
var am_syn = new DSX.AM ({
  mod_type: "sine", modfreq: 1200, depth: 0.5,
  gain: 0.8,
});
var sub_syn = new DSX.Subtract ({
  cutoff: 1230, resonance: 0, gain: 0,
});
```

그림 5. 신디사이저 컴포넌트
Fig. 5. synthesizer component

4-2 실시간 인터랙션을 위한 모바일 컨트롤러 맵핑

DrWebusic 컨트롤러는 두 가지 방법으로 데이터를 맵핑하여 네트워킹 한다. 첫째는 모바일에 구현된 마이크 인풋 컴포넌트이고 둘째는 모바일에서 제공하는 모션 센서 활용이다. DrWebusic 컨트롤러의 실시간 인터랙션에 필요한 맵핑 방법에 대해 살펴본다.

1) 마이크 인풋 활용 및 맵핑

DrWebusic 컨트롤러는 마이크 인풋 데이터를 받는다. 입력받은 데이터는 주파수 값과 음량 값을 분석하여 데이터를 실시간으로 메인 DrWebusic에 전달한다. 주파수 분석을 위

해 사용한 DrSax.js의 Tuner 인터페이스는 입력받은 사운드를 실시간으로 분석하여 주파수 값을 숫자로 제공한다. 또한 음량 값을 도출하기 위해 ampDomain() 인터페이스를 사용하였다. ampDomain()는 입력 사운드의 음량 값을 실시간으로 비주얼라이제이션 하고 콜백(call back)함수를 통해 음량 값을 제공한다. 그림 6은 각 데이터를 추출해 메인 DrWebusic에 데이터를 전송하는 소스코드이다.

```
var tune = new DSX.Tunner( "pitch", "note", (data) => {
  socket.emit( 'pitch',data );
});
var amp_canvas = new DSX.ampDomain( 'canv', "red",
200, 100, (data) => {
  var newData = data.toFixed(1);
  if (newData !== "0.5") {
    socket.emit( 'pitchGain',newData );
  }
});
```

그림 6. 신디사이저 컴포넌트
Fig. 6. synthesizer component

DrWebusic 컨트롤러를 통해 전송된 데이터는 메인 DrWebusic의 신디사이저 주파수 값을 제어한다. 주파수에 따라 제어되는 신디사이저 종류는 Basic Oscillator, Frequency Modulation, Amplitude Modulation, Subtractive Modulation 이다.

```
socket.on('pitch', function(str) {
  oscFirst.frequency.value = str;
  $("#pitch_val").html(str);
})
socket.on('pitchGain', function(str) {
  let dataGain = str -0.6;
  dataGain = dataGain.toFixed(1);
  if(dataGain < 0.11){
    dataGain = 0.0;
  }
  gainOscFirst.gain.value = dataGain;
  fm.gain.value = dataGain;
  sub_syn.gain.value= dataGain;
  amGain.gain.value= dataGain;
})
```

그림 7. 마이크 인풋 데이터 맵핑
Fig. 7. data mapping of mic input

입력받은 주파수 값의 범위는 0 ~ 20000 이고 각 신디

사이저의 오실레이터 주파수를 제어한다. 그림 7은 입력받은 메인 DrWebusic의 주파수 값 매핑을 나타내고 있다. 음량 값이 일정 값을 넘지 않는 경우 음량 값을 0으로 설정 된다.

2) 모바일 센서 활용 및 데이터 매핑

모바일은 모션 센서를 통해 각 축의 움직임에 따른 회전각도 값을 제공한다. 이 회전 값을 제어하기 위해 자바스크립트에서 제공하는 DeviceOrientationEvent 함수를 사용해 본 연구에 활용하였다. 그림 8은 모바일 센서의 데이터 매핑을 나타내고 있다.

