

안드로이드 SDK 개발 플랫폼 활용하기

# 액티비티 간 화면 전환하기

- 안드로이드에서는 화면간 전환을 처리하기 위해 Intent라는 클래스를 제공
  - Intent는 목적이나 의도라는 뜻으로 화면간의 메시지 역할을 수행
- 화면을 전환 할 때에는 아래의 명령과 같이 화면을 생성하고 전환하면 됨
  - `Intent myIntent = new Intent(FirstScreen.this, SecondScreen.class);`  
// 인텐트 생성
  - `startActivity(myIntent);`  
// 새로운 화면 생성 요청

- 첫 화면은 아래와 같이 시작 버튼만을 제공하는 화면을 구현한다.
  - 화면은 총 3개의 화면으로 구성
- 일반적인 게임 콘텐츠 앱을 개발한다고 가정
- 1. 처음에는 타이틀 화면 표시
  - 게임 시작하기 버튼
- 2. 게임 진행 화면
  - 게임 시작하기 버튼을 누르면 게임 진행 화면으로 넘어간다.
- 3. 게임오버 화면
  - 플레이어는 게임을 진행하다가 게임을 끝내게 되면 게임 오버 화면으로 넘어 간다.
  - 사용자에게 계속 게임을 진행할지 혹은 끝내고 첫 화면으로 돌아갈지 물어 볼 수 있다.



- 처음 타이틀 화면

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="안드로이드 APK
파일의 스키마 경로는 생략합니다."
    xmlns:app="안드로이드 APK 파일의 스키마 경로는 생략"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="#ccccFF"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Main Activity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Play"
/>
</android.support.constraint.ConstraintLayout>
```

- 게임 플레이 화면

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="안드로이드 APK
파일의 스키마 경로는 생략합니다."

    xmlns:app="안드로이드 APK 파일의 스키마 경로는 생략"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".GamePlayActivity"
    android:background="#FFFF00"
    >
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Die!"
    />
</android.support.constraint.ConstraintLayout>
```

- 게임이 종료되었을 때 표시할 게임오버 화면

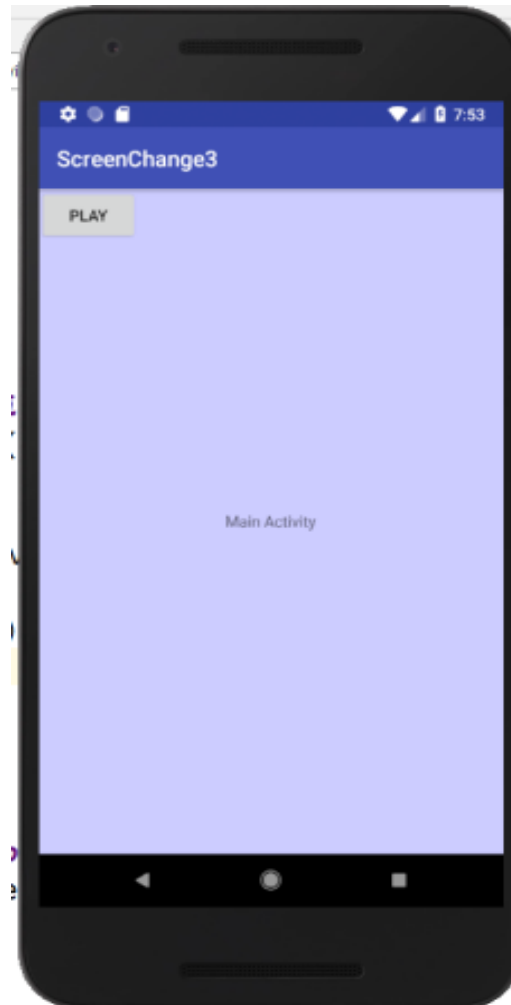
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="안드로이드
APK 파일의 스키마 경로는 생략합니다."
```

```
    xmlns:app="안드로이드 APK 파일의 스키마 경로는 생략"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".GameOverActivity"
    android:background="#FF0000"
  >
  <Button
      android:id="@+id/buttonMain"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Go Back to Main"
  />
```

