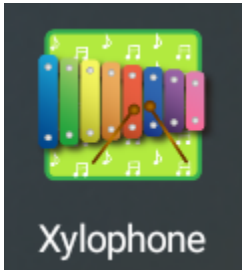

Kotlin을 이용한 Android 프로그래밍

실로폰 앱 만들기

Contents

- I. 실로폰 앱은 가로화면으로 고정된 한 액티비티에 도, 레, 미, 파, 솔, 라, 시, 도 음판이 있고 누르면 해당 음이 재생



실로폰 앱 만들기

▶ 프로젝트 명 : Xylophone

▶ 앱의 기능

▶ 화면의 음판을 누르면 소리가 재생

▶ 프로젝트 설계

▶ 실로폰 앱은 가로로 고정된 액티비티에 구성된 음판을 누르면 해당음을 재생

▶ 음판은 텍스트 뷰 속성을 수정하여 디자인

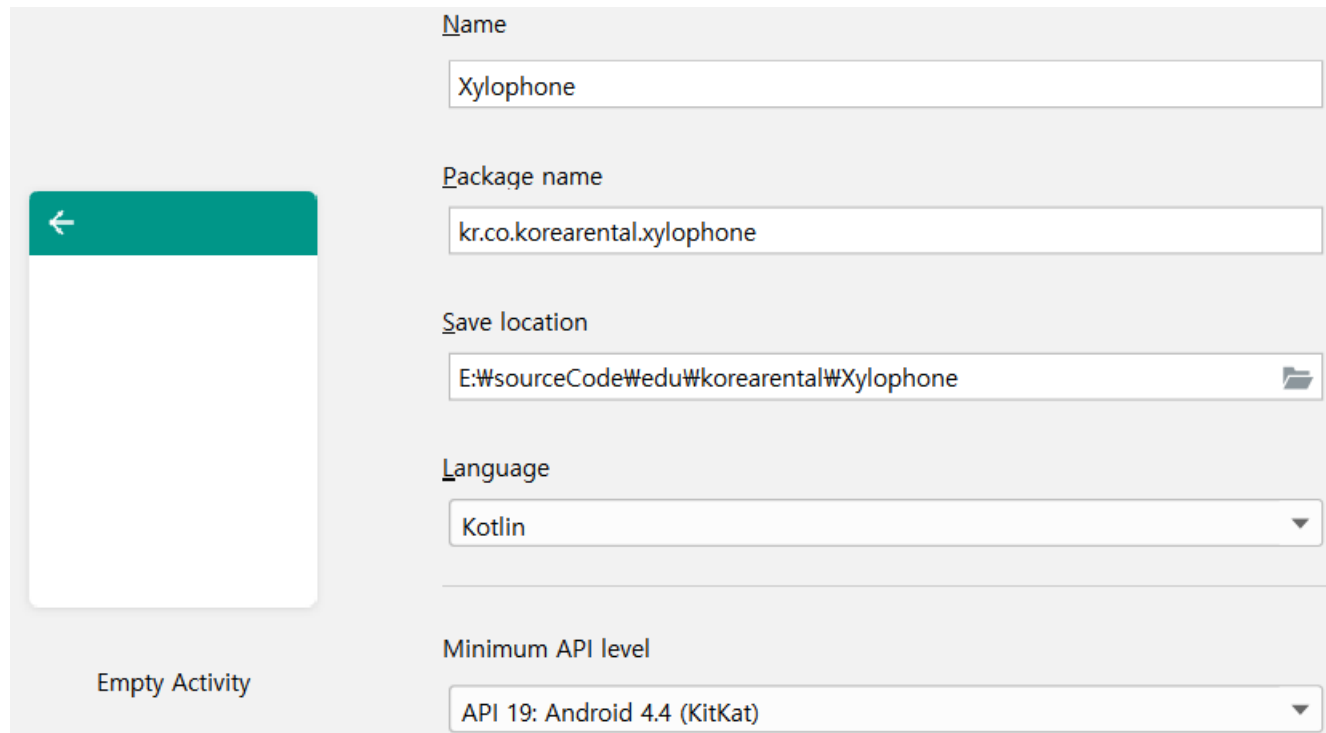
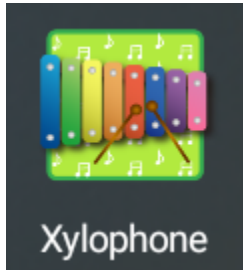
▶ 소리는 SoundPool 클래스를 사용하여 재생

▶ 5.0부터 API가 변경되었으므로 신/구 버전 모두 잘 동작하도록 분기를 적용

실로폰 앱 만들기

▶ 프로젝트 생성

- ▶ 프로젝트 명 : Xylophone
- ▶ minSdkVersion : 19(Android 4.4 KitKat)
- ▶ 기본 액티비티 : Empty Activity
- ▶ 프로젝트 생성한 후 아이콘 변경 : kr_xylophone.png



A screenshot of the 'New Project' dialog in Android Studio. The dialog is light grey and contains several input fields and a dropdown menu. On the left side, there is a preview of the app's main screen, which is a white rectangle with a teal header bar containing a white back arrow. Below the preview, the text 'Empty Activity' is visible. The right side of the dialog contains the following fields:

- Name:** A text field containing 'Xylophone'.
- Package name:** A text field containing 'kr.co.korearental.xylophone'.
- Save location:** A text field containing 'E:\sourceCode\Wedu\Wkorearental\WXylophone' with a folder icon on the right.
- Language:** A dropdown menu with 'Kotlin' selected.
- Minimum API level:** A dropdown menu with 'API 19: Android 4.4 (KitKat)' selected.