```

window.addEventListener('deviceorientation',
handleOrientation, false);
function handleOrientation(event) {
    var absolute = event.absolute;
    var alpha    = event.alpha;
    var beta     = event.beta;
    var gamma    = event.gamma;
    let tilt_front = beta.toFixed(0);
    let tilt_left  = alpha.toFixed(0);
    let tilt_right = gamma.toFixed(0);
    tilt_front = (tilt_front - 75) * -1;
    tilt_front = tilt_front.toFixed(0);
    tilt_left  = tilt_left - 90;
    tilt_left  = tilt_left.toFixed(0);
    tilt_right = tilt_right;
}

```

그림 8. 모바일 센서 매핑

Fig. 8. sensor mapping of mobile

먼저 X 축은 DeviceOrientationEvent.beta를 통해 -180도부터 180도의 데이터를 제공하고 Y 축은 DeviceOrientationEvent.gamma를 통해 -90도부터 90도 데이터를 제공한다. Z 축은 DeviceOrientationEvent.alpha를 통해 0도부터 360도 데이터를 제공한다. DeviceOrientationEvent.beta 값은 75를 빼고 -1을 곱해 변수 tilt_front에 저장되며 데이터 활용 범위는 -30 ~ 0 이다. DeviceOrientationEvent.alpha 값은 90을 뺀 후 tilt_left에 저장되며 활용 범위는 0 ~ 100 이다. DeviceOrientationEvent.gamma 값은 데이터 매핑 없이 변수 tilt_right에 저장된다. 활용 범위는 0 부터 이다. 매핑된 데이터는 설정에 따라 신디사이저의 오실레이터와 이펙터의 파라미터를 제어한다. 그림 9는 모바일을 통해 전송된 데이터를 활용하여 신디사이저를 제어하는 코드이다. 매핑된 tilt_front는 dataTop에 저장되며 50을 곱해 주파수 값을 제어한다. 데이터가 -10 ~ 10이면 음량 값은 0으로 설정된다.

```

oscFirst.frequency.value = dataTop * 50;
fm.carrier.value = dataTop * 50; ;
if ( dataTop > -10 && dataTop < 10 ) {
    gainOscFirst.gain.value = 0;
    fm.gain.value = 0;
    sub_syn.gain.value = 0;
    amGain.gain.value = 0;
} else {
    gainOscFirst.gain.value = 0.9;
    fm.gain.value = 0.9;
    sub_syn.gain.value = 0.9;
    amGain.gain.value = 0.9;
}

```

그림 9. 전송받은 데이터 매핑

Fig. 9. data mapping received

그림 10은 모바일을 통해 전송된 데이터를 활용하여 딜레이 이펙터를 제어하는 코드이다. 매핑된 tilt_front는 data.top에 저장되며 3을 곱해 딜레이 타임을 제어한다. 딜레이 타임 값이 0.1보다 작으면 딜레이 타임 값은 0.1로 저장된다. 매핑된 tilt_right는 data.right에 저장되며 67로 나누어 피드백을 제어한다. 0.1 보다 작으면 0.1로 저장되고 0.9보다 크면 0.9로 저장된다.

