

# Cert-is

## Study report

### Android 스터디 3~4 월 보고서

---

Cert\_is

한 창 훈

2016년 4월 30일

## - 스터디 인원 -

- 6 명 (스터디장 포함)

### 1. 이름

- IT 12 한창훈
- IT 12 권대영
- IT 12 류영승
- 컴공 12 조호열
- 컴공 17 장정윤
- 컴공 17 이호정

## 2. 3~4 월차 스터디 보고서

### 2-1 보고서\_한창훈 (스터디 진행사항 기록)

1주차 3/20	첫 스터디 날이었으므로 OT를 진행하였다. 스터디 원들끼리의 첫 만남과 구체적인 목표설정 (학습동아리 및 해커톤), 개발환경 통일(Android Studio, JDK) 및 교재 선정 (Step by Step 안드로이드 프로그래밍)을 진행하고 종료 하였다.
2주차 3/27	프로그래밍에 익숙하지 않은 스터디들을 위해 간단한 언어별 차이 언급 및 웹 전반적인 흐름에 대하여 설명(많이 사용하는 언어를 base로 하여서 Front-end와 back-end의 개념, Client와 Server간의 connection에 대하여 개념적 설명.  XML에 대한 개념을 설명하고, Layout에 대하여 설명함(교재의 예제 코드를 활용하여서.) 또한 XML의 구성요소의 종류에 대해서 설명해주었고 각 구성요소에서 내부에 필요한 정보에 대하여 설명함 (ex id값, weight값 등등..)
3주차 4/3	프로그래밍에 숙달되지 않은 스터디 원들을 위하여 C언어와, 지금까지 배운 점들에 대해서 숙지하고 있는지, 간단하게 질문과 답변 시간을 가짐. 또한 앞으로 진행될 '객체 지향프로그래밍'에 대한 개념과 어플리케이션 개발 폴더 구조를 살펴보았다. '로딩'과 '하드코딩'에 대한 차이점을 알려주었으며 '로딩'을 통해서 사용하는 것이 효율적이란 것을 설명함. manifest, activity, xml에 대하여 알려주었고, 그 다음주차 과제로 교재 ~195page까지 보기 + 해당 페이지에 있는 모든 예제 실제로 하고 결과물 가져오기를 과제로 내어주어 실제 프로그래밍에 들어가게 되었음.
4주차 4/24	안드로이드의 생명주기(life cycle)에 대한 설명과 중요성을 공부하였으며. 마찬가지로 지금까지 배운 것들에 대해 질문 답변 시간을 가졌다. JAVA 프로그래밍을 경험해보지 못한 1학년 후배들은 객체와 클래스, 속성과 메서드에 대한 이해도가 부족하였기에 '상속' 개념을 포함하여서 같이 학습을 하였으며. 저번시간에 내 준 과제를 위주로 스터디를 진행 하였다.

모든 스터디원이 교재를 읽어왔고, 실습 예제를 해왔다. 처음이었기 때문에 오류를 수정하지 못하는 경우도 많았지만, 스터디 시간에 같이 해결 해 보면서 오류의 해결법에 대한 솔루션도 제시해 줄 수 있는 시간이었다. 시험이 끝나지 않은 시점에서 부담이 될 수 있었을 텐데 모든 과제를 하고 스터디에 참가해준 스터디 원들이었기 때문에 다음 주는 197~304page까지 짧은 분량의 교재 과제를 내 주었고, 추가적으로 Fragment라는 요소에 대해서 학습해보자고 하였다.

분량까지의 예제 (3개)

1. 제자리에서 회전하는 ‘바람개비’ 만들기

- Layout폴더에 이미지를 넣어 실제로 불러오고 rotation 해주는 예제

(책과 완벽히 같은 코드이고, 아래 스터디 원들이 넣었으므로 생략)

2. 터치를 하면 위치를 따라오며 진동을 주는 이미지를 만들기

- onTouchEvent를 활용하여 사용자가 이미지를 터치할 때의 좌표를 입력받고, 진동을 불러오는 코드를 작성하였다.

(책과 완벽히 같은 코드이고, 아래 스터디 원들이 넣었으므로 생략)

3. string.xml에 ‘시’에 관련된 항목을 넣고 ‘로딩’하는 방법 익히기

- xml에서 ‘textview’에서 ‘하드코딩’ 하여 직접 데이터를 입력하는 방법이 아니라 string.xml에 정보를 입력하고 ‘로딩’하는 방법에 대해 숙달하였음

(책과 완벽히 같은 코드이고, 아래 스터디 원들이 넣었으므로 생략)

## 2-2 보고서\_권대영

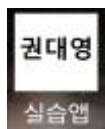
### 1. Android Studio 설치 및 기본 사용법 공부

app 개발 환경과 app의 실행원리(라이프 사이클)에 대한 공부, android studio의 프로젝트 파일 구조와 각 파일간의 연관성에 대한 공부.

