



# Android

Author : Do Phu Quy

Email : [phuquycntt@gmail.com](mailto:phuquycntt@gmail.com)

Phone : 0935 366 007

# ConstraintLayout

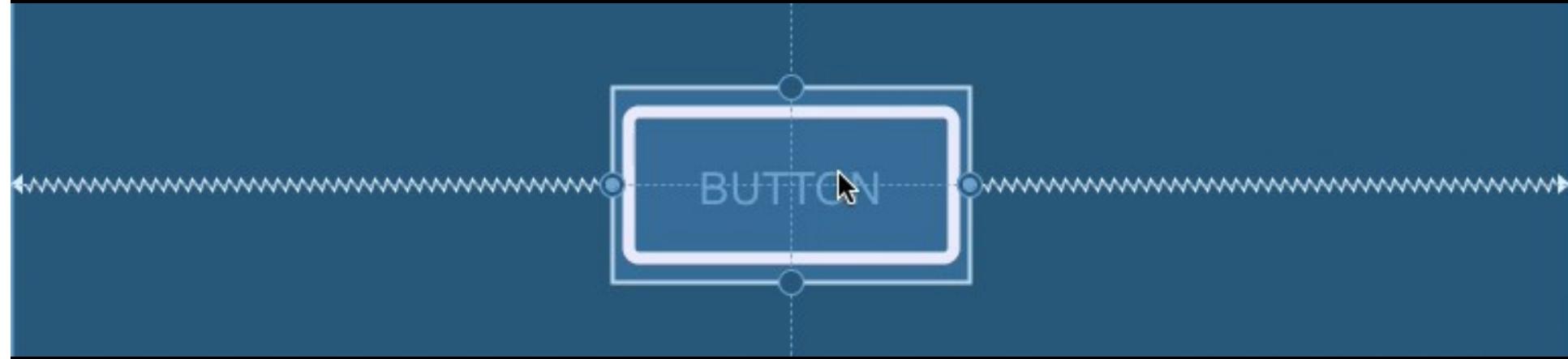
# What is it?

- Flexible Layout system
- Encourage flat hierarchies
- Tool support
- Compatible API 9 (99.99% of devices)

# Positioning



# Centering



# Dimensions

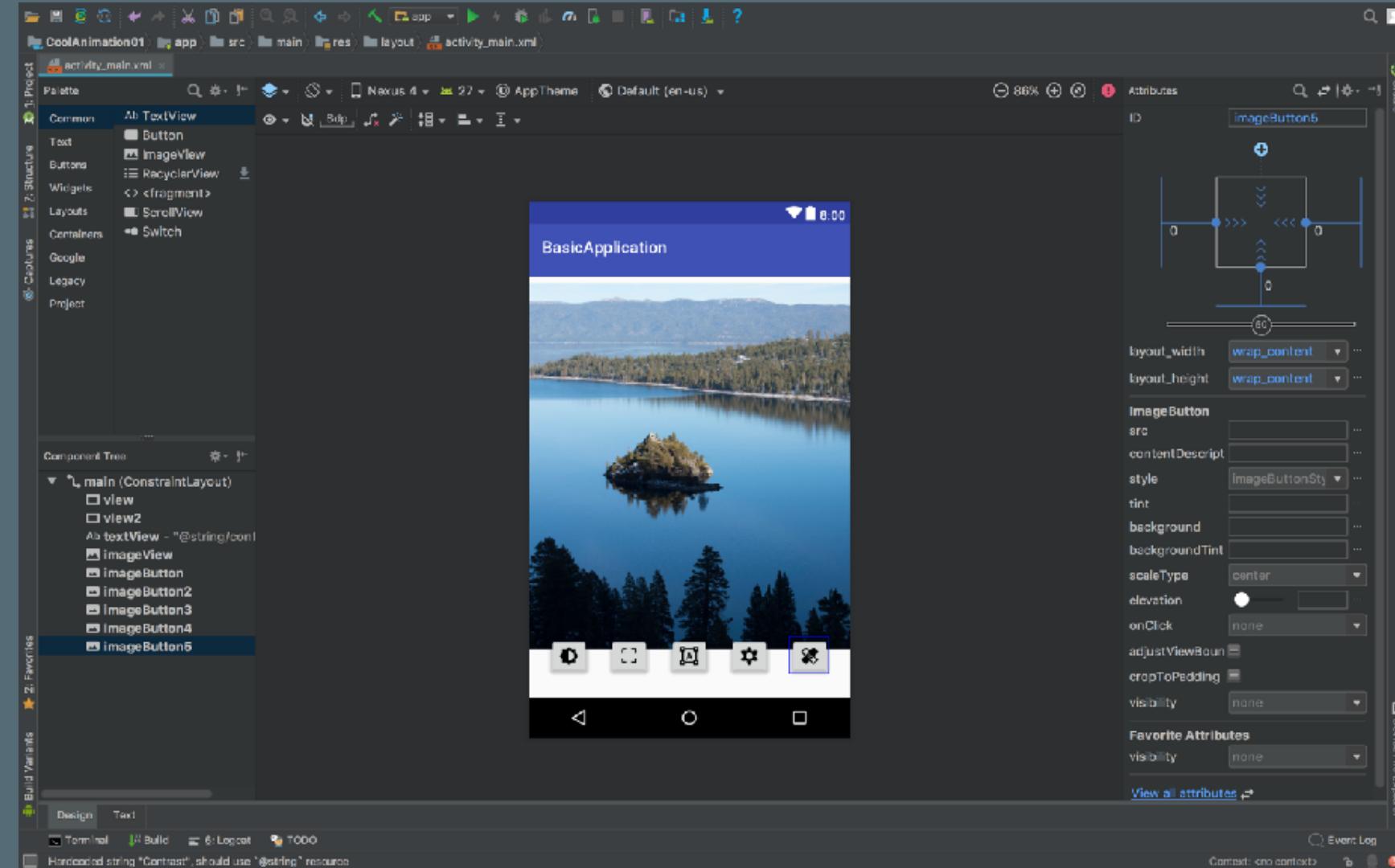
# Dimension

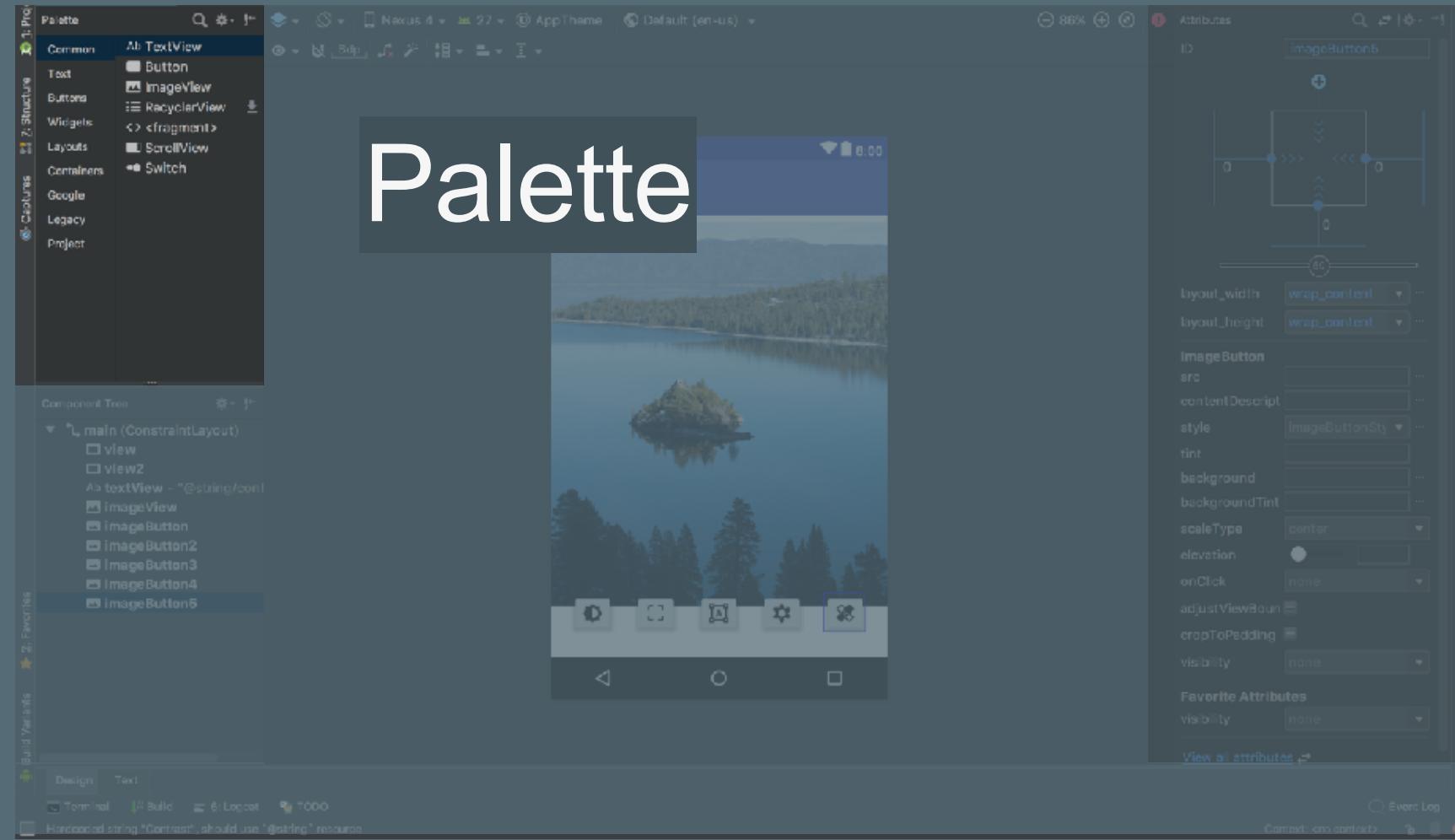
- Fixed
- wrap\_content
- Match Constraints (0dp)
  - takes available space
  - supports min/max/percent
  - ratio
- Match Parent

