

종합설계 프로젝트 수행 보고서

프로젝트명	AR기술을 활용한 전자 드럼
팀번호	S5-6
문서제목	수행계획서(O) 2차발표 중간보고서(O) 3차발표 중간보고서() 최종결과보고서()

2021.03.05

팀원 : 고현석 (팀장)

김태용

김현수

박승찬

지도교수 : 전광일 교수

지도교수 : 최진구 교수

문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2022.01.03	고현석(팀장)	1.0	수행계획서	최초작성
2022.03.05	박승찬	2.0	2차발표자료	설계서추가

문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (3월)	중간발표2 (5월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I
	II. 본론 (1~3)	II. 본론 (1~4)	II. 본론 (1~5)	II. 본론 (1~7)	II
	참고자료	참고자료	참고자료	참고자료	III

이 문서는 한국산업기술대학교 컴퓨터공학부의 “종합설계”교과목에서 프로젝트
 “AR기술을 활용한 전자 드럼”을 수행하는
 (S5-6, 고현석, 김태용, 김현수, 박승찬)들이 작성한 것으로 사용하기 위해서는
 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성
2. 기존 연구/기술동향 분석
3. 개발 목표
4. 팀 역할 분담
5. 개발 일정
6. 개발 환경

II. 본론

1. 개발 내용
2. 문제 및 해결방안
3. 시험시나리오
4. 상세 설계
5. Prototype 구현
6. 시험/ 테스트 결과
7. Coding & DEMO

III. 결론

1. 연구 결과
2. 작품제작 소요재료 목록

참고자료

서론

1. 작품 선정 배경 및 필요성

코로나 19로 인하여 실내 활동이 증가하고 문화생활은 제한되고 있다. 많은 사람들이 여가생활로 악기를 다루지만 드럼은 설치 공간, 가격, 소음 등으로 인하여 접근성이 다소 떨어진다. 그래서 나온 것이 전자드럼이다.

하지만 전자드럼은 드럼과 구성은 비슷하지만, 전문적으로 가르치는 커리큘럼이나 흥미는 실제 드럼보다 떨어진다. 따라서 초보자들이 전자드럼을 흥미롭고 쉽게 칠 수 있는 방법으로 AR 기술을 이용해 박자에 맞춰 치는 부분을 시각화함으로써 흥미도와 접근성을 도와주고자 한다.

2. 기존 연구/기술동향 분석



■ AR 피아노 Tutor (on Oculus Quest)

- (소개) AR 기술을 이용한 피아노 연습 어플.
- AR로 구현된 막대를 리듬게임처럼 타이밍에 맞게 알맞은 건반을 눌러서 연주하게 하여 피아노 연습을 도와주는 어플이다.



■ 비트세이버

- (소개) VR 안에서 비트에 맞춰 블록을 깨는 리듬 게임.
- 광선검을 이용해 리듬에 맞춰 나오는 블록을 깨며 점수를 얻는 게임이다.



■ Moplay smart drum

- (소개) 모바일 상에서 악보에 맞춰 드럼을 치는 게임.
- 악보에 맞게 내려오는 막대들에 맞게 터치를 하면 드럼이 연주되는 방식

3. 개발 목표

전자 드럼을 연주해 본 경험이 없거나 익숙하지 않은 유저들에게 애플리케이션을 실행해서 연주를 가능하도록 만들어주는 Android 기반 소프트웨어로서, 실제 드럼 연주의 공간적 제약을 넘어서서 작은 크기의 전자 드럼을 원하는 장소에서 모바일 기기만 있다면 연주할 수 있고 난이도별 음악을 선택하여 Score 기능을 제공하기 때문에 단순한 연주가 아닌 재미 요소까지 함께 가져갈 수 있는 Software 개발을 목표로 두고 있습니다.