### 2. Android Studio를 통해 'Step by Step 안드로이드 프로그래밍' 책의 예제 프로그램 작성

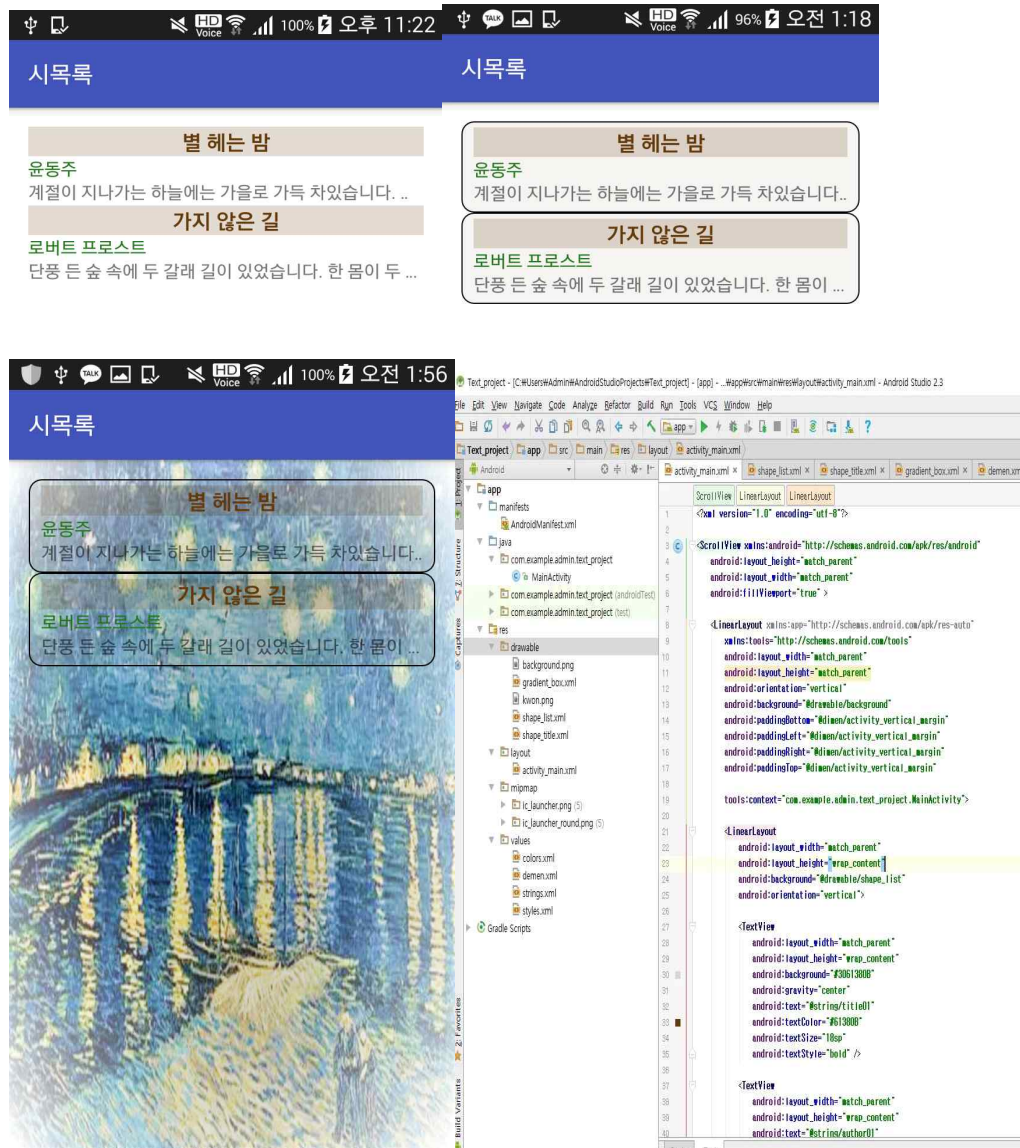
#### 결과물

1. 회전하는 이미지파일 + 터치를 따라오도록 + 진동 구현



## 2. 시 목록 앱

### 결과물



## 2-3 보고서\_류영승

### Android 스터디 보고서

#### 1. 개요

- 4학년이라는 좋은 구실(?)로 다른 스터디원에 비해 많이 참석하지 못했다. 반성중이고 급한 불인 중간고사는 지났으므로 앞으로 열심히 해나갈 계획이다.