실로폰 앱 만들기

▶ 가로 모드 고정

- ▶ 코틀린에서 고정하는 방법과 매니페스트에서 고정하는 방법이 있음
- ▶ 첫번째 방법은 수평 측정기 앱에서 사용한 방식을 적용
- ▶ MainActivity.kr의 onCreate()에 아래 코드 추가

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        requestedOrientation = ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

실로폰 앱 만들기

▶ 가로 모드 고정

▶ 두번째 방법은 매니페스트에 코드 추가

▶ 이번 예제에서는 매니페스트를 수정하여 가로 모드로 고정

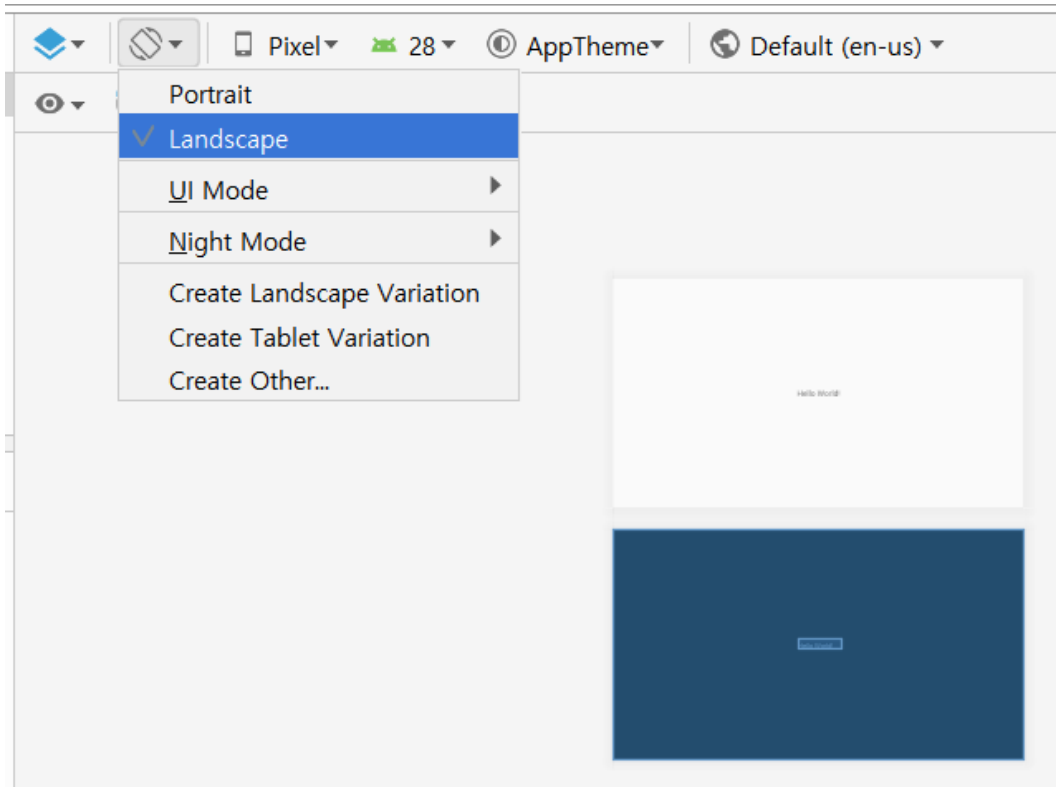
```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Xylophone"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity"
        android:screenOrientation="landscape">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>

            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
</application>
```

실로폰 앱 만들기

▶ 가로 모드 고정

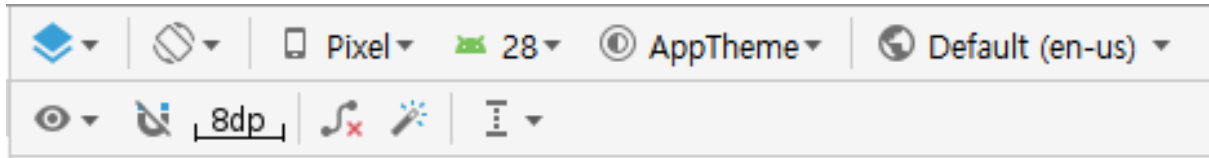
- ▶ 레이아웃 에디터에서 미리보기를 가로 모드로 변경
- ▶ activity_main.xml 파일을 열고 [Orientation for Preview] - [Landscape]