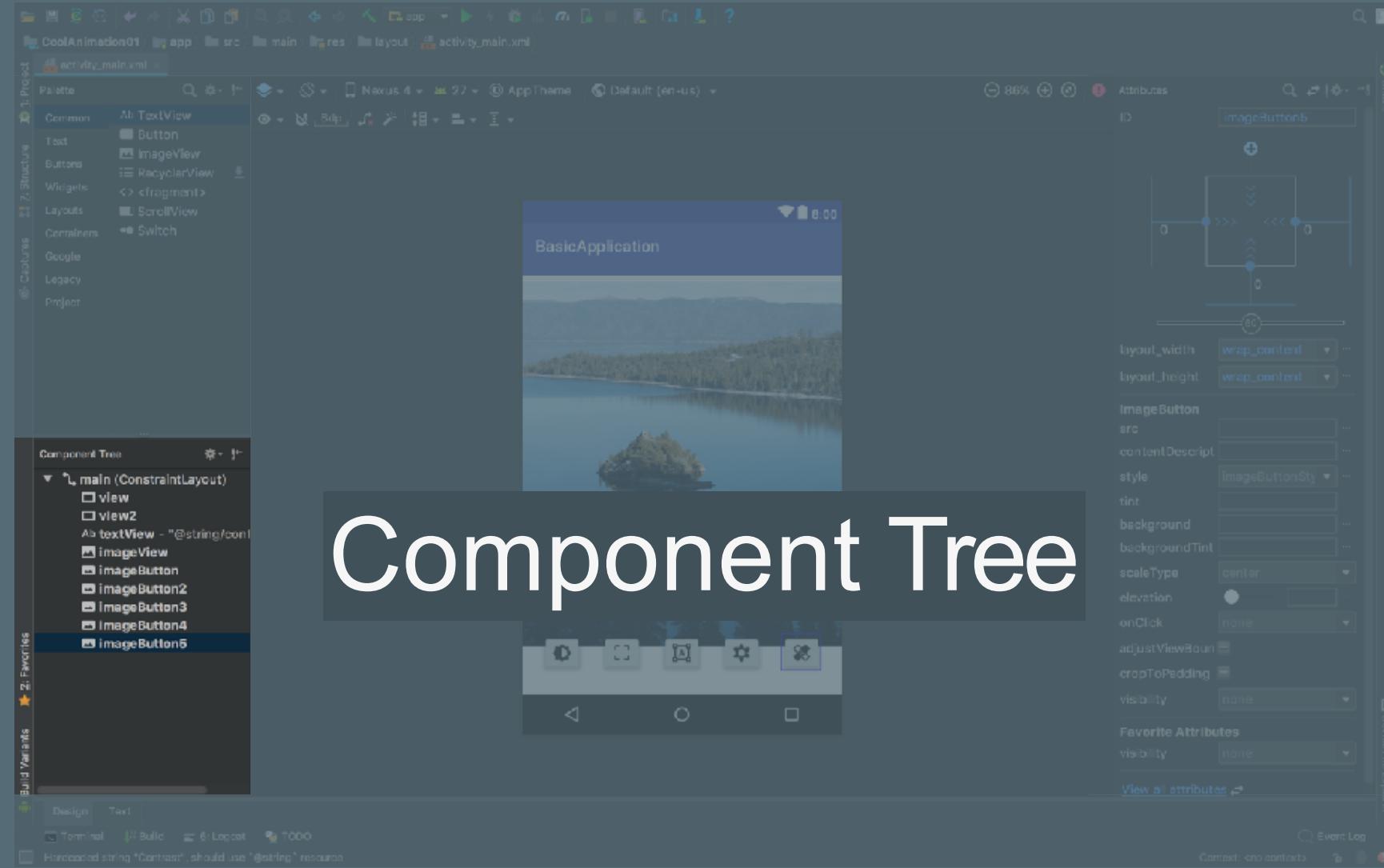
# Tooling



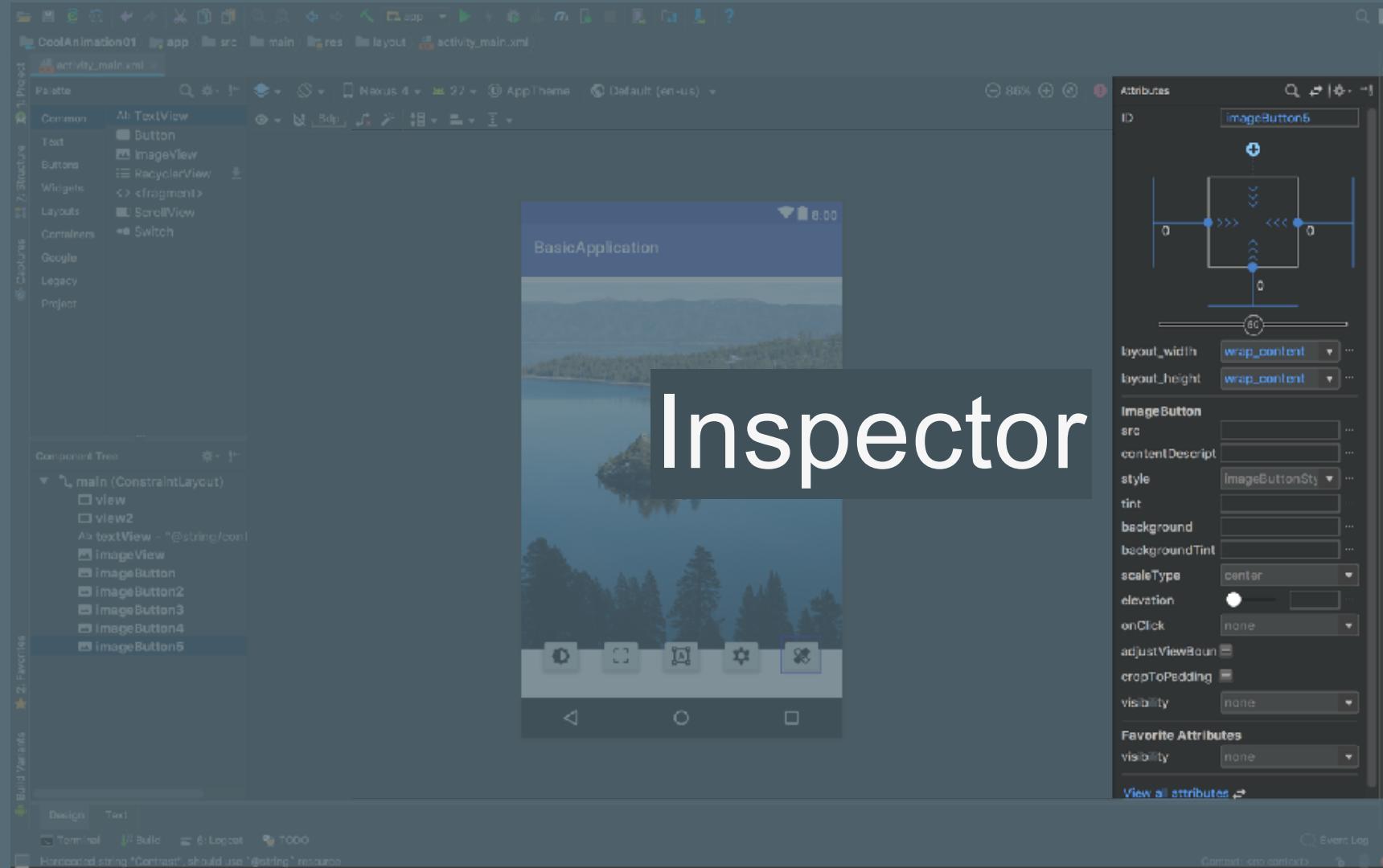
# Android Studio



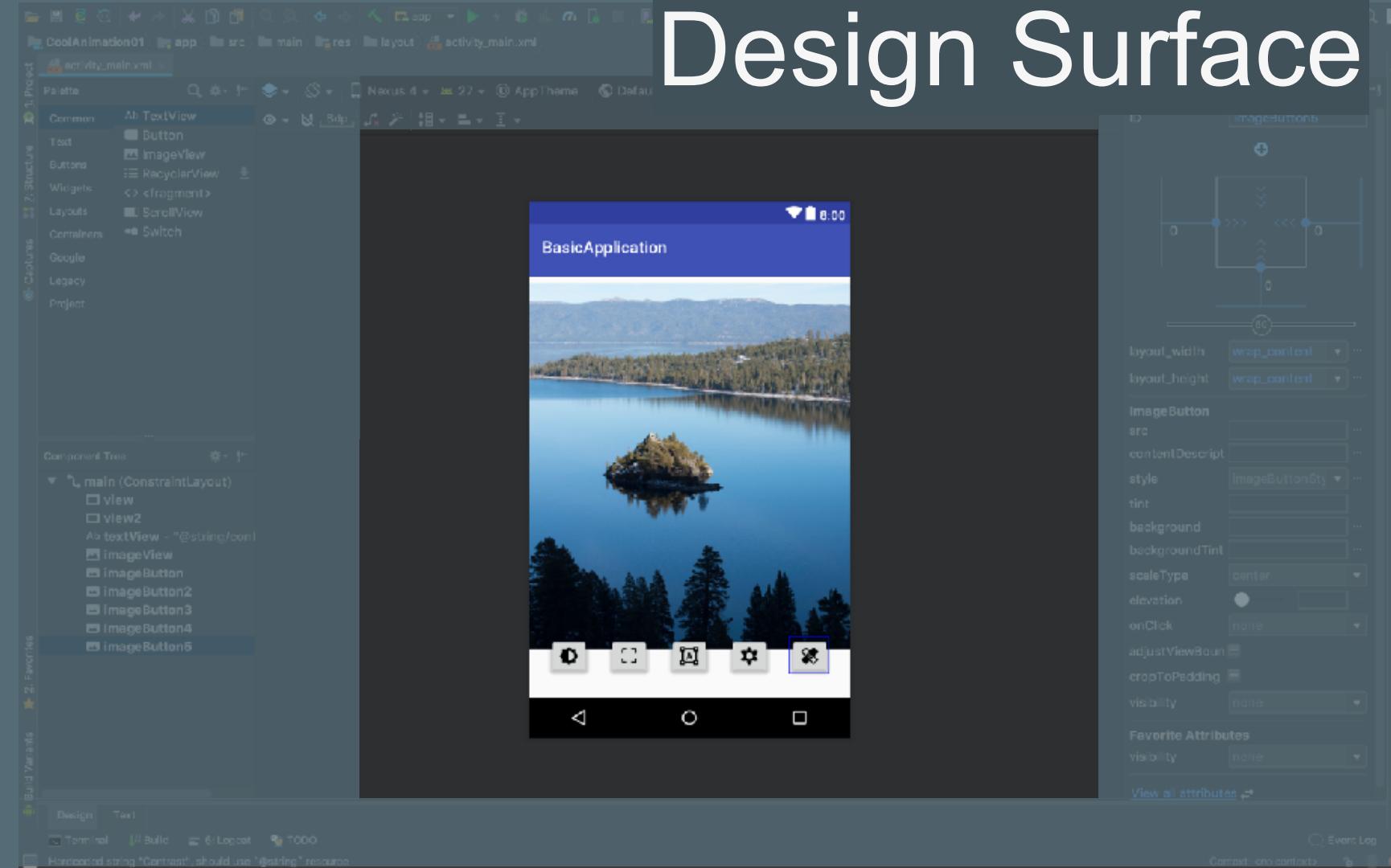


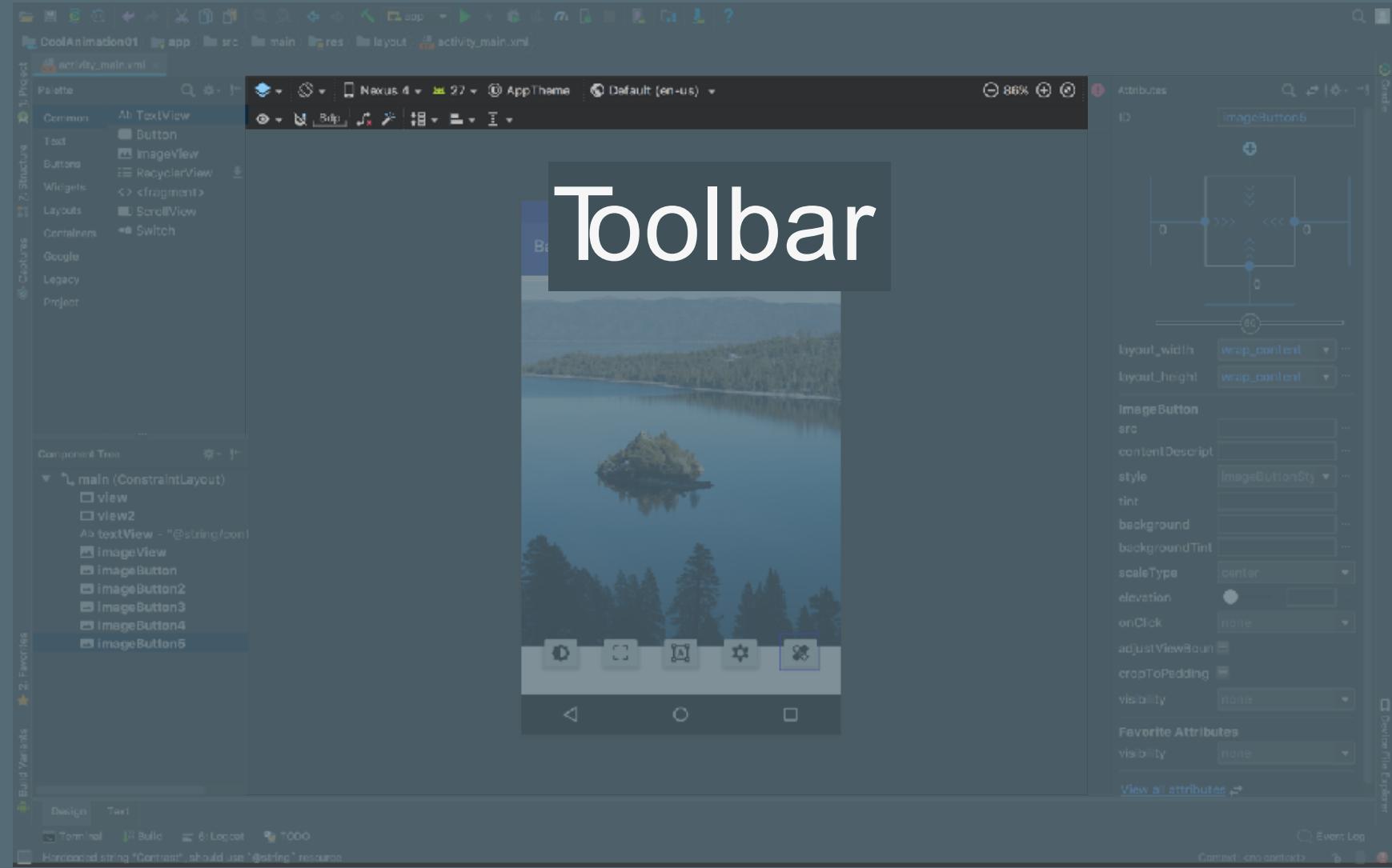


# Inspector

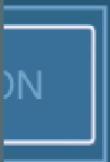
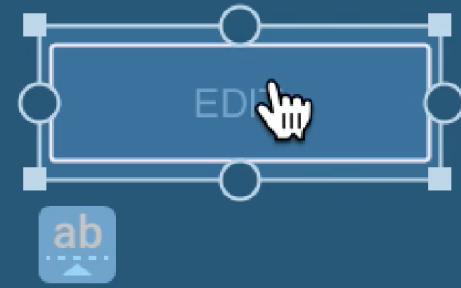


# Design Surface

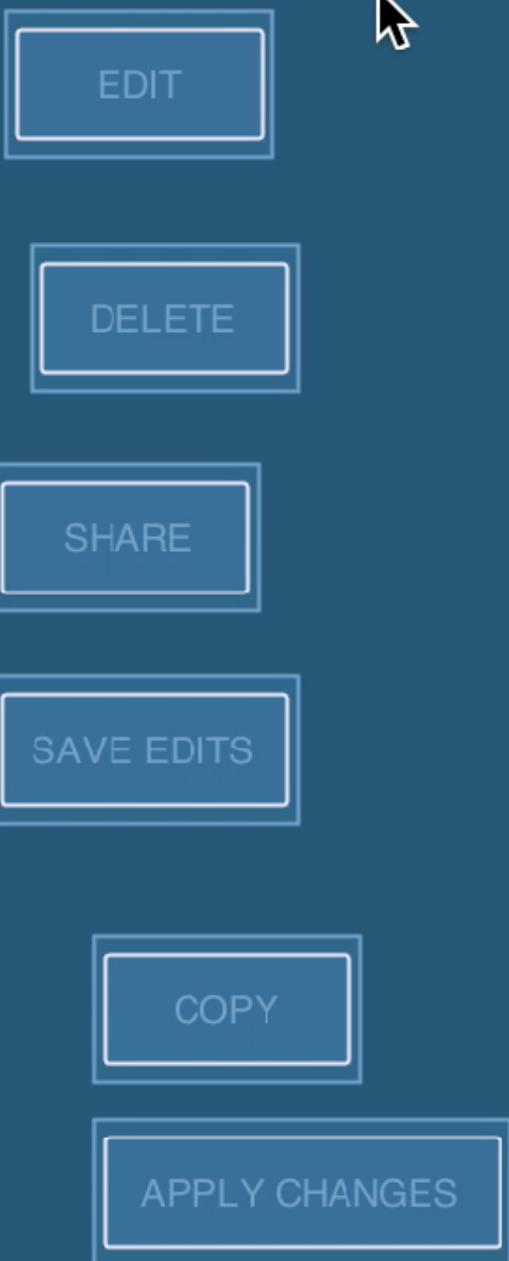




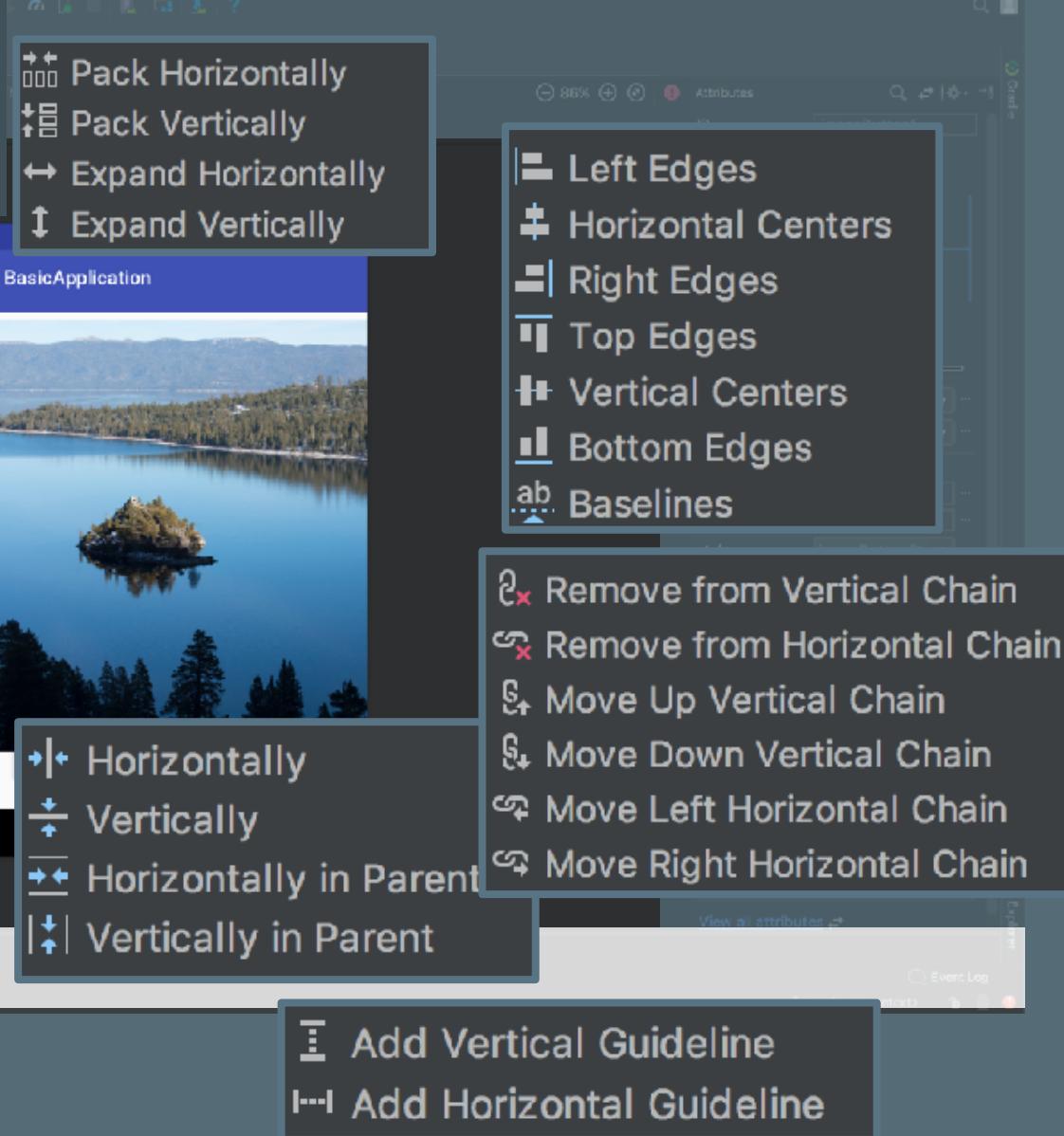
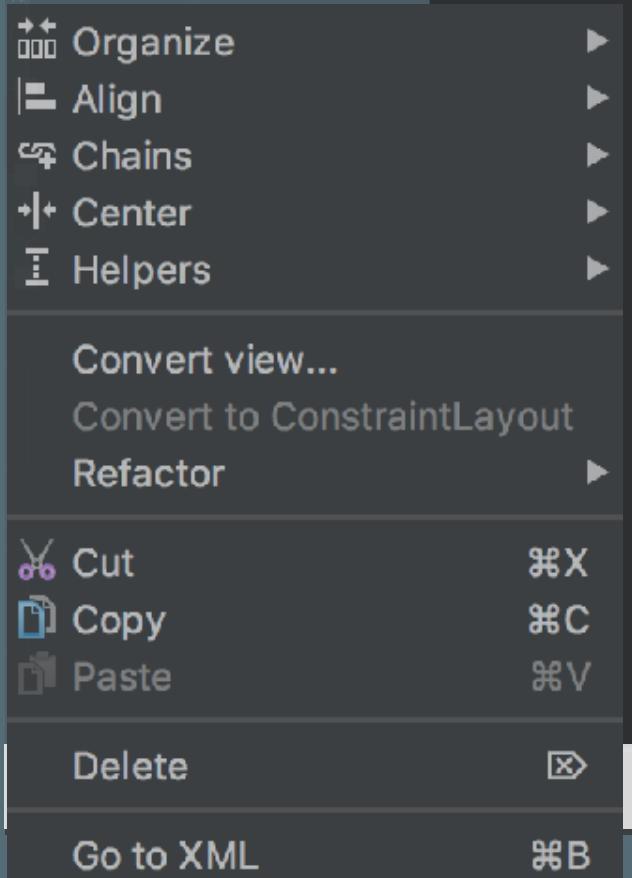
# Simple Operations



