

# 소셜 연주를 지원하는 스마트폰기반 다중 악기 애플리케이션 구현

황병곤\*

## 요약

최근 전 세계적으로 모바일 디바이스 및 통신 기술의 발전으로 전화와 문자 메시지만 주로 사용하던 기본 휴대폰은 스마트폰으로 급격하게 진화하여 사회의 각 영역에 다양한 서비스를 제공하고 있다. 본 논문은 스마트폰을 이용하여 밴드 연주에 들어가는 기본 악기인 기타, 피아노, 베이스 그리고 드럼을 기반으로 한 멀티유저 악기 연주 애플리케이션을 제안한다. 이 애플리케이션은 스마트폰에서 개별적으로 연주한 음악은 서버의 시간 동기화 기능에 의해서 통합된 음악 데이터 스트림으로 생성된다. 더불어 실제 악기와 유사한 그래픽 유저 인터페이스 실현으로 사용자는 효율적으로 스마트폰에 의한 악기 연주를 할 수 있다.

## An Implementation of Smartphone-based Multiple Musical Instruments Application supporting Social Playing

Hwang Byung-kon \*

## Abstract

Due to the development of mobile devices and communication technology, cell phones have radically evolved from portable phones into smartphones and provided variety of services to a society.

This paper presents Multi-user Musical Instrument application for smart phones based on band instruments such as piano, guitar, bass guitar, and drum. This application makes music playing from smartphone to integrative music data stream using the synchronization function of server. In addition, user can play it of smartphone efficiently by implementing a graphic user interface similar to real musical instruments.

Key Words : Smartphone, User Interface, Mobile, Musical Instrument

## 1. 서론

최근 정보통신 기술의 발전에 따라 통신 네트워크 기능 및 사용자 요구를 만족시킬 수 있는 다양한 애플리케이션을 갖춘 스마트폰 수요가 기하급수적으로 늘

고 있으며, 스마트폰의 그 쓰임새도 다양하다.[1]

이러한 스마트폰의 쓰임새가 다양해지면서 사용자는 시간과 장소의 제약을 받지 않고 다양한 정보를 서로 공유하고 제공할 수 있다. 즉, 스마트폰은 일반 컴퓨터와 같이 사용자가 원하는 다양한 애플리케이션을 동작시킬 수 있다. 이러한 스마트폰 애플리케이션의 급격한 증가에 따라 애플리케이션 콘텐츠도 다양해지고 있는 실정이다. 특히 소셜 네트워킹 애플리케이션은 나이와 계층에 관계없이 커뮤니케이션의 장을 열게하고 있는 실정이다. 이러한 시대적 변화에 비추어보면 사용자들이 생활에서 다양하게 이용할 수 있는 소셜 네트워킹 애플리케이션 개발은 날로 더욱 요구되어진다고 할 수 있다.[2,3]

이러한 측면에서 본 논문은 스마트폰을 이용하여 밴

※ 제일저자(First Author) : 황병곤  
접수일:2011년 12월 08일, 수정일:2011년 12월 21일  
완료일:2011년 12월 26일

\* 대구대학교 컴퓨터공학부 교수

[bkhwang@daegu.ac.kr](mailto:bkhwang@daegu.ac.kr)

▣ 이 논문은 2008년도 대구대학교 교내 연구비 지원에 의해 수행되었음

드 연주에 필요한 기타, 피아노, 베이스, 드럼 악기를 사용하여 멀티유저 악기 연주 애플리케이션을 구현한다. 4명의 개별 사용자가 각자 스마트폰을 이용하여 맡은 악기를 연주하면, 개별적으로 연주한 음악은 시간 동기화 기능을 이용하여 통합한 후 하나의 음악 데이터 스트림을 생성한다. 여기서 개별 악기는 유저 인터페이스를 통하여 연주할 수 있고 연주 한 곡을 녹음, 저장, 재생, 삭제할 수 있다. 이 시스템은 안드로이드(Android) SDK를 이용하여 구현 되었으며 실제 큰 악기들을 휴대하기 힘든 상황에서 간편하게 연주할 수 있도록 실제 악기와 유사한 형태의 인터페이스로 구현한다.[4-6]

