

상세 설계서

Leap Motion을 이용한 가상 악기 연주 및 음악 창작 공유 플랫폼 (Harmony)

Ver. 1.2

2018. 11. 15

한국외국어대학교

정보통신공학과

2팀(ENSEMBLE)

상세설계서: Leap Motion을 이용한 가상 악기 연주 및 음악 창작 공유 플랫폼 (Harmony)

문서 정보

구 분	소 속	성 명	날 짜	서 명
작성자	한국외국어대학교	김세용	2018. 11. 15	
	한국외국어대학교	김준영	2018. 11. 15	
	한국외국어대학교	김진우	2018. 11. 15	
	한국외국어대학교	김정민	2018. 11. 15	
	한국외국어대학교	유진솔	2018. 11. 15	
검토자	한국외국어대학교	김세용	2018. 11. 15	
	한국외국어대학교	김준영	2018. 11. 15	
	한국외국어대학교	김진우	2018. 11. 15	
	한국외국어대학교	김정민	2018. 11. 15	
	한국외국어대학교	유진솔	2018. 11. 15	
사용자				
승인자	한국외국어대학교	홍진표	2018. 11. 15	

상세설계서: Leap Motion을 이용한 가상 악기 연주 및 음악 창작 공유 플랫폼 (Harmony)

머리말

본 문서는 Leap Motion을 이용하여 가상 악기 연주 및 합주와 음악 창작 공유 플랫폼을 웹으로 제공하는 서비스들을 구축하기 위한 시스템들의 상세한 설계 기술을 설명한다.



목 차

목차

목 차	4
1. 개요	5
1.1 목적	5
1.2 범위	5
2. 시스템 구성	6
3. 기능 소개 및 설계	7
3.1 Arduino	7
3.2 Software	9
3.2.1 HM Software	9
3.2.2 Play Drum	10
3.2.3 Play Keyboard	12
3.2.4 Play LaunchPad	13
3.2.5 합주 모듈	14
3.3 Web Server & Database	15
4. 팀원 담당업무	17
5. 프로젝트 세부 추진 계획	18

1. 개요

본 장에서는 leap motion을 이용한 악기 연주 및 음악 창작과 공유 플랫폼인 Harmony에 대한 목적과 범위를 제시한다.

1.1 목적

본 프로젝트는 언제 어디서나 가상으로 악기 연주가 가능하고 여러 사용자의 연주 음악을 병합하여 새롭게 창작한 음악의 업로드, 다운로드가 가능하도록 하는 HM software를 설계한다.

이를 통하여 사용자가 직접 악기를 구매하지 않아도 HM software, leap motion, PC가 있는 환경이라면 언제든지 가상으로 드럼, 건반악기, 런치패드 연주가 가능하고 녹음 기능 또한 제공하여 자신의 연주를 녹음할 수 있는 환경을 구축하는 데 목적을 둔다.

마지막으로 HM web site에 음원 공유 기능을 제공하여 업로드 된 다른 사용자들의 음악을 다운로드 하여 자신의 음악과 병합이 가능하게 해주는 데에 목적이 있다.