```

case 'delay':
    let delayTime = data.top * 3;
    if ( delayTime < 0.1 ) { delayTime = 0.1; }
    saxDelay.delayTime = delayTime;
    let newdata = data.right / 67
    newdata = newdata.toFixed(1);
    if ( newdata < 0.1 ) { newdata = 0.1; }
    if ( newdata > 0.9 ) { newdata = 0.9; }
    saxDelay.feedback = newdata;

```

그림 10. 이펙터 데이터 매핑

Fig. 10. data networking in real time

V. 결 론

HTML5의 등장과 더불어 다양한 API를 제공하는 Web은 브라우저 처리속도 향상과 자바스크립트의 발전으로 영상, 음악 등의 멀티미디어를 실시간으로 제어하는 Web기반의 애플리케이션 개발을 가능하게 하였다. Web 애플리케이션은 접근성과 호환성의 이점 있으며 특히 모바일은 센서와 카메라, 마이크 등의 기능을 제공해 활용성이 좋다. 본 논문은 이러한 Web 기술을 활용한 실시간 인터랙티브 오디오 시스템 개발을 목적으로 한다. 이에 본 연구는 다

음과 같이 진행되었다.

첫째, Web 기반의 오디오 라이브러리인 DrSax.js를 활용하여 Web 오디오 애플리케이션 DrWebusic을 개발하였다. 15개의 오디오 컴포넌트를 기능에 따라 Sound Input, Synthesis, Effector로 분류하여 구현하였다. Sound Input는 악기 연주나 노래 등의 마이크 입력 값을 제어하는 컴포넌트로 구성되었고 Synthesis는 오실레이터 기반의 소리 합성 사운드를 제공한다. Effector는 사운드를 실시간으로 프로세싱 한다.

둘째, DrWebusic 컨트롤러를 개발하여 메인 DrWebusic와의 실시간 인터랙션 시스템을 구현하였다. 접근성이 좋은 모바일의 화면 크기를 고려하여 UI를 디자인하였고 실제 인터랙티브 공연에 사용될 목적으로 모바일의 모션 센서와 마이크기능을 활용하였다.

본 연구를 통하여 제작된 실시간 인터랙티브 Web 오디오 시스템 DrWebusic은 인터랙티브 멀티미디어 시스템에 사용될 목적으로 개발되었다. 또한 미디어 설치작품, 다양한 융복합 콘텐츠 전시 등에도 효과적으로 활용될 것으로 기대되며 퍼지컬 컴퓨팅(Physical Computing)기술과 인공지능(Artificial Intelligence)기술을 활용한 멀티미디어 콘텐츠 시스템 개발을 향후연구로 계획 중이다.

참고문헌

- [1] G. Wang. Designing smule's iphone ocarina. In Proceedings of the International Conference on New Interfaces for Musical Expression, Pittsburgh, 2009.
- [2] J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang. Evolving mobile phone orchestra. In Proceedings of the International Conference on New Instruments for Musical Expression, June 2010.
- [3] Rohs, M. and Essl, G. "Camus2 - Collaborative Music Performance with Mobile Camera Phones." In Proceedings of the International Conference on Advances in Computer Entertainment Technology (ACE). Salzburg, 2007.
- [4] Freeman, J. "Large Audience Participation, Technology, and Orchestral Performance." In Proceedings of the International Computer Music Conference (ICMC). Barcelona, 2005.
- [5] Essl, G., & Muller, A. (2010, June). Designing Mobile Musical Instruments and Environments with urMus.
- [6] Bianciardi, D., Igoe, T., and Singer, E. "Eos Pods : Wireless Devices for Interactive Musical Performance." In Proceedings of the Fifth Annual Conference on Ubiquitous Computing. Seattle, 2003.
- [7] Essl, G. and Rohs, M. "Interactivity for Mobile Music-Making." *Organised Sound*. 14, no. 2 (2009): 197-207.
- [8] Dahl, L., Herrera, J., and Wilkerson, C. "TweetDreams : Making Music with the Audience and the World Using Real-Time Twitter Data." In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME). Oslo, 2011.
- [9] Lee, S.W. and Freeman, J. Echobo : A Mobile Music Instrument Designed for Audience To Play. In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME). 2013. Daejeon, Korea.
- [10] N. Weitzner, J. Freeman, S. Garrett, and Y. Chen. massMobile an audience participation framework. Proceedings of the New Interfaces For Musical Expression Conference, 2012.
- [11] web audio api-w3c working draft. Web audio API: <https://www.w3.org/TR/webaudio>.
- [12] Roberts, C., Wakefield, G., Wright, M., & Kuchera-Morin, J. (2015). Designing musical instruments for the browser. *Computer Music Journal*.
- [13] Hindle, A. (2013). SWARMED: Captive Portals, Mobile Devices, and Audience Participation in Multi-User Music Performance. In NIME.
- [14] 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.
- [15] 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, 2013. Web audio api extension.
- [16] E. Hong and J. Kim. DrSax.js : a JavaScript based Unified Web Audio Library and Framework. ICACS 2017, Jeju, Korea.
- [17] E. Hong and J. Kim. Interactive Web Audio Networking System and Method with DrSax.js. ICCIP 2017, Nov, Tokyo, Japan.

02

홍의식(Euy shick Hong)



2011년 : 동국대학교 영상대학원 멀티미디어학과 컴퓨터음악 석사

2019년 : 동국대학교 영상대학원 멀티미디어학과 컴퓨터음악 박사

2019년~현 재: ICMT(Institute for Computer Music and Technology) 연구소장

2018년~현 재: Market Designers 개발팀 프론트엔드 팀장

※관심분야 : 멀티미디어 콘텐츠, 컴퓨터음악, 인터랙티브 퍼포먼스, Web 애플리케이션, 사운드 합성