기존의 단일 사용자 악기 연주 애플리케이션의 경우 한 악기만의 연주가 가능하기 때문에 다중 악기에 의한 밴드 연주의 현실감을 느끼기 어려웠다. 본 논문은 이러한 기존 악기 연주 애플리케이션의 단점을 극복하기 위한 다중 악기연주 애플리케이션을 위해 3가지의 통신 모드를 지원한다.[7] 여기서 3개 통신 모드는 1) Ad Hoc 모드(전철이나 버스와 같이 비교적 근거리에서 모인 사용자간 협동 연주를 위한 모드(PowerAgent: 4개 단말기간의 공평한 전력 소모를 위한 S/W 모듈)), 2) Wifi 모드(Wifi 가능 지역에서의 통신 모드), 3) 3G 모드(1, 2번 모드가 불가능한 경우)로서 다수의 사용자와 함께 하는 일종의 소셜 네트워크를 실현할 수 있으며 편한 연주를 위해 멀티 터치스크린을 구현한다.

## 2. 개발내용

### 2.1 시스템 구성

본 시스템은 첫째, 안드로이드 2.0 플랫폼으로 구성이 되어 멀티터치의 구현이 가능하다. 그리고 이 악기 연주 애플리케이션은 기본적으로 사용자에게 포크 기타 (Folk Guitar), 그랜드 피아노 (Grand Piano), 일렉트릭 베이스 기타 (Electric Bass Guitar) 및 스틸 드럼 (Steel Drum)같은 주요 4가지 악기를 지원한다.

둘째, 안드로이드 기반 녹음(Recording) 시스템 구현 부분으로 각 녹음된 데이터는 SD 저장장치를 이용하여 관리하며 녹음된 데이터에 접근 및 재생, 삭제를 할 수 있다. 셋째, 서버-클라이언트간 통신 구현 부분으로 다수 사용자를 효율적으로 관리하기 위해서 다중 연주를 위해 채널(또는 Room)을 서버에서 제공하고 사용자는 이를 신청하여 사용할 수 있도록 한다. 넷째, 서버에서 클라이언트 관리 구현 부분으로, 사용자 클라

이언트는 서버에 접속하여 접속된 다른 사용자와 함께 연주할 수 있으며, 이때 서버는 사용자간의 지연시간을 줄이기 위해 100ms내로 각 사용자들의 음원 코드를 저장한 후 특정 시간에 통합 전송한다.

### 2.2. 시스템 기능

#### 2.2.1 멀티터치 센서 지원 악기연주

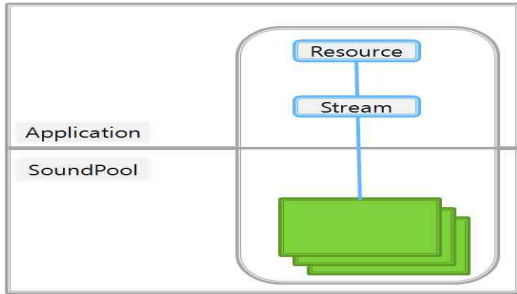
##### (1)음원 로드

본 연구에 기반이 되는 안드로이드는 사운드를 재생하기 위해 다음과 같은 클래스를 제공한다.

