

# Mobile Programming

## Lec 6. Piano App



Ewha Womans University

# Image Button, Image background

2

ImageButton



Image background  
of anything(Button)

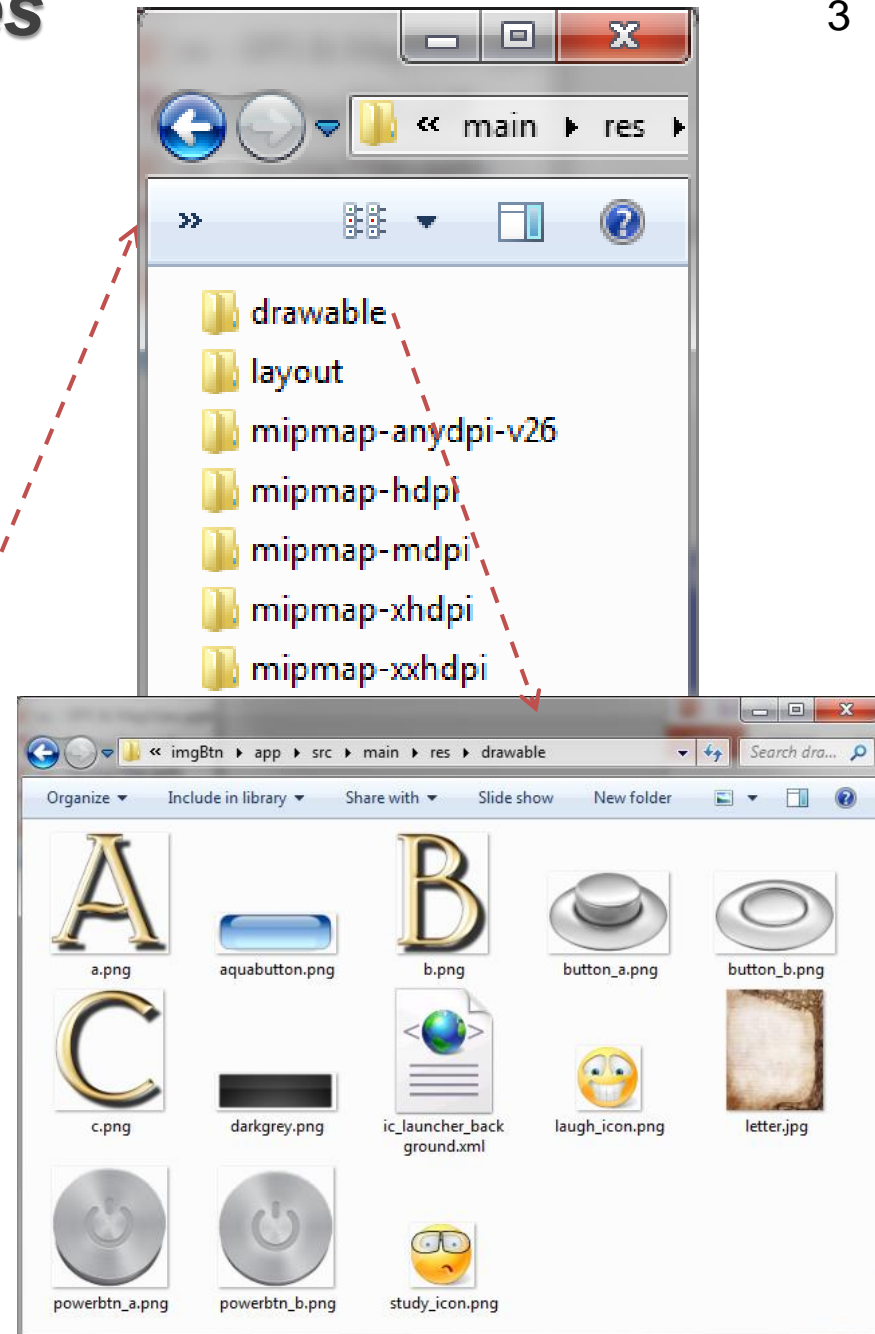
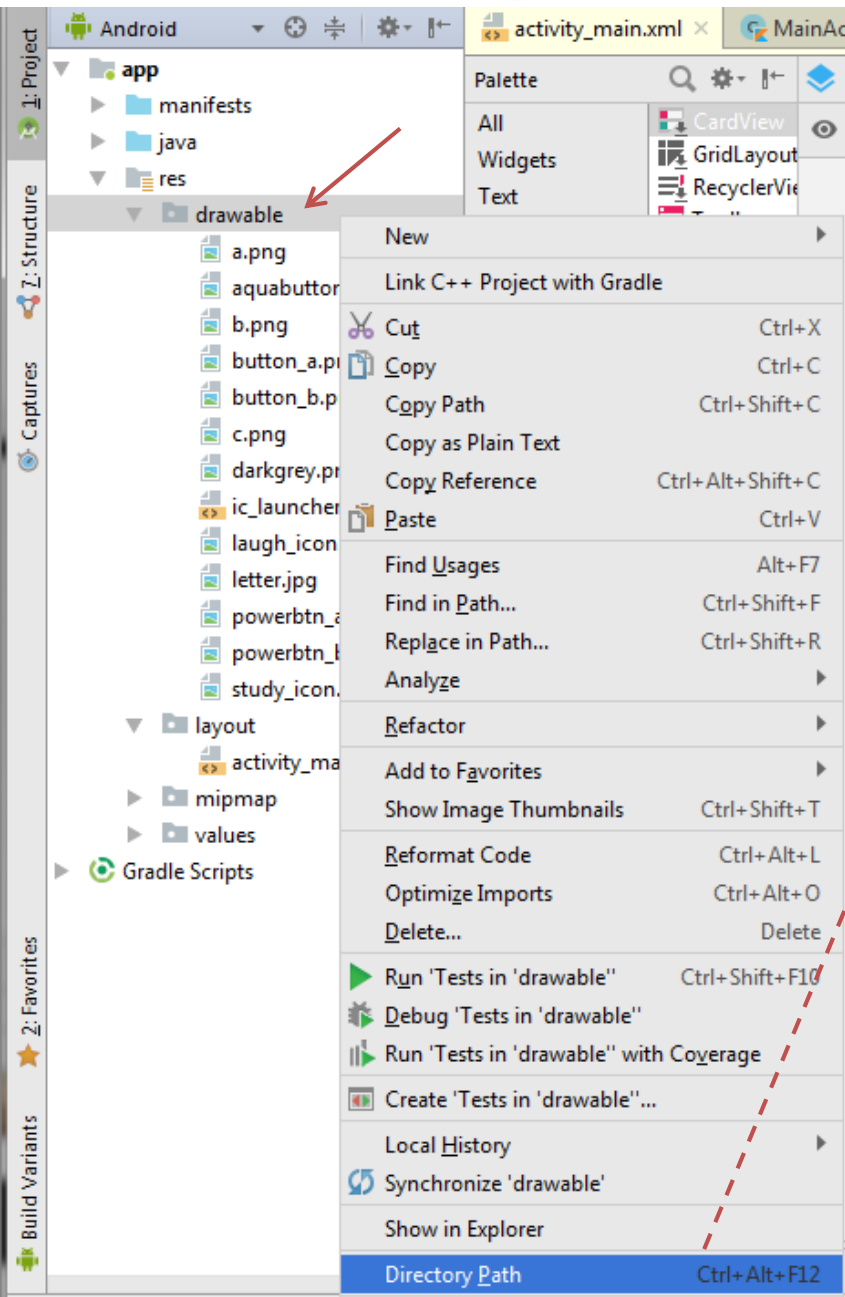


Image background  
of anything(EditText)