```
<Button
    android:id="@+id/buttonContinue"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Continue"
    app:layout_constraintTop_toBottomOf="@id/buttonMain"
/>
</android.support.constraint.ConstraintLayout>
```



- 타이틀 화면의 표시
  - PLAY 버튼을 눌러도 화면이 안넘어감 → 이어서 Java 코드 구현



```
package com.iot.screenchange3;
import android.content.Intent; // 인텐트 활용
import android.support.v7.app.AppCompatActivity; // 액티비티 부모 클래스
import android.os.Bundle; // 액티비티 생성 번들
import android.view.View; // 뷰계열 최상위 클래스
import android.widget.Button; // 버튼 뷰 사용

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); 표시
        Button button = (Button)findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
                    GamePlayActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

- 새로 생성된 화면은 Manifest 파일에 등록할 것

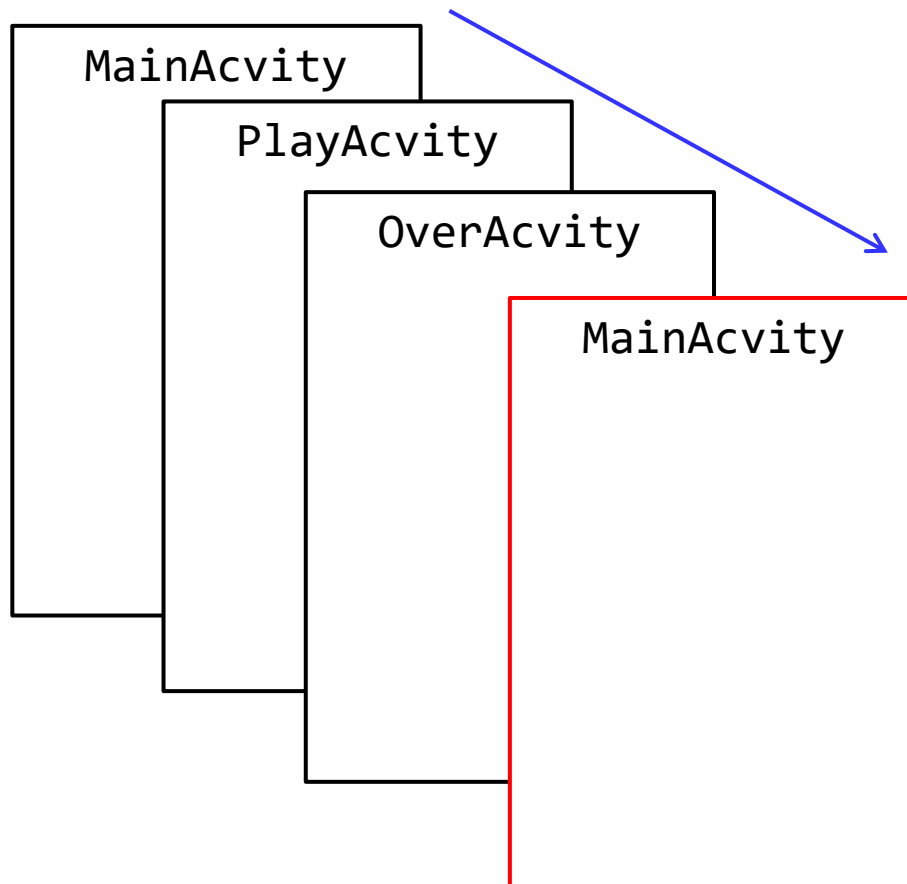