<표 1> 사운드 재생 클래스

Package	Class Name	Description
android.media	Media Player	<ul style="list-style-type: none"> <li>음원 파일을 재생</li> <li>하나의 음원을 재생하는 용도로 사용</li> <li>음악이나 동영상 데이터를 재생</li> <li>seeking, pause, play 등 지원</li> </ul>
	Sound Pool	<ul style="list-style-type: none"> <li>음원 파일을 재생</li> <li>음원을 16bit PCM 디코딩으로 리소스 화</li> <li>리소스 화 된 음원을 재생함으로써 CPU사용량 감소</li> <li>여러 개의 스트림 로드 가능</li> <li>동시 재생 및 관리 가능</li> <li>연속적으로 자주 사용되는 짧은 음원에서 주로 사용</li> </ul>

위 <표 1>에서 두 가지의 클래스 중 본 시스템에 사용된 클래스는 SoundPool이다. 본 시스템은 악기 연주가 주가 되므로 각 악기에 대한 음원의 재생이 매우 중요하다. 일반적인 악기는 다수의 음이 자주 반복적으로 재생되는 것과 같이 본 시스템에서도 다수의 음원이 자주 반복적으로 재생이 된다. 그리고 재생되는 음원이 길지 않기 때문에 MediaPlayer 클래스 보다 SoundPool 클래스를 사용하여 구현하였다. (그림 1)은 SoundPool 클래스를 사용한 음원 구현이다.



(그림 1) SoundPool의 리소스화 된 음원

(2) 사용자 인터페이스

사용자 인터페이스는 실제와 같은 모양을 가지고 있다.

○ 전체적인 인터페이스

- 각각의 악기를 선택하기 쉬운 인터페이스
- 각 악기마다 있는 간편한 녹음 및 재생 버튼
- 실제 악기 모양 이미지

○ Piano 인터페이스

- 실제 건반과 같은 형태로 구현
- 옥타브 조절키에 의한 옥타브 변경
- 터치에 의한 건반의 이미지 변화

피아노의 경우 실제와 같은 건반 형식으로 터치 시 건반이 눌린 것과 같은 효과를 제공하여 준다. 멀티 터치 방식을 이용하여 제작되었기 때문에 스마트폰의 특성상 작은 화면으로 인해 연주하기 어려울 수 있으나 조금만 익숙해진다면 편하게 연주를 할 수 있다.

○ Guitar 인터페이스

- 6줄의 현이 각각의 음을 가지도록 구현
- 코드 등록 후 원하는 코드로 연주 가능
- 드래그로 인한 연속적인 음 연주가 가능

기타는 보통 코드 연주를 많이 한다. 하지만 스마트폰으로 실제와 같은 기타 연주는 불가능하다. 그래서 위쪽에 코드 버튼을 만들고 아래쪽에는 6개의 현을 배치함으로써 코드 연주를 쉽게 할 수 있다. 코드 버튼을 통해 코드를 선택하고 현을 터치하면 실제와 같은 기타 연주가 가능하다. 코드를 바꿔 가면서 손으로 드래그 하여 연주하면 리듬 기타의 사운드를 낼 수 있다.

○ Bass 인터페이스

- 4줄의 현이 각각의 음을 가지도록 구현
- 코드 등록 후 원하는 코드로 연주 가능
- 드래그로 인한 연속적인 음 연주 가능
- 플렛의 변경으로도 연주 가능

베이스는 기본적으로 기타와 거의 동일하다. 코드 선택 버튼이 있고 아래에는 베이스 4줄의 현이 있다. 하지만 보통 베이스는 코드를 잡고 기타처럼 치지는 않는다. 그래서 베이스에는 코드 중에 플렛을 넣어서 그 플렛을 잡은 상태에서 각 현이 소리를 낼 수 있도록 하였다.

○ Drum 인터페이스

- 5기통의 드럼이 각각 실제와 같이 배치
- 드래그를 이용하여 연주가 가능

드럼의 경우는 기본적인 5기통의 드럼으로 구성되어 있으며 각각 마다 드럼의 고유의 소리를 간직하고 있으며 실제 드럼을 치는 것과 같은 배치로 연주를 할 수 있도록 구현되었다. 작은 화면상에서 연속적으로 연주해야 되는 경우 가까운 거리는 드래그를 이용하여서 연주할 수 있도록 되어 있다.

(3) 음원 및 코드 지원

본 시스템은 풍부한 음원과 다양한 코드를 지원한다. 여기서 코드는 몇 개의 음으로 구성되어진 화음을 말하며 음악의 기본바탕이다. 각 악기에 대해 제공하는 음원은 <표 2>와 같다.