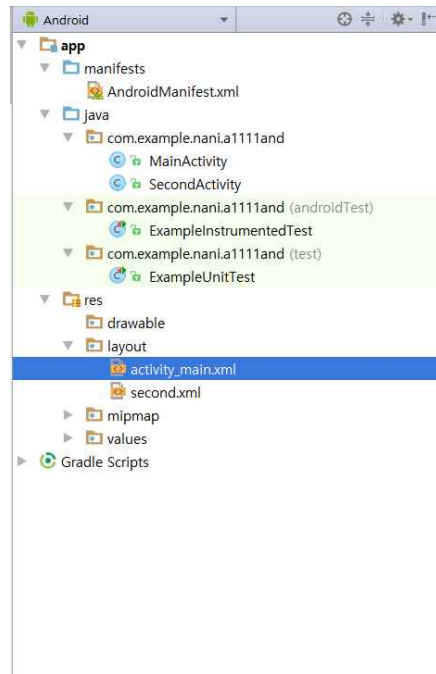
#### 2. 공부한 내용들

1) XML : 안드로이드에서 애플리케이션의 디자인 부분인 xml에 대해 공부했다. 그중 Linear Layout과 Relative Layout의 차이점에 대해 이해하고 신입생들이 이해가 힘든 부분에 대해 설명해주기 위해 다음과 같은 예제를 작성하였다.



Linear Layout에서도 vertical과 horizontal 그리고 weight에 대해 공부했다. 위의 예제는 Linear Layout 안에 작은 Linear Layout을 주고 각각 가로방향 weight 세로방향 weight를 줘서 직관적으로 봐서 이해하기 쉽게 만들어 놓은 예제이다.





2) Fragment : 이때는 스터디에 불참하여 혼자 따로 공부를 하였다

다수의 xml과 activity를 서로 연결해주기 위해 Manifest에 econdActivity를 연결해주었다.