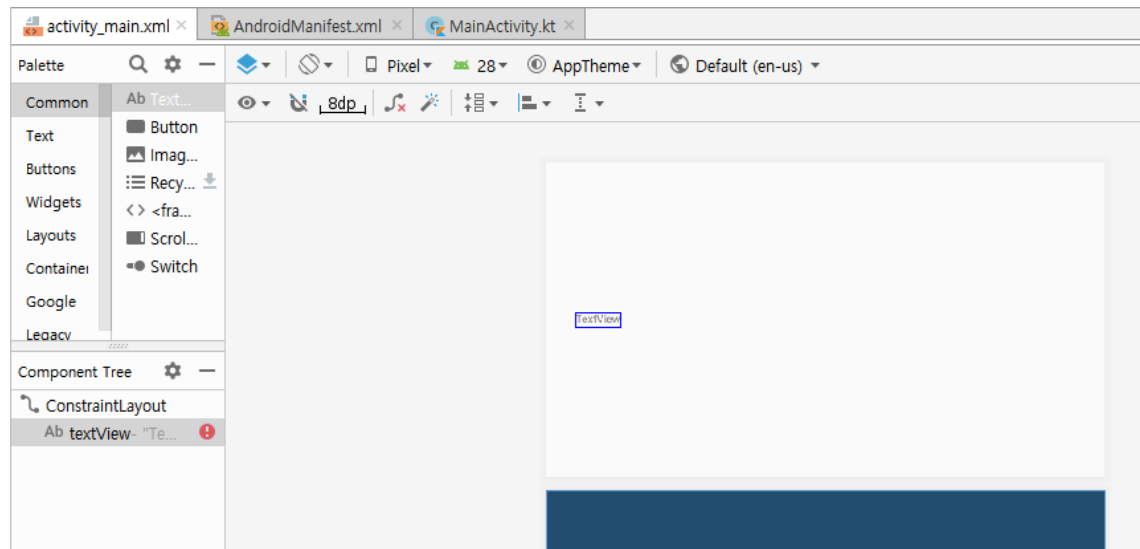
실로폰 앱 만들기

▶ 텍스트 뷰로 음판 만들기

- ▶ 레이아웃의 디자인 과정에서 음판을 적당히 배치한 후에 한번에 제약을 추가할 수 있도록 Autoconnect 모드 비활성화



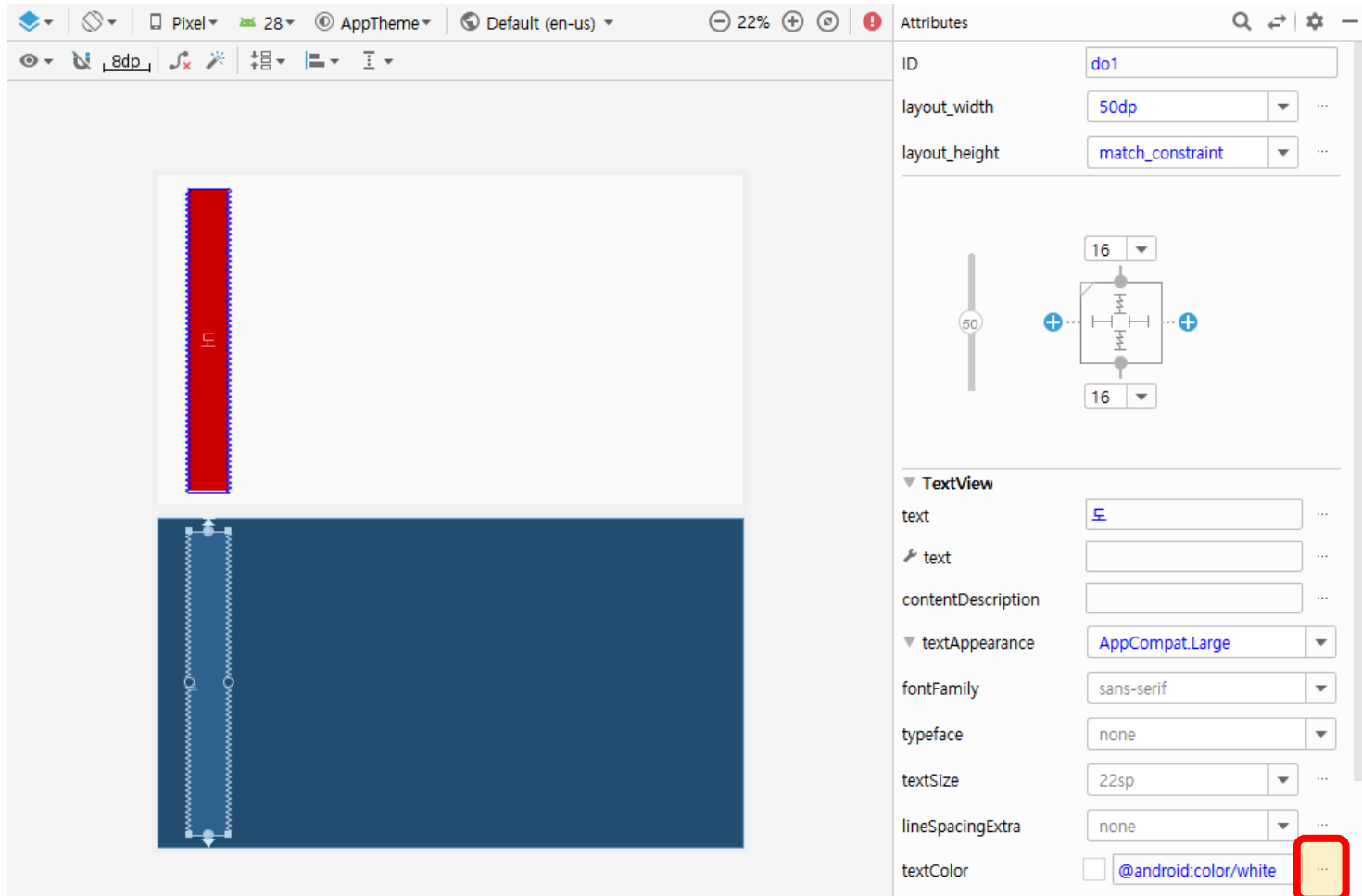
- ▶ 음판을 추가하기 위하여 기본으로 배치된 텍스트 뷰를 삭제하고 팔레트 창에서 TextView를 적당한 위치에 추가