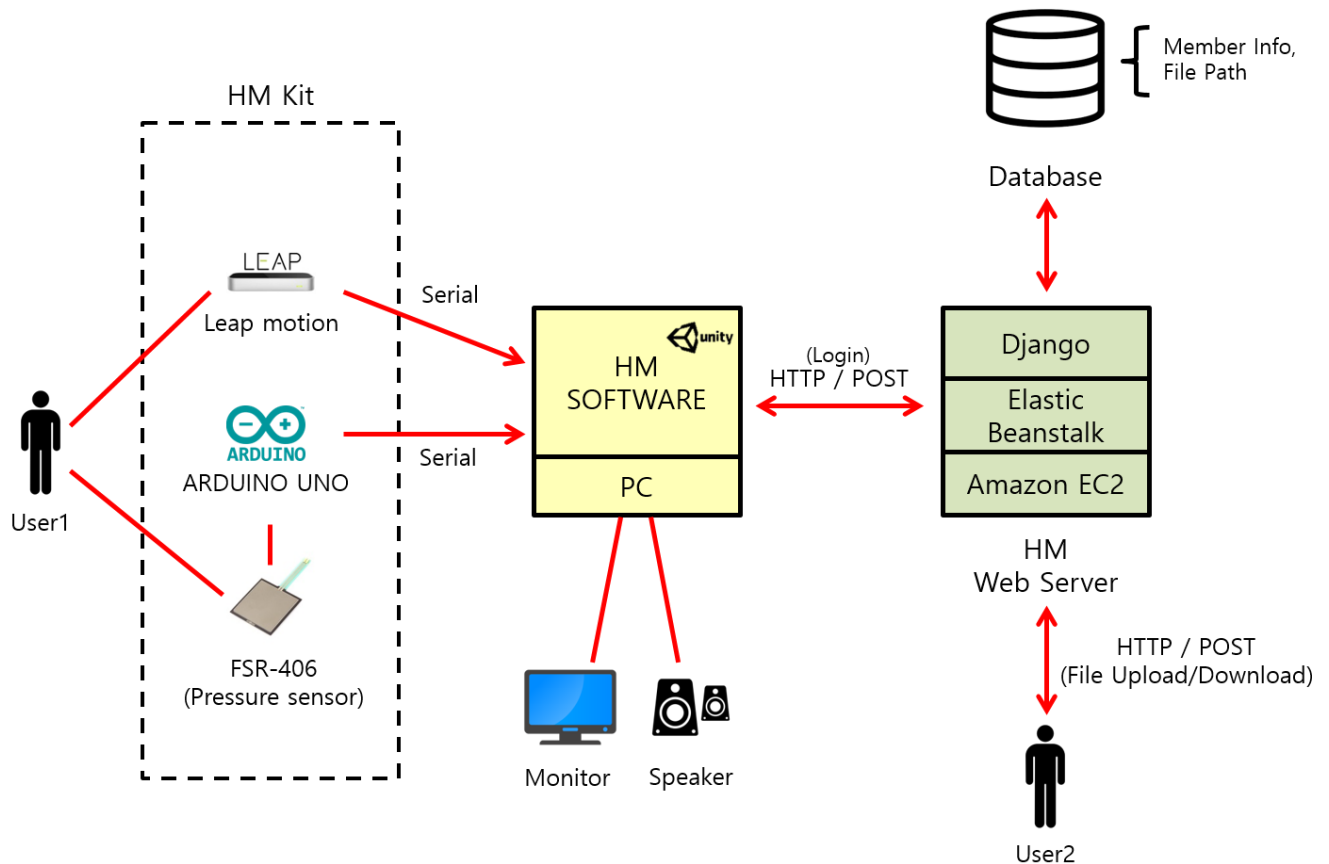
1.2 범위

‘Harmony’ 서비스는 leap motion, arduino와 압력센서로 구성된 HM kit을 이용하여 가상으로 악기를 연주하고, 음악을 창작하고 공유하는 환경을 제공한다.

본 프로젝트 개발 진행에 있어서 다음과 같은 범위를 둔다.

- Leap motion과 arduino를 이용한 악기 연주
: Leap motion으로 가상의 손을 화면에 띄우고, arduino로 압력 센서 값을 받아와 드럼에서는 킥 드럼의 역할로, 키보드에서는 서스테인 페달의 역할로, 런치 패드에서는 루프 스테이션의 역할로써 가상 악기 연주가 가능하다.
- 재생/정지, 녹음, mute/solo 기능으로 구성된 합주모듈 제공
: HM software를 실행하여 만든 악기 연주 음원 뿐만 아니라 외부의 어떤 음악 파일과도 mixing이 가능하다.
- HM website에서 음악 공유 기능 제공
: HM website에 업로드 되어 있는 음악들을 다른 사용자들과 공유할 수 있다.

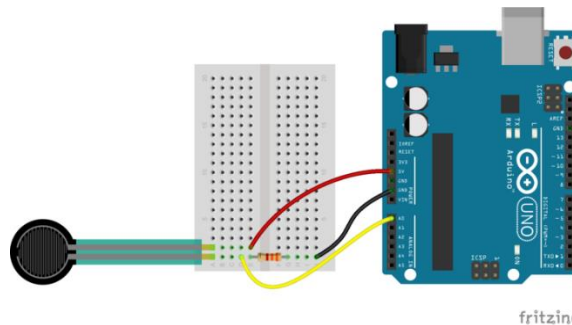
2. 시스템 구성



구성 요소	설명
User	User는 Harmony Service를 받는 대상이다.
HM Kit	HM Kit은 Leap Motion과 Arduino(FSR-406)으로 구성되어 있다.
HM Software	HM Software에서는 가상 악기를 연주하고 녹음 할 수 있으며 다른 곡과 합주를 할 수 있다.
HM Web Server	HM Server는 로그인 기능, 음원파일 업/다운로드 기능 등을 통해 음악 공유 플랫폼 서비스를 제공한다.
Database	Database는 HM Server와 정보를 주고 받는다.