상세 목표

Aa 구분	≡ 내용	≡ 설명
<u>Augmented Reality(AR)</u>	AR 연주	모바일 기기 카메라나 HMD 장치를 통해서 드럼을 연주할 때, 정확한 박자에 어느 위치를 쳐야 할지 알 수 있습니다.
<u>DRUM Basic Strok</u>	기본 연주	드럼 연주에 익숙하지 않은 유저들을 위해서 기본 박자를 익힐 수 있습니다.
<u>실시간 영상 처리</u>	악보 읽어오기	악보를 웹 서버로부터 받아와서 음악에 맞게 실시간으로 악보를 확인하면서 연주할 수 있습니다.

4. 팀 역할 분담

	고현석	김태용	김현수	박승찬
자료수집	악보변환 알고리즘	알고리즘 사용 라이브러리 조사	유니티 안드로이드 연계	유니티 리듬 게임
설계 및 구현	악보변환 알고리즘	데이터 변환 및 설계	Android 설계 및 유니티 구현	유니티로 박자에 맞게 게임 제작
테스트	기본 스트록 테스트 녹음기능 테스트 데이터 변환 테스트 악보변환 테스트			

5. 개발 일정

활동 내용	12월				1월				2월				3월			
	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주
[기획]																
팀 구성	●															
아이디어 회의 및 정리	●	●	●													
개발 아이디어 선정			●	●												
과제명 선정				●												
[요구사항 도출]																
기존 연구/기술 조사 및 분석				●	●											
작품 배경 및 필요성 분석				●	●	●										
제안 사항 도출					●	●	●									
[기본설계 및 상세설계]																
구현(개발)해야 할 내용 분석							●	●	●							
시나리오 설계								●	●	●	●					
시스템 구조 설계								●	●	●	●					
최종기획 보고서 완료												●	●			

6. 개발 환경

- 개발 언어 : python3.5, Kotlin, C#
- 사용 프레임워크 : Unity, Android(7~12)
- 주요 라이브 : Music 21, AR Core
- 개발 방법 : Scrum Framework
- 전자 드럼: SERD-1

본론

1. 개발 내용

	제안사항
Frontend	<ul style="list-style-type: none"> • Unity를 이용하여 드럼 구성별 스트록을 타격 타이밍에 맞게 화면에 표기한다. • 화면에 표기된 스트록을 칠 타이밍에 따라 정확도 분류한다. • Android를 이용하여 메뉴, 프로필, 모드선택등의 전반적인 UI 디자인한다. • 본인의 연주를 리뷰할 수 있게 녹음기능을 연주완료 후 선택 가능하게 한다.
알고리즘	<ul style="list-style-type: none"> • Python을 이용하여 악보에서 드럼 타겟별 데이터 추출 & 변환하여 이용한다.