실로폰 앱 만들기

▶ 텍스트 뷰로 음판 만들기

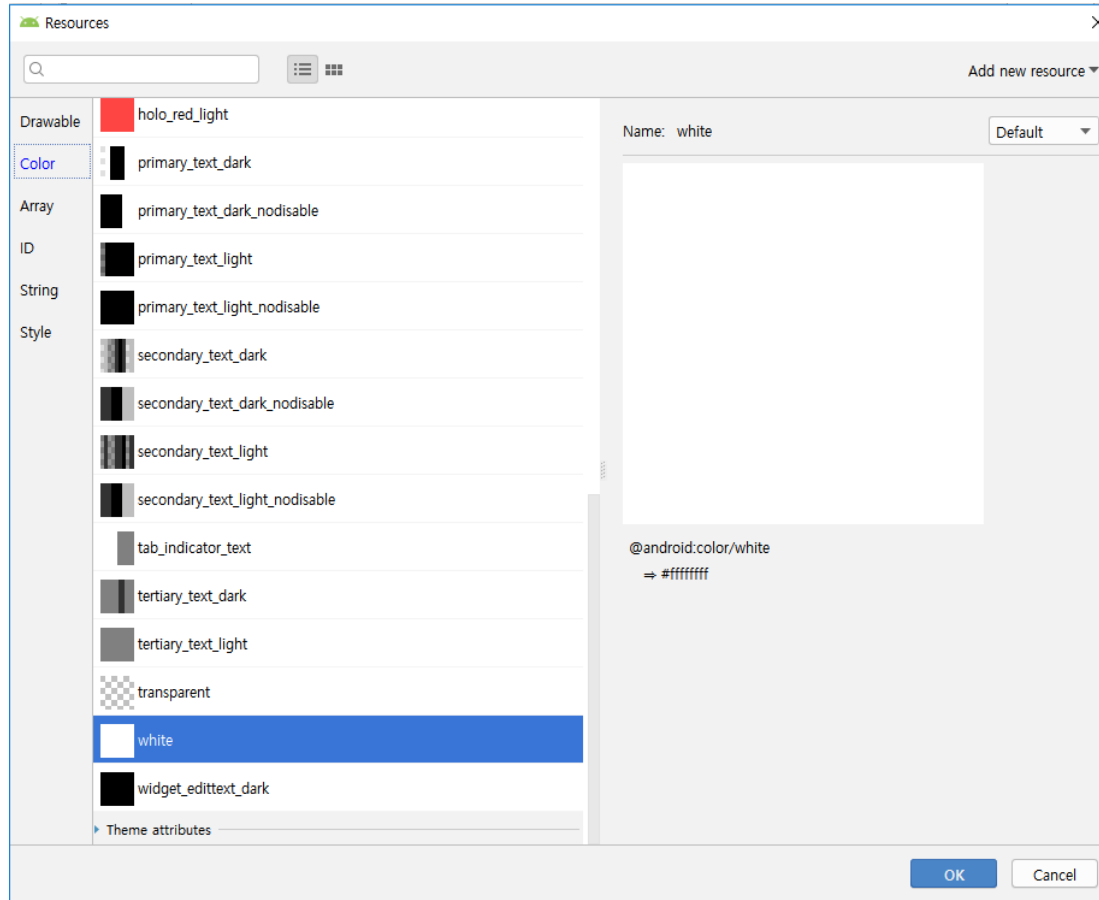
▶ 속성을 다음과 같이 설정



실로폰 앱 만들기

▶ 텍스트 뷰로 음판 만들기



▶ 글자 색상 선택 : @android:color/white

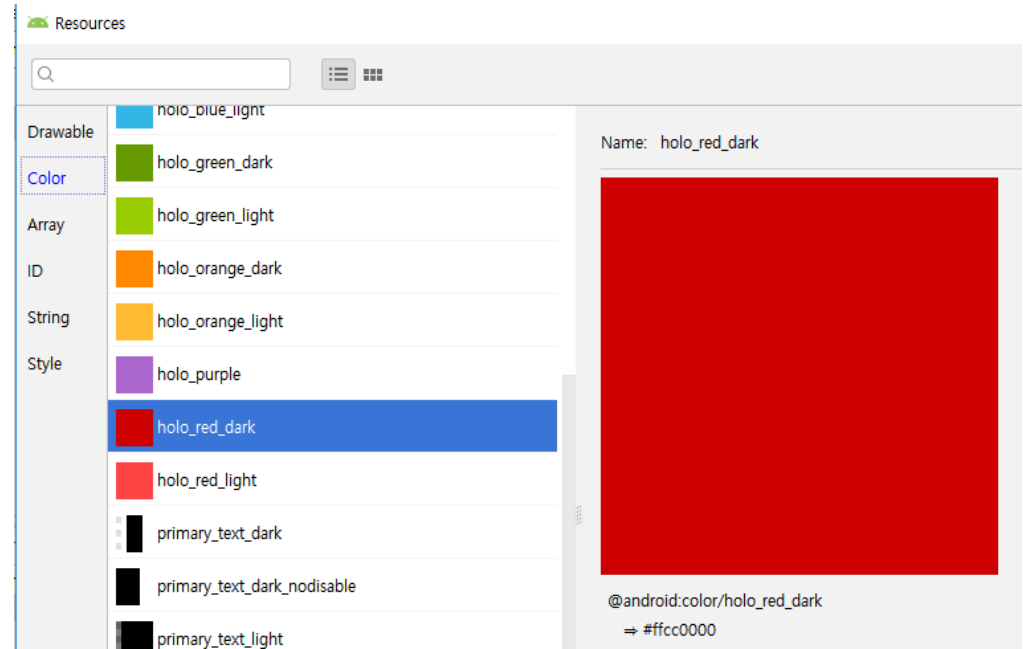


실로폰 앱 만들기

▶ 텍스트 뷰로 음판 만들기

- ▶ 배경색 선택을 위하여 [View all attributes](#) ⇄ 클릭 후 background에서 글자색과 동일한 과정으로 선택 : @android.color/holo_red_dark

Attributes	
id	do1
layout_width	50dp
layout_height	0dp
▶ Constraints	
▶ Layout_Margin	[?, ?, 16dp, ?, 16dp]
▶ Padding	[?, ?, ?, ?, ?]
▶ Theme	
background	 @android:color/hc ... 
▶ gravity	[center]
layout_editor_absoluteX	




실로폰 앱 만들기

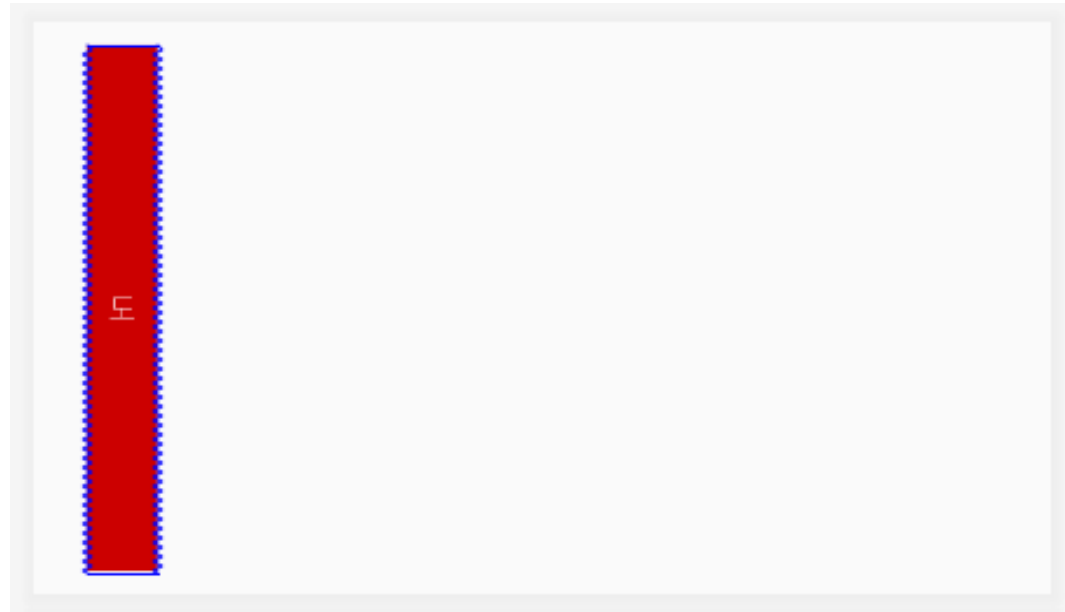
▶ 텍스트 뷰로 음판 만들기

▶ 글자를 가운데로 정렬하기 위하여 gravity 속성을 center로 설정

▶ '도' 완성!