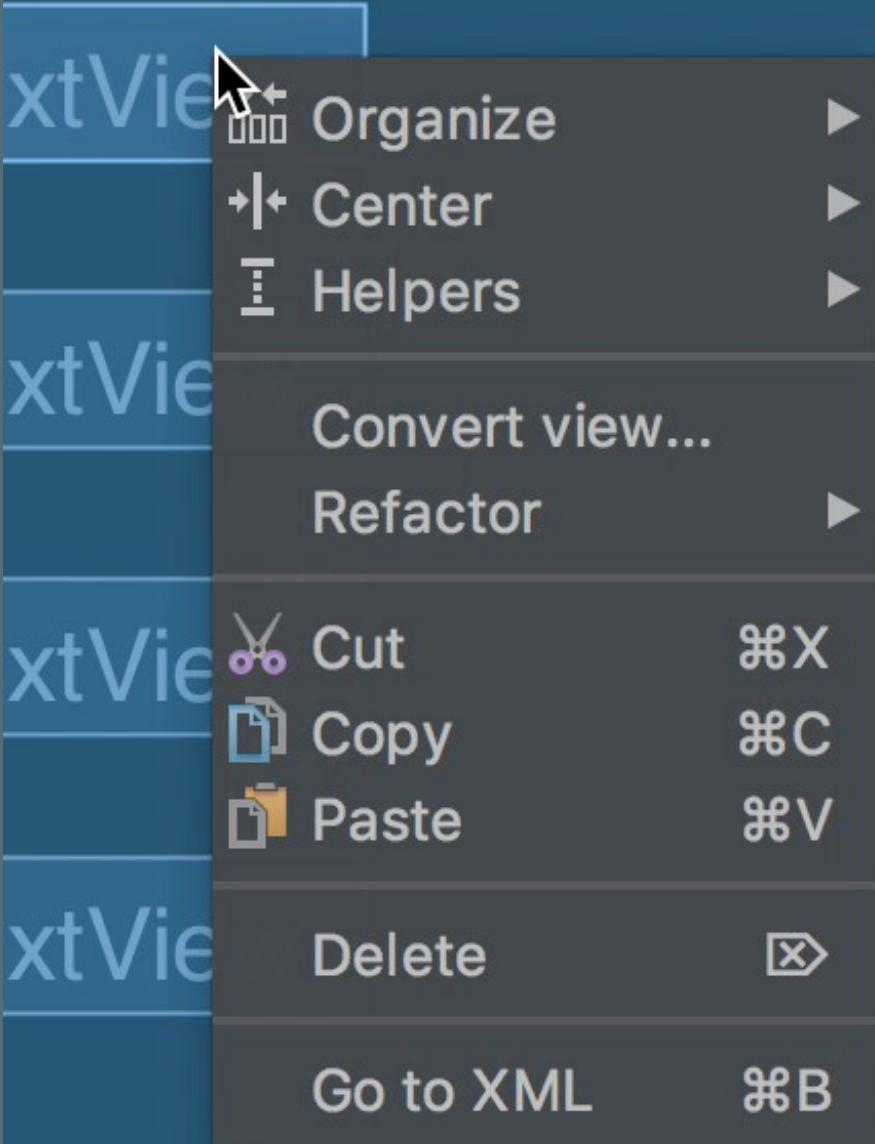
# Group Operations



# Right-click Menu

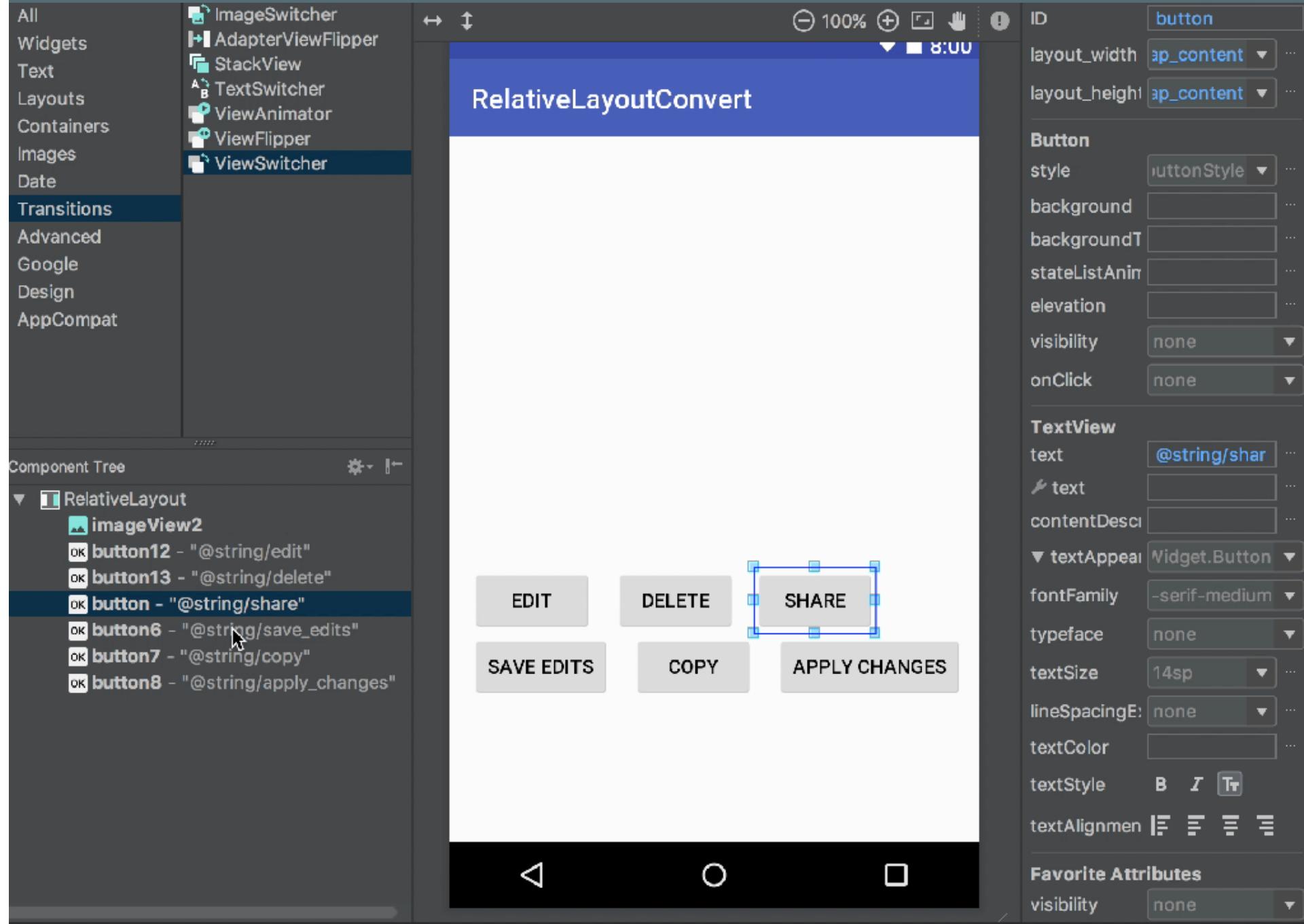


# Context Sensitive Menu



# Inference





# Performances

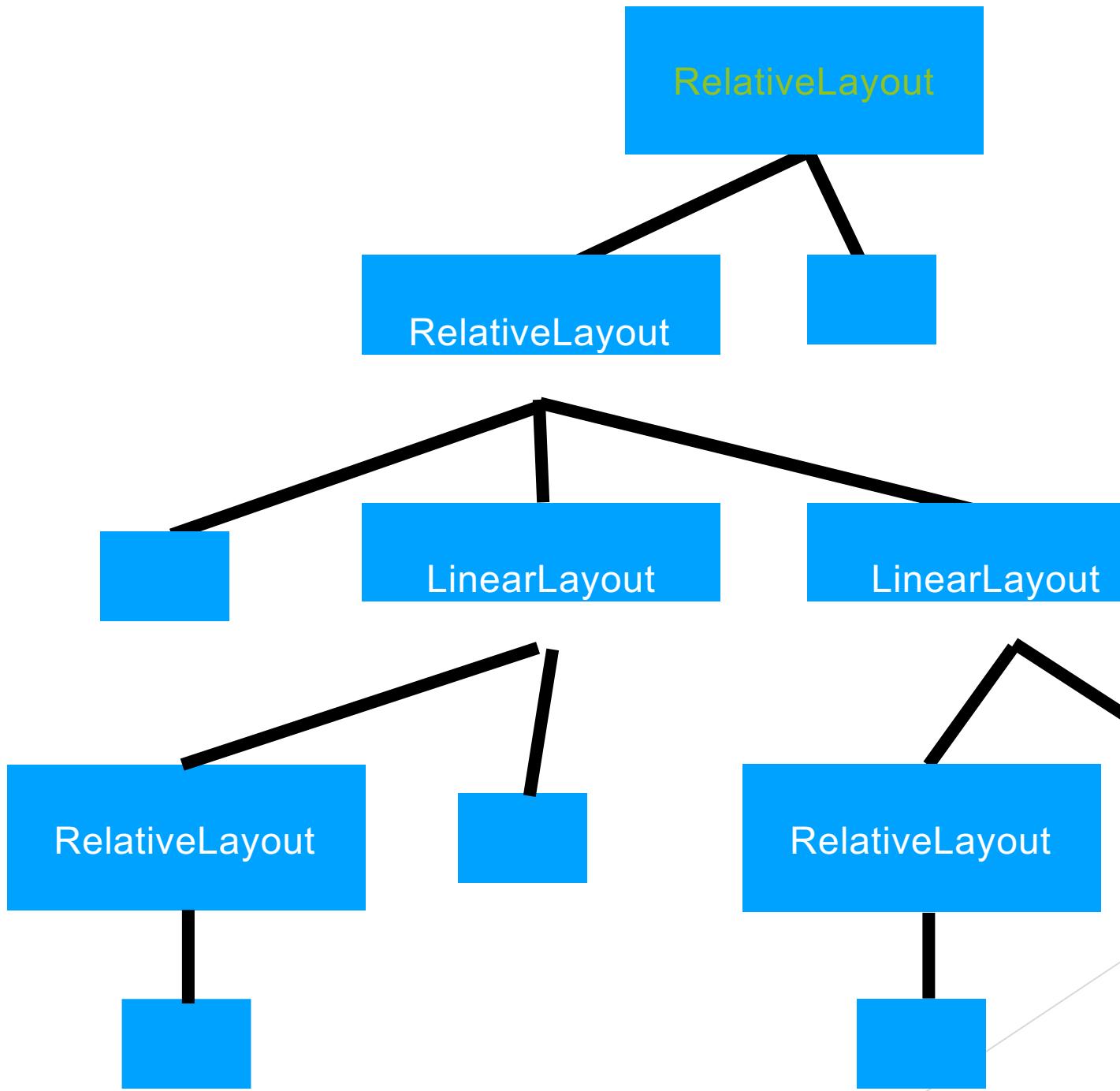


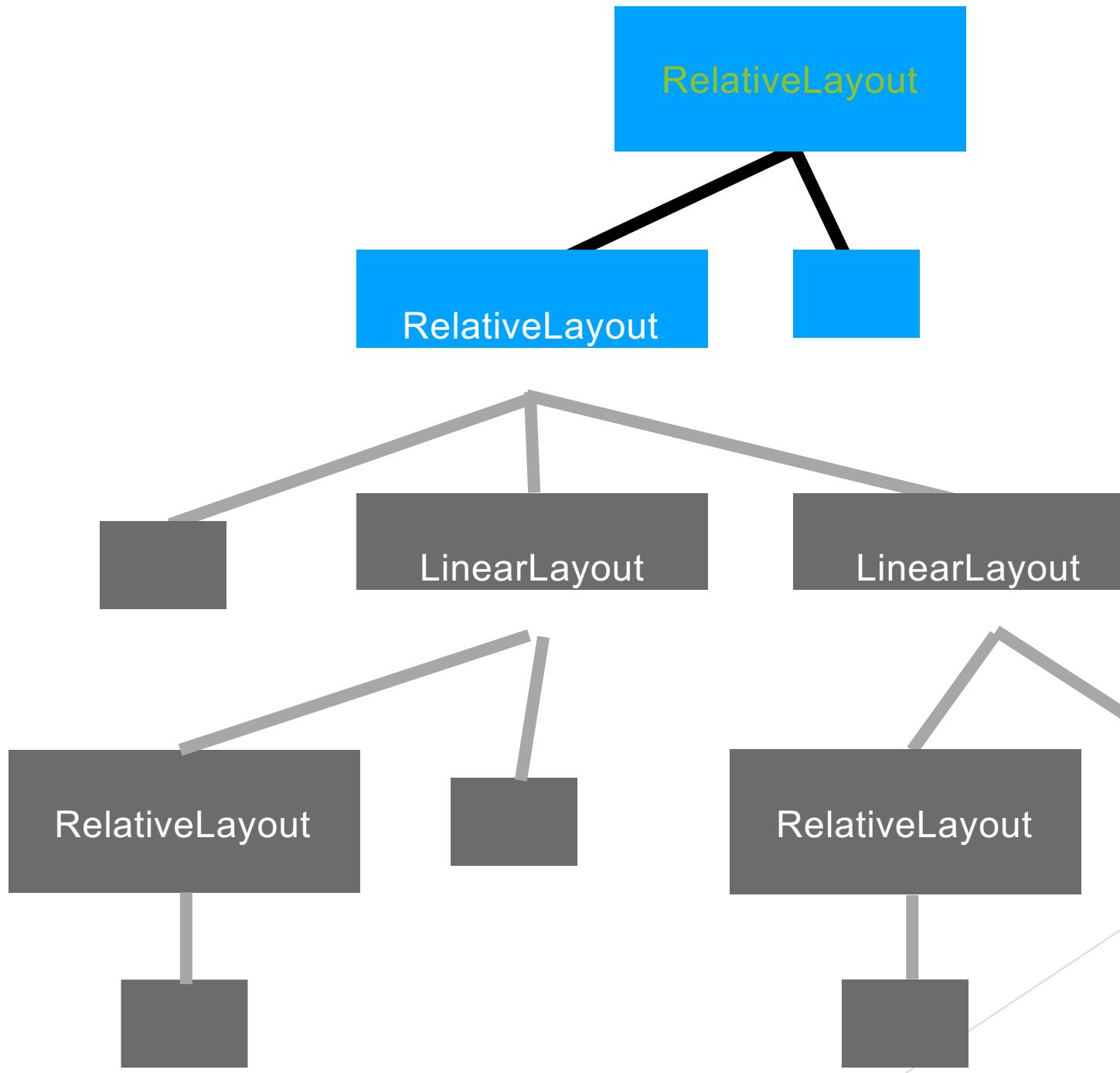
# A ViewGroup does

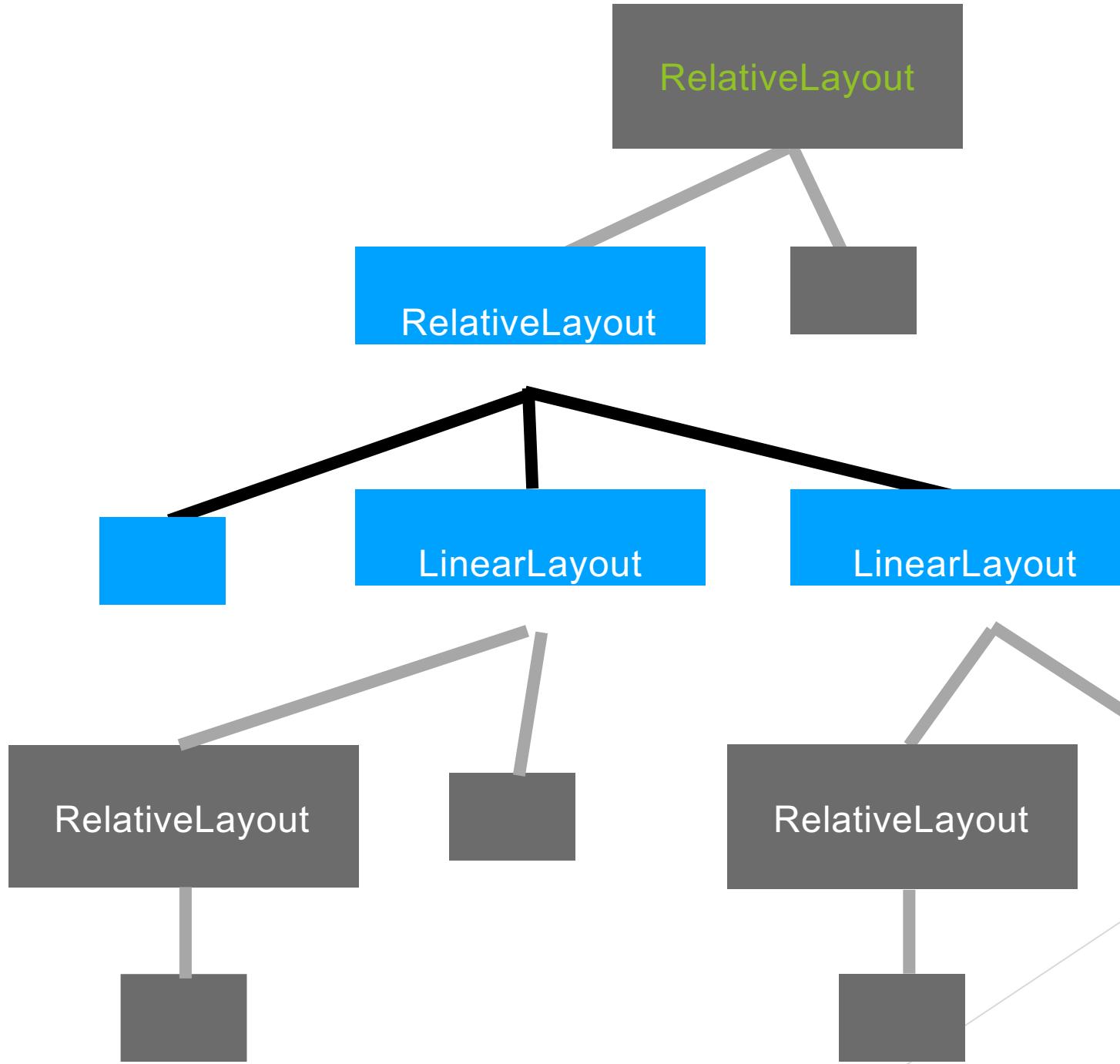
Measure

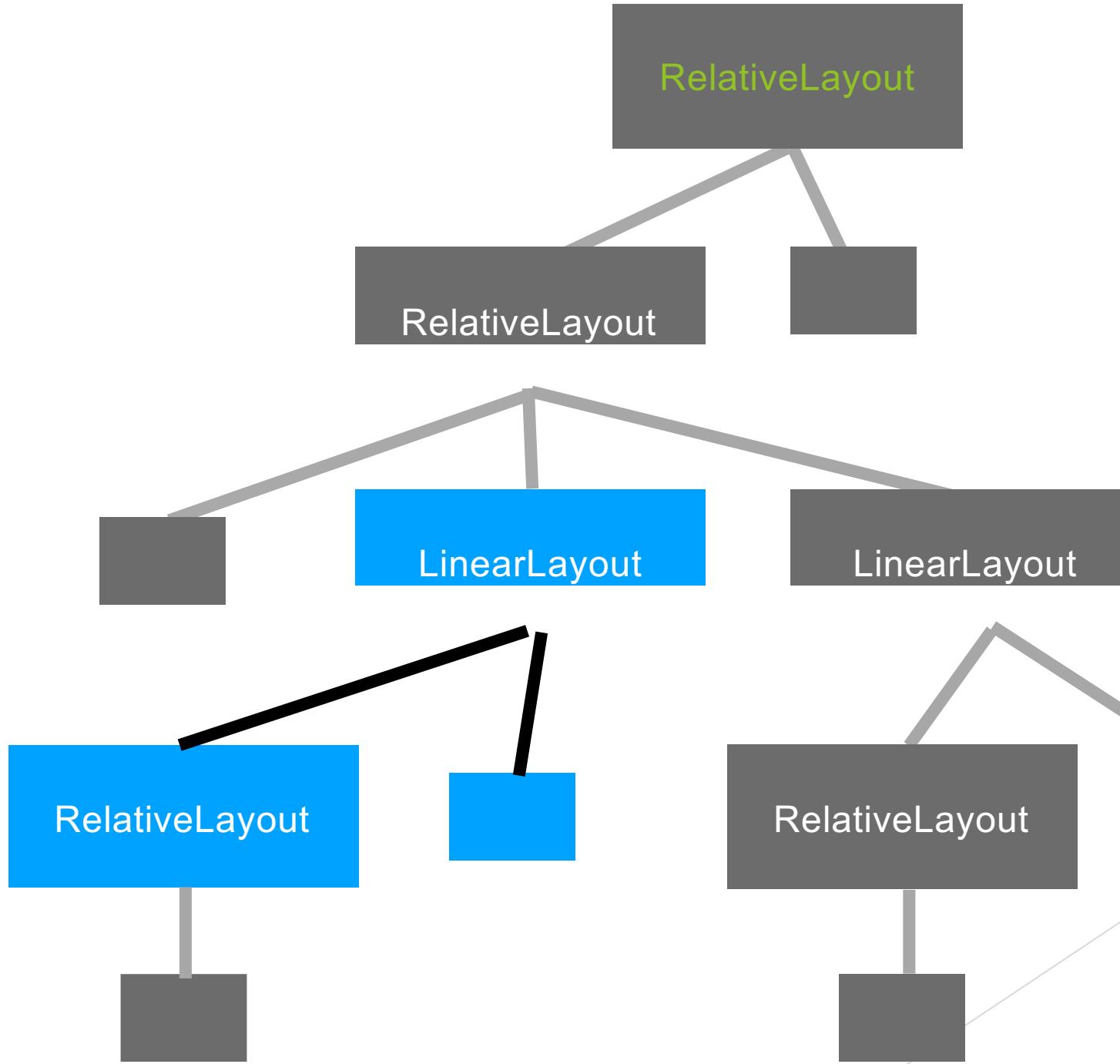
Layout

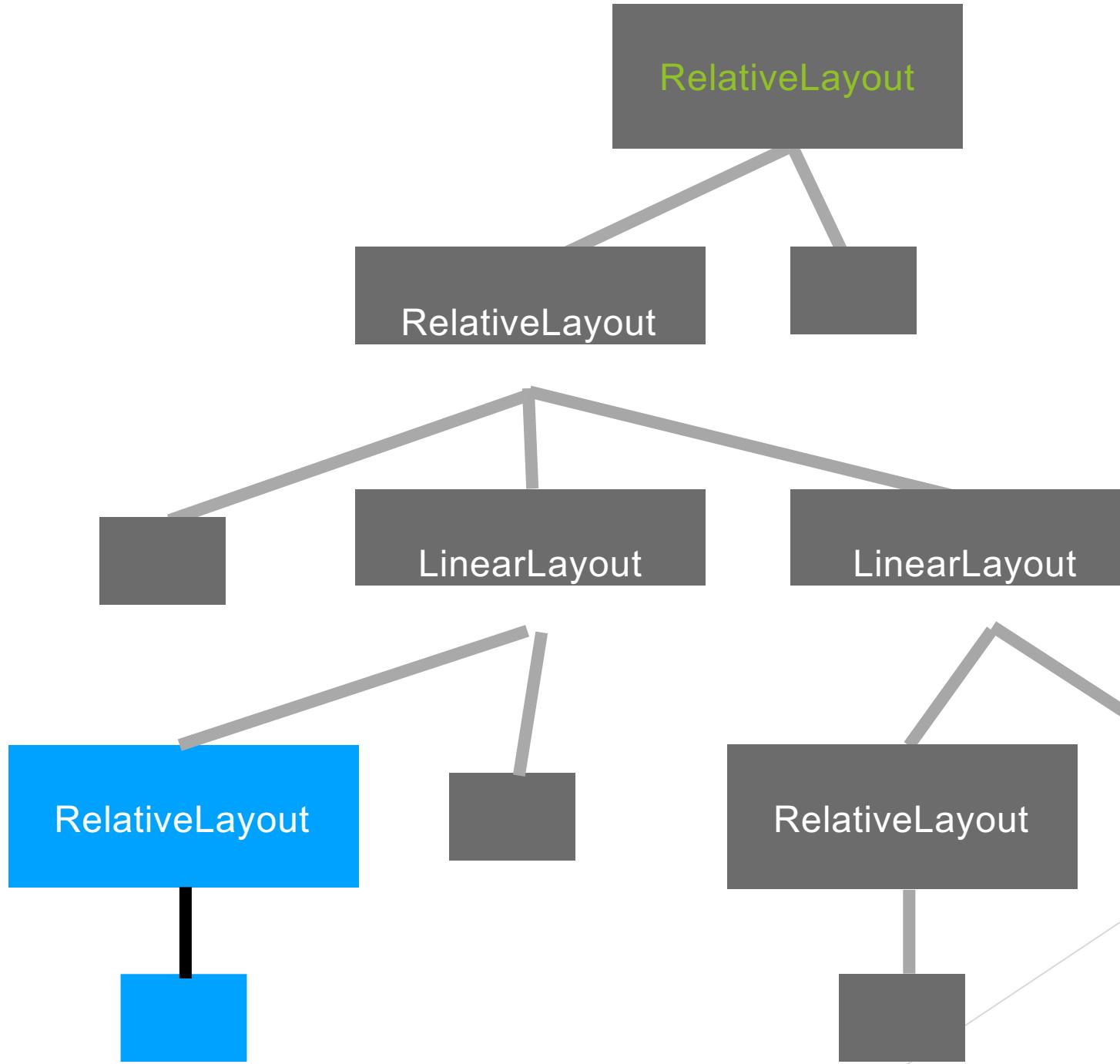
Draw

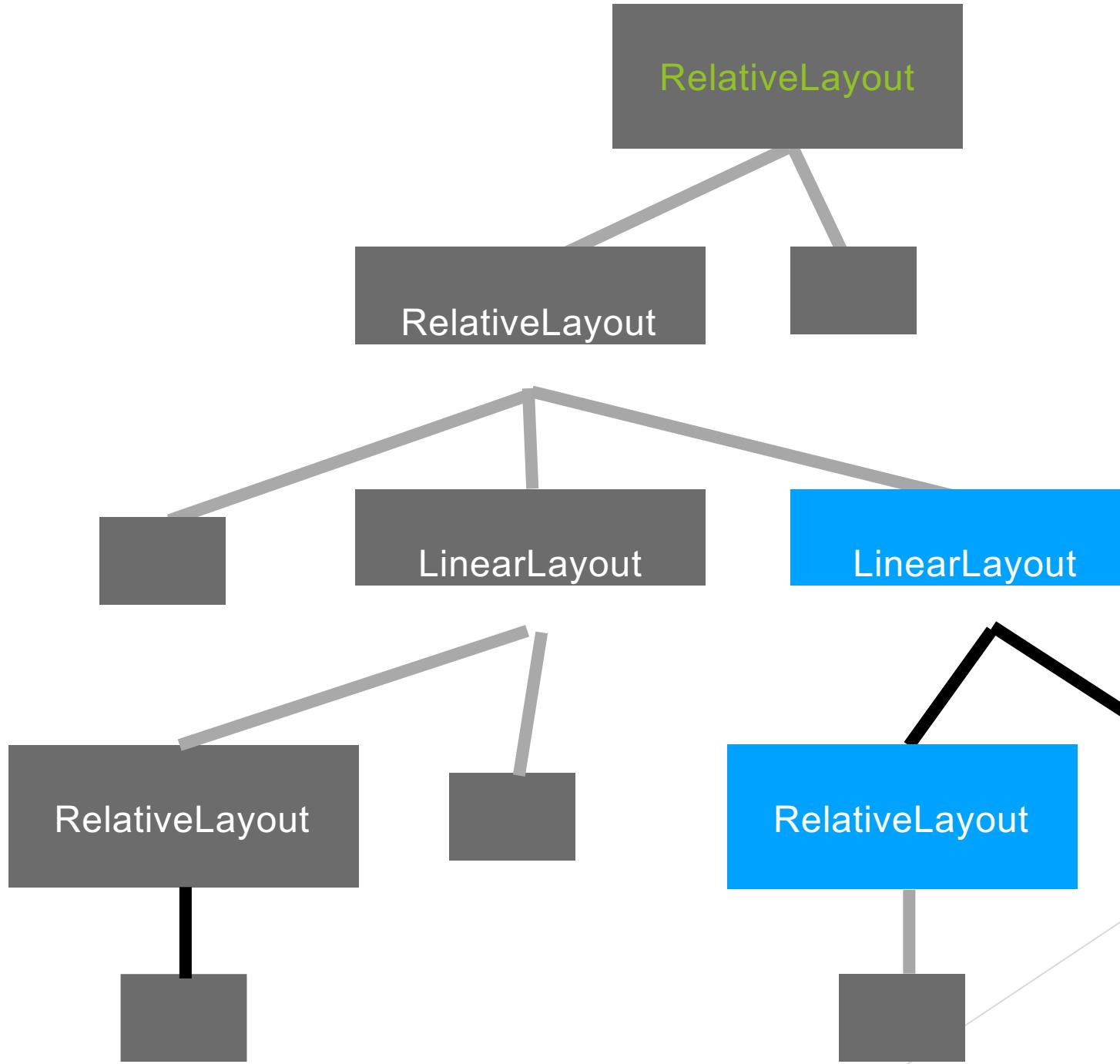


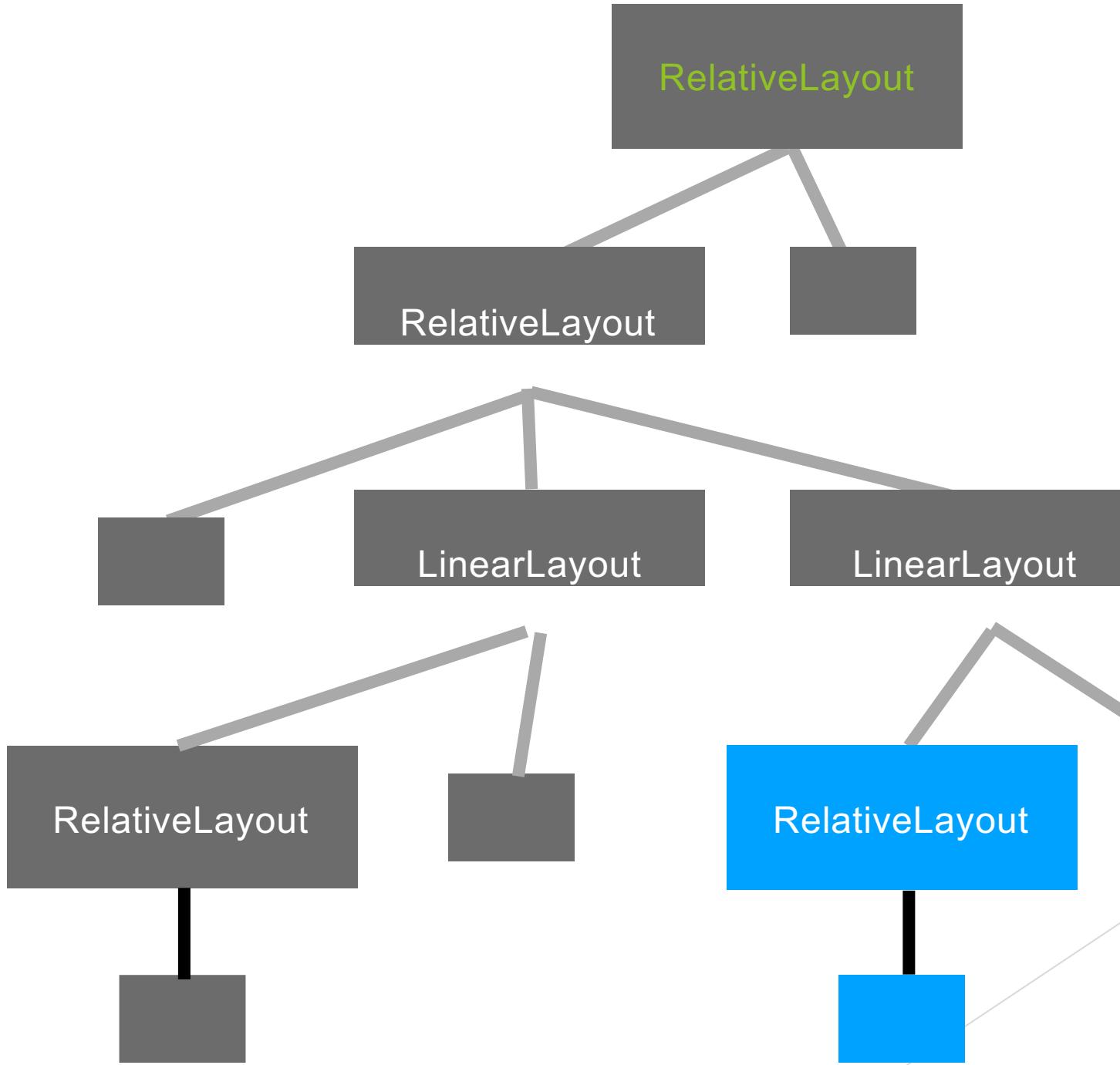












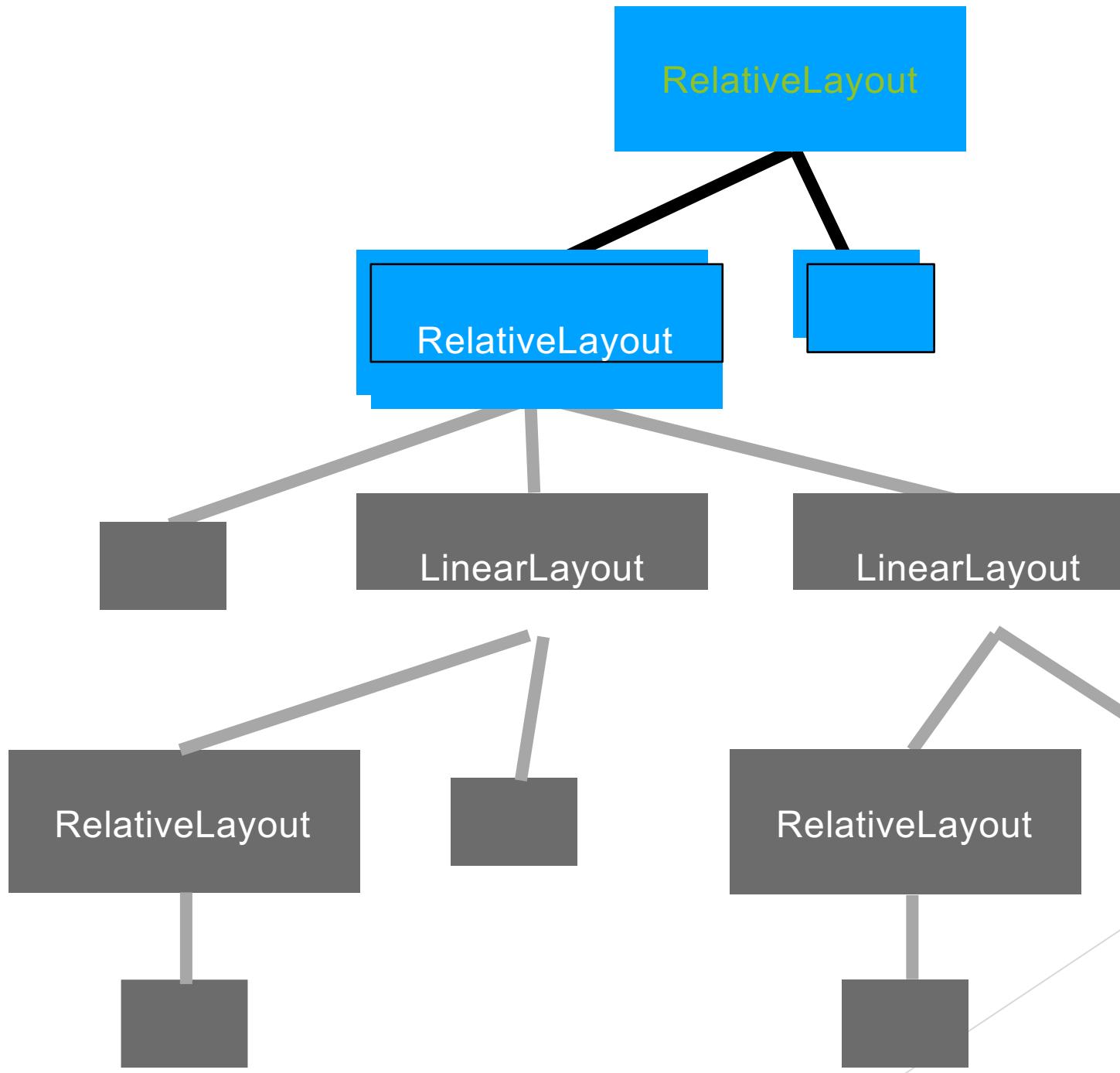
Besides