```
package com.iot.screenchange3;
import android.content.Intent; // 인텐트 활용
import android.support.v7.app.AppCompatActivity; // 액티비티 부모 클래스
import android.os.Bundle; // 액티비티 생성 번들
import android.view.View; // 뷰계열 최상위 클래스
import android.widget.Button; // 버튼 뷰 사용

public class GamePlayActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_game_play);
        Button button = (Button)findViewById(R.id.button);

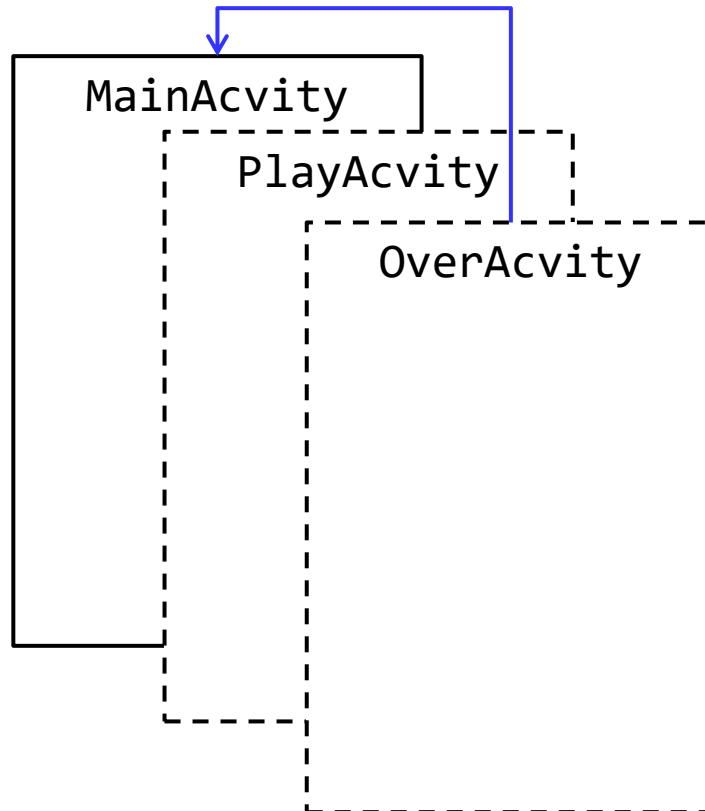
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(GamePlayActivity.this,
                    GameOverActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

# 액티비티는 항상 겹쳐지는 특징이 있다.

- 게임 오버 이후 메인으로 가고 싶은데
  - Main이 두개가 되어 버렸다.
  - Over를 닫고 Play를 닫아도 되지만, 잠깐이지만 눈에 보이기도 한다.
  - 바로 갈 수 있는 방법은 없을까?



- 액티비티의 플래그를 이용하여 특정 액티비티로 바로 돌아가게 할 수 있다.
  - Intent.**FLAG\_ACTIVITY\_CLEAR\_TOP**