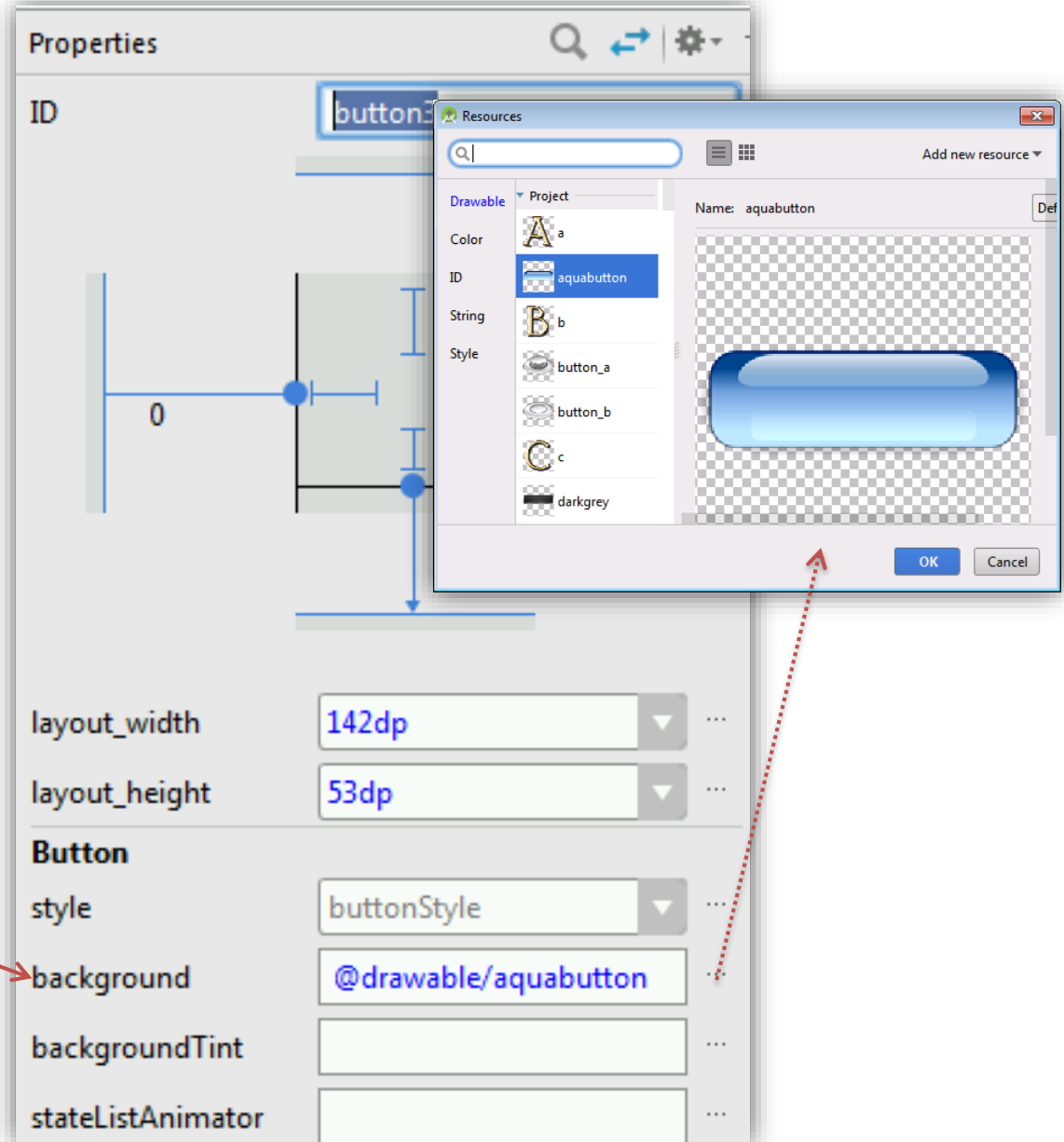
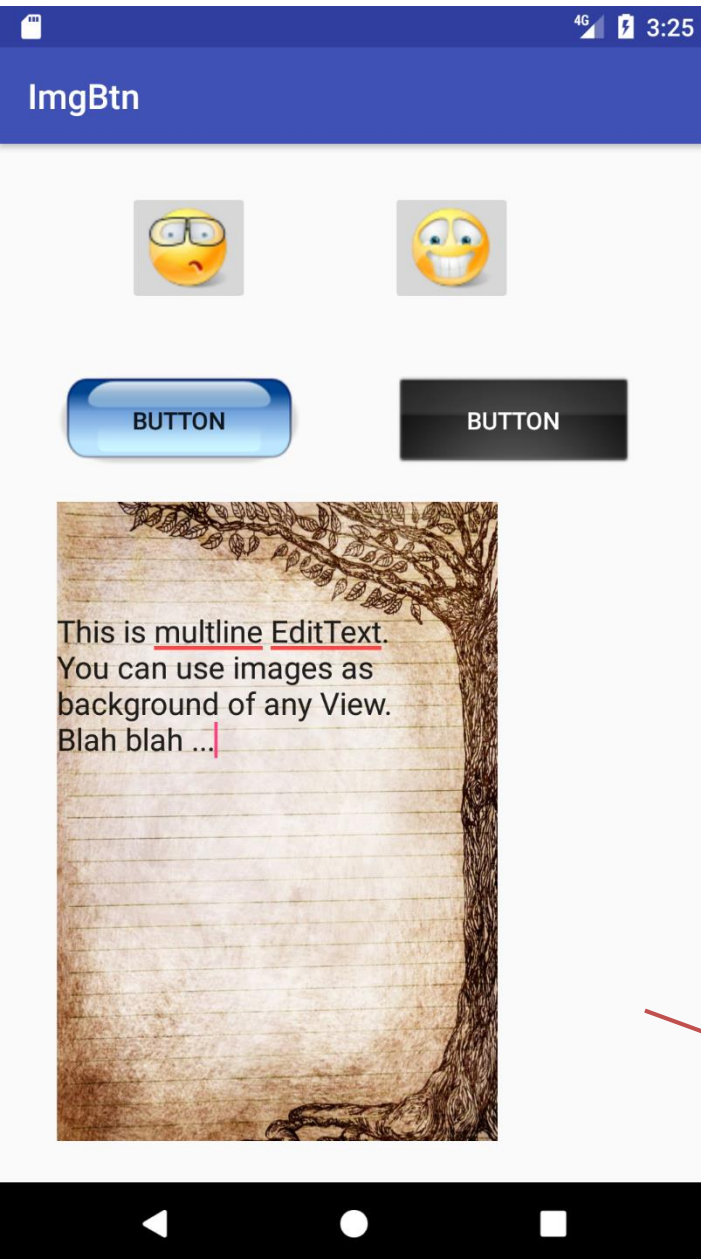
# Another way to put images

3

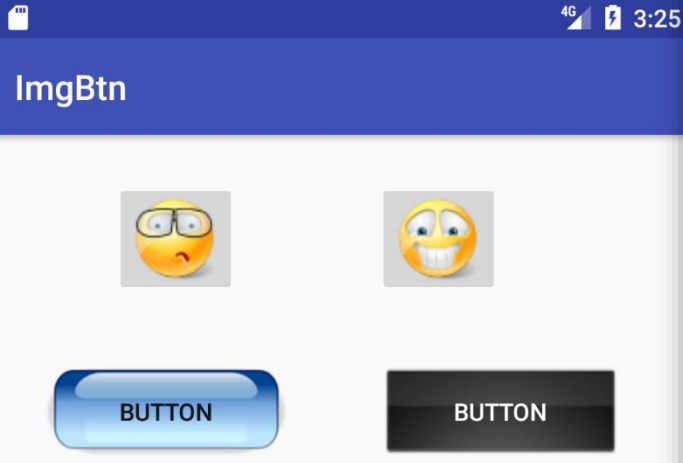


# Set image background of anything

4



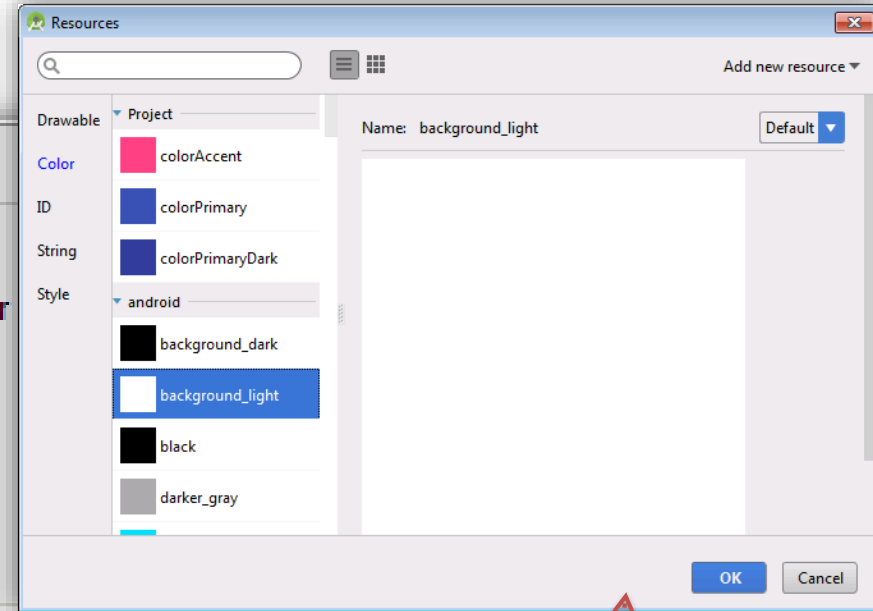
# Select color



Properties  
backgroundTint  
stateListAnimator  
elevation  
visibility  
onClick

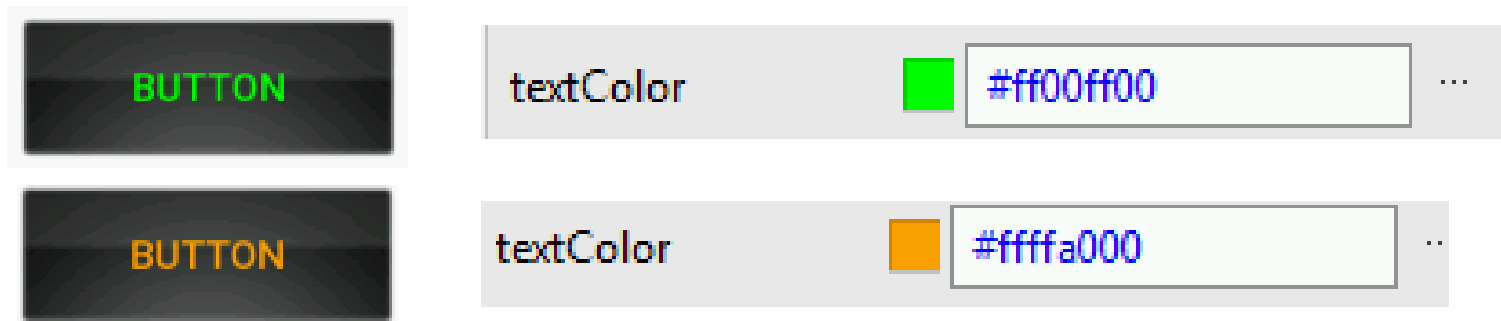
## TextView

text Button ...  
text ...  
contentDescription ...  
▼ textAppearance AppCompat.Widget.Button ...  
fontFamily sans-serif-medium ...  
typeface none ...  
textSize 14sp ...  
lineSpacingExtra none ...  
textColor @android:color/black ...