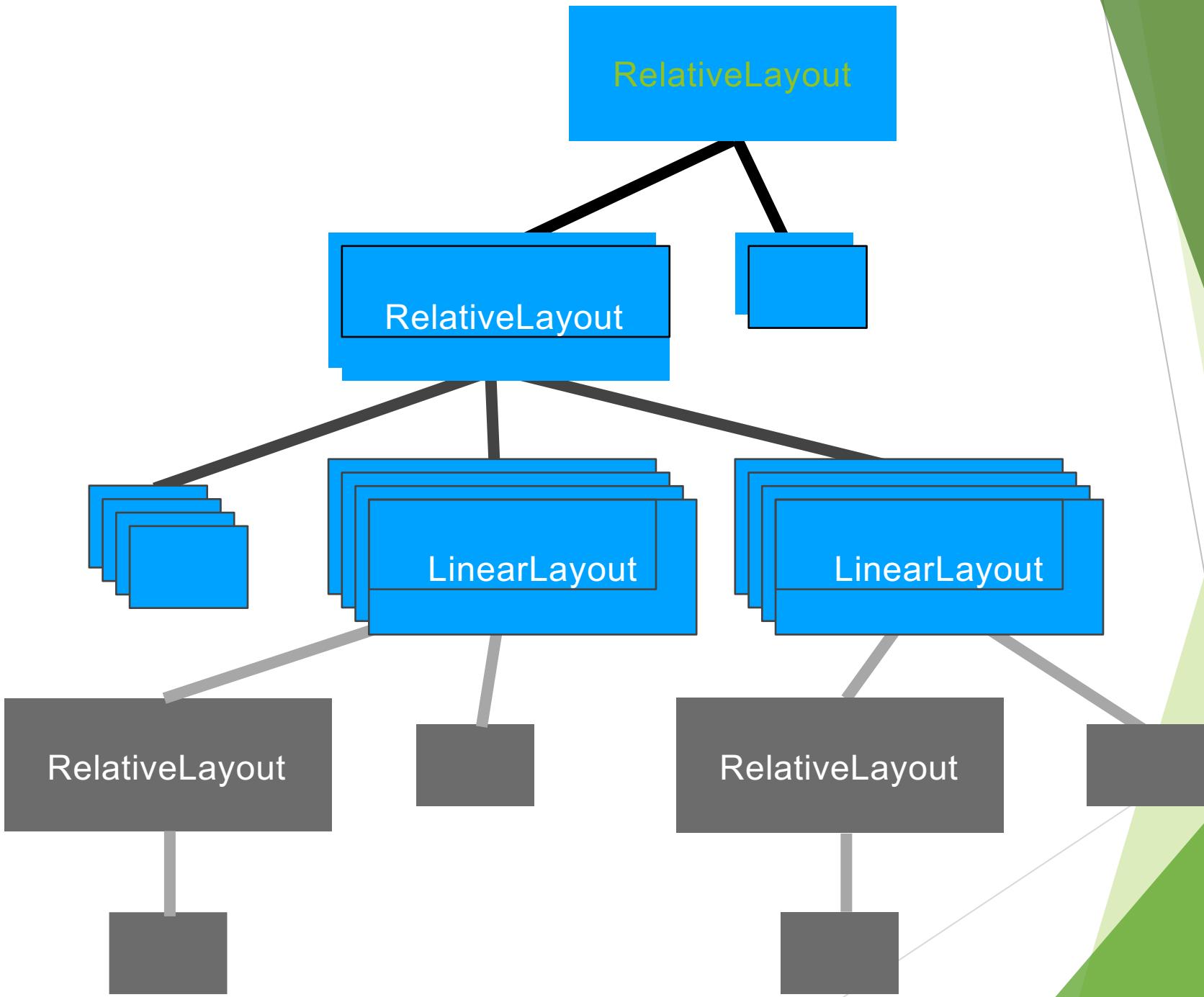
## **RelativeLayout**

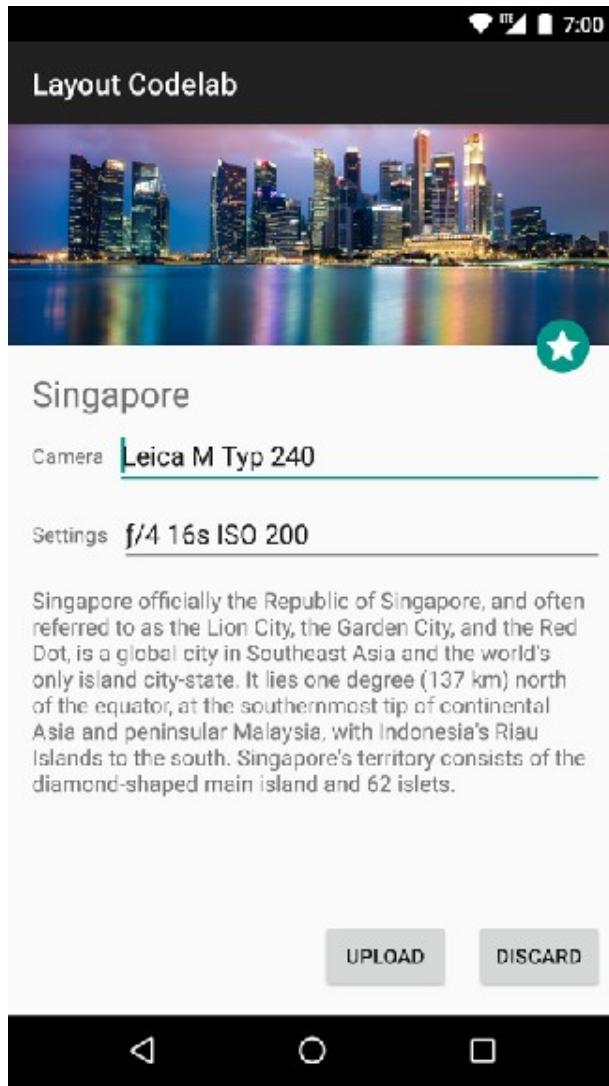
- Measures children twice

## **LinearLayout**

- Measures children twice if weight used



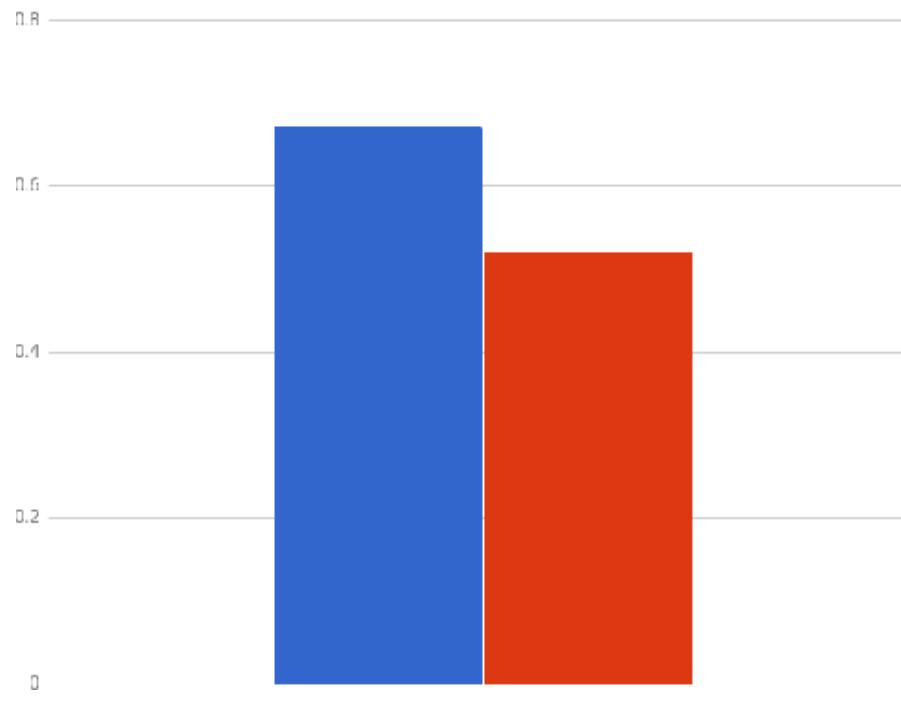




```
<RelativeLayout>
    <ImageView />
    <ImageView />
    <RelativeLayout>
        <TextView />
        <LinearLayout >
            <TextView />
        </LinearLayout>
        <RelativeLayout >
            <EditText />
        </RelativeLayout>
    </LinearLayout>
    <LinearLayout >
        <TextView />
        <RelativeLayout >
            <EditText />
        </RelativeLayout>
    </LinearLayout>
    <Textview />
</RelativeLayout>
<LinearLayout >
    <Button />
    <Button />
</LinearLayout>
</RelativeLayout>
```



```
<android.support.constraint.ConstraintLayout>
    <ImageView />
    <ImageView />
    <TextView />
    <EditText />
    <TextView />
    <TextView />
    <EditText />
    <Button />
    <Button />
    <TextView />
</android.support.constraint.ConstraintLayout>
```



CL 1.0.2,  
Nexus 5X

<https://android-developers.googleblog.com/2017/08/understanding-performance-benefits-of.html>