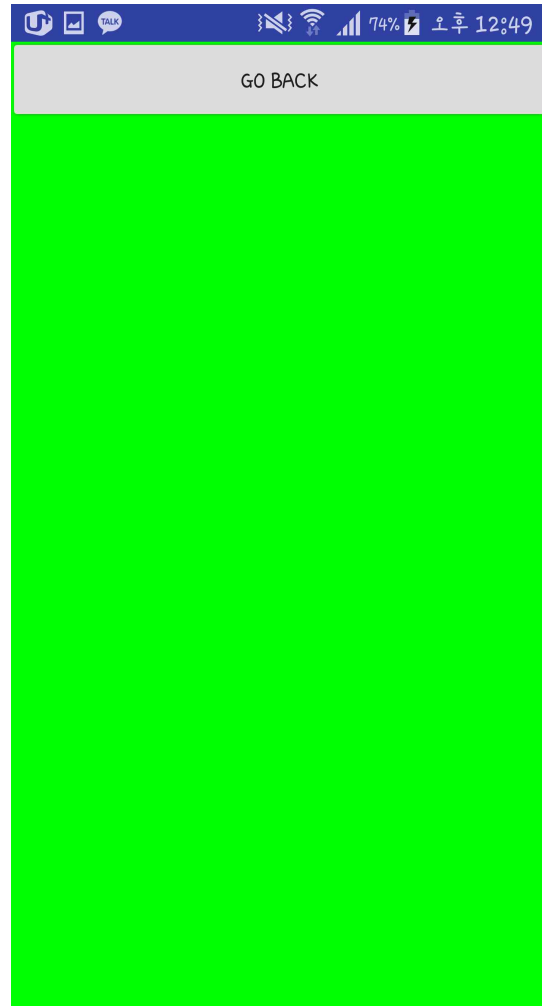
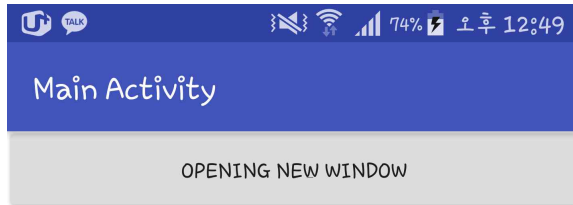
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.nani.a1111and">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="1111And"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SecondActivity"
            android:label="Second Activity"/>
    </application>
</manifest>
```

그리고 secondActivity에 뿌려줄 secondxml를 작성하였고 버튼을 눌렀을 때 두 번째 xml을 호출하는 기능을 추가하였다

실행화면



## 2-4 보고서\_조호열

### 1. 앱 프로젝트 구조와 실행 원리

#### 1.1 안드로이드는 프로젝트 생성->파일 편집->프로젝트 실행 3단계로 진행

- 프로젝트 생성은 5단계로 구성 됨.

- ① 프로젝트 시작
- ② 프로젝트 구성
- ③ 실행 디바이스
- ④ 액티비티 유형
- ⑤ 액티비티 구성파일 이름 설정

- 프로젝트 실행

Run메뉴에서 Run 'app'을 클릭하거나 메뉴 아이콘 Run 'app' 을 클릭하면 나타나는  
디바이스 선택 창의 목록에서 실행할 디바이스(AVD 또는 스마트폰)을 선택하고 실행

#### 1.2 자바와 xml 파일로 구성되는 기본 파일들의 연관성을 이용하여 실행

- activity\_main.xml은 화면을 설계하며, dimen.xml 파일에 정의된 여백의 크기 등을 설정한다.
- MainActivity.java 파일은 액티비티를 생성하고 activity\_main.xml에 정의된 레이아웃을 출력한다.
- AndroidManifest.xml은 어플리케이션 아이콘과 라벨, 화면 스타일 등을 지정하고, 액티비티를 구성하는 자바 클래스를 지정한다.
- Gradle Scripts는 컴파일 / 빌더 정보를 담고 있으며 실행파일을 만드는 역할을 한다.

## 2. 안드로이드 레이아웃 구성

### 2.1 RelativeLayout 과 LinearLayout의 개념과 차이점

- RelativeLayout은 따로 위치를 정하지 않으면 뷰가 0.0에 계속 쌓이는 레이아웃
- LinearLayout은 뷰가 가로 또는 세로로 순차적으로 나열되는 레이아웃

Android:orientaton 으로 가로(horizontal), 세로(vertical)로 배열 설정

### 2.2 View 사용

- layout\_width, layout\_height로 뷰의 크기를 설정

- ① match\_parent = 부모 뷰의 크기에 맞춤
- ② wrap\_content = 생성한 뷰의 내용물 크기에 맞춤
- ③ dp = 직접 뷰의 크기를 설정

- android: id

뷰 고유의 식별자(이름)를 설정해줌.

- android: src

Drawable 폴더에 있는 이미지 파일을 불러옴.

- android: gravity

뷰 안에 들어있는 content 들의 위치 설정(bottom ,top ,right ,left, center)

- android: layout\_gravity

현재 뷰를 상위 뷰 안의 위치 설정

- android: layout\_weight

현재 뷰의 가중치를 설정 (클수록 공간차이 많이 함)

- android: text

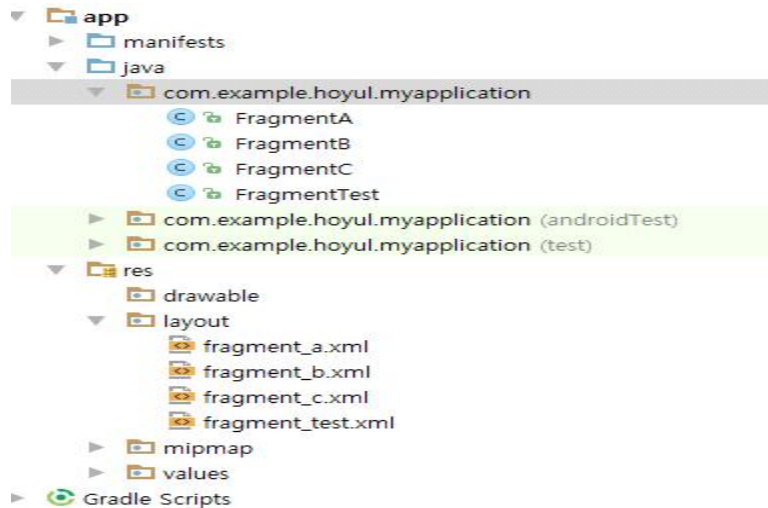
android: textSize,

android: textStyle

android: textColor

텍스트의 내용, 크기, 글꼴, 색깔 설정

## 1. Fragment를 이용해서 activity 화면 분할해보기



구성도 ↑