3. 기능 소개 및 설계

3.1 Arduino



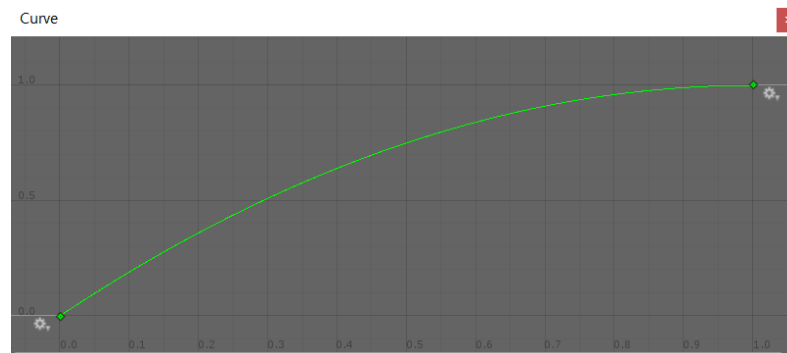
- 압력 센서(FSR-406) 회로도 -

▶ '아두이노를 이용한 드럼 Kick sound 출력' 요구사항(SFR-H-9)에 대한 상세 설계



위와 같은 페달에 FSR-406 센서를 붙여 발로 밟으면서 센서에 압력을 가한다. 아두이노에서 센서의 값을 받아와 입력 받은 값의 최댓값을 구해 변수에 저장한다. 아두이노에서는 Ardunity가 제공하는 ArdunityApp.process(); 함수를 통해 Unity에 Serial 통신으로 전달한다. Unity에서는 Ardunity가 제공하는 Analog Input component 로 아두이노에서 값을 받아온다. 값을 Ardunity에서 제공하는 Analog Input component 로 받아와 Ardunity가 제공하는 Mapping Input component를 통해 압력 센서의 값을 volume에 대한 값으로 logarithm mapping 시킨다.

이때, 사람의 청력이 logarithm 으로 진폭을 측정하기 때문에 volume에 대한 mapping도 logarithm 으로 한다. Unity에서 mapping 된 값을 parameter로 Kick sound를 출력하는 함수에 넘겨준다. Kick sound를 출력하는 함수는 parameter로 받은 값으로 volume을 설정하여 kick sound를 출력한다.



▶ ‘아두이노를 이용한 피아노 소리 Reverb 효과 출력’ 요구사항(SFR-H-11)에 대한 상세 설계



위와 같은 페달에 FSR-406 센서를 붙여 발로 밟으면서 센서에 압력을 가한다. 아두이노에서는 페달의 압력 유무를 Boolean 값으로 변수에 저장한다. Ardunity가 제공하는 ArdunityApp.process(); 함수를 통해 Unity에 Serial 통신으로 전달한다.

Unity에서는 Ardunity가 제공하는 Analog Input component 로 아두이노에서 값을 받아온다. 받은 값이 True이면 Unity에서 제공하는 Script API인 AudioReverbFilter를 이용하여 Reverb level 값을 최대치로 조절하여 Reverb 효과를 준다. False 값 일 때는 Reverb level 값을 0으로 조절하여 reverb 효과를 없앤다.

▶ ‘아두이노를 이용한 런치 패드 루프스테이션’ 요구사항(SFR-H-13)에 대한 상세 설계



위와 같은 페달에 FSR-406 센서를 붙여 발로 밟으면서 센서에 압력을 가한다. 아두이노에서는 압력 센서로부터 0이 아닌 값이 들어오면 Ardunity에서 제공하는 ArdunityApp.process(); 함수를 통해 Serial 통신으로 Unity에 ‘ON’값을 전달한다. ON 값을 받은 HM Software는 Record 기능이 활성화되어 녹음이 시작된다. 토크 방식으로 구현된 아두이노는 0이 아닌 값이 다시 한번 들어오게 되면 ‘OFF’를 전달하여 녹음을 종료한다.

녹음을 끝난 후 녹음된 파일을 unity 상에서 자동 재생시켜 주며 .loop를 통해 그 파일을 반복 재생 시켜준다.