# More and More Features...



# 1.0

Relative positioning

Center positioning & bias

Gone

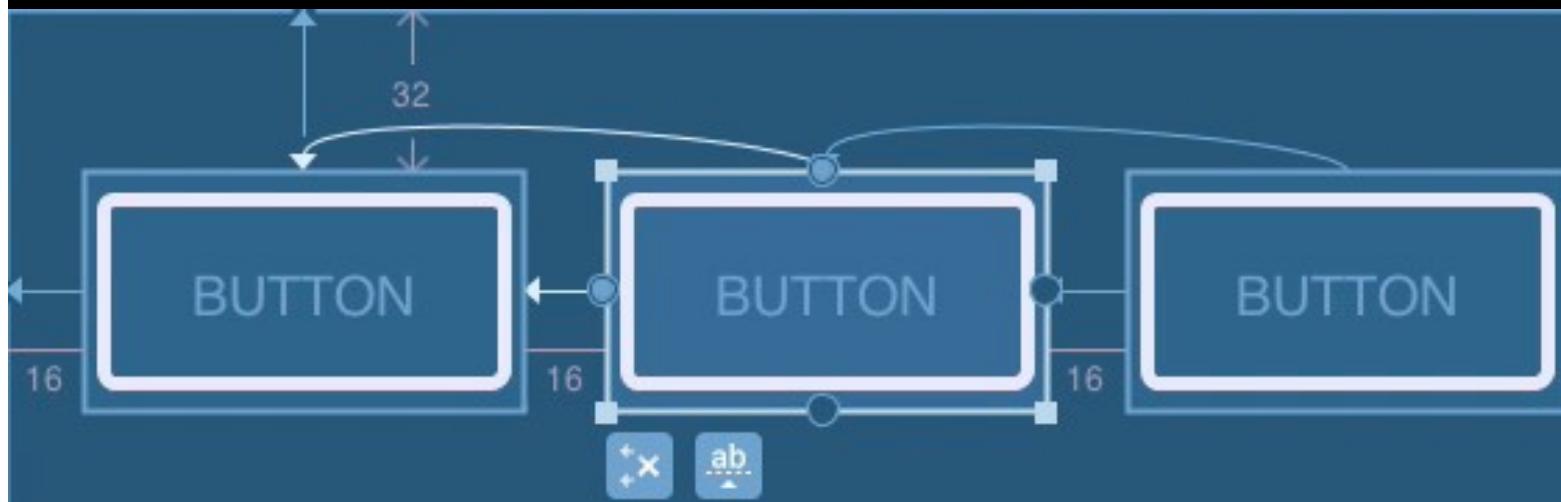
Guidelines - helper objects

Chains

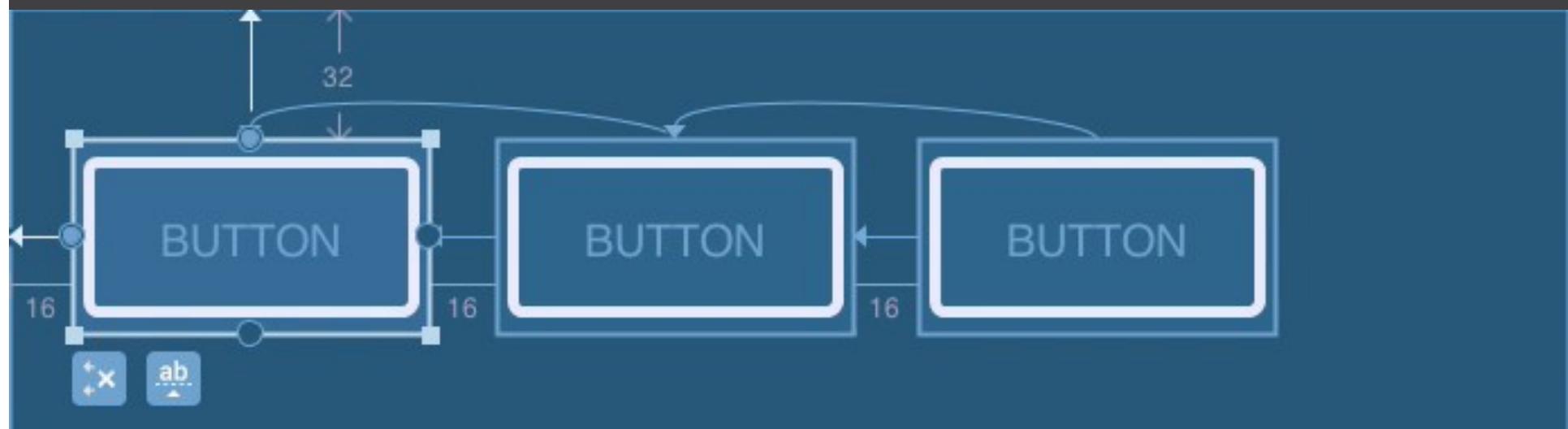
Dimension constraints: min/max, Ratio

ConstraintSet

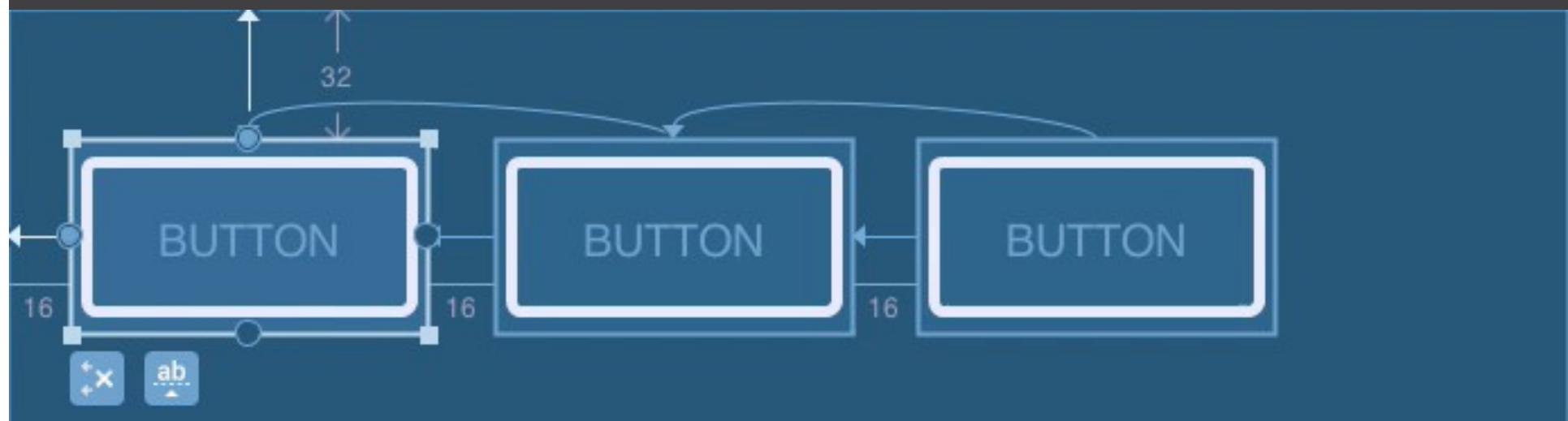
# Gone Behavior



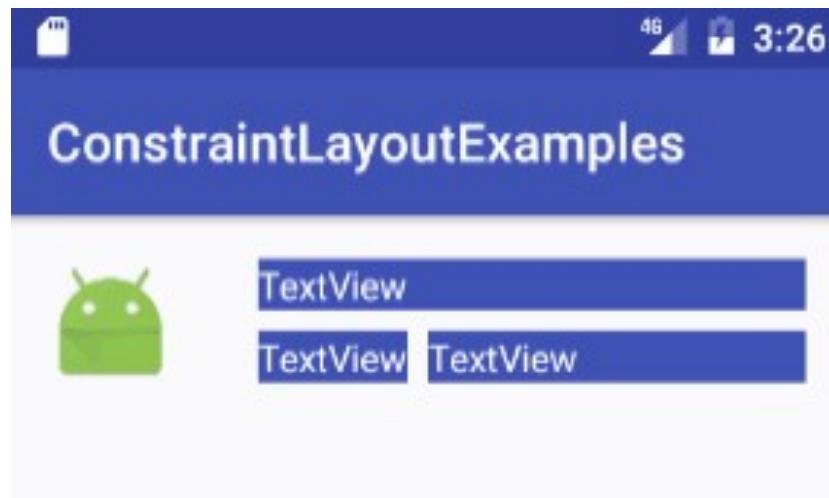
# Inadapted Margin



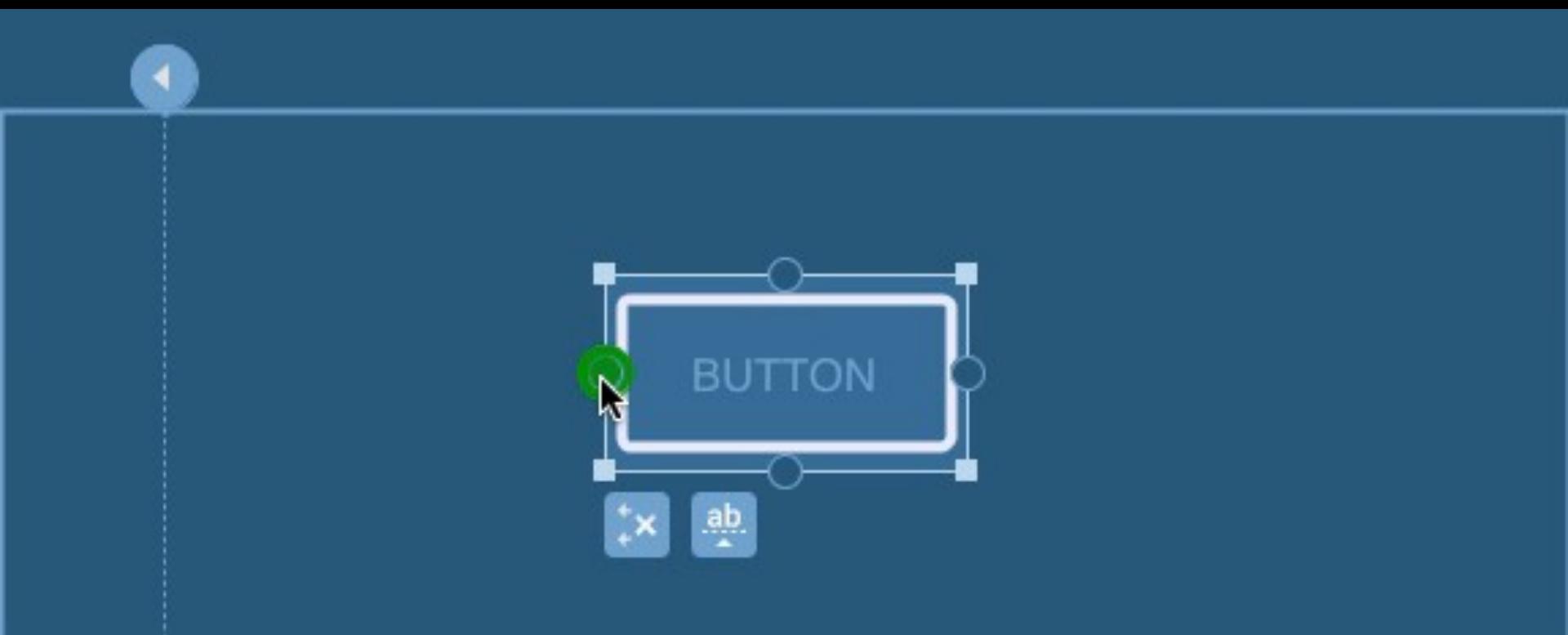
# Gone Margins

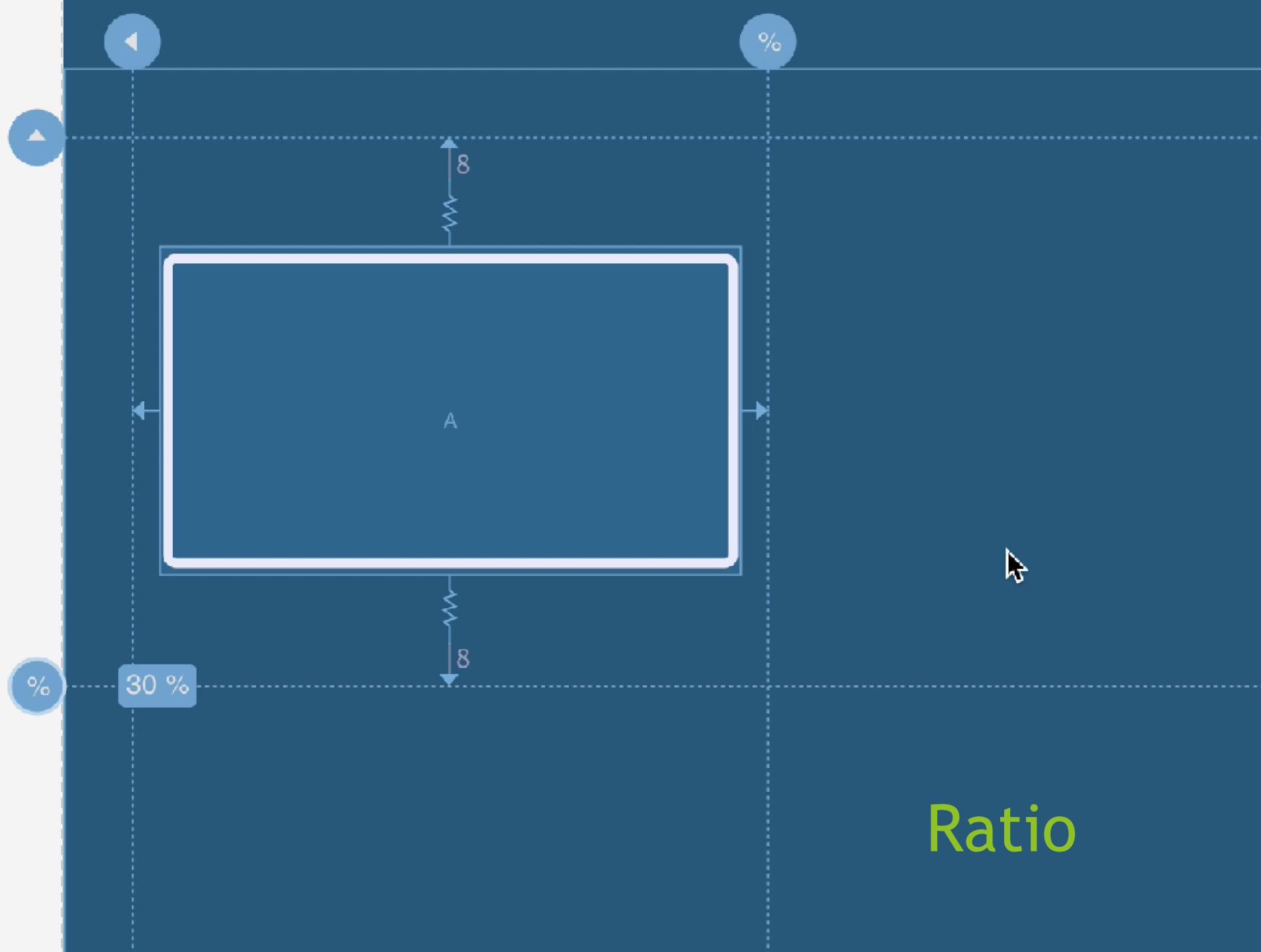


# Example



# Guidelines

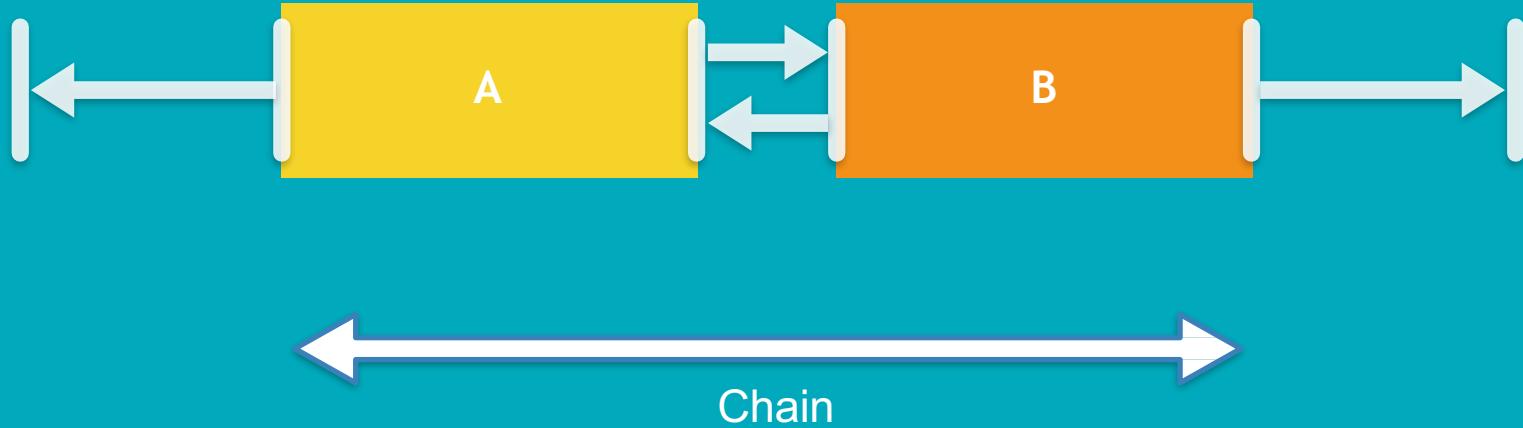




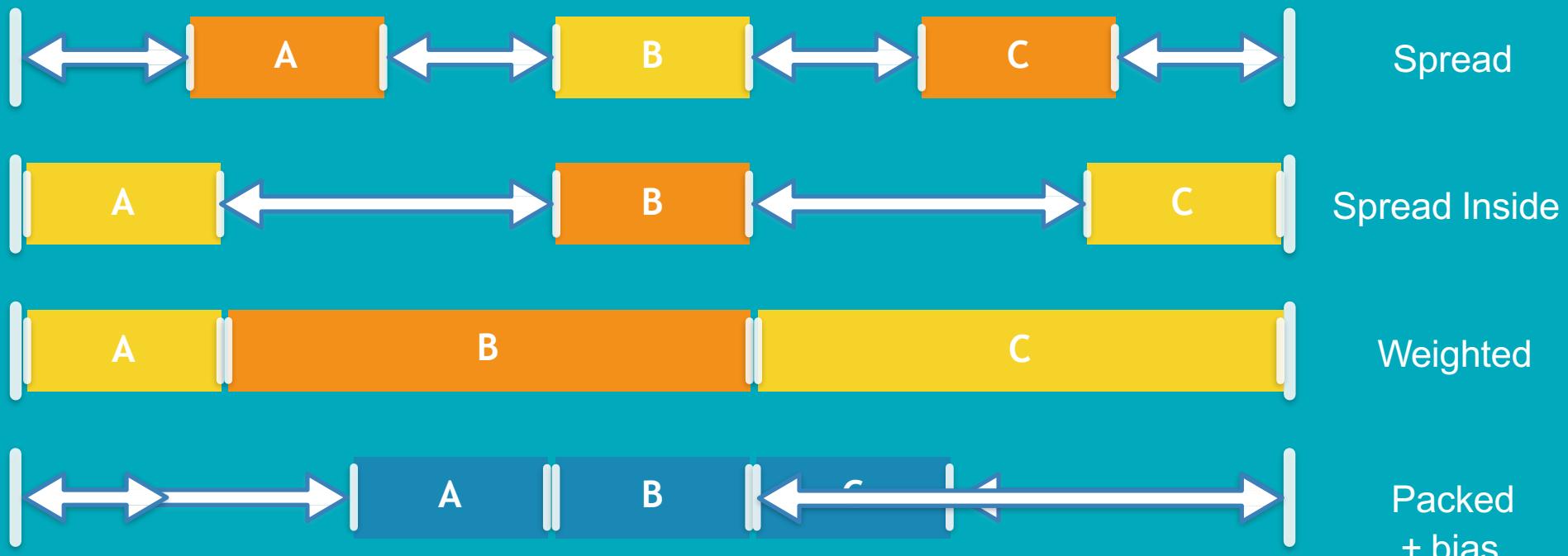
Ratio

# Chains

bi-directional constraints



# Different Chain Styles



# 1.1.0

Barriers

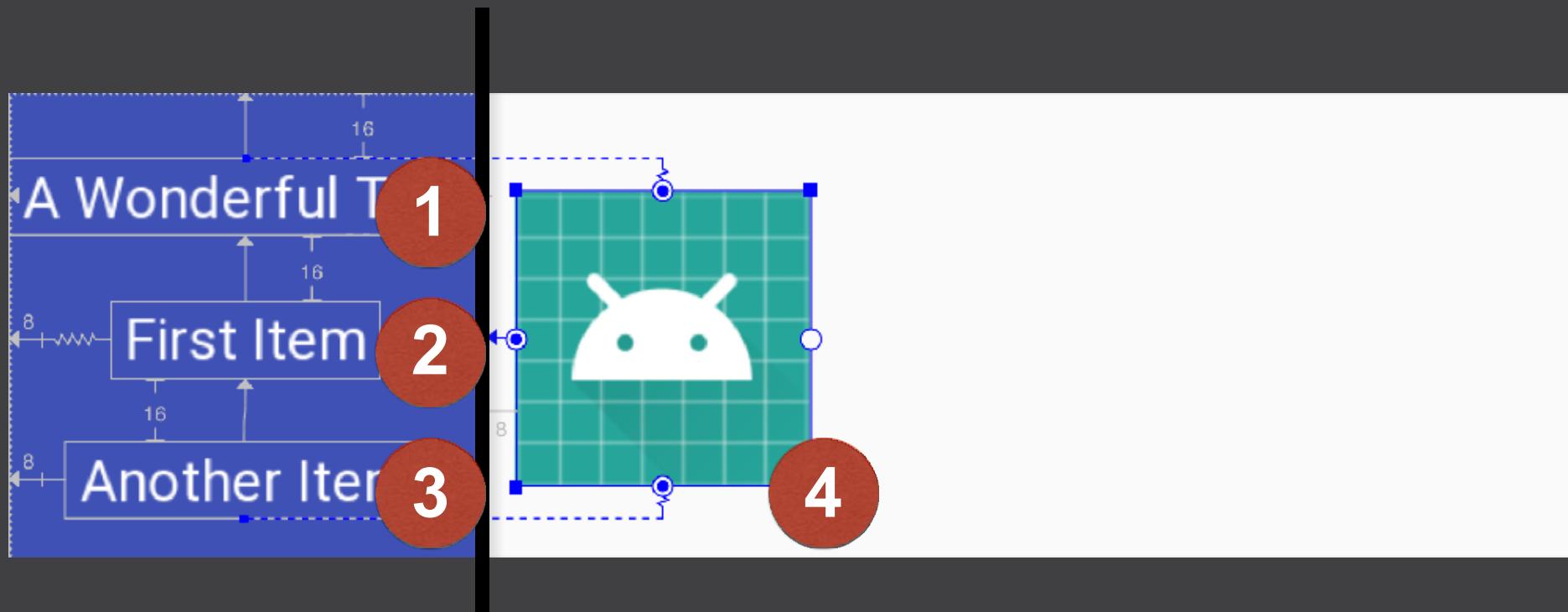
Groups : apply visibility to a set of widgets