```
import ...
public class FragmentA extends Fragment{
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_a, container, false);
    }
}
```

FragmentA.java ↑ (화면A 버튼을 누르면 fragment\_a를 프래그먼트에 호출)

```
import ...
public class FragmentB extends Fragment{
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_b, container, false);
    }
}
```

FragmentB.java ↑ (화면B 버튼을 누르면 fragment\_b를 프래그먼트에 호출)

```
import ...  
public class FragmentC extends Fragment {  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.fragment_c, container, false);  
    }  
}
```

FragmentC.java ↑ (화면C 버튼을 누르면 fragment\_c를 프래그먼트에 호출)

```
package com.example.hoyul.myapplication;
import ...
public class FragmentTest extends Activity{
    FragmentManager manager;
    FragmentTransaction trans;
    FragmentA fa;
    FragmentB fb = new FragmentB();
    FragmentC fc = new FragmentC();
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fragment_test);
        Button btn_a = (Button)findViewById(R.id.btn_a);
        Button btn_b = (Button)findViewById(R.id.btn_b);
        Button btn_c = (Button)findViewById(R.id.btn_c);
        manager = getFragmentManager();
        fa = (FragmentA)manager.findFragmentById(R.id.frg_a);
        btn_a.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                trans = manager.beginTransaction();
                trans.replace(R.id.frg_a, fa);
                trans.commit();
            }
        });
        btn_b.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                trans = manager.beginTransaction();
                trans.replace(R.id.frg_a, fb);
                trans.commit();
            }
        });
        btn_c.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                trans = manager.beginTransaction();
                trans.replace(R.id.frg_a, fc);
                trans.commit();
            }
        });
    }
}
```

FragmentTest.java ↑ (FragmentA,B,C의 객체를 생성하여 레이아웃을 프래그먼트에 호출)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.hoyul.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".FragmentTest"
            android:label="My Application">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>

</manifest>
```



AndroidManifest.xml ↑ (액티비티를 구성하는 자바 클래스를 지정한다.)

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#ffdd00">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="노란색"/>
</RelativeLayout>
```

fragment\_a.xml ↑ (화면A 버튼을 누르면 나오는 화면)

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#ff2321">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="빨간색"/>
</RelativeLayout>
```

fragment\_b.xml ↑ (화면B 버튼을 누르면 나오는 화면)

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#22741C" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="초록색"/>
</RelativeLayout>
```

fragment\_c.xml ↑ (화면C 버튼을 누르면 나오는 화면)

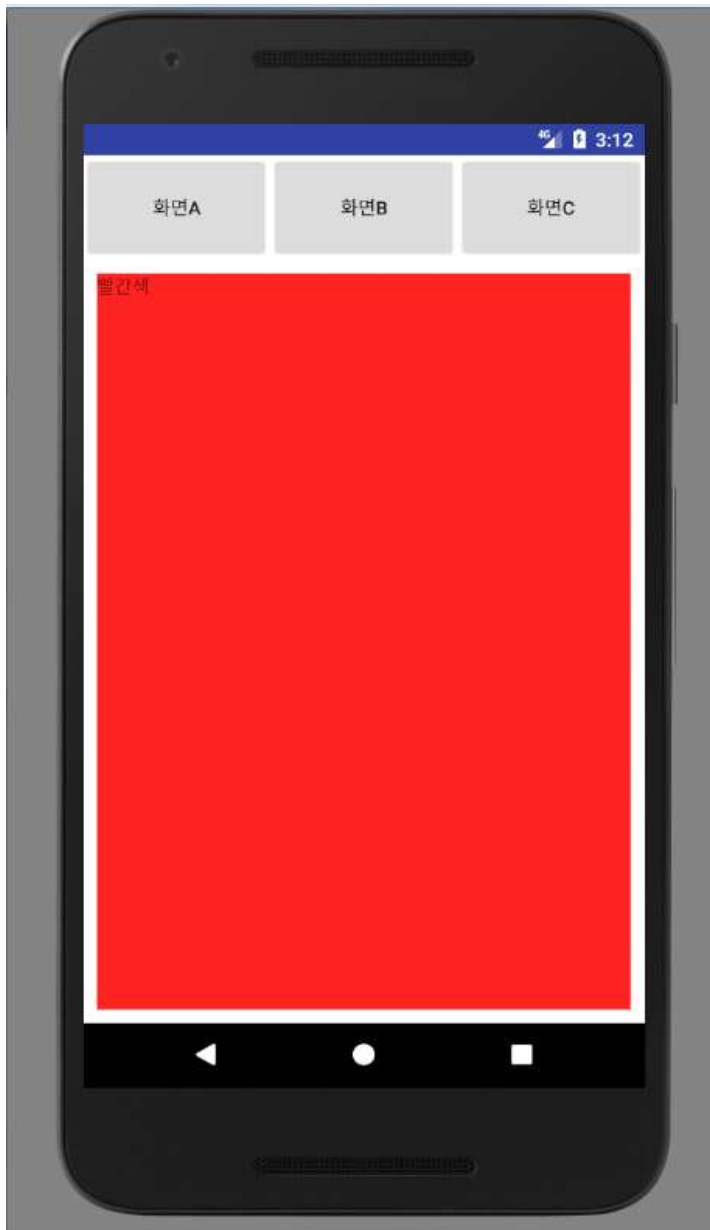
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_horizontal">
        <Button
            android:id="@+id/btn_a"
            android:layout_width="0dp"
            android:layout_height="80dp"
            android:layout_weight="1"
            android:text="화면A"/>
        <Button
            android:id="@+id/btn_b"
            android:layout_width="0dp"
            android:layout_height="80dp"
            android:layout_weight="1"
            android:text="화면B"/>
        <Button
            android:id="@+id/btn_c"
            android:layout_width="0dp"
            android:layout_height="80dp"
            android:layout_weight="1"
            android:text="화면C"/>
    </LinearLayout>
    <fragment
        android:name="com.example.hoyul.myapplication.FragmentA"
        android:id="@+id/frg_a"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp"/>
</LinearLayout>
```