# Or, select color by RGB number

6



RGB Color in Hexadecimal number (16진수)

16진수 수자: 0 1 2 3 4 5 6 7 8 9 A B C D E F  
10 11 12 13 14 15

#23 →  $2 \times 16 + 3 = 35$

#AB →  $10 \times 16 + 11 = 171$

#00 ~ #FF → 0 ~ 255

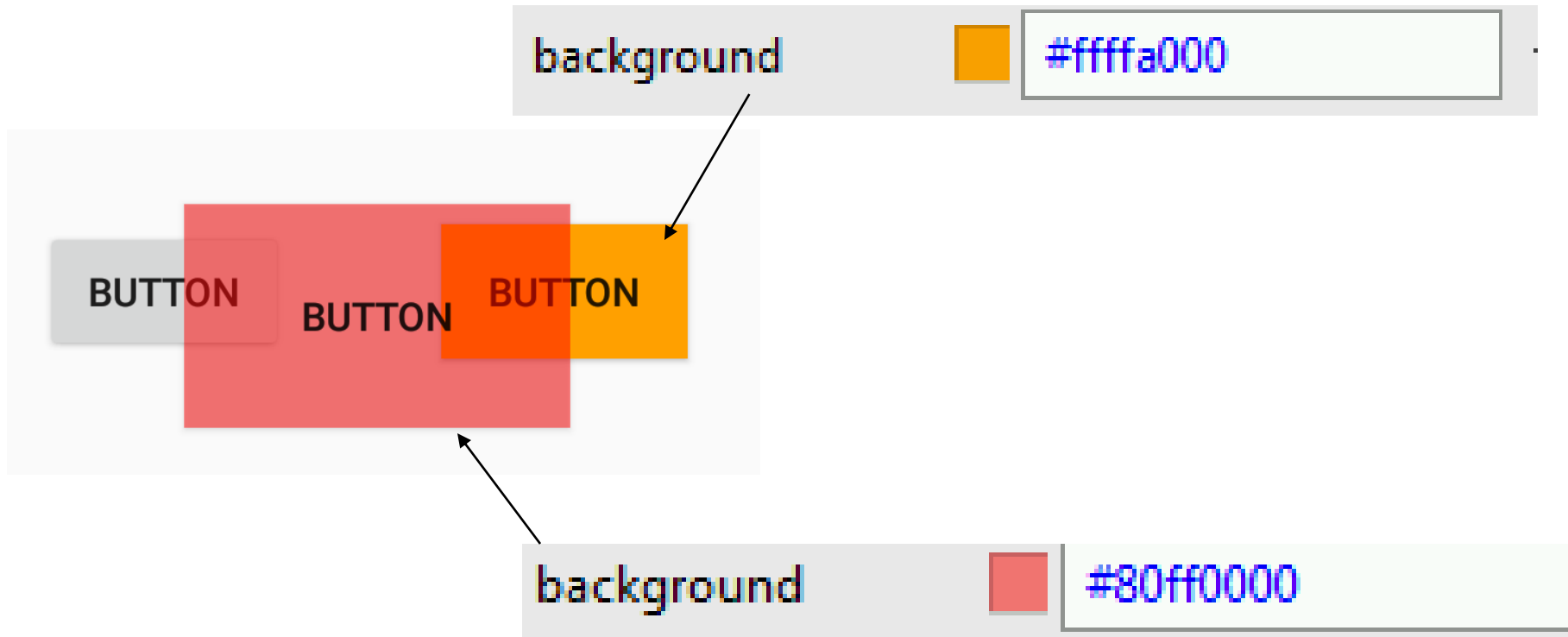
#FF00FF00

RGB color

Alpha(불투명도) : 0 ~ 255

# *background can be just a color*

7



# Image selector for button

8

<res/drawable/custom\_btn.xml>

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true" android:drawable="@drawable/button_b" />
    <item android:drawable="@drawable/button_a" />
</selector>
```





# Image selector for button

9

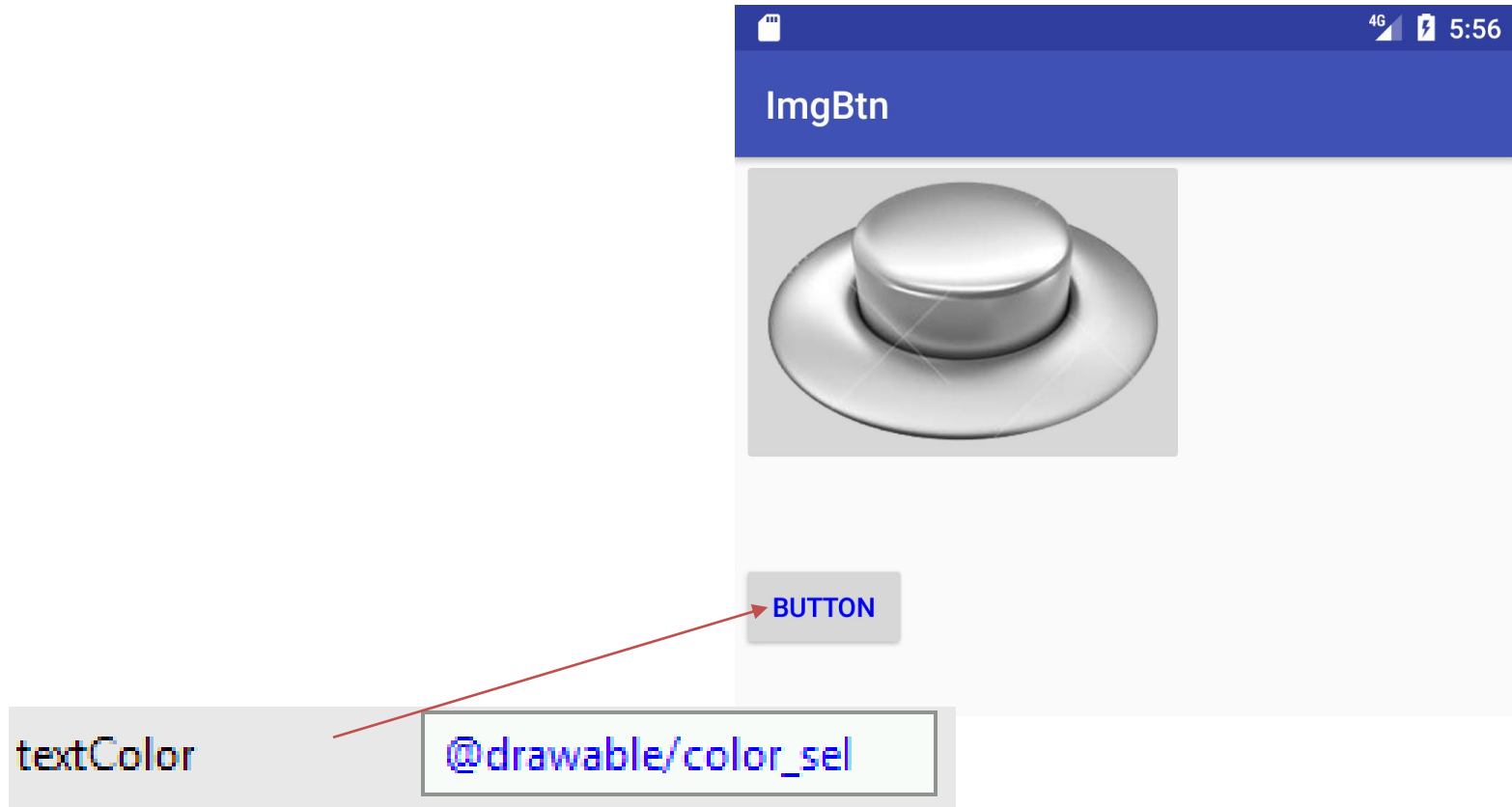
The screenshot illustrates the steps to create a drawable resource file in Android Studio:

- File Explorer:** The `res/drawable` directory is selected. Existing files include `box_mf_04.png`, `button_click.png`, and `button_hover.png`.
- Context Menu:** A right-click context menu is open, showing options like `New`, `Cut`, and `Copy`. The `New` option is expanded, showing `Drawable resource file` as the selected choice.
- New Resource File Dialog:** A dialog box is open with the following fields:
  - File name:** `custom_btn`
  - Source set:** `main`
  - Directory name:** `drawable`
- Qualifiers:** The `Available qualifiers` list includes `Country Code`, `Network Code`, `Locale`, `Layout Direction`, and `Smallest Screen Width`. The `Chosen qualifiers` list is empty, showing "Nothing to show".
- XML Editor:** The `custom_btn.xml` file is open, showing the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
</selector>
```