Placeholder

Percent Dimensions

Circular Constraints

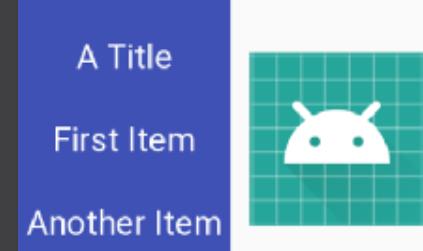
# Barriers



# Barriers



# Barriers



# Groups

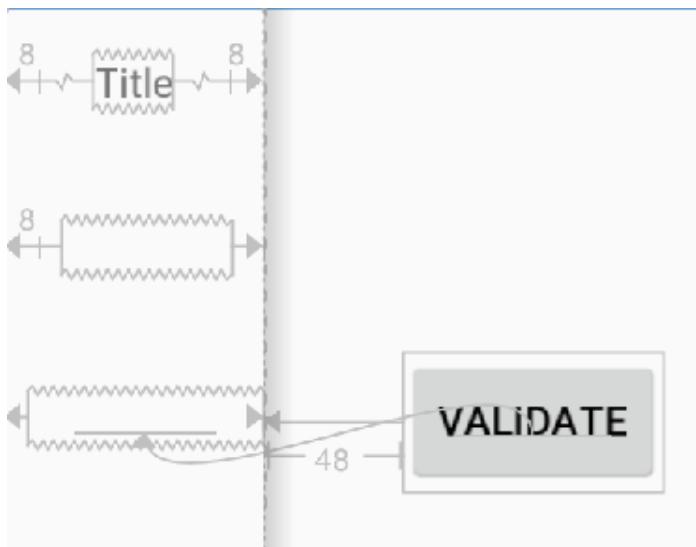
**textview2**

**textview3**

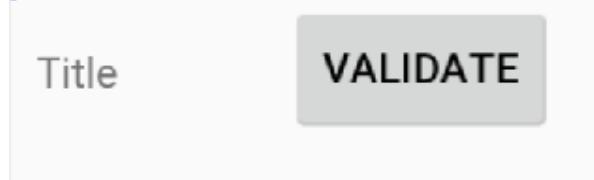
---

Title	
First item	
Another Item	<b>VALIDATE</b>

# Groups



Invisible



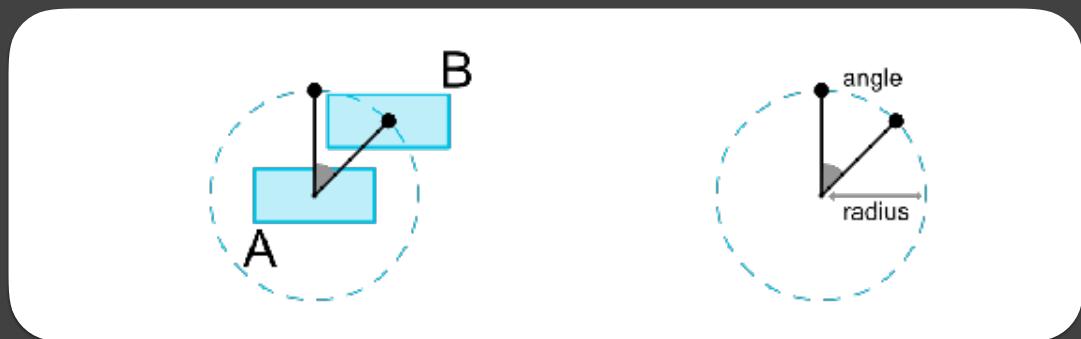
Gone

# Placeholder

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
  
    fun select(v: View) {  
        TransitionManager.beginDelayedTransition(main_layout)  
        placeholder.setContentId(v.id)  
        main_title.text= v.tag as CharSequence?;"";  
    }  
}
```



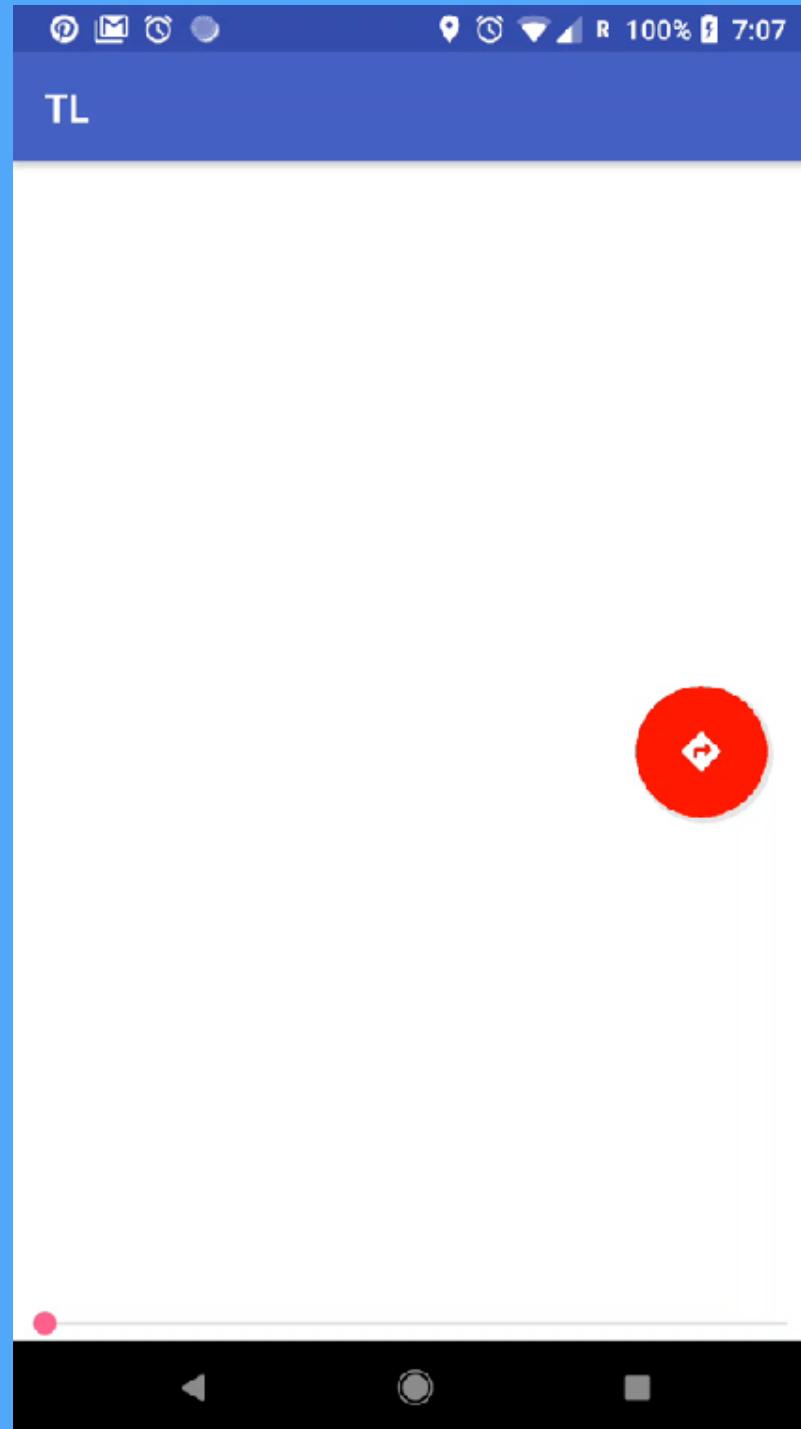
# Circular Constraints



```
<Button android:id="@+id/buttonA" ...>
<Button android:id="@+id/buttonB" ...
    app:layout_constraintCircle="@+id/buttonA"
    app:layout_constraintCircleRadius="100dp"
    app:layout_constraintCircleAngle="45" />
```

# Example

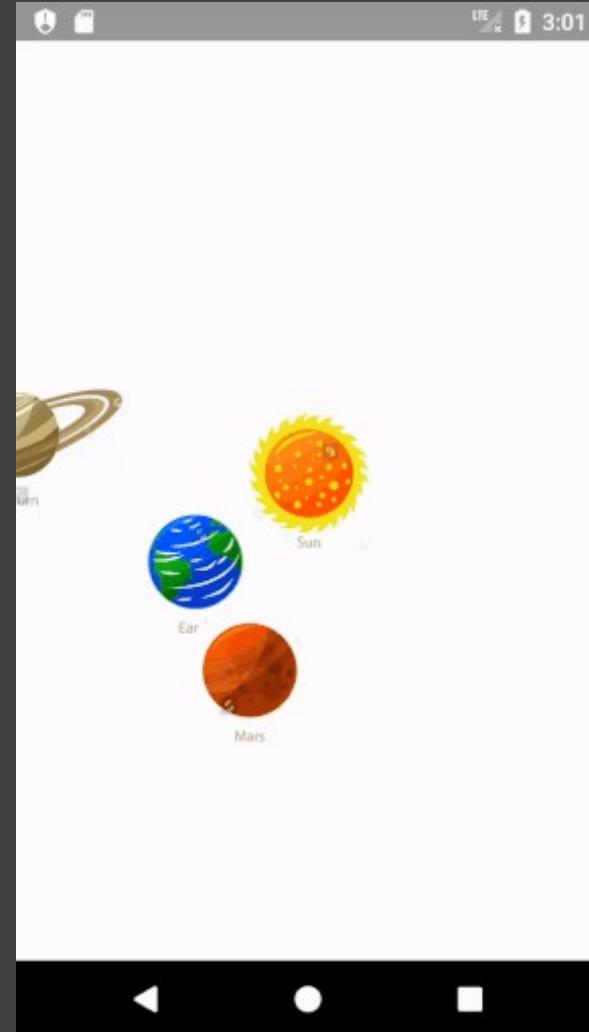
Circular  
Menu



# Circular Constraints

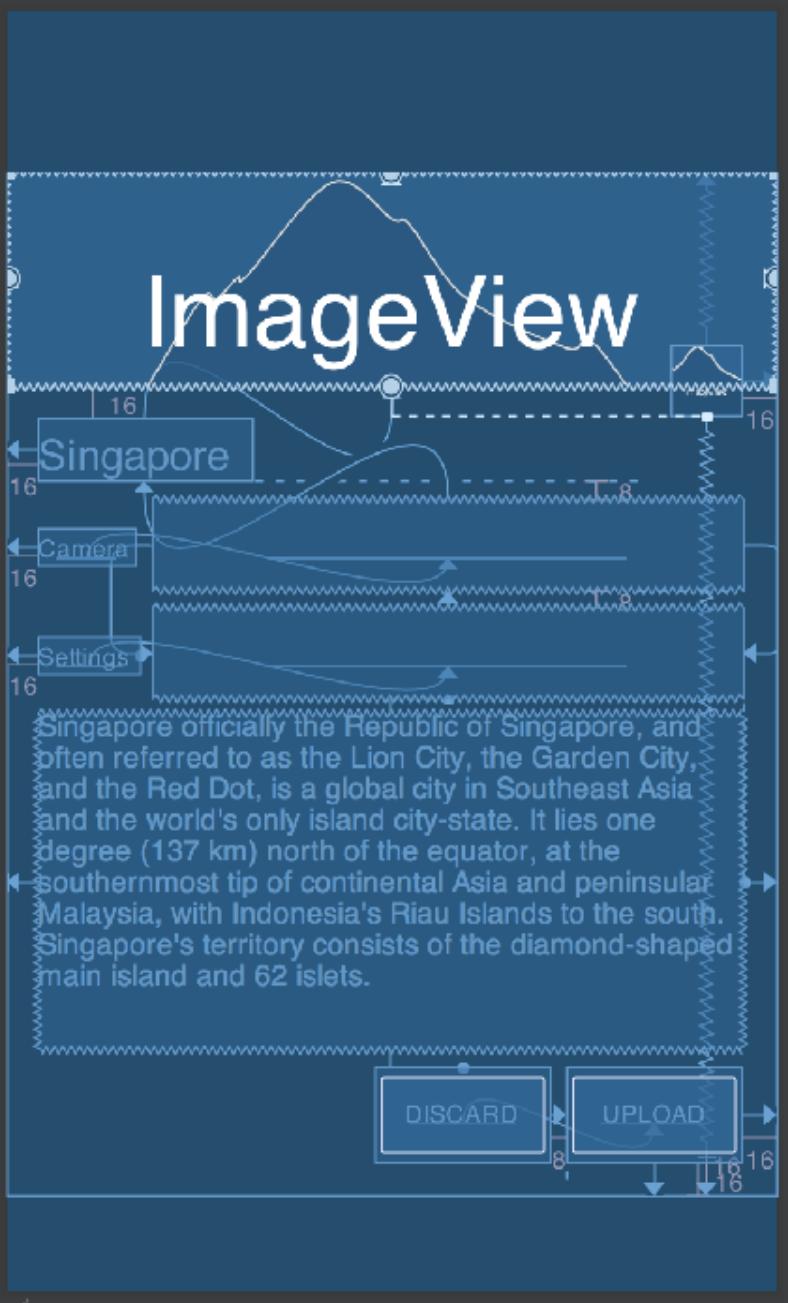
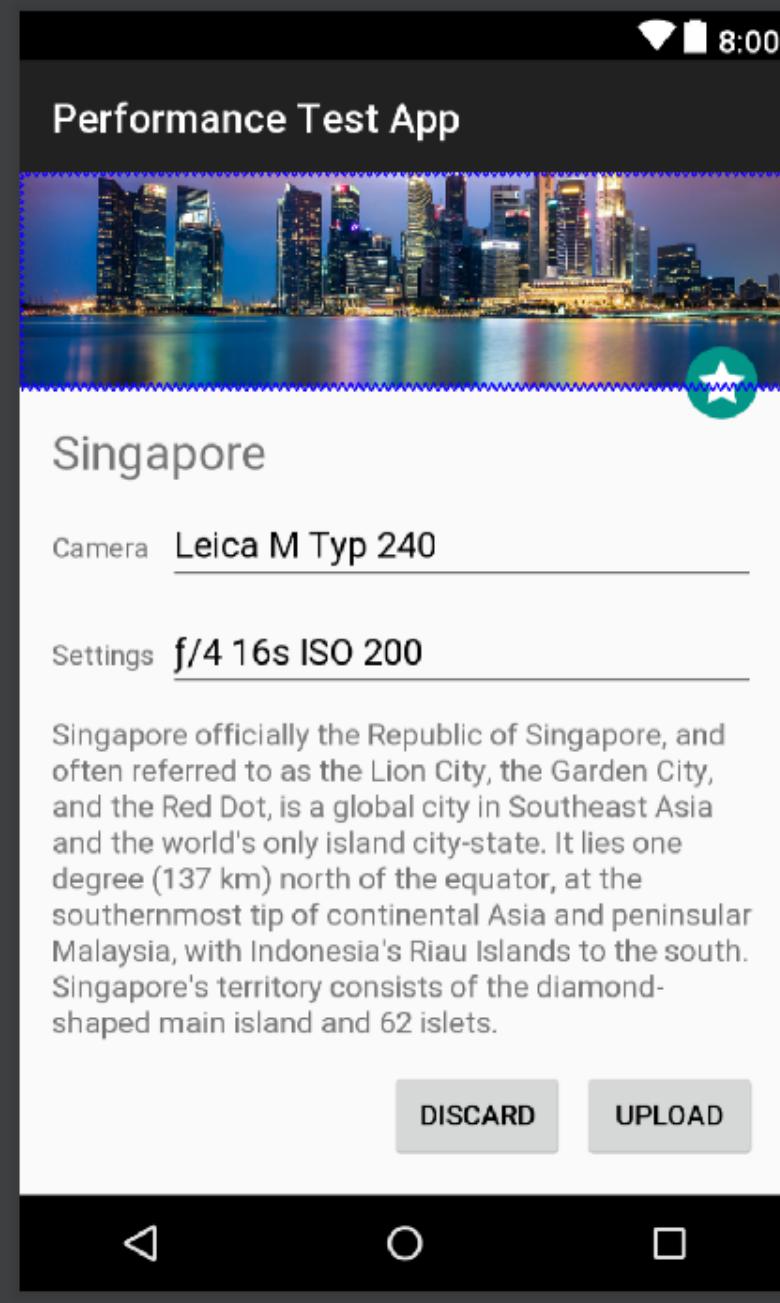
Andrew Kelly

[https://medium.com/devnibbles/  
constraintlayout-circular-  
positioning-9489b11cb0e5](https://medium.com/devnibbles/constraintlayout-circular-positioning-9489b11cb0e5)



# Our First Example





# Image View

Singapore

Camera

Settings

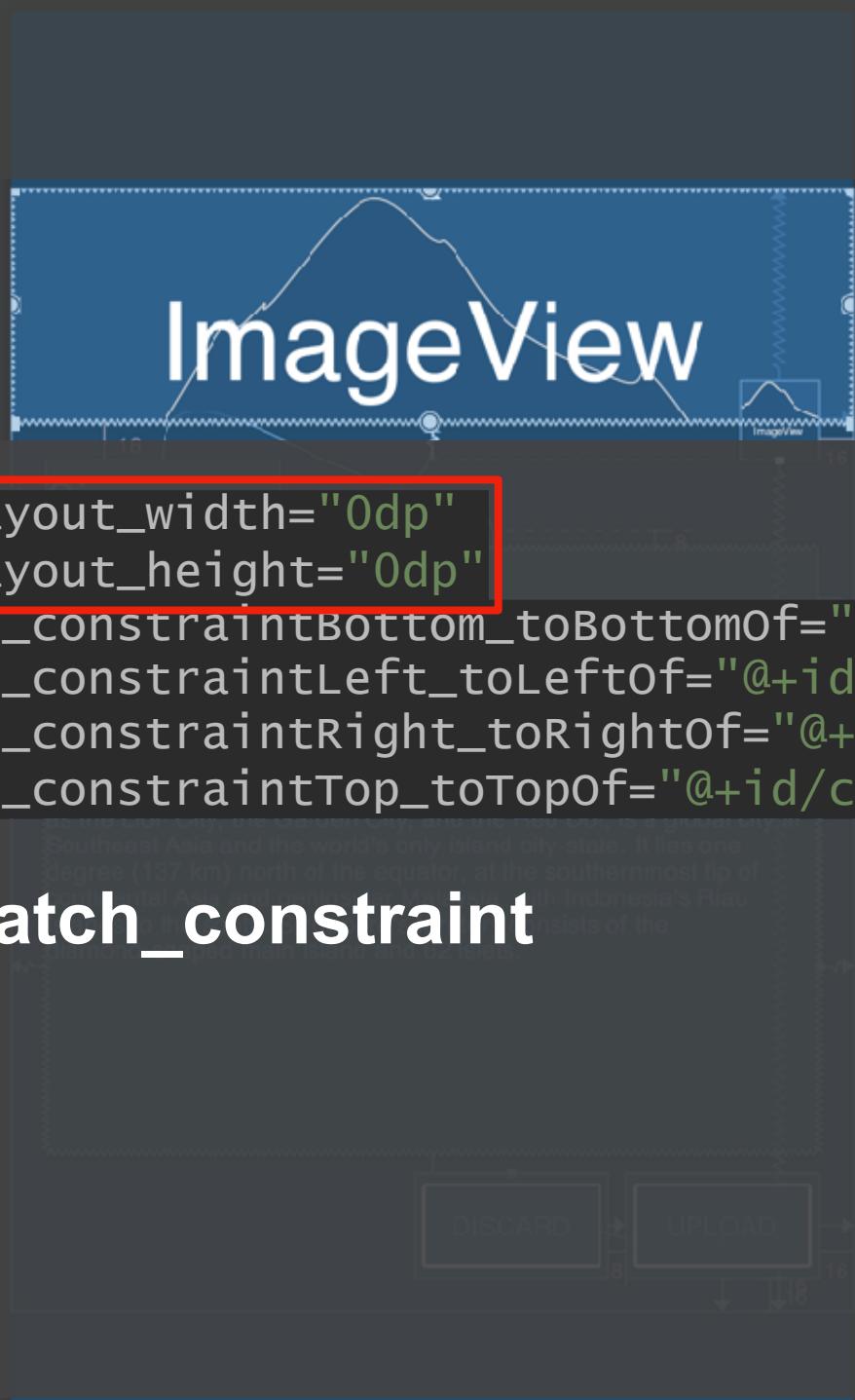
Singapore officially the Republic of Singapore, and often referred to as the Lion City, the Garden City, and the Red Dot, is a global city in Southeast Asia and the world's only island city-state. It lies one degree (137 km) north of the equator, at the southernmost tip of continental Asia and peninsular Malaysia, with Indonesia's Riau Islands to the south. Singapore's territory consists of the diamond-shaped main island and 62 islets.

DISCARD

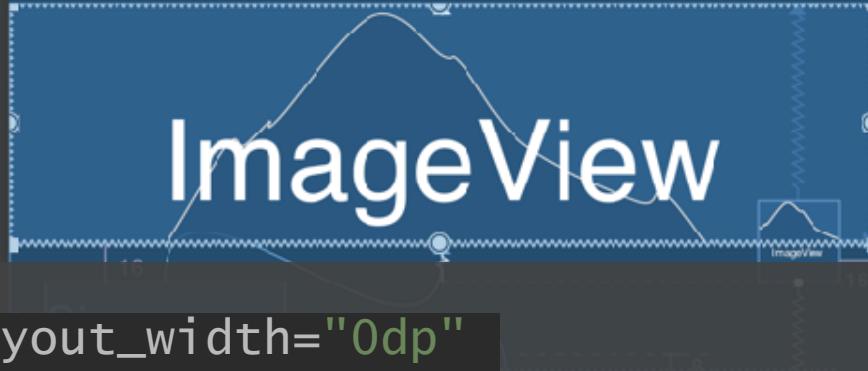
UPLOAD

# ImageView