Attributes

id	do1
layout_width	50dp
layout_height	0dp
▶ Constraints	
▶ Layout_Margin	[?, ?, 16dp, ?, 16dp]
▶ Padding	[?, ?, ?, ?, ?]
▶ Theme	
background	 @android:color/holo_red_dark
▼ gravity	[center]
bottom	<input type="checkbox"/>
clip_horizontal	<input type="checkbox"/>
center	<input checked="" type="checkbox"/>
clip_vertical	<input type="checkbox"/>
start	<input type="checkbox"/>
right	<input type="checkbox"/>
center_horizontal	<input checked="" type="checkbox"/>
fill	<input type="checkbox"/>



실로폰 앱 만들기

▶ 텍스트 뷰로 음판 만들기

▶ 같은 방식으로 음판을 7개 더 배치

▶ 공통 속성

▷ layout_width : 50dp

▷ layout_height : match_constraint

▷ textAppearance : AppCompat.Large

▷ textColor : @android:color/white

▷ gravity : center

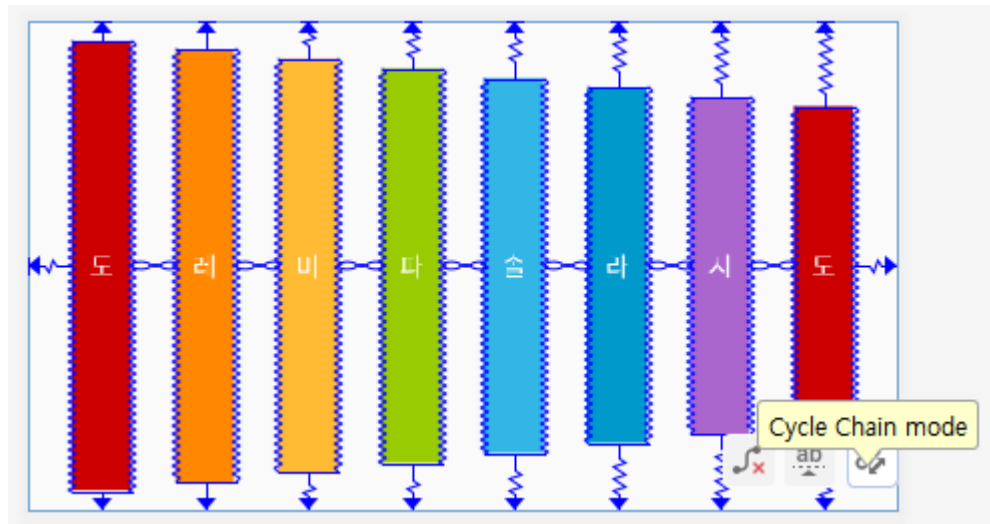
▶ 개별 속성

ID	위, 아래 여백	Text	background
re	24	레	@android:color/holo_orange_dark
mi	32	미	@android:color/holo_orange_light
fa	40	파	@android:color/holo_green_light
sol	48	솔	@android:color/holo_blue_light
la	56	라	@android:color/holo_blue_dark
si	64	시	@android:color/holo_purple
do2	72	도	@android:color/holo_red_dark

실로폰 앱 만들기

▶ 텍스트 뷰로 음판 만들기

- ▶ 음판을 정렬하기 위하여 모든 음판을 화면에 적당히 배치하고 ctrl 버튼을 누른 상태에서 모든 텍스트뷰를 선택
 - ▷ 선택된 모든 뷰는 수평 방향 체인으로 설정됨
- ▶ 체인으로 연결된 뷰 중에서 아무거나 선택한 상태로 Cycle chain mode 아이콘을 클릭하면서 아래와 같이 좌우에 균등한 여백이 설정되는 모양이 될 때까지 클릭

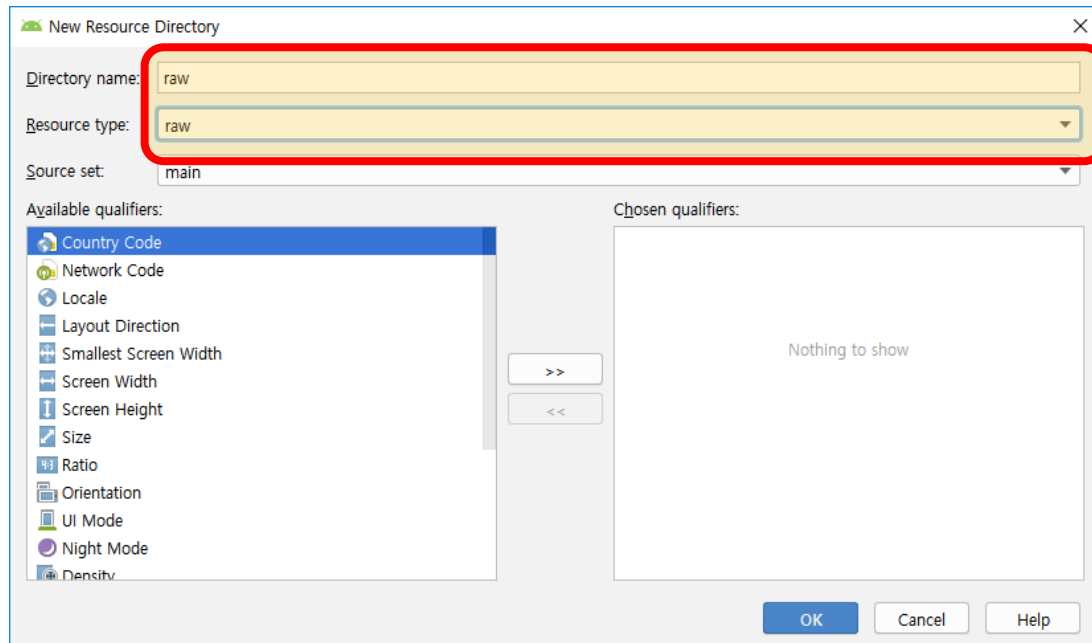


실로폰 앱 만들기

▶ 소리 재생하기

▶ raw 리소스 디렉터리 추가

- ▶ wav, mp3 등의 사운드 파일은 raw 리소스 디렉터리에서 관리
- ▶ 해당 디렉터리는 기본 생성이 되지 않기 때문에 추가
- ▶ raw 디렉터를 생성하려면 프로젝트 창에서 res 폴더를 선택하고 마우스 우 클릭 후 new - Android Resource Directory 클릭