fragment\_test.xml ↑ (앱 실행시 가장 먼저 나오는 화면)

## 2. 실행화면



앱 실행시 가장 먼저 나오는 화면↑ (기본 설정을 fragment\_a로 설정해서 노란색이 나옴)



화면B 버튼을 누르면 나오는 화면 ↑



화면C 버튼을 누르면 나오는 화면 ↑

## 2-5 보고서\_장정윤

날짜	스터디 내용
3/20	앞으로의 스터디 진행 방향과 목표에 대해 토의하고 스터디 교재로 'Step by Step 안드로이드 프로그래밍'을 선정하였습니다.
3/27	'back-end', 'front-end' 개념과 안드로이드 스튜디오의 레이아웃 유형을 알아보고 'linearlayout' 과 'relativelayout' 의 소스에 대해 공부하였습니다(교재 112p~119p).
4/3	질문과 답을 통한 지난 스터디 복습을 시작으로, 순차지향, 객체지향 개념과 프로젝트 모듈의 구조와 기능, 모듈별 소스들의 기능에 대해 공부하였으며(교재 72p), 프로젝트가 'xml', 'manifest', 'activity' 이 세 가지로 구성되어 있다는 것을 배웠습니다(교재 102p~103p). 다음 스터디 과제로 교재 2p~195p를 예습해오기를 내주셨습니다.
4/24	지난 시간에 내주셨던 과제를 검사 받고, 과제를 하며 헛갈렸던 부분인 객체(Object)와 클래스(Class), 속성(Attribute), 메서드(Method)에 대해 배웠습니다. 추가적으로, 자료형(기본자료형과 참조자료형)에 대해 공부하였습니다. 또한, 직접 예제 실습('안녕, 안드로이드!' 출력하기, 회전하는 바람개비 만들기, 터치 위치를 따라오는 이미지 만들기, 시 앱 만들기, 이미지 출력하기 등)을 해봄으로써 안드로이드 앱에 대한 흥미와 이해도가 높아졌고 스스로 무언가를 만들 수 있어서 기쁘고 신기했습니다.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.jjy02.helloandroid.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="안녕, 안드로이드!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

▲ '안녕, 안드로이드!' 코드



▲ '안녕, 안드로이드!' 실행



```
package com.example.jjy02.pinwheel1;

import android.animation.ObjectAnimator;
import android.animation.ValueAnimator;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.animation.LinearInterpolator;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ImageView iv_pinwheel =(ImageView)findViewById(R.id.pinwheel);

        ObjectAnimator object = ObjectAnimator.ofFloat(iv_pinwheel, "rotation", 360);
        object.setInterpolator(new LinearInterpolator());
        object.setDuration(2000);
        object.setRepeatCount(ValueAnimator.INFINITE);
        object.start();
    }
}
```