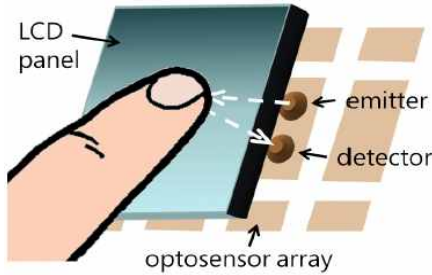
<표 2> 악기의 음원

	Piano	Drum	Guitar	Bass	Total
음원 개수	43개	8개	61개	56개	168개
용량	8.94KB	1.81KB	10.4KB	13.0KB	34.2KB

코드 버튼에 Long Click 이벤트가 적용 되면 자신이 연주하고자 하는 코드로 변경할 수 있는 인터페이스가 제공되며, 변경된 코드들은 데이터화된 후 저장되므로 다음 실행 시에도 변경된 상태가 유지된다.

(4)멀티 터치 지원

안드로이드 2.0 에서는 (그림 2)와 같은 멀티 터치를 지원한다. 멀티 터치란 동시에 여러 개의 터치 포인트를 인식하는 기술로, 일반적인 하나의 터치 포인트만 인식을 하는 것보다 더 다양한 조작을 할 수 있게 된다.



(그림 2) 멀티 터치의 원리

본 시스템은 동시에 3개까지 멀티 터치를 지원한다. 더 많은 수의 멀티 터치를 지원하면 좋을 수 있으나 사실 스마트폰 조건에서는 악기 연주를 하는데 충분하다. 멀티터치와 기타 관련된 공통적인 이벤트들은 PublicEvent.class에서 관리한다.

O PublicEvent.class의 특성

- 4종류의 악기 연주에 있어서 공통적으로 포함되어 있는 터치 관련 이벤트들과 공통으로 존재하는 변수들 선언.
- 볼륨 키의 업, 다운 이벤트를 오버라이드 해서 미디어 볼륨만 조정이 가능하도록 설정
- 아이디 값을 이용하는 방법으로 터치하면서 이동할 때마다 소리가 계속 나오는 것이 아니라 이동하여 터치했을 때에만 소리가 나도록 함

2.2.2 레코딩 시스템

(1)연주 녹음 및 재생

악기 인터페이스 좌측 상단에 녹음 및 재생 버튼이 위치하고 있다. 이 버튼을 통하여 사용자는 현재 연주 중인 음원을 녹음 할 수 있으며, 즉시 재생도 가능하다. 안드로이드에서는 녹음 기능을 사용하기위해 <표 3>과 같은 클래스를 제공한다.

<표 3> 안드로이드 미디어 클래스

Package	Class Name	Description
android.media	AudioRecorder	<ul style="list-style-type: none"> <li>• 애플리케이션에서 리소스를 관리하는 클래스</li> <li>• 플랫폼 하드웨어에서 입력되는 오디오를 기록하는 Java 애플리케이션을 위해 오디오 리소스 관리함</li> <li>• 샘플링 및 세부적인 정보를 지정 가능</li> </ul>
	MediaRecorder	<ul style="list-style-type: none"> <li>• 비디오와 오디오를 기록할 때 사용</li> <li>• 간단한 상태 머신을 기초로 기록</li> </ul>

(2)파일 관리

녹음 되는 데이터는 SD카드 특정 디렉터리에 저장된다. 저장되는 데이터의 형식은 <표 4>와 같다.

<표 4> 녹음 데이터

악기 종류	형식
Piano	• Piano_"count"_. 3gp
Drum	• Drum_"count"_. 3gp
Bass	• Bass_"count"_. 3gp
Guitar	• Guitar_"count"_. 3gp
합주	• "Name_Day"_"count"_. 3gp

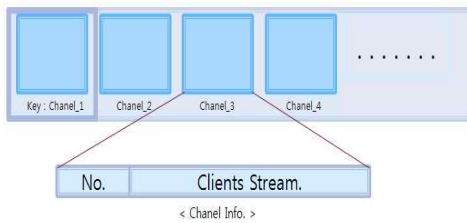
파일 관리 내의 MediaPlayer 클래스를 이용하여 재생, 일시정지 그리고 삭제기능을 제공한다.