실로폰 앱 만들기

▶ 소리 재생하기

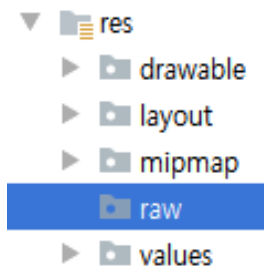
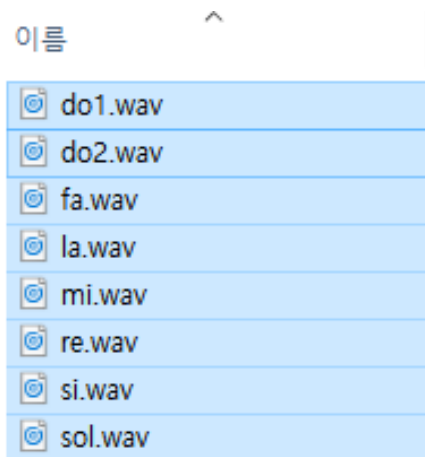
▶ 실로폰 소리 파일을 강의 사이트에서 다운로드

▶ scale.zip

▶ 다운로드 후 소리 파일의 이름을 변경해도 상관없지만 파일명은 소문자와 '_'기호, 숫자로 구성

■ 파일명에 숫자는 사용 가능하지만 숫자로 시작은 불가

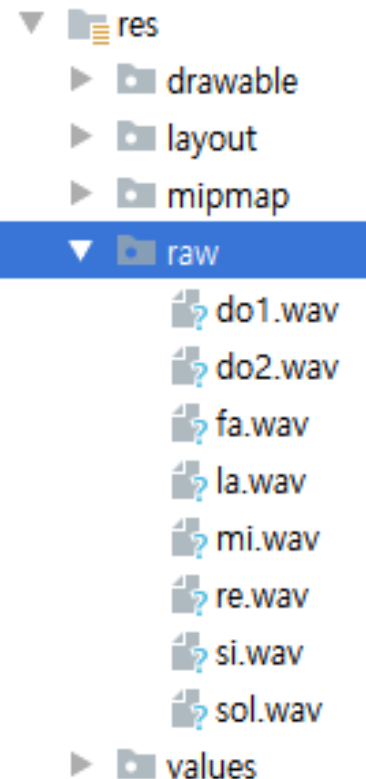
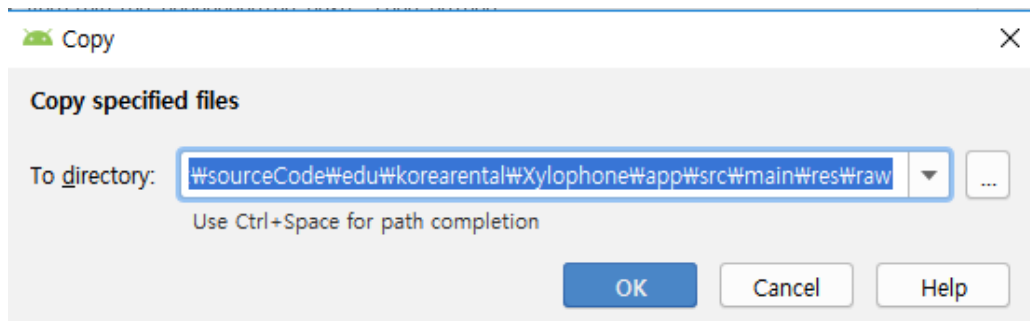
▶ 파일 전체 선택 후 raw 디렉터리에 ctrl + c ➔ ctrl + v



실로폰 앱 만들기

▶ 소리 재생하기

▶ OK ➔ 복사 확인



실로폰 앱 만들기

▶ 소리 재생하기

- ▶ 안드로이드에서 사운드를 재생하기 위한 대표적인 방법은 MediaPlayer 클래스를 사용하거나 SoundPool 클래스를 사용하는 방법이 있음
- ▶ MediaPlayer는 일반적인 사운드 파일 재생과 비디오 파일 재생에 사용
 - ▶ 재생이 끝나면 반드시 release() 메소드를 호출하여 자원을 해제해야 함

```
//raw 디렉터리의 파일을 사용하는 예
val mediaPlayer = MediaPlayer.create(this, R.raw.do1)
button.setOnClickListener{mediaPlayer.start()}
...
//사용이 끝나면 자원 해제
mediaplayer.release()
```

- ▶ 소리를 한번만 재생하는 경우나 노래, 배경음에 주로 사용

실로폰 앱 만들기

▶ 소리 재생하기

- ▶ 실로폰과 같이 음계판을 클릭할 때 마다 연속해서 소리를 재생해야 할 경우에는 SoundPool 클래스가 더 유용
- ▶ Builder().build() 메서드로 SoundPool 객체를 생성하고 load() 메서드로 소리 파일을 로드하여 해당 아이디를 반환

```
val soundPool = SoundPool.Builder().build()

val soundId = soundPool.load(this, R.raw.do1, 1)
button.setOnClickListener{soundPool.play(soundId, 1.0f, 1.0f, 0, 0, 1.0f)}
```

▶ load()원형

```
//음원을 준비하여 id를 반환
load(context: Context, resId: Int, priority: Int):
```