# color selector for button text

10

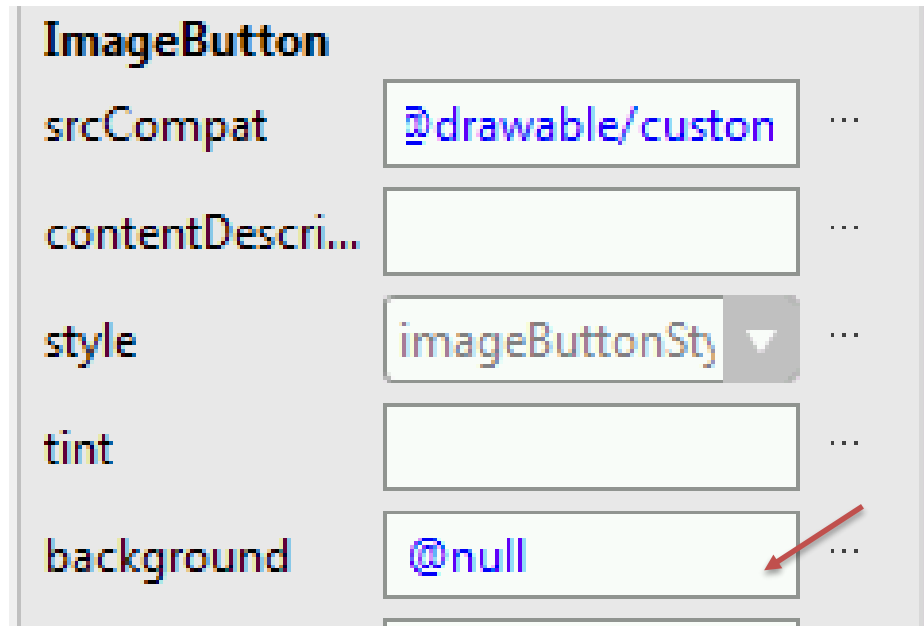


<res/drawable/color\_sel.xml>

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <selector xmlns:android="http://schemas.android.com/apk/res/android">
3     <item android:state_pressed="true" android:color="#FF0000" />
4     <item android:color="#0000FF" />
5 </selector>
```

# Set background to “@null” for transparent

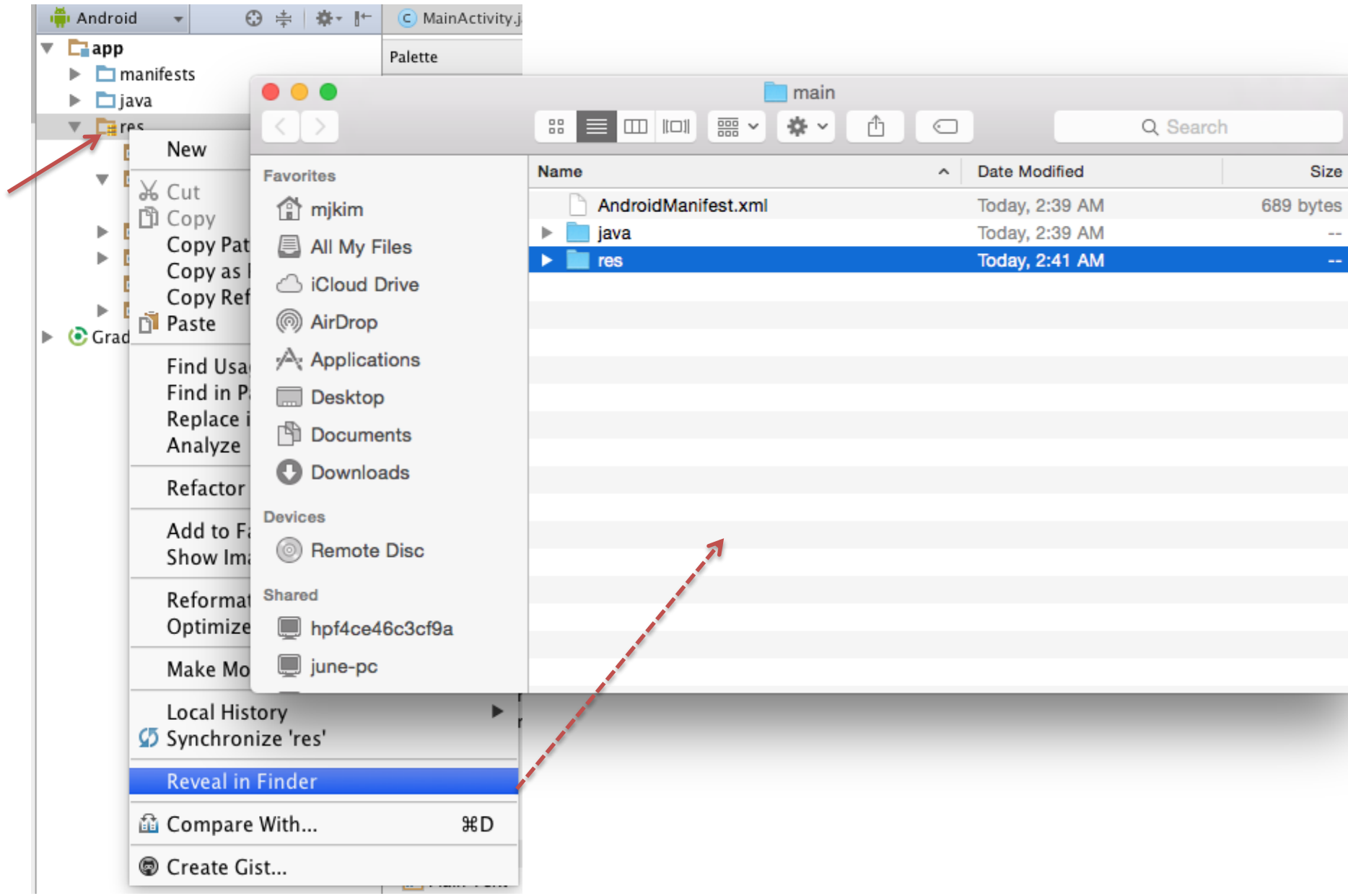
11



# Piano App

# Open the resource folder

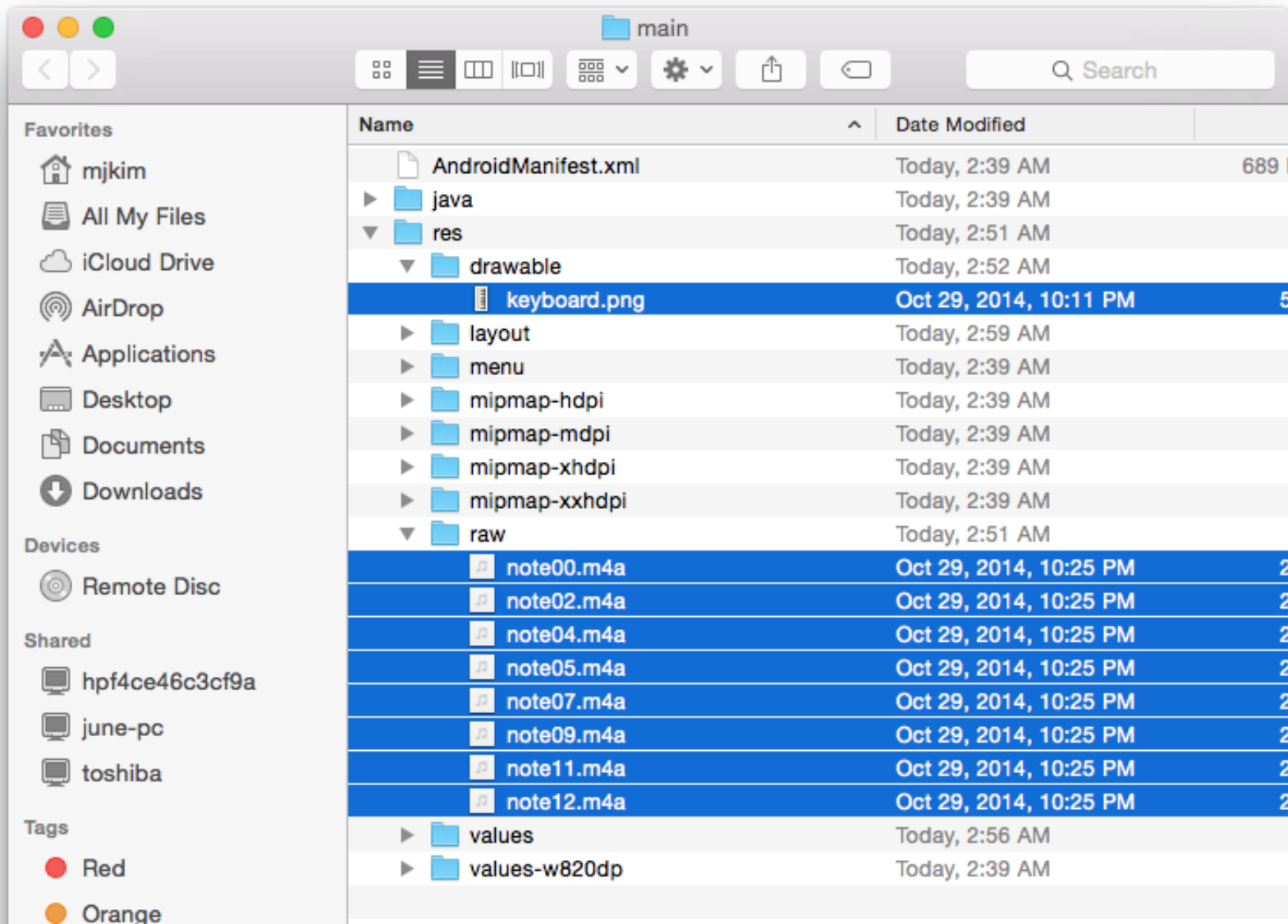
13

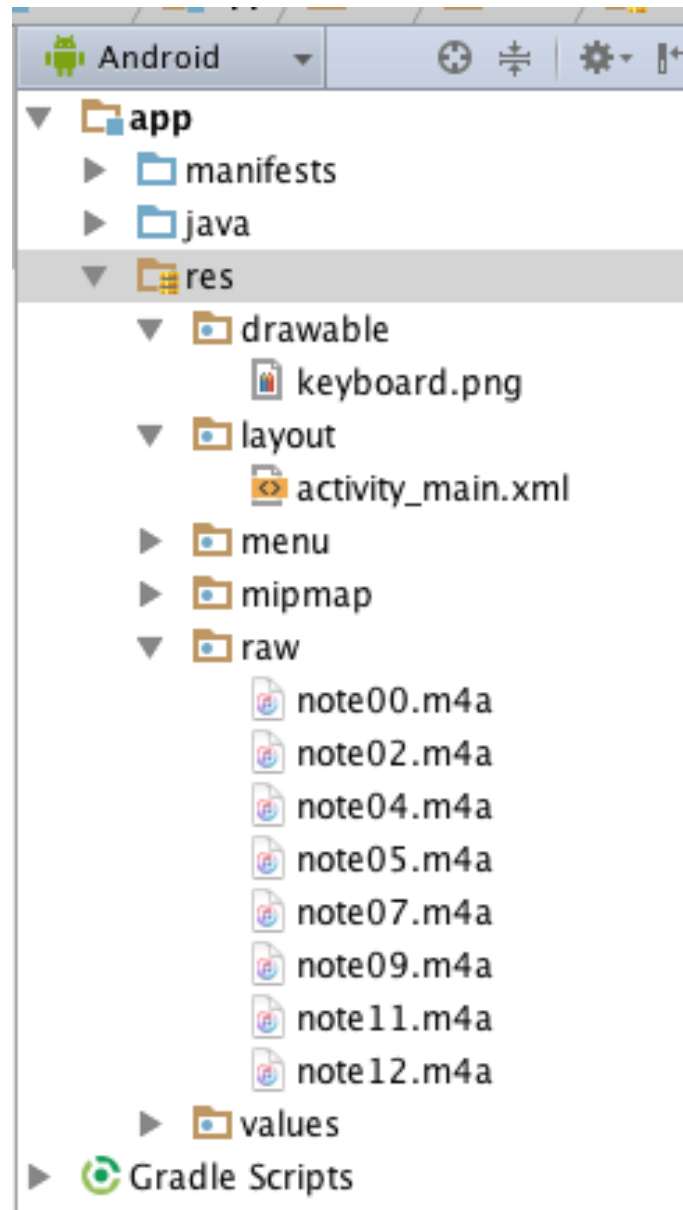


# Add mp3 files to resource

14

- make a new folder 'raw'






# Press a button to play

16