2. 문제 및 해결방안

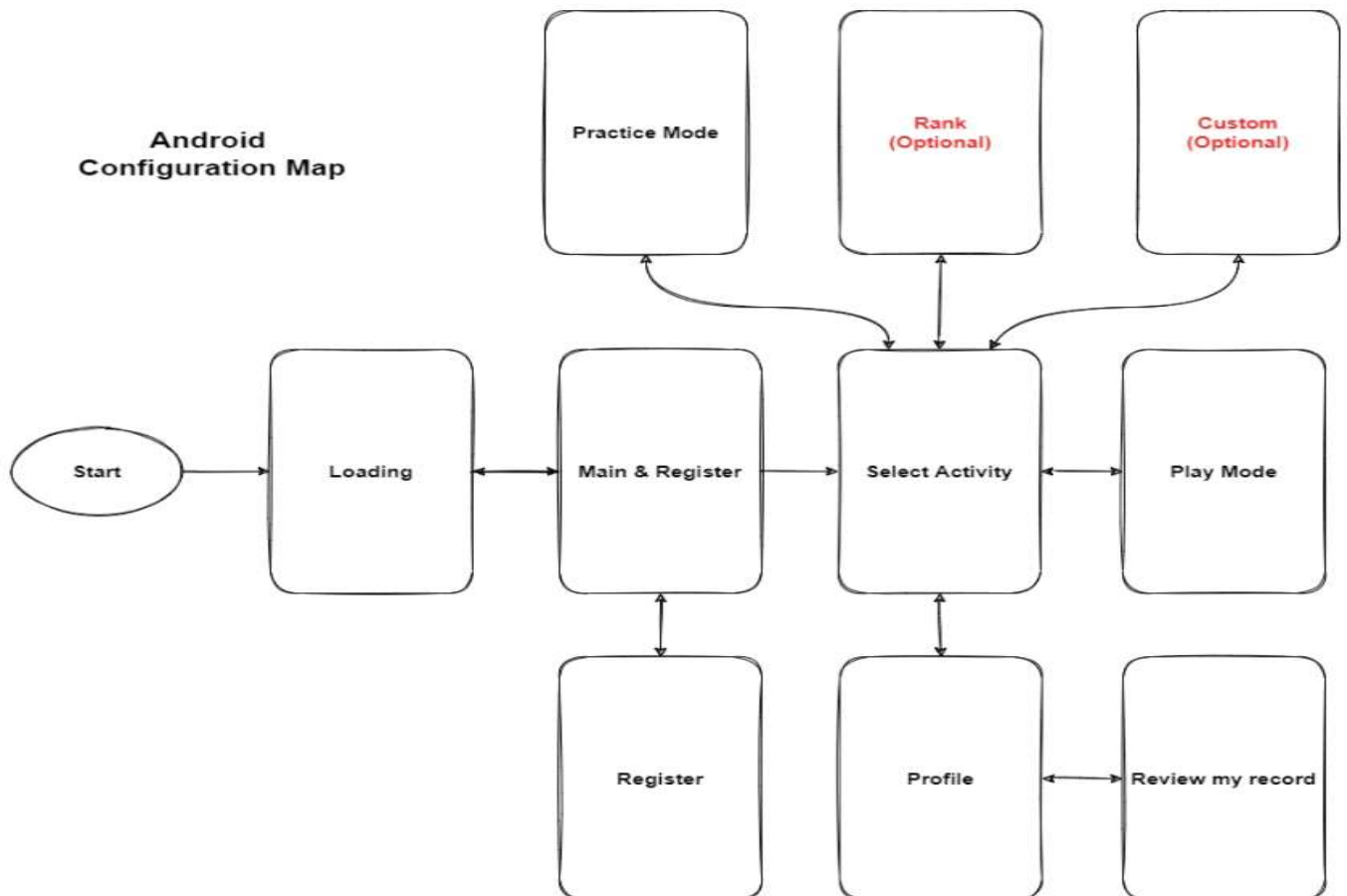
순번	제안사항
1	■ 기본 스트록을 추가 <ul style="list-style-type: none"> • 기본 스트록 추가로 인해 드럼 초심자가 드럼을 배울 수 있는 기회 제공 • 기본적인 몸 풀기가 가능
2	■ 녹음기능 <ul style="list-style-type: none"> • 녹음기능을 추가하여 본인의 연주를 리뷰할 수 있도록 한다. • 메모리 크기를 생각하여 원할 때에만 녹음을 할 수 있도록 한다.
3	■ 악보를 게임으로 변환 <ul style="list-style-type: none"> • 원하는 드럼 악보를 이미지로 입력하면 게임을 변환할 수 있다. • 드럼 악보를 입력하여 원하는 연주를 연습해볼 수 있다.
4	■ 진행 속도 변환 <ul style="list-style-type: none"> • 음악과 게임의 진행속도를 조절하여 원하는 난이도로 드럼 연주를 연습할 수 있도록 한다.