2.2.3 클라이언트의 관리 및 통신

(1)채널 제공

각 클라이언트 간 원활한 통신을 위해선 서버에서 각 클라이언트를 관리하는 것이 중요하다. 본 시스템에서는 채널은 클라이언트를 관리하여 각 사용자의 연주를 통합한다. 서버는 사용자의 채널에 대한 정보(Key)와 스트림 정보를 클래스화 하여 HashMap으로 List화한 후 메모리에 저장하고 클라이언트에는 채널 정보를 제공한다. 저장되는 채널정보는 (그림 3)과 같다.

Hash Map List



(그림 3) 서버의 채널 구성도

(2)클라이언트 간 통신

같은 채널에 접속 중인 클라이언트들의 통신은 TCP/IP를 사용하며, 직렬화된 객체를 교환한다. 교환되는 직렬화된 객체는 (그림4)와 같다.

Code	Message	Flag	Chanel Number	Client Info
← ArrayList Variable Length →	← String Variable Length →	← Int 4Byte →	← Int 4Byte →	← Int 4Byte →

(그림 4) 교환되는 객체의 구조

객체 내에는 음원에 대한 코드(Code)를 시작으로 Message, Flag, Channel Number, Client Info 순의 구조로 되어있다.

○ Code

- 음원에 대한 타입 정보를 가지고 있는 부분으로서 그 종류로는 지정된 0~168까지의 음원에 대한 번호로 이루어진다.

○ Message

- 클라이언트 간의 커뮤니케이션에 대한 메시지 정보를 포함한다.

○ Flag

- 교환되는 객체가 어떤 종류인지 알려주는 부분으로서 전역 상수를 참고한다.

○ Channel Number

- 클라이언트가 서버에 채널의 대한 정보를 요청한다.

○ Client Info

- 다수의 클라이언트를 구분하기 위해 각 클라이언트

의 고유한 정보를 저장한다.

(3) 채널 내 사용자간 연주 동기화

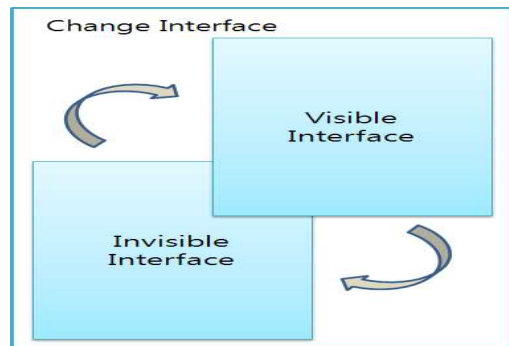
악기들의 합주를 위해서는 각 사용자 악기 연주 동기화가 필요하다.[8] 각 클라이언트에서 전송되는 음원 코드는 송신처의 클라이언트를 제외한 채널 내 나머지 클라이언트에게 전송하여야 한다. 이때, 발생 되는 문제는 다음과 같다.

- 무선망과의 통신 상태가 원활하지 않을 경우
- 서버에서 단시간에 수많은 데이터 패킷이 클라이언트에 도착하는 경우

각 클라이언트에서 수신되는 다수의 음원 코드를 저장하는 List가 담긴 객체는 상황에 따라 10ms ~ 100ms 정도 지연시킨 후 전송을 하게 된다.