3.2 Software

3.2.1 HM Software

HM Software는 LeapMotion과 Arduino가 연결된 PC에서 실행된다. Software상에서 악기를 선택하여 가상 공간에서 연주를 할 수 있으며 연주한 곡을 녹음하거나, 녹음된 연주 위에 새로운 연주를 얹어 각각의 연주들을 병합해주는 기능을 제공한다. 또한 HM Software는 키보드와 마우스 그리고 Leap Motion을 통해 생성된 가상의 커서(Cursor)로도 메뉴 조작 가능하다.

▶ ‘모션인식을 이용한 인터페이스 조작’ 요구사항(SFR-H-2)에 대한 상세 설계



Leap Motion의 Finger Class에서 검지 손가락(Index Finger)의 위치와 방향을 `finger.TipPosition()` 함수와 `selectedDirection(finger.TipPosition())` 함수를 통해 값을 받는다. 이 값은 `ToVector3()` 함수를 통해 벡터 값으로 변환한다. UI를 보여주는 가상의 공간에 위치한 2차원 Canvas 평면과 검지 손가락 끝의 직선 벡터의 교점을 구하여 Cursor를 생성한다.

클릭을 할 수 없는 가상의 커서(Cursor)는 `OnCollisionStay()`를 통해 UI BUTTON과 만나있음을 확인하고, 이 때 Timebar를 제공하여 Timeout이 되면 클릭 함수 (`OnButtonClick()`)를 호출하여 Button을 활성화 한다.

▶ ‘로그인/로그아웃 기능’ 요구사항(SFR-H-3)에 대한 상세 설계

사용자가 HM Web Site에서 ID와 PW를 치고 Login 버튼을 누르면 입력 받은 ID와 PW를 딕셔너리에 담아 WWWfrom 클래스의 인스턴스에 `form.AddField()` 하여 웹 서버에 전송할 양식 데이터를 생성한다.

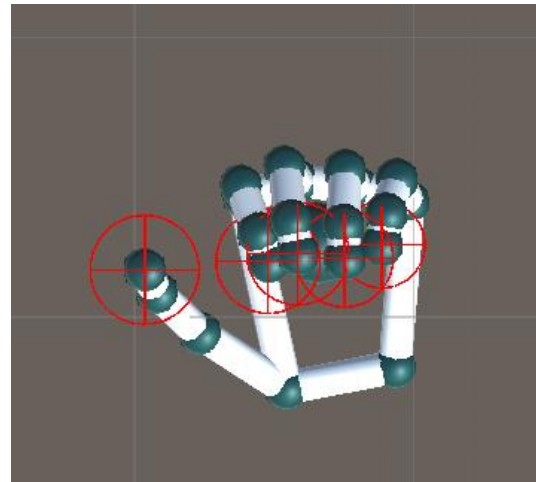
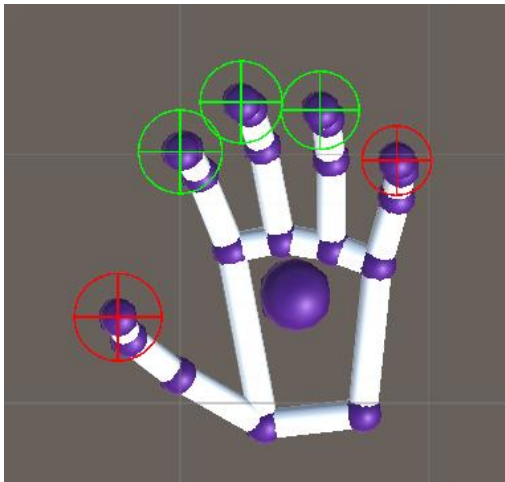
그 후 WWW클래스의 `WWW(url, form)` 형식으로 인스턴스를 생성하여 POST하여 Server로 ID,PW를 보낸다. Server는 ID와 PW가 유효한지 Database에 질의를 하여 검사 후 유효하면 사용자의 악기 구매 현황 정보를 전달해 주고 Software상에서 다음 Scene로 넘어간다. 또한 사용자가 로그아웃 버튼을 누르면 세션이 끊기고 Login Scene으로 넘어간다.

3.2.2 Play Drum

HM Software는 립 모션 위에서 주먹을 쥐면, 드럼 스틱 객체를 얻는다. HM Software는 얻어진 스틱 객체를 이용하여, 가상환경에 띄워진 드럼 객체들을 연주할 수 있다. HM Software의 드럼은 심벌 3개와 북 4개를 가지며 드럼 스틱으로 내려치는 세기에 따라 악기 볼륨이 조절된다.