```
package com.iot.screenchange3;

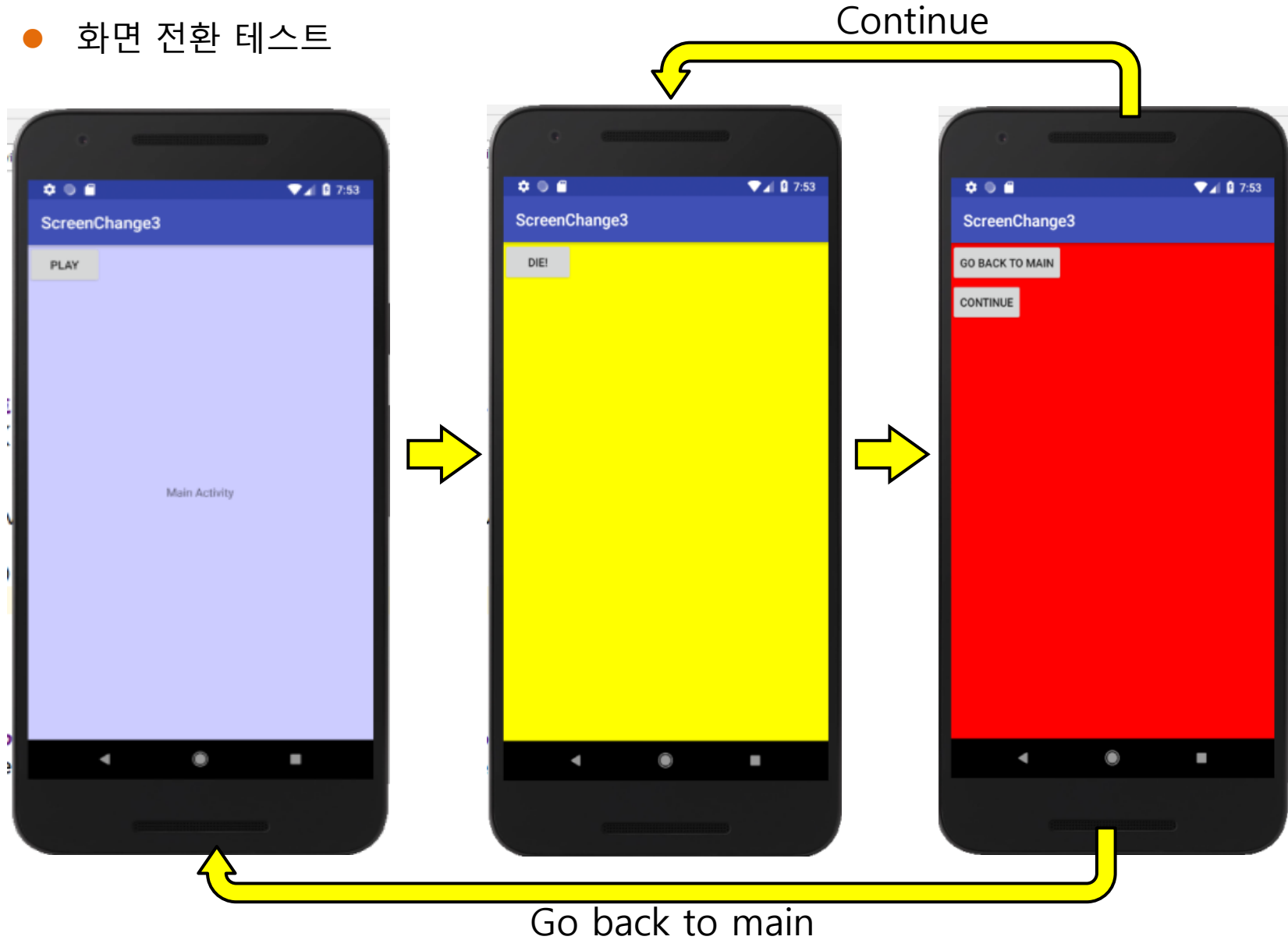
import android.content.Intent; // 인텐트 활용
import android.support.v7.app.AppCompatActivity; // 액티비티 부모 클래스
import android.os.Bundle; // 액티비티 생성 번들
import android.view.View; // 뷰계열 최상위 클래스
import android.widget.Button; // 버튼 뷰 사용

public class GameOverActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_game_over);
        Button buttonMain = (Button)findViewById(R.id.buttonMain);
        buttonMain.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(GameOverActivity.this,
                    MainActivity.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP); // 바로 돌아가기
                startActivity(intent);
            }
        });
    }
};
```

```
Button buttonContinue = (Button)findViewById(R.id.buttonContinue);

buttonContinue.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}
```

- 화면 전환 테스트





- 게임 오버 화면에서는 게임을 계속 이어서 할지 아니면 끝내고 첫 화면으로 돌아갈지 물어본다.
  - 이때 인텐트를 이용해서 결과 반환
  - 이전 화면은 첫 화면으로 갈지 게임을 계속 진행할지 결정할 수가 있게 된다.
- 왼쪽 그림이 게임 진행 화면인데 게임 오버 화면으로 넘어가기 위해 die 버튼을 누른다.

그러면 캐릭터가 사망하게 되고 게임 오버 화면으로 넘어 간다.
- 여기서 사용자는 게임을 종료 할지 또는 이어서 할지 결정
  - 게임을 이어서 하는 경우에는 바로 이전 화면으로 넘어가면 된다.
  - 첫 화면으로 돌아가기 버튼을 누른 경우 직전 화면으로 돌아가면 안되고 두 번의 화면을 닫아야 첫 화면으로 돌아갈 수 있다