2.2.4 사용자간 커뮤니케이션

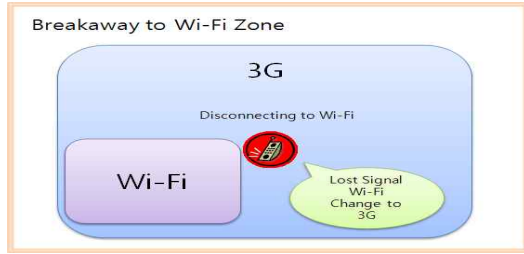
밴드 합주를 하기 위해서 커뮤니케이션을 위한 채팅 기능을 제공한다. 악기 인터페이스의 리사이징에 대한 부담을 줄이기 위해 채팅과 악기 인터페이스는 메뉴 버튼을 이용하여 교환된다. (그림 5)는 악기와 채팅 인터페이스 전환 기능이다.



(그림 5) 체인징되는 인터페이스

2.2.5 무선 망 전환

사용자가 WiFi를 이용하여 악기 연주 애플리케이션을 사용 중에 3G망으로의 이동이 발생하였을 경우 이를 사용자에게 (그림 6)와 같은 메시지 형태의 경고창으로 알리며, 사용자의 결정에 따라 무선망 전환이나 애플리케이션을 종료시킨다.



(그림 6) 무선망 전환 경고창

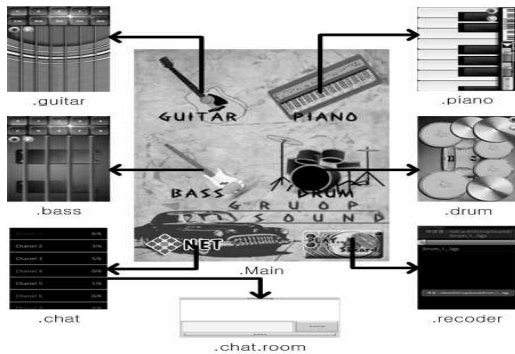
No	Function
1	<ul style="list-style-type: none"> <li>일반적인 피아노와 동일한 형식의 건반</li> <li>화면상에 한 옥타브를 표현</li> </ul>
2	<ul style="list-style-type: none"> <li>녹음 및 재생에 대한 버튼</li> </ul>
3	<ul style="list-style-type: none"> <li>옥타브를 변환시키는 버튼 부분</li> <li>총 3옥타브 까지 변경 가능</li> </ul>
4	<ul style="list-style-type: none"> <li>반주 음을 조정하는 버튼</li> <li>속도 조절 및 반주의 형태 변경 가능</li> </ul>

(그림 8) 피아노 인터페이스

## 2.3 애플리케이션 구현

### 2.3.1 메인 인터페이스

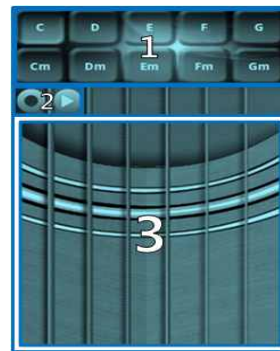
(그림 7)은 구현된 시스템의 메인 인터페이스이다.



(그림 7) 메인 인터페이스

### 2.3.3 기타 인터페이스

기타의 경우는 코드 변경과 연주 부분, 녹음 및 재생으로 이루어져 있으며 아래와 같은 기능을 제공한다.

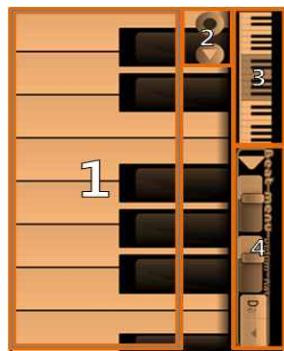


No	Function
1	<ul style="list-style-type: none"> <li>기본적으로 10개의 코드를 선택할 수 있도록 구성</li> <li>Long Click시 다른 코드로 교체</li> </ul>
2	<ul style="list-style-type: none"> <li>녹음 및 재생에 대한 버튼</li> </ul>
3	<ul style="list-style-type: none"> <li>일반적인 기타와 동일한 6현</li> <li>위의 코드부분에서 설정한 소리를 인식하여 작동</li> </ul>

(그림 9) 기타 인터페이스

### 2.3.2 피아노 인터페이스

피아노의 경우 크게 4가지 항목으로 이루어져 있으며 각각의 버튼은 아래와 같은 기능을 제공한다.