▶ '드럼 스틱 조작' 요구사항(SFR-H-7)에 대한 상세 설계

ExtendedFingerDetector 함수를 통해 각 손가락 정보(thumb, index, middle, ring, pinky)들의 상태를 stateArray에 담아 펼쳐진 손가락의 개수를 판단한다. Extended finger의 개수가 0이 되면 주먹을 쥐 것으로 activate하고 가상의 손에 가상 드럼 스틱 객체를 생성하여 쥐어준다.



▶ '드럼 악기 연주 제공' 요구사항(SFR-H-8)에 대한 상세 설계

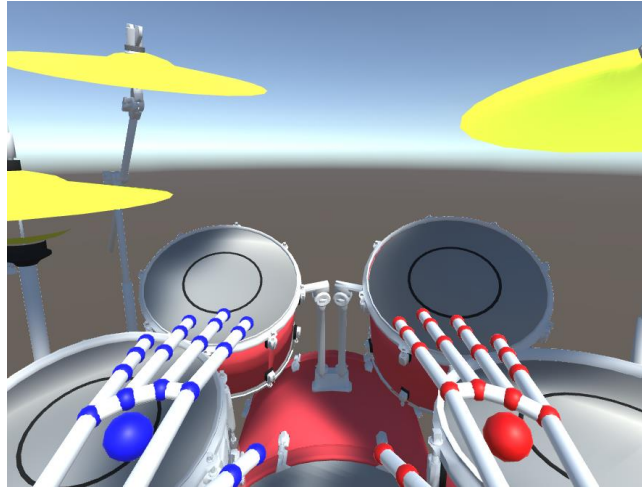
1. 연주방식

HM Software는 사용자의 손을 립 모션을 통해 인식하여 가상 환경에 가상 손 객체를 생성한다. HM Software는 드럼스틱 객체와 가상환경에 구현된 드럼 객체들의 충돌을 감지하여 사운드를 출력한다. 드럼 스틱과 각 드럼 객체 간의 충돌에 의한 사운드 출력은 드럼 스틱 객체의 rigidbody컴포넌트와 드럼 객체의 mesh collider의 충돌을 감지하여 일으킨다. 이때 드럼의 세기는 사용자가 휘두르는 드럼 스틱 객체와 드럼 객체의 상대적인 속도를 감지하여, 그 값에 비례하게 사운드를 출력한다.

```
audioSource.volume = collision.relativeVelocity.magnitude
```

상세설계서: Leap Motion을 이용한 가상 악기 연주 및 음악 창작 공유 플랫폼 (Harmony)

2. 악기 구조



HM Software는 다음과 같이 1인칭 시점에서 가상 악기를 보여준다. HM Software는 LeftTom, RightTom, Snare, FloorTom, Hi-hat Cymbal, Crash Cymbal, Ride Cymbal으로 드럼을 구성한다. 악기의 현실성을 높이기 위해 Cymbal들에 물리적 특성 중의 하나인 spring joint를 드럼에 적용한다. 이를 통해 HM Software는 치는 방향과 세기에 따라 다른 움직임을 보여준다.

3. 세기

소리의 세기는 정적인 드럼 객체와 동적인 드럼스틱 객체의 상대 속도에 비례하여 출력한다.

```
audioSource.volume = collision.relativeVelocity.magnitude
```

3.2.3 Play Keyboard

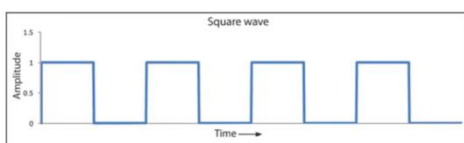
▶ '건반 악기 연주 제공' 요구사항(SFR-H-10)에 대한 상세 설계

1. 립모션을 통해 인식되어 생성된 가상 손 객체로 건반을 쳐서 연주가 가능하다. 건반은 실제 건반악기와 흡사하게 눌러지며 일정 각도 이상이 되면 소리를 낸다. 또한 건반이 눌리는 속도에 의해 사운드의 볼륨이 조절된다..
 - ① 건반과 손 object에 물리 제어로 동작하게 하는 rigidbody component와 mesh형태를 따라서 콜라이더를 생성한다. 건반들 간에 충돌을 방지하기 위해
Physics.IgnoreLayerCollision(LayerMask.NameToLayer("piano"), LayerMask.NameToLayer("piano"), true)를 적용한다.
 - ② transform.rotation.eulerAngles를 이용하여 건반이 눌렸을 때 회전 각도를 알아내고 transform.rotation=Quaternion.Euler()를 이용하여 눌러지는 각도를 설정하고, 눌러진 뒤 원래 상태로 회복되도록 한다.
 - ③ 충돌이 일어났을 경우 실행되는 OnCollisionEnter 함수에서 transform.rotation.eulerAngles.z, 즉 z축의 회전 각도가 1.5를 넘어가면 소리를 출력한다.
2. 건반 악기는 2가지 종류의 건반악기 사운드와 소리를 커스텀하게 조절이 가능한 신디사이저를 제공한다.
3. 2가지 종류의 건반악기 사운드는 changeSound 함수를 이용하여 각 버튼을 누르면 모든 건반에 mapping된 소리가 버튼의 키보드 종류에 맞게 바뀐다.
4. 신디사이저는 Triangular Wave, Sine Wave, Square Wave 3가지 파형을 기반으로 주기, 진폭, 위상 등을 조절하여 다양한 사운드를 만든다.