3. 시험 시나리오

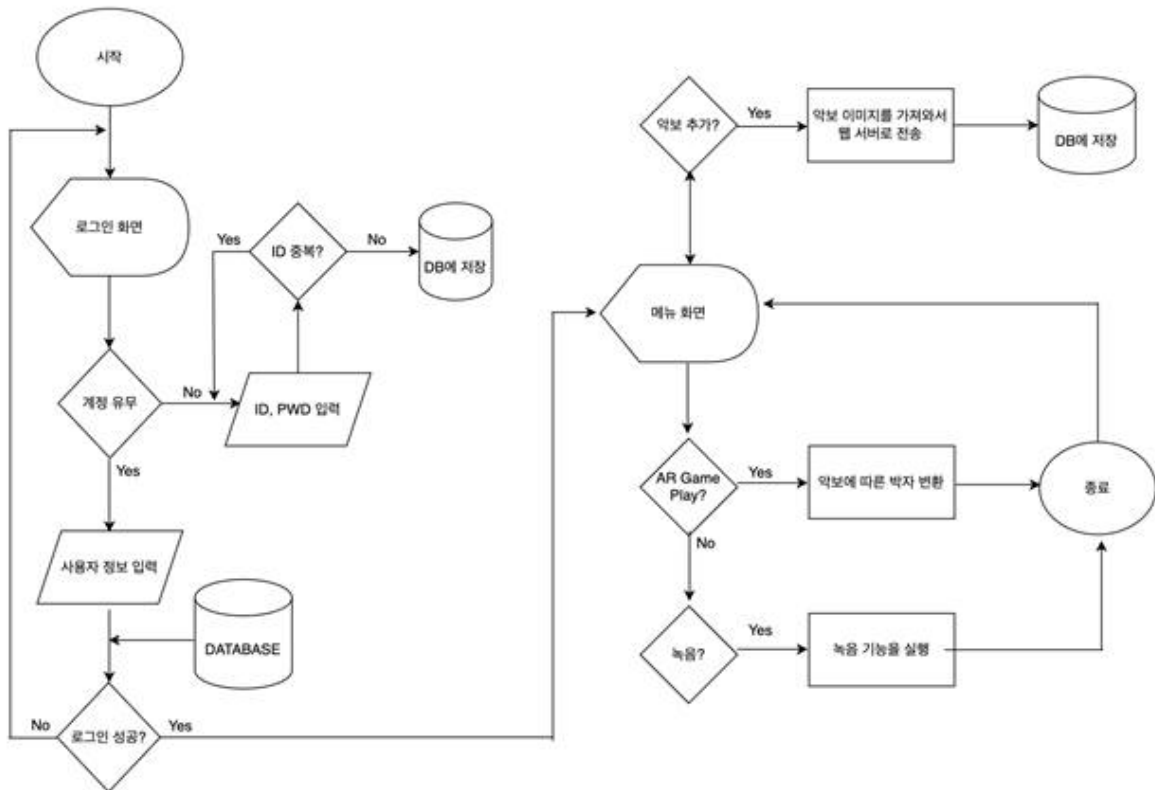
시나리오 명	내용
드럼 인식	<ul style="list-style-type: none"> 핸드폰 카메라를 통해 드럼을 인식하여 AR이 작동되는지 확인함.
기본 스트록	<ul style="list-style-type: none"> 인식한 드럼위에 기본적인 스트록을 떨어트려 사용자가 연습을 할 수 있도록 함. 난이도에 따라 원하는 기본 스트록을 정해서 연습할 수 있도록 함.
연주 녹음	<ul style="list-style-type: none"> 사용자가 원하는 노래를 연주 후 자신의 연주를 확인하기 위해 녹음 기능을 사용할 수 있도록 함.
악보 입력	<ul style="list-style-type: none"> 원하는 악보를 입력하여 스트록으로 데이터를 변환 하여 원하는 노래를 연습할 수 있도록 함.
연주 속도 조절	<ul style="list-style-type: none"> 연습하는 노래의 속도를 변환하여 난이도를 조절할 수 있다.