### 2.3.4 베이스 인터페이스

베이스의 경우는 코드 변경 부분에서 기타와는 약간 차이 나는 인터페이스 구성을 가지지만 전반적으로는 동일한 기능을 제공한다. 기능에 대한 설명은 아래와 같다.

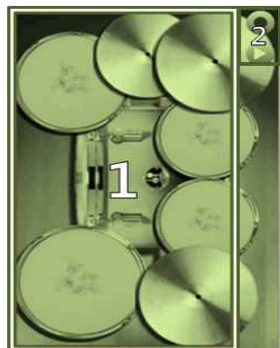


No	Function
1	<ul style="list-style-type: none"> <li>기본적으로 10개의 코드를 선택할 수 있도록 구성</li> <li>Long Click시 다른 코드로 교체</li> <li>기타와는 다르게 플랫 지정도 가능</li> </ul>
2	<ul style="list-style-type: none"> <li>녹음 및 재생에 대한 버튼</li> </ul>
3	<ul style="list-style-type: none"> <li>일반적인 베이스와 동일한 4현</li> <li>위의 코드부분에서 설정한 소리를 인식하여 작동</li> </ul>

(그림 10) 베이스 인터페이스

### 2.3.5 드럼 인터페이스

가장 단순한 인터페이스 형태를 가진 부분으로서 2개의 항목으로 이루어져 있다. 기능에 대한 설명은 아래와 같다.



No	Function
1	<ul style="list-style-type: none"> <li>5기통의 드럼을 그대로 재현</li> <li>터치시 실제와 같이 울리는 효과 구현</li> </ul>
2	<ul style="list-style-type: none"> <li>녹음 및 재생에 대한 버튼</li> </ul>

(그림 11)드럼 인터페이스

### 2.3.6 파일재생 인터페이스

위의 악기 연주 인터페이스에서 녹음된 각각의 파일을 재생할 수 있는 부분으로서 다음과 같이 크게 3가지 기능을 제공하여 준다.



No	Function
1	<ul style="list-style-type: none"> <li>재생중인 음원의 정보를 알려주는 정보 창</li> </ul>
2	<ul style="list-style-type: none"> <li>전체 길이를 알려주며 앞뒤로 이동 가능한 Seek bar</li> </ul>
3	<ul style="list-style-type: none"> <li>현재 저장된 파일의 목록을 표시</li> <li>터치 시 해당 파일을 재생</li> </ul>

(그림 12) 파일 재생 인터페이스

### 2.3.7 채널정보 인터페이스

서버에 접속하였을 시 보이는 인터페이스는 채널 리스트 및 채널 상태를 표시해준다. 기능에 대한 설명은 아래와 같다.



No	Function
1	<ul style="list-style-type: none"> <li>5기통의 드럼을 그대로 재현</li> <li>터치시 실제와 같이 울리는 효과 구현</li> </ul>
2	<ul style="list-style-type: none"> <li>녹음 및 재생에 대한 버튼</li> </ul>

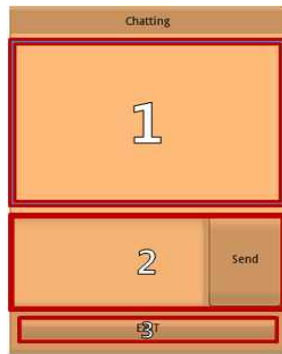
(그림 11)드럼 인터페이스

No	Function
1	<ul style="list-style-type: none"> <li>• 색에 맞추어 채널의 상태 확인가능</li> <li>• 채널 리스트 표시</li> <li>• 터치 시 채널에 대한 접속</li> </ul>
2	<ul style="list-style-type: none"> <li>• 채널의 수용인원을 표시하는 부분</li> </ul>

(그림 13) 채널정보 인터페이스

### 2.3.8 채팅 인터페이스

사용자간의 채팅을 위한 인터페이스는 일반적인 채팅 시스템과 유사하며 기능은 아래와 같다.