Square Wave

Replace "data[i] = (float)(gain * Mathf.Sin((float)phase));" with:

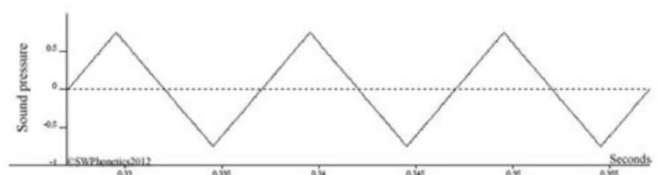
```
if (gain * Mathf.Sin((float)phase) >= 0 * gain)
{
    data[i] = (float)gain * 0.6f;
}
else
{
    data[i] = -(float)gain * 0.6f;
}
```



Triangle Wave

Replace "data[i] = (float)(gain * Mathf.Sin((float)phase));" with:

```
data[i] =
(float) (gain * (double) Mathf.PingPong ((float) phase, 1.0f));
```



3.2.4 Play LaunchPad

HM Software는 사용자의 런치패드 악기 선택이 일어난 후, 화면에 런치패드 악기와 립 모션 위의 사용자 손을 인식하여 만들어진 가상의 손을 보여준다. HM Software는 가상환경의 손 객체와 런치패드 객체 간의 충돌을 감지하여 사운드를 제공하고, 런치패드를 치는 손의 속도를 감지하여 사운드 볼륨을 조절한다. 연주가 일어나는 중에도 임의로 개체에 속해 있는 음악파일이 변경될 수 있고, 루프 스테이션을 통해 다양한 연주가 가능하다.

▶ ‘런치패드 악기 연주 제공’ 요구사항(SFR-H-12)에 대한 상세 설계

1. 연주 방식

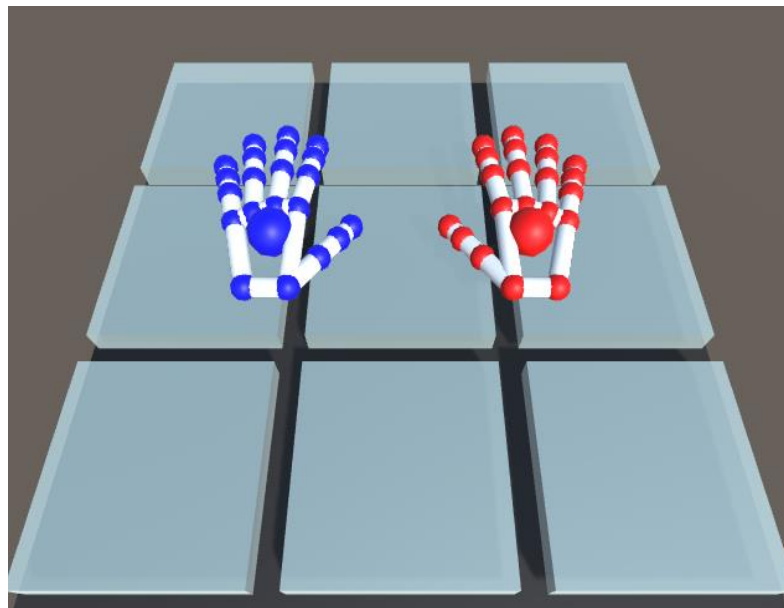
Leap motion을 통해 사용자의 손을 인식하고 가상환경에 손 객체를 생성한다. Leap motion이 인식한 물리적 성질을 갖는 손가락인 rigidfinger에 런치패드 객체의 콜라이더와 반응할 tag를 심어준다.

```
if (bones[i] != null) {  
    CapsuleCollider capsule = bones[i].GetComponent<CapsuleCollider>();  
    bones[i].tag = "target";  
}
```