```
class MainActivity : AppCompatActivity() {  
  
    var sp = SoundPool(5, AudioManager.STREAM_MUSIC, 0)  
  
    var note = IntArray(8)  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        note[0] = sp.load(this, R.raw.note00, 1)  
        note[1] = sp.load(this, R.raw.note02, 1)  
        note[2] = sp.load(this, R.raw.note04, 1)  
        note[3] = sp.load(this, R.raw.note05, 1)  
        note[4] = sp.load(this, R.raw.note07, 1)  
        note[5] = sp.load(this, R.raw.note09, 1)  
        note[6] = sp.load(this, R.raw.note11, 1)  
        note[7] = sp.load(this, R.raw.note12, 1)  
    }  
}
```



priority



# Numbers in Kotlin

17

```
var a = 123  
var b = 12.34  
var c = 12.34f
```

```
var d = 1f  
var e = 1
```

```
textView1.text = b.toString()  
textView2.text = c.toString()  
textView3.text = (b/3).toString()  
textView4.text = (c/3).toString()  
textView5.text = (d/3).toString()  
textView6.text = (e/3).toString()
```

Type	Bit width
Double	64
Float	32
Long	64
Int	32
Short	16
Byte	8

12.34

12.34

4.113333333333333

4.113333

0.33333334

0

# Numbers in Kotlin

18

```
var a = 123  
a = 1234567890123456  
  
var b = 123L  
b = 1234567890123456
```

Type	Bit width
Double	64
Float	32
Long	64
Int	32
Short	16
Byte	8

# Numbers in Kotlin

19

```
var a = 123
a = 1234567890123456

var b = 123L
b = 1234567890123456
```

```
var a : Int = 123
a = 1234567890123456

var b : Long = 123
b = 1234567890123456

var c : Float = 12.34

var d : Double = 12.34

var e : Double = 12.34f
```

# Array : collection of indexed variables

20

```
class MainActivity : AppCompatActivity()  
  
    var sp = SoundPool(5, AudioManager.STREAM_MUSIC, 1)  
  
    var note = IntArray(8)  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        note[0] = sp.load(this, R.raw.note0)  
        note[1] = sp.load(this, R.raw.note1)  
        note[2] = sp.load(this, R.raw.note2)  
        note[3] = sp.load(this, R.raw.note3)  
        note[4] = sp.load(this, R.raw.note4)  
        note[5] = sp.load(this, R.raw.note5)  
        note[6] = sp.load(this, R.raw.note6)  
        note[7] = sp.load(this, R.raw.note7)  
    }
```

```
var note0 = 0  
var note1 = 0  
var note2 = 0  
var note3 = 0  
var note4 = 0  
var note5 = 0  
var note6 = 0  
var note7 = 0
```

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    note0 = sp.load(this, R.raw.note0)  
    note1 = sp.load(this, R.raw.note1)  
    note2 = sp.load(this, R.raw.note2)  
    note3 = sp.load(this, R.raw.note3)  
    note4 = sp.load(this, R.raw.note4)  
    note5 = sp.load(this, R.raw.note5)  
    note6 = sp.load(this, R.raw.note6)  
    note7 = sp.load(this, R.raw.note7)  
}
```

```
fun key0(view : View) {  
    sp.play(note[0], 1.0f, 1.0f, 0, 0, 1.0f)  
}  
  
fun key1(view : View) {  
    sp.play(note[1], 1.0f, 1.0f, 0, 0, 1.0f)  
}
```