- ▶ context : 컨텍스트를 지정하거나 액티비티를 지정
- ▶ resId : 재생할 raw 디렉터리의 소리 파일 리소스를 지정
- ▶ priority : 우선 순위를 지정하며 숫자가 높을 수록 우선 순위가 높음

실로폰 앱 만들기

▶ 소리 재생하기

▶ play()원형

//음원을 재생

```
play(soundId: Int, leftVolume: Float, rightVolume: Float, priority: Int, loop: Int, rate: Float):
```

- ▶ soundId : load()메서드에서 반환된 음원의 id를 지정
- ▶ leftVolume : 왼쪽 볼륨을 0.0 ~ 1.0사이에서 지정
- ▶ rightVolume : 오른쪽 볼륨을 0.0 ~ 1.0사이에서 지정
- ▶ priority : 우선순위를 지정하며 0이 가장 낮은 우선순위
- ▶ loop : 반복을 여부를 지정하며 -1이면 반복하고 0이면 반복하지 않음
- ▶ rate : 재생속도를 지정하며 1.0이면 보통이고 0.5는 0.5배속, 2.0은 2.0배속임

실로폰 앱 만들기

▶ 소리 재생하기

▶ SoundPool 클래스를 사용하여 소리 재생을 구현

▶ SoundPool 초기화 버전 분기

▶ MainActivity파일을 열고 SoundPool 객체를 초기화하는 코드를 작성

▶ setMaxStreams()는 한꺼번에 재생하는 음원의 개수를 지정

- 본 예제에서는 8개를 동시에 재생 가능하도록 설정

```
class MainActivity : AppCompatActivity() {  
    private val soundPool = SoundPool.Builder().setMaxStreams(8).build()
```

▶ 본 프로젝트는 minSdkVersion을 19로 설정하여 생성하였고 위 코드는 안드로이드 API 21(안드로이드 5.0 롤리팝)부터 지원되기 때문에 코드에 오류 표시

▶ 따라서 위와 같은 코드를 작성하면 5.0 미만의 버전에서는 런타임 에러로 종료됨

실로폰 앱 만들기

▶ 소리 재생하기

▶ SoundPool 클래스를 사용하여 소리 재생을 구현

▶ SoundPool 초기화 버전 분기

▶ 이전 버전에서는 아래와 같은 코드로 사운드 리소스를 처리

```
//안드로이드 5.0 미만에서 사용하는 코드  
private val soundPool = SoundPool(8, AudioManager.STREAM_MUSIC, 0)
```

▶ 첫번째 인자는 최대 재생 스트림 개수

▶ 두번째 인자는 음원의 종류

▶ 세번째 인자는 음질을 나타내고 기본값은 0

▶ 위 코드를 편집기에 입력할 경우 생성자에 줄이 표시되는데, deprecated 되었기 때문

■ 더 이상 사용되지 않는다는 메시지

실로폰 앱 만들기

▶ 소리 재생하기

▶ SoundPool 클래스를 사용하여 소리 재생을 구현

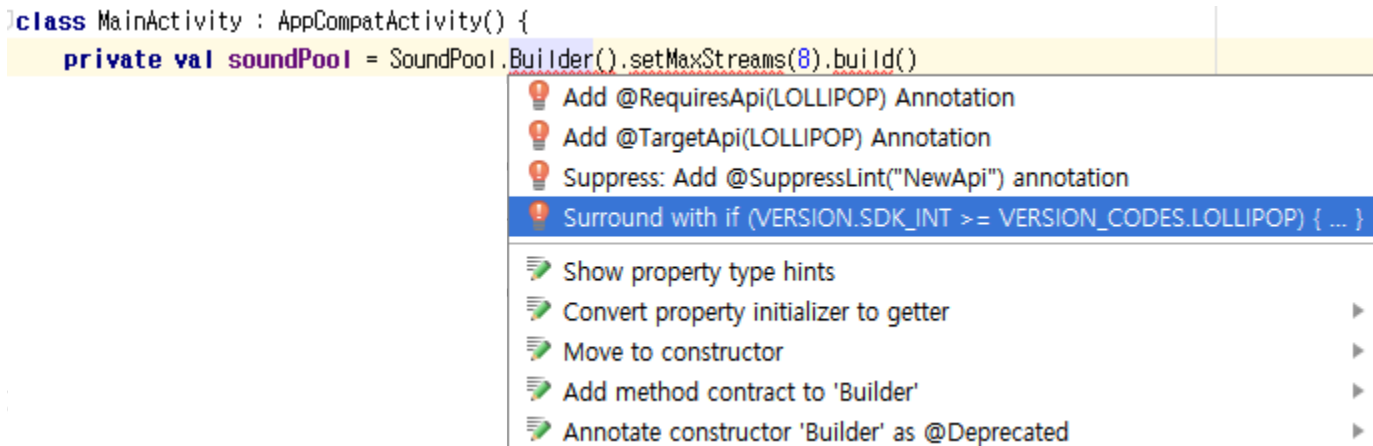
▶ SoundPool 초기화 버전 분기

▶ 버전에 따라 다른 사운드 리소스를 처리하는 코드를 추가

```
class MainActivity : AppCompatActivity() {  
    private val soundPool = SoundPool.Builder().setMaxStreams(8).build()  
}
```

▶ 아래 코드의 에러 표시인 빨간 줄에서 단축키 Alt + Enter를 눌러서 제안사항을 확인

▶ 제안사항 중 Surround with로 시작하는 항목을 클릭



실로폰 앱 만들기

▶ 소리 재생하기

▶ SoundPool 클래스를 사용하여 소리 재생을 구현

▶ SoundPool 초기화 버전 분기

▶ 구버전에서 사용하는 SoundPool 객체 초기화 코드를 추가

```
class MainActivity : AppCompatActivity() {  
    private val soundPool = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {  
        SoundPool.Builder().setMaxStreams(8).build()  
    } else {  
        SoundPool(maxStreams: 8, AudioManager.STREAM_MUSIC, srcQuality: 0)  
    }  
}
```