런치패드의 mesh collider와 손 객체의 target tag가 충돌했다는 값을 받으면 sound를 출력한다.

```
if (collision.gameObject.CompareTag("target"))  
    audioSource.PlayOneShot(play, audioSource.volume);
```

2. 악기 구조



HM Software는 다음과 같은 런치패드의 1인칭 연주화면을 제공한다. 총 9개의 연주 패드가 있고 각각의 패드에는 사용자의 선택에 따른 audiosource를 받고 그에 따라 사운드를 출력한다. 메뉴에 있는 또 다른 소스들이 패드의 소스들과 변경되어 다양한 사운드들이 출력된다.

3. 세기

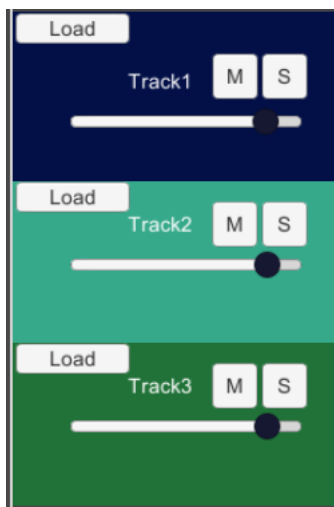
소리의 세기는 정적인 런치패드 객체와 동적인 손 객체의 상대 속도에 비례하여 출력한다.

```
audioSource.volume = collision.relativeVelocity.magnitude
```

3.2.5 합주 모듈

합주 모듈은 각 악기의 연주화면에 숨어 있다가 연주화면의 좌측 상단에 합주 모듈 버튼을 클릭 할 때 나타난다. 합주 모듈은 사용자가 선택한 외부 음원 파일을 SW에 불러와 Mixer기능을 통해 생성된 트랙에 일대일 Mapping시킨다. 각 트랙을 독립적으로 제어(MUTE기능, SOLO기능, Volume Control기능) 할 수 있는 기능을 가지고, 이를 바탕으로 편리하게 사용자의 연주와 합주 할 수 있게 하고, 녹음 기능을 통해 외부 음원에 사용자의 연주가 덧입혀진 하나의 wav파일을 생성하여 사용자에게 제공한다.

▶ ‘사용자가 원하는 음원 파일 load기능 제공’ 요구사항(SFR – H -15)에 대한 상세 설계

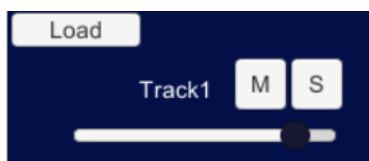


1. 그림과 같이 트랙 별로 Load버튼을 생성한다. Load버튼 클릭 시에 Unity의 OpenFileDialog() 함수를 실행시켜 File Browser를 띄우고, 사용자의 파일 선택에 의해 해당 파일의 Path를 얻어낸다.

2. 얻어낸 파일의 Path를 Unity의 www객체에게 Parameter로 전달하고, www객체는 Path를 바탕으로 파일을 다운로드하여 GetAudioClip()함수를 통해 SW에서 사용할 수 있는 오디오 파일로 불러온다.

3. 불러온 오디오 파일을 해당 트랙의 AudioClip으로 추가하여 오디오를 사용자가 제어할 수 있도록 한다.

▶ ‘트랙 별 독립적 제어 기능 제공’ 요구사항(SFR – H - 16)에 대한 상세 설계



트랙 별 독립적 제어를 하기 위해, 기본적으로 3개의 트랙을 Mixer에서 생성한다.

- Mute 기능

1. 그림과 같이 트랙 별로 Mute버튼을 생성한다. Mute 버튼 클릭 시에 해당 트랙의 현재 Volume을 저장하고 Volume을 Unity Mixer상의 최솟값인 -80으로 변경한다.
2. Mute 버튼을 한 번 더 클릭 시에 저장했었던 Volume으로 변경한다.

- Solo 기능

1. 그림과 같이 트랙 별로 Solo버튼을 생성한다. Solo 버튼 클릭 시에 나머지 트랙의 현재 Volume을 저장하고 Volume을 Unity Mixer상의 최솟값인 -80으로 변경한다.
2. Solo 버튼을 한 번 더 클릭 시에 나머지 트랙의 Volume을 저장했었던 Volume으로 변경한다.