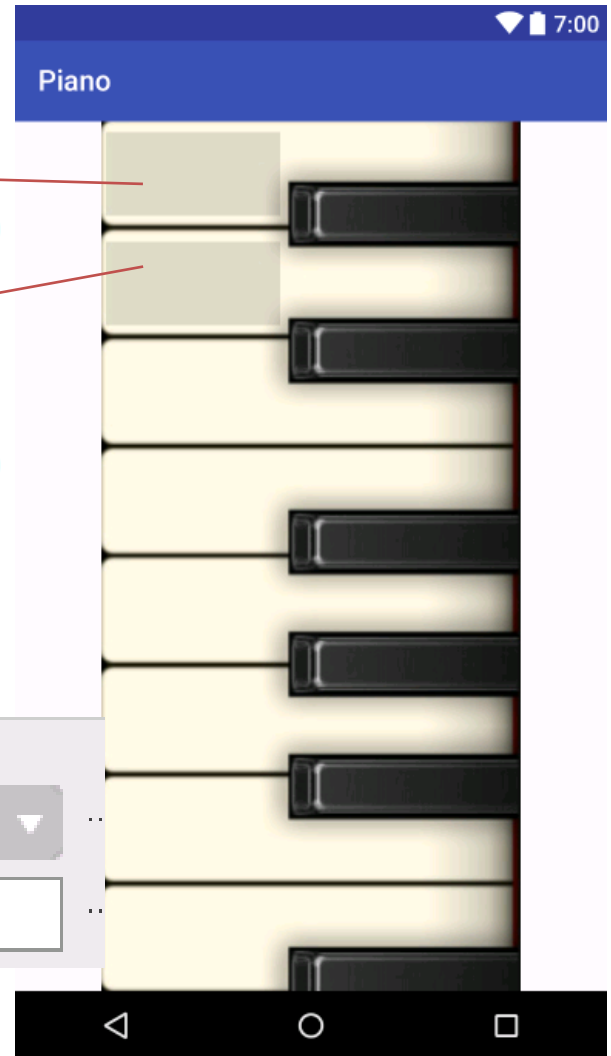
#### Button

style

buttonStyle

background

#20000000



`play(int soundID, float leftVolume, float rightVolume, int priority, int loop, float rate)`  
Play a sound from a sound ID.

# Add *nocolor(transparent)*, *transparent blue*

22



<res/values/colors.xml>

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
    <color name="colorPrimary">#3F51B5</color>
```

```
    <color name="colorPrimaryDark">#303F9F</color>
```

```
    <color name="colorAccent">#FF4081</color>
```

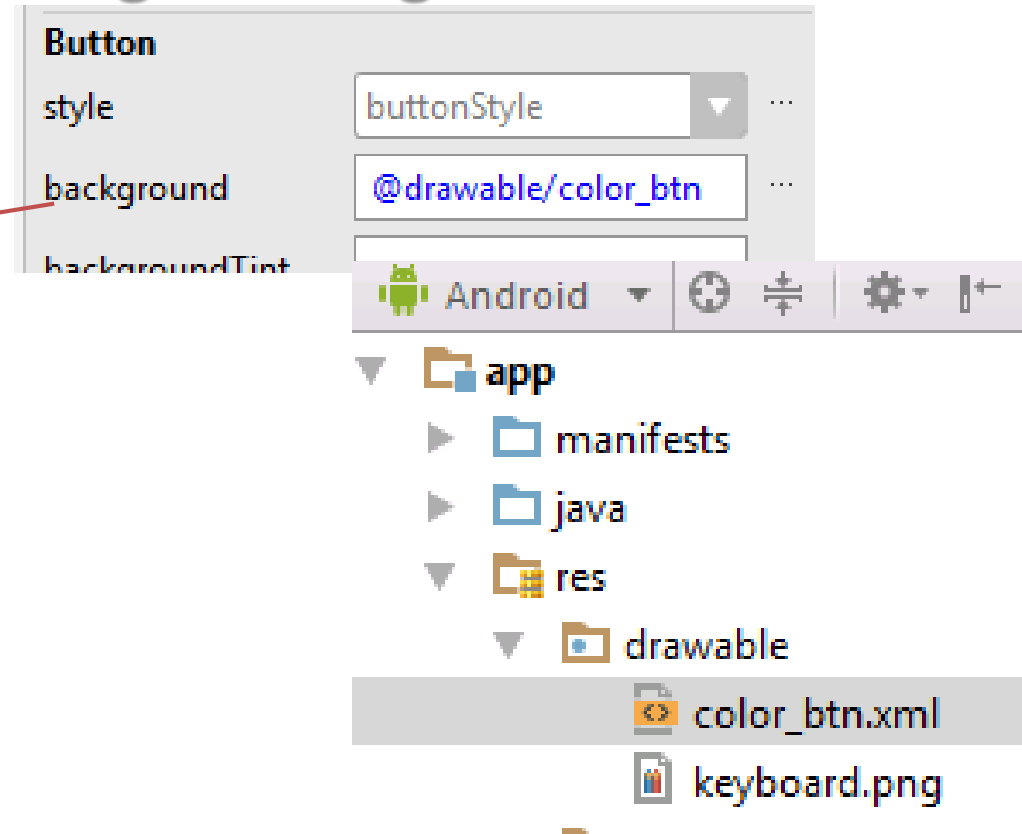
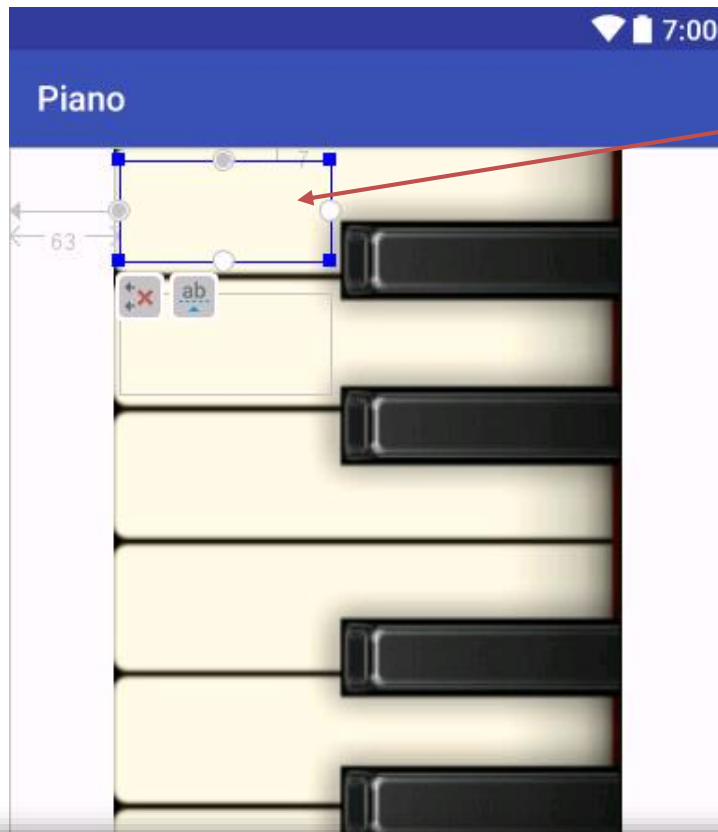
```
    <color name="nocolor">#00000000</color>
```

```
    <color name="transblue">#400000FF</color>
```

```
</resources>
```

# Button with no-text, image-background

23

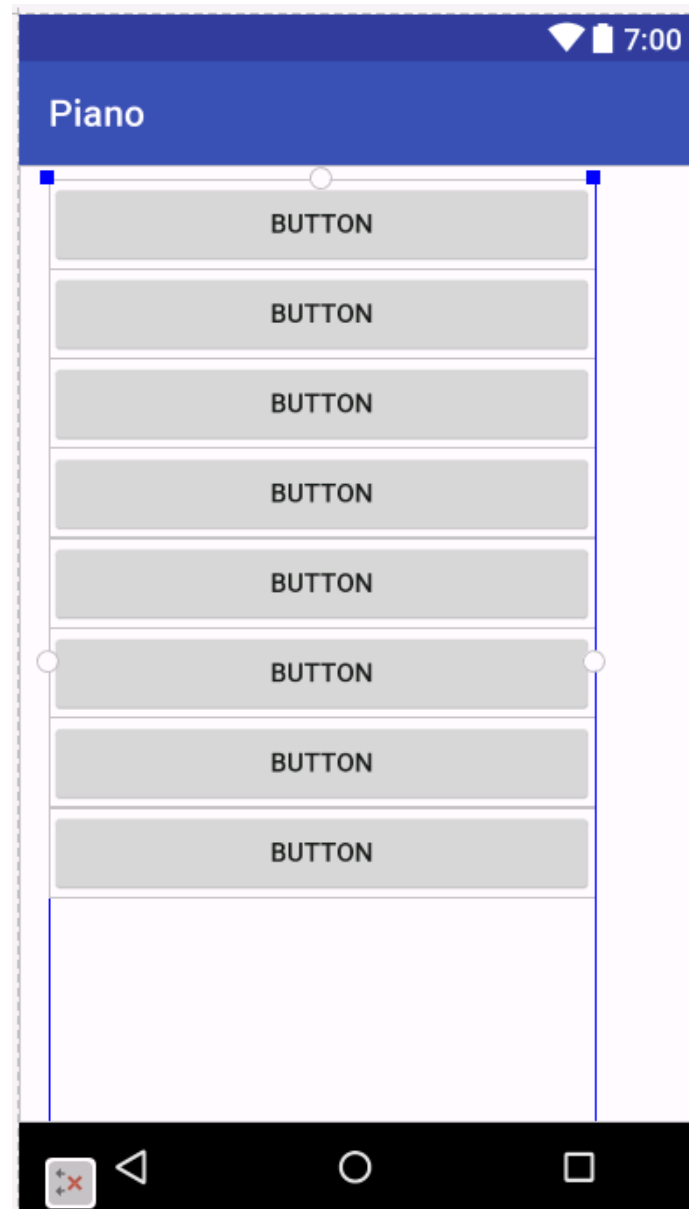
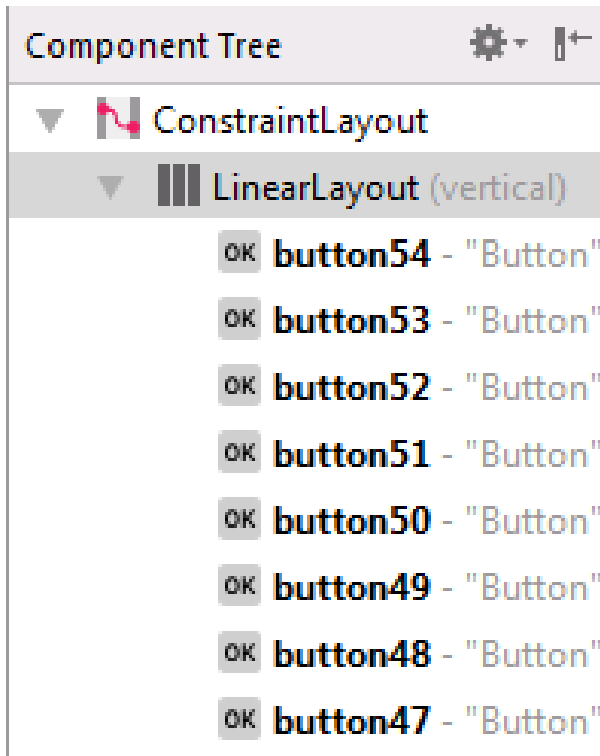


<res/drawable/color\_btn.xml>

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true" android:drawable="@color/transblue" />
    <item android:drawable="@color/nocolor" />
</selector>
```