▲ '바람개비1' 코드



▲ '바람개비1' 실행

```
package com.example.jjy02.followme1;

import android.animation.ObjectAnimator;
import android.content.Context;
import android.os.Bundle;
import android.os.Vibrator;
import android.support.v4.app.FragmentActivity;
import android.view.MotionEvent;
import android.view.WindowManager;
import android.widget.ImageView;

public class MainActivity extends FragmentActivity {

    ImageView iv_smile;
    float previous_x = 0;
    float previous_y = 0;
    Vibrator mVibe;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_main);
        iv_smile = (ImageView) findViewById(R.id.smile);
        mVibe = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
    }

    public boolean onTouchEvent(MotionEvent event) {

        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                break;

            case MotionEvent.ACTION_MOVE:
                int touch_x = (int) event.getX();
                int touch_y = (int) event.getY();

                ObjectAnimator smileX = ObjectAnimator.ofFloat(iv_smile, "translationX", previous_x, touch_x);

                smileX.start();
                ObjectAnimator smileY = ObjectAnimator.ofFloat(iv_smile, "translationY", previous_y, touch_y);

                smileY.start();

                mVibe.vibrate(50);

                previous_x = touch_x;
                previous_y = touch_y;
                break;

            case MotionEvent.ACTION_UP:
                break;
        }
        return false;
    }
}
```

▲ '터치 위치를 따라오는 이미지' 코드



▲ '터치 위치를 따라오는 이미지' 실행

그리고 다음 스터디 과제로 교재 197p~304p를 예습해오기를 해주셨습니다.

--	--

## 2-6 보고서\_이호정

### 1.앱

기본적인 안드로이드 개발은 다음단계로 이루어진다.

- 1.프로젝트 시작
- 2.프로젝트 구성
- 3.실행 디바이스
- 4.엑티비티 유형
- 5.엑티비티 구성파일 이름설정

### 2. 객체 지향적

C언어 같은 순차지향적과는 달리 안드로이드 개발은 객체지향적이다

순차지향적은 코드내용을 차례로 읽고 객체지향적은 기능별로 구분하여 필요할 때 그 코드 내용을 뽑아서 쓸수있다 객체지향적은 JAVA, Java script같은 언어가 있다.

### 3. mipmap

앱의 아이콘 관련 이미지를 넣는 폴더이다. 그 이외에 안드로이드 앱을 만들 때 쓰이는 이미지들은 drawable 폴더 에 넣어야 한다.

또한 androidmanifest.xml파일은 앱의 아이콘과 라벨, 화면스타일등을 지정하고,엑티비티를 구성하는 자바 클래스를 지정한다. 앱제목 과 프로젝트 제목은 다른것이니 헷갈리지 말자.

### 4.activity\_main.xml은 화면설계와 관련되어있다. 화면을 설계할 수 있다

특히 dimens.xml은 여백크기를 설정한다. 여백크기는 일단 손대지 않지만 숙련되면 손대보자.

### 5.이미지 복사하는 방법

이미지 선택하고 복사를 한 다음에 폴더를 선택하고 경로 선택한 다음 이미 지명을 정해 준다(이미지 이름을 설정해주는것이 나중에 편하다)

## 6.회전하는 오브젝트 만들기

1.Drawable 폴더에 이미지를 복사한다(mipmap 이미지에 넣으면 안된다)

2.어플리케이션이름을 바람개비 1로하기위해서 app\_name속성 값에 해당되는 데이터를 수정하기

3.이미지 뷰의 id를 pinwheel로 지정한다

(androud:id="@+idWpinwheel")

4 액티비티 제어

엑티비티를 생성하고 activity\_main.xml를 출력하기

valueanimator.infinite는 애니메이션을 무한번 반복하도록 설정하도록 하게만드는것임.

Object.setDuration(2000); 애니메이션 시간을 2초로 설정 (2초라고 2하면 안되고 꼭 2000을 해야한다.)

Object.start(); 애니메이션을 실행함

성공!

## 7.움직이는 모션 오브젝트 만들기에서..

Case문을 이용한다

Action\_down 누르기 시작할때

Action\_move 터치하고 움직이는상태

Int touch\_x=(int)event.getx(); 코드는 터치하고 있는 x의 위치 인식하는 것

nib.viration(50); ()안의 밀리초동안 진동하는 코드

## 8.java의 자료형. 기본자료형

정수

실수

문자형

논리형

(기본적으로 c언어랑 비슷하다)

## 9.xml

Xml은 인터넷에 연결된 시스템끼리 데이터를 쫓게 주고 받을수있게 할 목적으로 만들어진 것이다. 선언과 엘리먼트로 나누어진다

## 10.자바클래스

객체와 속성 메소드가 있는데 메소드는 일종의 함수라고 생각하면된다.