- Volume Control 기능

1. Volume Slider를 생성하고, Mixer에서의 해당 트랙 Volume Fader와 Mapping하여 Slider를 조절하면 트랙의 볼륨이 조절되도록 한다.

▶ ‘소프트웨어 내부 소리 녹음 기능 제공’ 요구사항(SFR-H-17)에 대한 상세 설계

1. Unity Main Camera에 AudioListener는 소프트웨어에서 나는 모든 소리를 출력해주는 역할을 하므로 AudioListener에서 출력되는 사운드를 wav파일로 변환하는 Script를 Component로 추가한다.
2. Script는 OnAudioFilterRead를 통해 오디오 청크(chunk)가 해당 필터에 통과될 때마다 오디오 데이터를 받아들이고 FileStream.write를 이용해 이를 wav파일로 변환하는 함수를 추가한다.

3.3 Web Server & Database

HM Web Server에서는 HM Web Site에서 사용자가 회원가입, 로그인, 음악 파일 업/다운로드, 음악 재생 등의 요청을 했을 시 요청에 대한 응답을 해주는 기능을 제공한다. 또한 HM Software에서 사용자가 로그인 한 후 악기를 선택 할 때 자신이 구매한 악기 내에서 선택하도록 제한하는 기능을 담당한다.

▶ ‘회원가입’ 요구사항(SFR-W-1, SFR-H-1)에 대한 상세 설계

사용자가 HM Web Site에서 회원가입 버튼을 누르면 Server는 회원 가입을 할 수 있는 페이지의 HTML file을 띄워준다. 사용자는 ID, PW, Serial Number를 입력하고 해당 내용이 유효할 경우 Database에 ID, PW를 INSERT한다. Member 테이블 구조는 다음과 같다.

ID	PW	Drum	Keyboard	LaunchPad
----	----	------	----------	-----------

Drum, keyboard, LaunchPad 애트리뷰트는 사용자가 악기를 구매했는지의 여부를 기입한다.

상세설계서: Leap Motion을 이용한 가상 악기 연주 및 음악 창작 공유 플랫폼 (Harmony)

▶ ‘로그인/로그아웃 기능’ 요구사항(SFR-W-2)에 대한 상세 설계

사용자가 HM Web Site에서 ID와 PW를 치고 Login 버튼을 누르면 Server는 입력 받은 ID와 PW가 유효한지 Database에 질의를 하여 검사 후 유효하면 로그인 세션이 연결된다. 또한 사용자가 로그아웃 버튼을 누르면 세션이 끊기고 홈 화면 HTML 파일을 띄워준다.

▶ ‘음악파일 업/다운로드 기능’ 요구사항(SFR-W-4)에 대한 상세 설계

사용자가 audio file을 upload 요청하면 시스템은 filetransfers API를 사용하여 parameter를 통해 파일을 server에 저장하기 위한 파일경로와 파일 확장자를 받아와 데이터베이스의 Audio테이블에 저장한다. 이 후 POST method로 서버에 파일을 전송 받는다.

Audio 테이블 구조는 다음과 같다.

PostNum	Path	FileName	FileExtension
---------	------	----------	---------------

Audio file을 download 하는 기능을 FileResponse의 parameter로 as_attachment = True로 설정, content type을 지정해주어 Server가 저장한 audio file을 제공한다. 사용자가 audio file을 downloading 할 수 있게 html 파일을 통해 UI를 제공한다.

▶ ‘음악 파일 재생’ 요구사항(SFR-W-5)에 대한 상세 설계

음원 파일은 HTML <audio> 태그를 통해서 Browser의 기본 Player로 재생된다



상세설계서: Leap Motion을 이용한 가상 악기 연주 및 음악 창작 공유 플랫폼 (Harmony)

4. 팀원 담당업무

이름	정	부	비고
김세용	Software 구축 & UI	Web Sever Database	
김진우	드럼 구현 런치패드 구현	Software UI	
김준영	합주 모듈 구현	건반 악기 구현	
김정민	Arduino Web Server	Database	
유진솔	건반 악기 구현	런치패드 구현	



상세설계서: Leap Motion을 이용한 가상 악기 연주 및 음악 창작 공유 플랫폼 (Harmony)

5. 프로젝트 세부 추진 계획

구분	내용	9주차	10주차	11주차	12주차	13주차	14주차	15주차
설계	Server 기본구성							
	Server 핵심기능							
	WebSite 기본구성							
	Web 디자인							
	Software 기본구성							
	Software 디자인							
	드럼							
	런치패드							
	건반악기							
	Arduino							
테스트	악기보완							
	점검							
종료	최종							