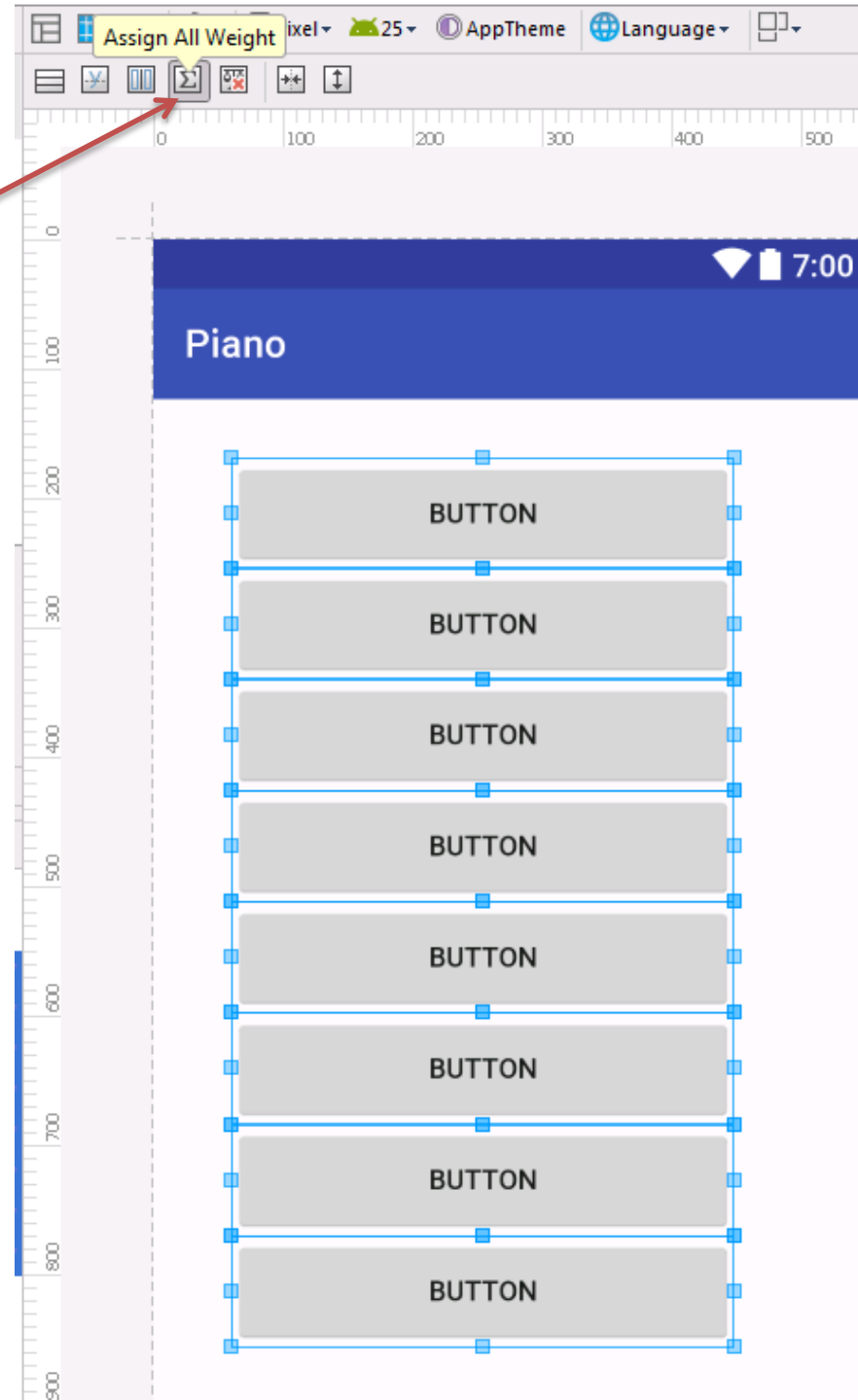
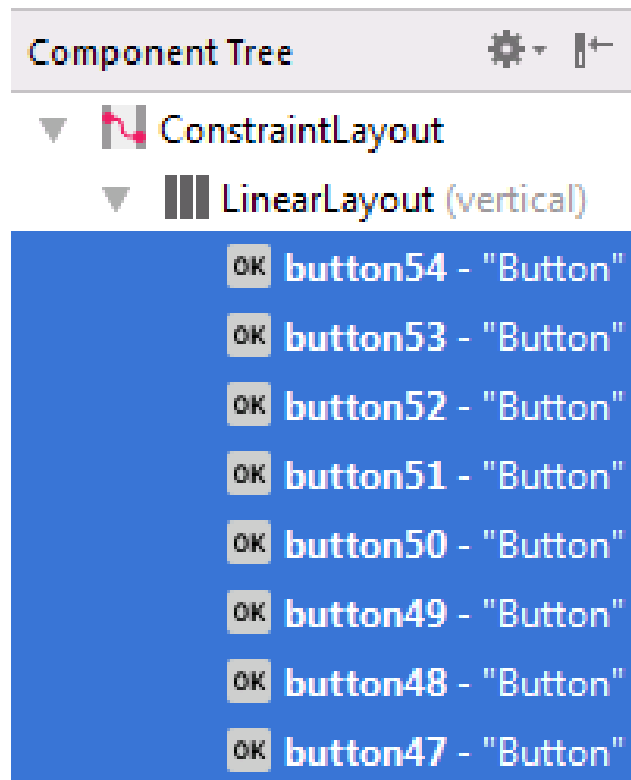
# *(Better UI layout) 8 buttons in a LinearLayout*