▶ 빌드 버전이 롤리팝을 기준으로 실행되는 코드가 달라짐

실로폰 앱 만들기

▶ 소리 재생하기

▶ 음판에 동적으로 클릭 이벤트 정의

- ▶ 음판을 눌렀을 때 해당 음계를 재생하도록 MainActivity.kt에 다음과 같이 코드를 작성
- ▶ 아래 코드는 listOf()를 사용하여 텍스트 뷰의 ID와 음원 파일의 리소스 ID를 연관 지은 Pair 객체 8개를 리스트 객체 sounds로 생성
- ▶ Pair 클래스는 두개의 연관된 객체를 저장

```
private val sounds = listOf(  
    Pair(R.id.do1, R.raw.do1),  
    Pair(R.id.re, R.raw.re),  
    Pair(R.id.mi, R.raw.mi),  
    Pair(R.id.fa, R.raw.fa),  
    Pair(R.id.sol, R.raw.sol),  
    Pair(R.id.la, R.raw.la),  
    Pair(R.id.si, R.raw.si),  
    Pair(R.id.do2, R.raw.do2)  
)
```

실로폰 앱 만들기

▶ 소리 재생하기

▶ 음판에 동적으로 클릭 이벤트 정의

▶ sounds 리스트를 forEach()를 사용하여 요소를 하나씩 꺼내서 tune()메서드에서 전달

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
  
    setContentView(R.layout.activity_main)  
  
    sounds.forEach { tune(it) }  
}
```

실로폰 앱 만들기

▶ 소리 재생하기

▶ 음판에 동적으로 클릭 이벤트 정의

- ▶ tune()메서드는 Pair 객체를 받아서 load()메서드로 음원의 ID를 얻고 findViewById() 메서드로 텍스트 뷰의 ID에 해당하는 뷰를 얻고 텍스트 뷰를 클릭했을 때 음원을 재생

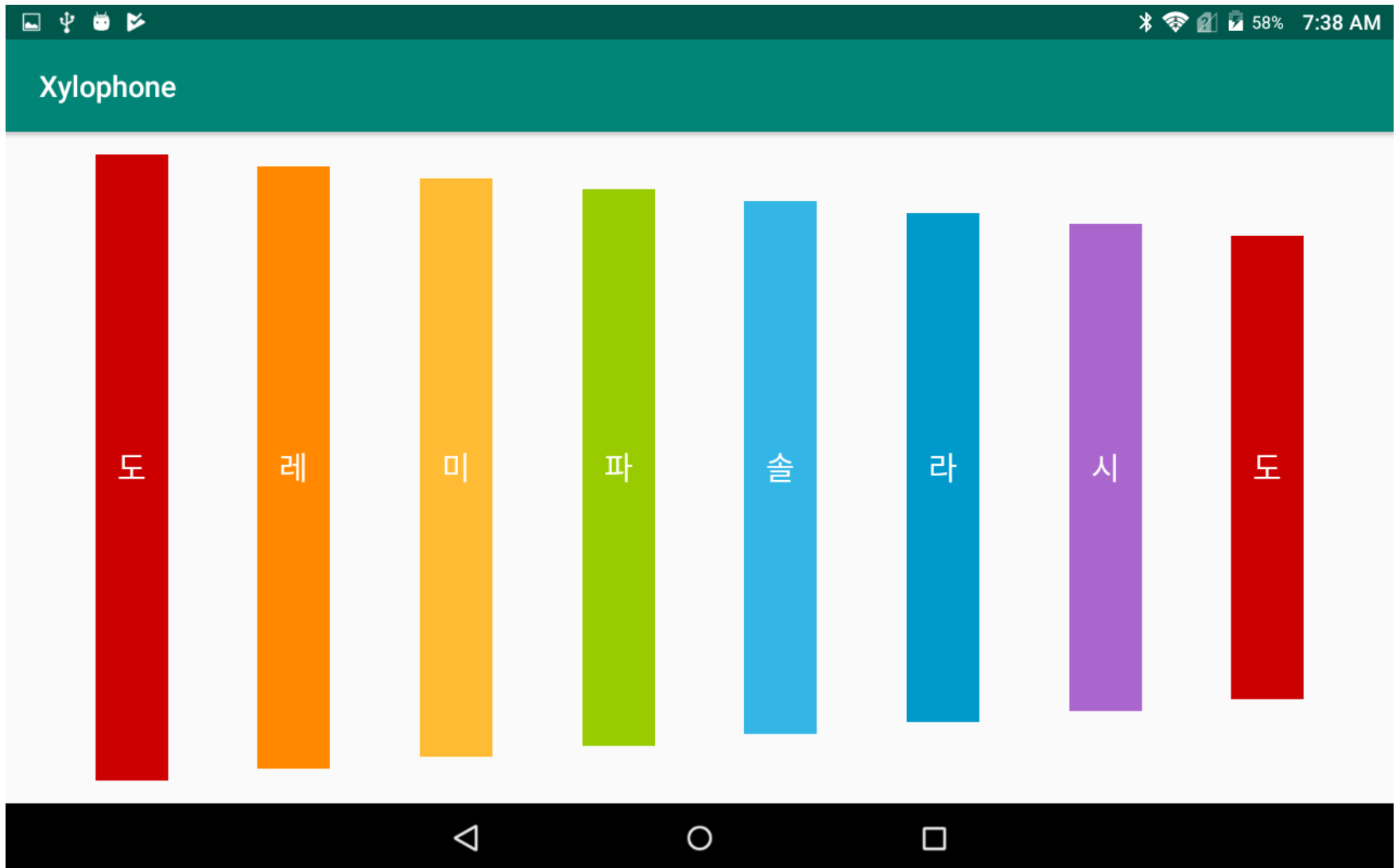
```
private fun tune(pitch: Pair<Int, Int>) {  
    val soundId = soundPool.load(context this, pitch.second, priority: 1)  
    findViewById<TextView>(pitch.first).setOnClickListener { it View!  
        soundPool.play(soundId, leftVolume: 1.0f, rightVolume: 1.0f, priority: 0, loop: 0, rate: 1.0f)  
    }  
}
```

- ▶ 앱을 종료할 때는 반드시 release()메서드를 호출하여 SoundPool 객체의 자원을 해제

```
override fun onDestroy() {  
    super.onDestroy()  
    soundPool.release()  
}
```

실로폰 앱 만들기

▶ 실행하기!



Q & A