4. 상세 설계

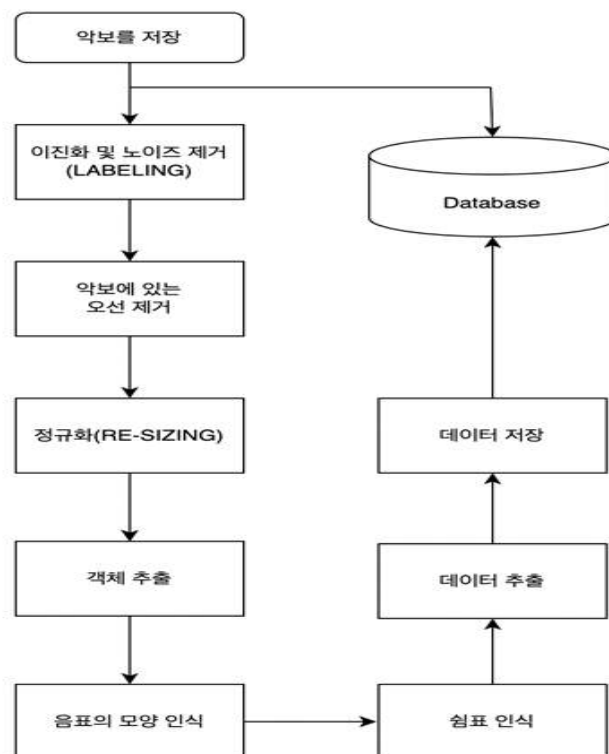
4-1 Android Configuration Map



4-2 Flow chart(Android App)



4-3 Flow chart(악보 변환 알고리즘)



4-4 기본 스트록

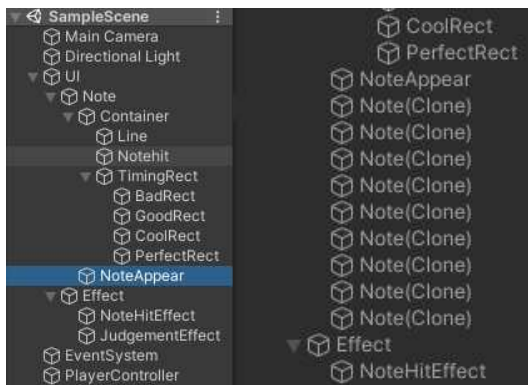
- 연습을 위한 기본 스트록을 구성
- 기본 스트록을 난이도에 따라 여러 단계로 나누어 실력을 늘려갈수록 자신의 실력에 맞는 연습을 할 수 있도록 구성

```

24 //노트를 hit할 경우 안보이게 하기.(이미지 없애서)
25 public void HideNote()
26 {
27     noteImage.enabled = false;
28 }

```

- 노트가 메모리를 계속 차지 하기 때문에 생성 후에 안보일 경우 삭제 해주기.



4-5 연주 속도 조절

```

5 public class Note : MonoBehaviour
6 {
7     public float noteSpeed = 400;
8
9     UnityEngine.UI.Image noteImage;

```

```

5 public class NoteManager : MonoBehaviour
6 {
7     public int bpm = 0;
8     double currentTime = 0d;
9
10    [SerializeField] Transform tfNoteAppear = null;
11    [SerializeField] GameObject goNote = null;
12
13    TimingManager theTimingManager;
14    EffectManager theEffectManager;
15
16    void Start()
17    {
18        theEffectManager = FindObjectOfType<EffectManager>();
19        theTimingManager = GetComponent<TimingManager>();
20    }

```

- 사용자가 쳐야할 노트의 속도를 조절하는 변수와 노래의 bpm을 입력받는 변수
- 이후에 나오는 노트들의 속도와 노래의 bpm을 조절 또는 입력하여 사용자가 연습할 때의 난이도를 조절 할 수 있다.

```

19 void Update()
20 {
21     transform.localPosition += Vector3.right * noteSpeed * Time.deltaTime;
22 }

```

```

24 void Update()
25 {
26     currentTime += Time.deltaTime;
27
28     if(currentTime >= 60d / bpm)
29     {
30         GameObject t_note = Instantiate(goNote, tfNoteAppear.position, Quaternion.identity);
31         t_note.transform.SetParent(this.transform);
32         theTimingManager.boxNoteList.Add(t_note);
33         currentTime -= 60d / bpm;
34     }
35 }

```