24



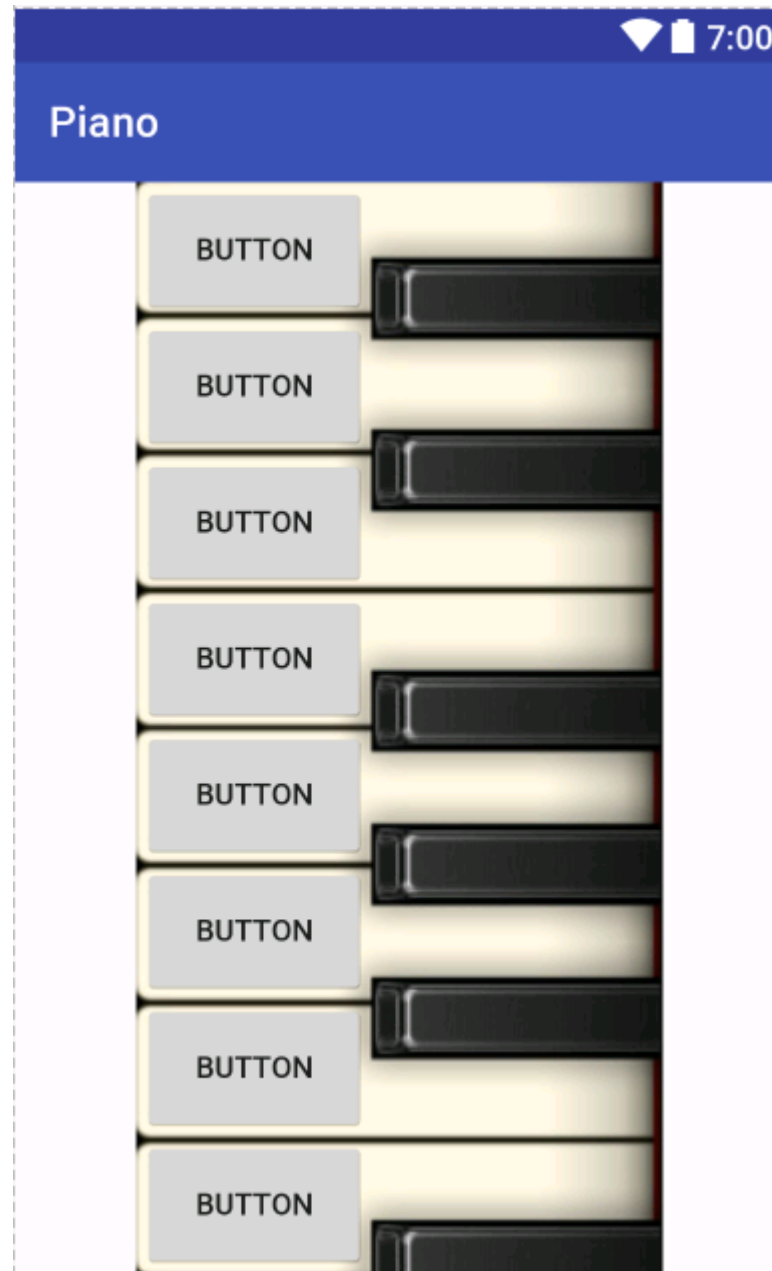
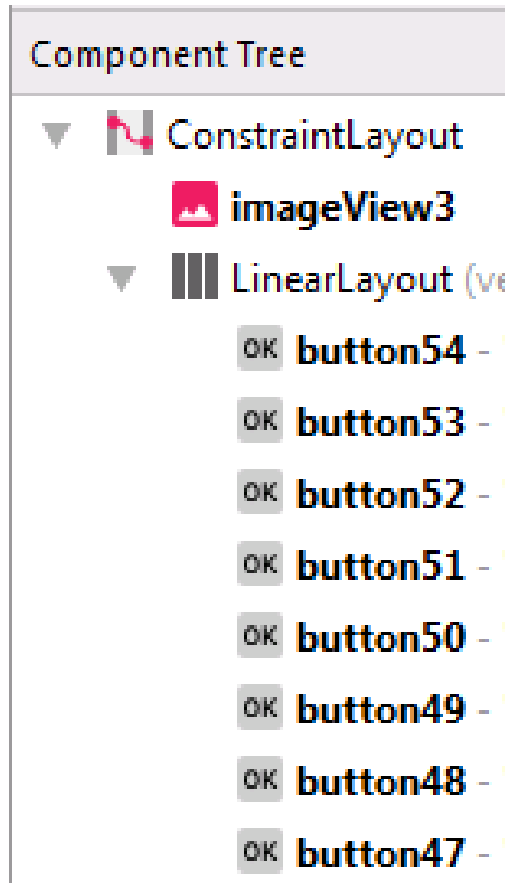


# Assign All Weight



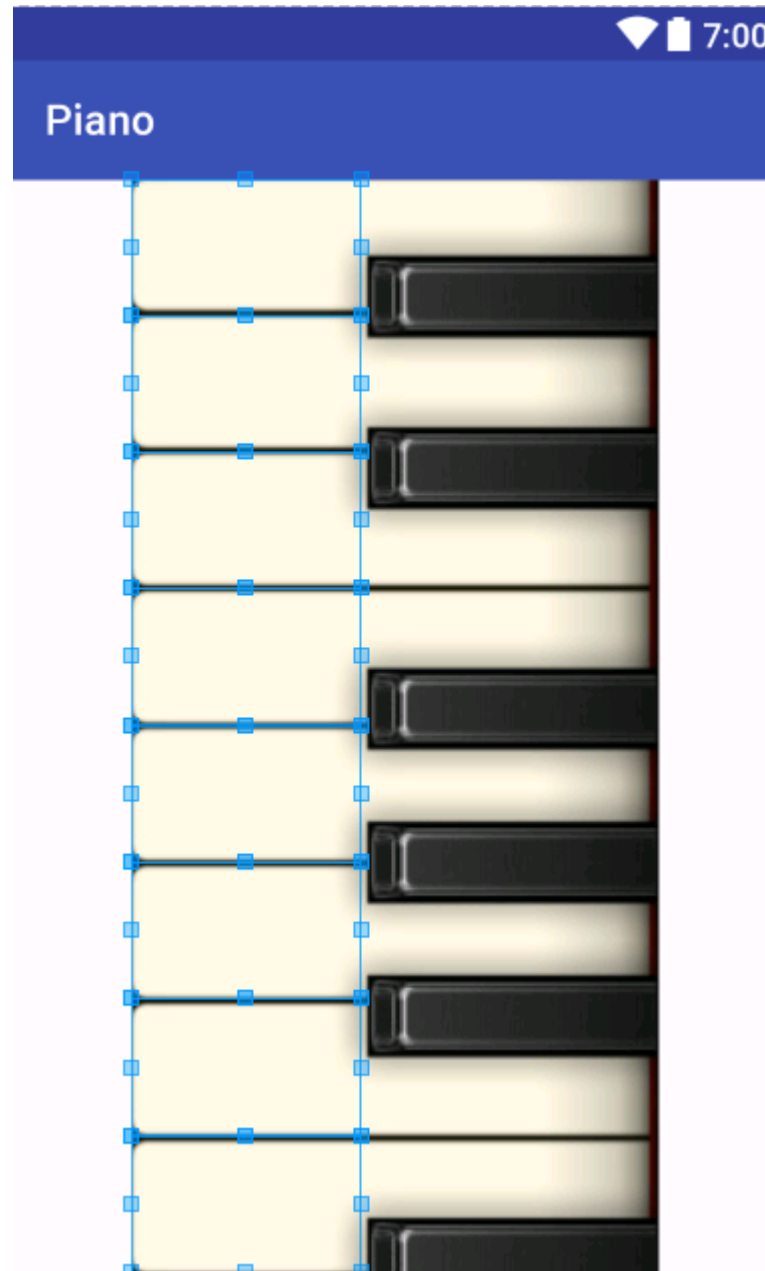
# Add keyboard *ImageView*

26



# *No text, image background (color\_btn)*

27



## *(Extra) Playing on touch down*

28

```
package com.mjkim.piano

import android.media.AudioManager
import android.media.SoundPool
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.view.MotionEvent
import android.view.View
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity(), View.OnTouchListener {

    var sp = SoundPool(5, AudioManager.STREAM_MUSIC, 0)
    var note = IntArray(8)
```

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    note[0] = sp.load(this, R.raw.note00, 1)  
    note[1] = sp.load(this, R.raw.note02, 1)  
    note[2] = sp.load(this, R.raw.note04, 1)  
    note[3] = sp.load(this, R.raw.note05, 1)  
    note[4] = sp.load(this, R.raw.note07, 1)  
    note[5] = sp.load(this, R.raw.note09, 1)  
    note[6] = sp.load(this, R.raw.note11, 1)  
    note[7] = sp.load(this, R.raw.note12, 1)  
  
    button54.setOnTouchListener(this)  
    button53.setOnTouchListener(this)  
    button52.setOnTouchListener(this)  
    button51.setOnTouchListener(this)  
    button50.setOnTouchListener(this)  
    button49.setOnTouchListener(this)  
    button48.setOnTouchListener(this)  
    button47.setOnTouchListener(this)  
}
```

```
override fun onTouch(view: View, event: MotionEvent): Boolean {  
    if(event.action == MotionEvent.ACTION_DOWN) {  
        if(view.id == R.id.button54) {  
            sp.play(note[0], 1.0f, 1.0f, 0, 0, 1.0f)  
        } else if(view.id == R.id.button53) {  
            sp.play(note[1], 1.0f, 1.0f, 0, 0, 1.0f)  
        } else if(view.id == R.id.button52) {  
            sp.play(note[2], 1.0f, 1.0f, 0, 0, 1.0f)  
        } else if(view.id == R.id.button51) {  
            sp.play(note[3], 1.0f, 1.0f, 0, 0, 1.0f)  
        } else if(view.id == R.id.button50) {  
            sp.play(note[4], 1.0f, 1.0f, 0, 0, 1.0f)  
        } else if(view.id == R.id.button49) {  
            sp.play(note[5], 1.0f, 1.0f, 0, 0, 1.0f)  
        } else if(view.id == R.id.button48) {  
            sp.play(note[6], 1.0f, 1.0f, 0, 0, 1.0f)  
        } else if(view.id == R.id.button47) {  
            sp.play(note[7], 1.0f, 1.0f, 0, 0, 1.0f)  
        }  
    }  
    return false  
}
```