같은 클래스끼리는 속성을 상속받는다

## 11.텍스트 리소스 만들기, 이미지를 통해 영화넣기, 오디오 넣기, 동영상넣기등의 예시를 재움.

<string name="title01">별헤는 밤</string>첫번째 시의 제목을 하는 코드다

이런식으로 텍스트 리소스를 편집해준다. 일단 코드 하나하나 보고 이해 하는 단계부터 시작해야함. 다 외우려고 하지말자.

## 12.데이터형

문자형

정수형

부동소수형

논리형

### 13. 배열

같은 이름과 같은 데이터형을 가진 연속적인 저장공간[]

### 14. 느낀점

처음안드로이드 개발을 하고 싶었을때에는 각종 스토어에있는 앱처럼 만들고 싶은 의욕이 많았다. 그러나 시중에 판매되거나 다운받은 앱의 클릭한번, 홈저튼 한번, 스크롤한번 내리는데 하나하나 코드를 짜야하고 생각해야하는 어려움이 있다는것을 알고 내가 할 수 있을까 하는 걱정이 앞섰다. 그러나 선배님들의 도움을 받으며 예제부터 하나하나 코드를 짜보면서 오류도 떠보고 빨간줄도 쳐지는것을 보며 생각보다 어렵구나 했지만 마침내 돌아가는 바람개비를 보니 성취감도 느껴지고 뿌듯하였다. 조금 더 공부해서 실제로 앱을 만들어보고 싶은 마음도 생기고 말이다. 배운것이 너무 많은데 그 대부분이 내가 코드를 이해하지 못하거나 또는 아예 처음 보는 생소한 개념인것이 많아 보고서에는 핵심개념과 내가 이해하지 못하였거나 헷갈렸던 것 중심으로 적어보았다. 이번에는 시험기간도 겹쳐 더 열심히 하지 못한 아쉬움도 남지만 앞으로의 시간에는 더 많이 코드를 쳐보고 완성되는 기쁨을 많이 느낄수 있으면 좋겠다. 무엇보다 c언어도 모르는 신입생이 많이 답답했을 터인데 매우 기초적인 질문에도 친절히 대답해주신 선배님들이 있어 바람개비라도 돌릴수 있었다. 오히려 'c언어도 질문해'라고 의욕 넘치는 모습을 보여주셔서 스터디를 계속 할 수 있었던 것 같다.

앞으로도 꾸준한 공부를 통해 어제보다 더 코드가 익숙해지면 좋겠다.



### 3. 3~4 월 스터디 결과

3~4 월은 각종 행사와 시험기간으로 스터디를 빠지는 날이 많았다. 또한 실제 프로젝트를 진행하기 이전에, 안드로이드를 처음 접해보는 스터디원이 있었기에, 부담스럽지 않은 기본 교재를 선정하여 8 주 과정에 맞추어 진행하기로 하였다. 그 결과로 나온 보고서가 상술되어있는 보고서이며, 프로그래밍을 올해 처음 시작한 새내기들도 있기 때문에, C 언어와 자바의 클래스 등등, 프로그래밍 전반적인 스터디를 진행하는 방향이 되었다. 소수의 스터디이고 고학년과 1 학년이 섞여있는 스터디라서 다소의 걱정이 되었으나, 그만큼의 열정을 보여주고 스터디원끼리 친밀감을 보여주어서 보람 깊은 2 달간의 스터디였다고 생각한다. 특히 신입생 스터디원은 객체 지향 프로그래밍과 자바를 처음 배우기 때문에 힘들었을 텐데 버거워 하지 않고 주도적으로 질문을 하여 함께 하는 스터디를 만들어주었기 때문에 3,4 월의 스터디의 분위기를 조성하는데 많은 도움을 주었다. 또한 12 학번 스터디원들은 경험해보거나 혹은 기반지식을 가지고 의욕적으로 프로젝트를 제안하고 1 학년들의 학습을 도와주었으며, 추가적으로 내 준 과제도 부담을 가지지 않고 해결하여서 매끄러운 스터디를 진행하는데 큰 도움이 되었다. 차후 책의 진도를 마무리하고 간단하게 서버와 DB 를 연동시켜 회원가입과 로그인기능을 품은 어플리케이션을 제작 해 볼 예정이며, 희망자에 한하여 여름 방학 때 자체 혹은 기관에서 주최하는 해커톤을 진행 해 볼 예정이다.