No	Function
1	<ul style="list-style-type: none"> <li>• 대화 내용이 표시되는 부분</li> <li>• 일반적인 채팅화면의 중심</li> </ul>
2	<ul style="list-style-type: none"> <li>• 문자를 입력하는 부분</li> <li>• 문자를 위의 채팅화면으로 전송하기 위한 버튼</li> </ul>
3	<ul style="list-style-type: none"> <li>• 채팅 창에서 나가기 위한 종료 버튼</li> <li>• 터치 시 메인 화면으로 복귀</li> </ul>

(그림 14) 채팅 인터페이스

## 3. 결론

기존의 악기연주 애플리케이션은 개인이 연주하거나 다수의 사용자들이 모여서 각각 다른 애플리케이션을 사용하여 연주하였으나, 본 논문에서는 각 악기가 나타낼 수 있는 고유한 코드들을 설정하여 조금 더 풍성한 음원을 가질 수 있도록 구현되었고, 한편 WiFi를 이용한 원거리 통신을 이용하여 멀리 떨어져 있는

사용자들도 함께 연주 할 수 있도록 구현되었다.

네트워크의 가장 큰 문제점인 용량의 확대에 인한 싱크로 율의 불일치는 음원파일자체가 아닌 음원코드를 각 클라이언트에게 전달함으로써 해결할 수 있다. 각 사용자는 하나의 채널을 부여하여 채널을 관리함으로써 다른 사용자 연주와 구별될 수 있도록 구현되었다. 연주가 시작된 상황에서 WiFi지역을 이탈하였을 경우 추가적인 데이터 요금 부가 경고메시지가 발생한다. 동의하였을 경우는 3G로 변형되어 계속 연주를 할 수 있도록 구현되었다.

이 애플리케이션이 소셜 네트워크의 다른 방안으로 변화되어지면 광고 마케팅 등 다양한 분야에서 애플리케이션이 창출되어 질 수 있다. 한편 채널 네트워크 응용 모델 적용 및 안드로이드 구조 및 자바프로그래밍, C/S 관련 교육 자료로 활용되며 그리고 미래 지향적인 악기 발명도 가능하다.

## 참 고 문 헌

[1] Verkasalo,H.,Lopez-Nicolas,C.,Molina-Castillo,F.J. & Bouwman,H, Analysis of users and non-users of smartphone application, 『Telematics and Information』,27:242-256. 2010.

[2] 김.민.식. 정현준. 휴대폰 산업의 탈추격형 대응전략 스마트폰을 중심으로 『방송통신정책』,22(1):1-16. 2010.

[3] 정종기, 스마트폰 애플리케이션을 활용한 국내외 대학도서관의 모바일 서비스에 관한 연구, 『한국 바이블학회지』,22(1):171-180. 2011.

[4] Frank Ableson Charlie Collins Robi Sen/김진석 · 국중진 옮김, “ 언로킹 안드로이드 ”, FREELEC, 2010.

[5] 김상형, “ 안드로이드 프로그래밍 정복 ”, 한빛미디어, 2010.5.10.

[6] SAYED Y. HASHIMI · SATYA KOMATINENI, 김지원 옮김 “안드로이드 마스터 북”,제이펍, 2010.

[7] 정석용, “ 정석용의 TCP/IP소켓 프로그래밍 ”, 주식회사 프리렉 , 2009.

[8] J. K. Yun, H. R. Lee, and K. R. Park, Multiple Devices Synchronization and Control Architecture for Orchestral Media Service System, IEEE,2009.



## 황 병 곤



1980년 : 경북대학교 대학원(공학석사)  
1990년 : 경북대학교 대학원 (공학박사)

1975년~1978년 해군장교

1978년~1979년 한국기계금속시험연구소 연구원

1980년~현재: 대구대학교 컴퓨터IT 공학부 교수

관심분야 : 영상처리, 컴퓨터그래픽스, 멀티미디어 콘텐츠