```
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="@+id/favorite"
    app:layout_constraintLeft_toLeftOf="@+id/container"
    app:layout_constraintRight_toRightOf="@+id/container"
    app:layout_constraintTop_toTopOf="@+id/container"
```



**0dp = match\_constraint**



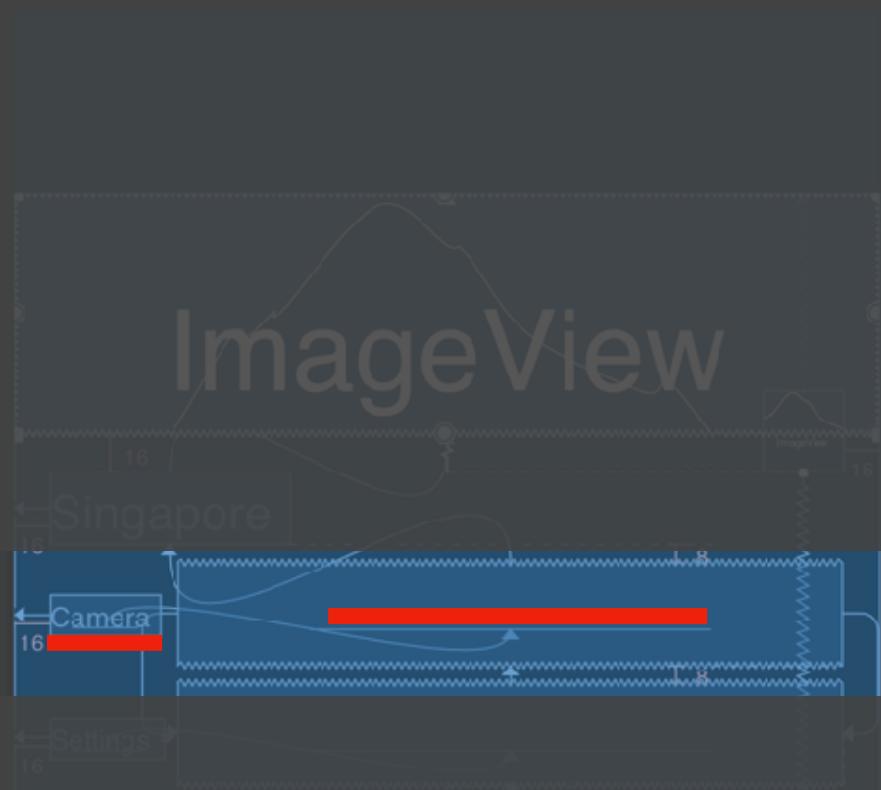
```
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="@+id/favorite"
    app:layout_constraintLeft_toLeftOf="@+id/container"
    app:layout_constraintRight_toRightOf="@+id/container"
    app:layout_constraintTop_toTopOf="@+id/container"
```

**0dp = match\_constraint**

**Size is defined by its constraints  
(expanded same as parent)**



```
app:layout_constraintLeft_toLeftOf="@+id/container"  
app:layout_constraintBaseline_toBaselineOf="@+id/cameraType"
```



```
app:layout_constraintLeft_toLeftof="@+id/container"  
app:layout_constraintBaseline_toBaselineof="@+id/cameraType"
```

**Baseline is constrained to another baseline**

# ImageView

```
app:layout_constraintLeft_toLeftof="@+id/container"  
app:layout_constraintRight_toRightof="@+id/container"
```

Singapore officially the Republic of Singapore, and often referred to as the Lion City, the Garden City, and the Red Dot, is a global city in Southeast Asia and the world's only island city-state. It lies one degree (137 km) north of the equator, at the southernmost tip of continental Asia and peninsular Malaysia, with Indonesia's Riau Islands to the south. Singapore's territory consists of the diamond-shaped main island and 62 islets.

DISCARD

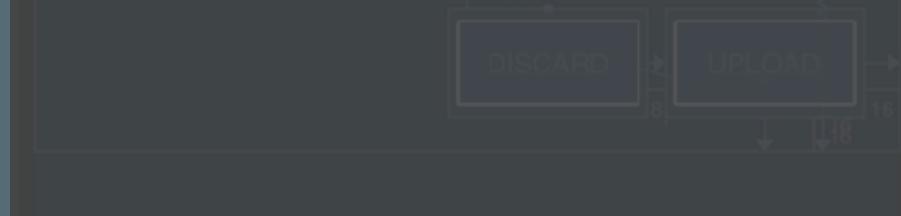
UPLOAD

ImageView  
app:layout

app:layout\_constraintRight\_toRightOf="@+id/container"

Horizontal constraints + no bias = Centered

constraintLeftToRightOf="@+id/container" r"



# ConstraintSet



# ConstraintSet

Contains all your constraints

Programming API

Clone from live layout or XML

# Initializing a ConstraintSet

**Layout resource**   `set.clone(context, R.layout.main)`

**Live layout**        `set.clone(findViewById(R.id.main))`

**XML file**            `set.load(context, R.xml.main)`

# Applying a ConstraintSet

`set.applyTo(layout)`

# ConstraintSet

## Initialization

```
ConstraintSet mConstraintSet1 = new ConstraintSet(); // create a Constraint Set  
ConstraintSet mConstraintSet2 = new ConstraintSet(); // create a Constraint Set
```

## onCreate

```
mConstraintSet2.clone(context, R.layout.state2); // get constraints from layout  
setContentView(R.layout.state1);  
mConstraintLayout = (ConstraintLayout) findViewById(R.id.activity_main);  
mConstraintSet1.clone(mConstraintLayout); // get constraints from ConstraintSet
```

## To change state

```
mConstraintSet1.applyTo(mConstraintLayout);
```

# ConstraintSet

## Initialization

```
ConstraintSet mConstraintSet1 = new ConstraintSet(); // create a Constraint Set  
ConstraintSet mConstraintSet2 = new ConstraintSet(); // create a Constraint Set
```

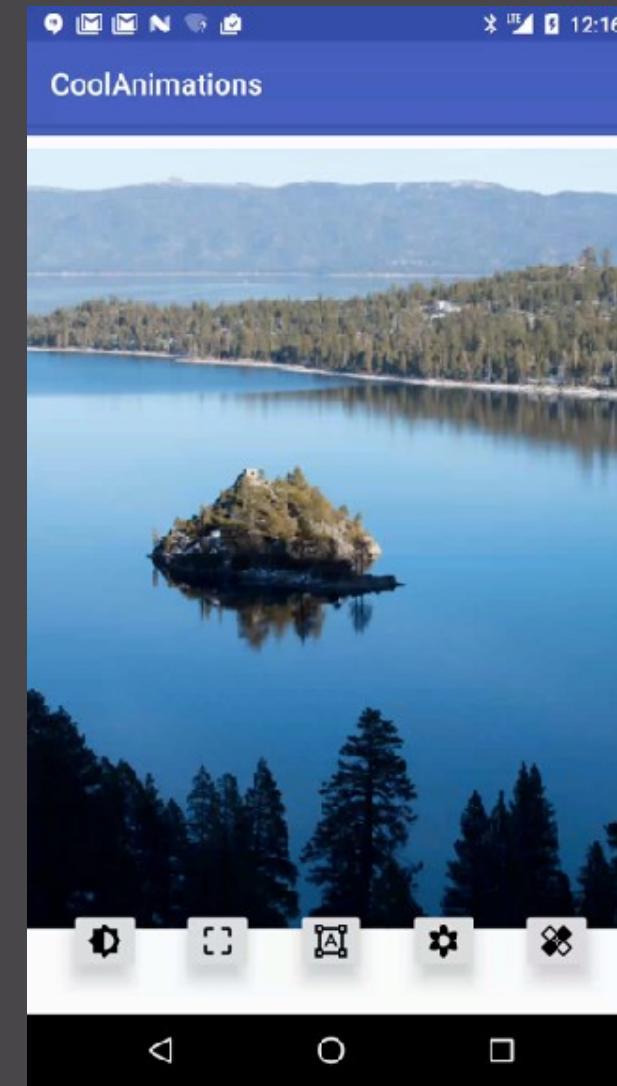
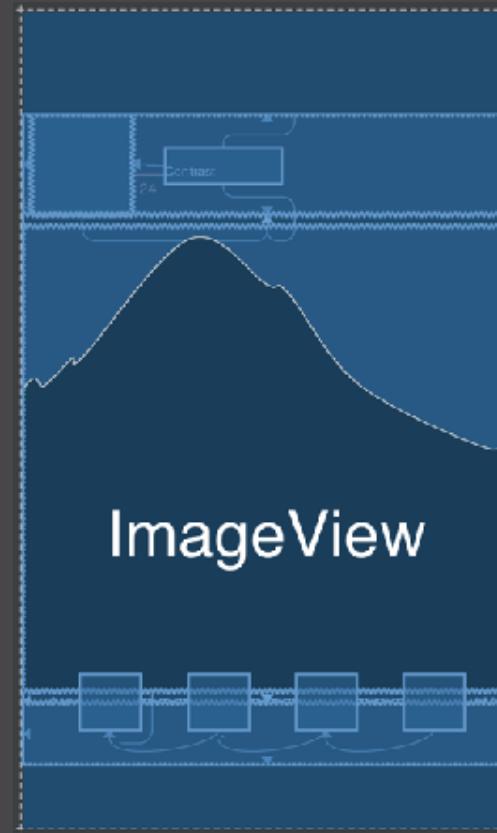
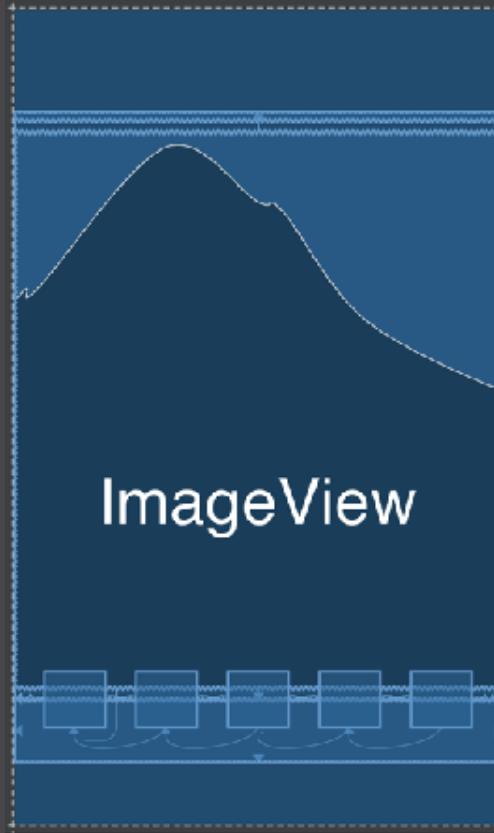
## onCreate

```
mConstraintSet2.clone(context, R.layout.state2); // get constraints from layout  
setContentView(R.layout.state1);  
mConstraintLayout = (ConstraintLayout) findViewById(R.id.activity_main);  
mConstraintSet1.clone(mConstraintLayout); // get constraints from ConstraintSet
```

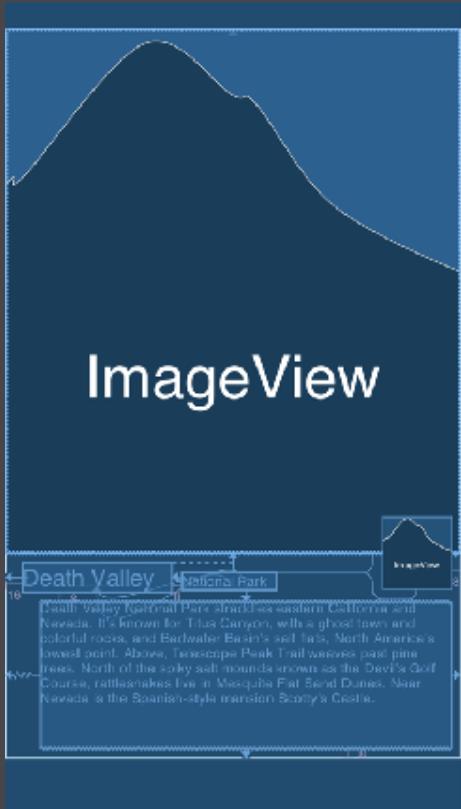
## To change state

```
TransitionManager.beginDelayedTransition(mConstraintLayout);  
mConstraintSet1.applyTo(mConstraintLayout);
```

# Build two layouts



# Build two layouts



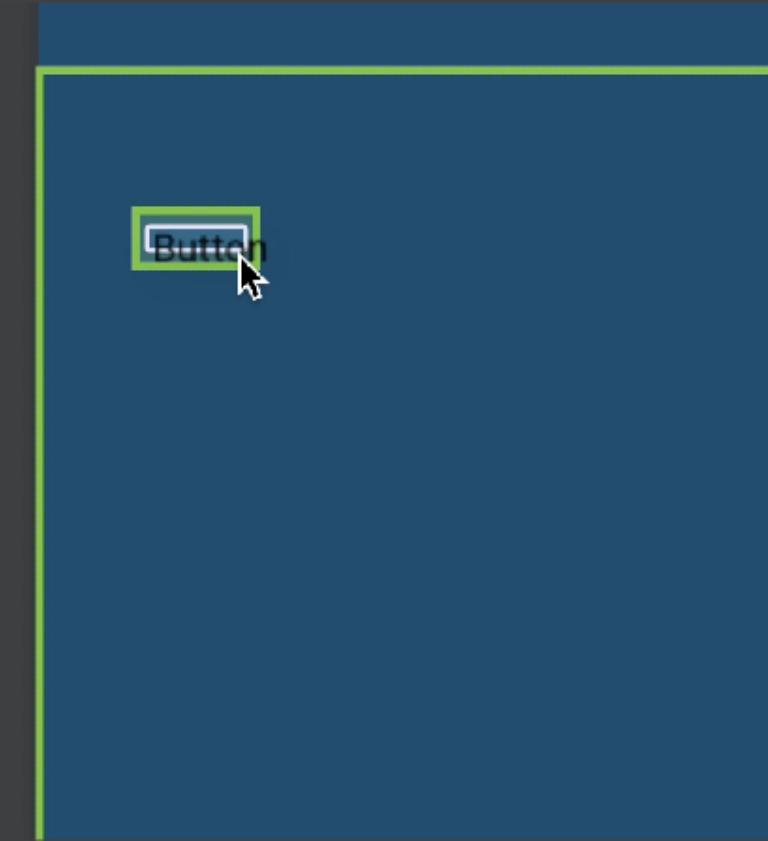
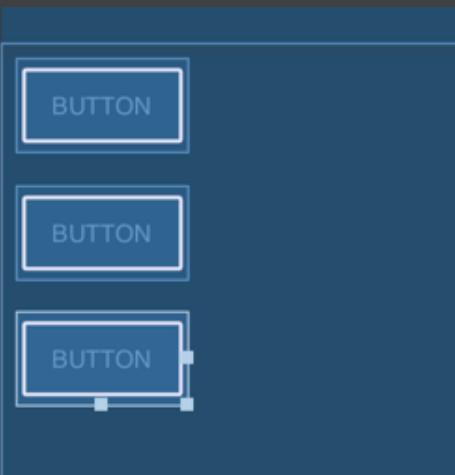
# More Real-life Examples



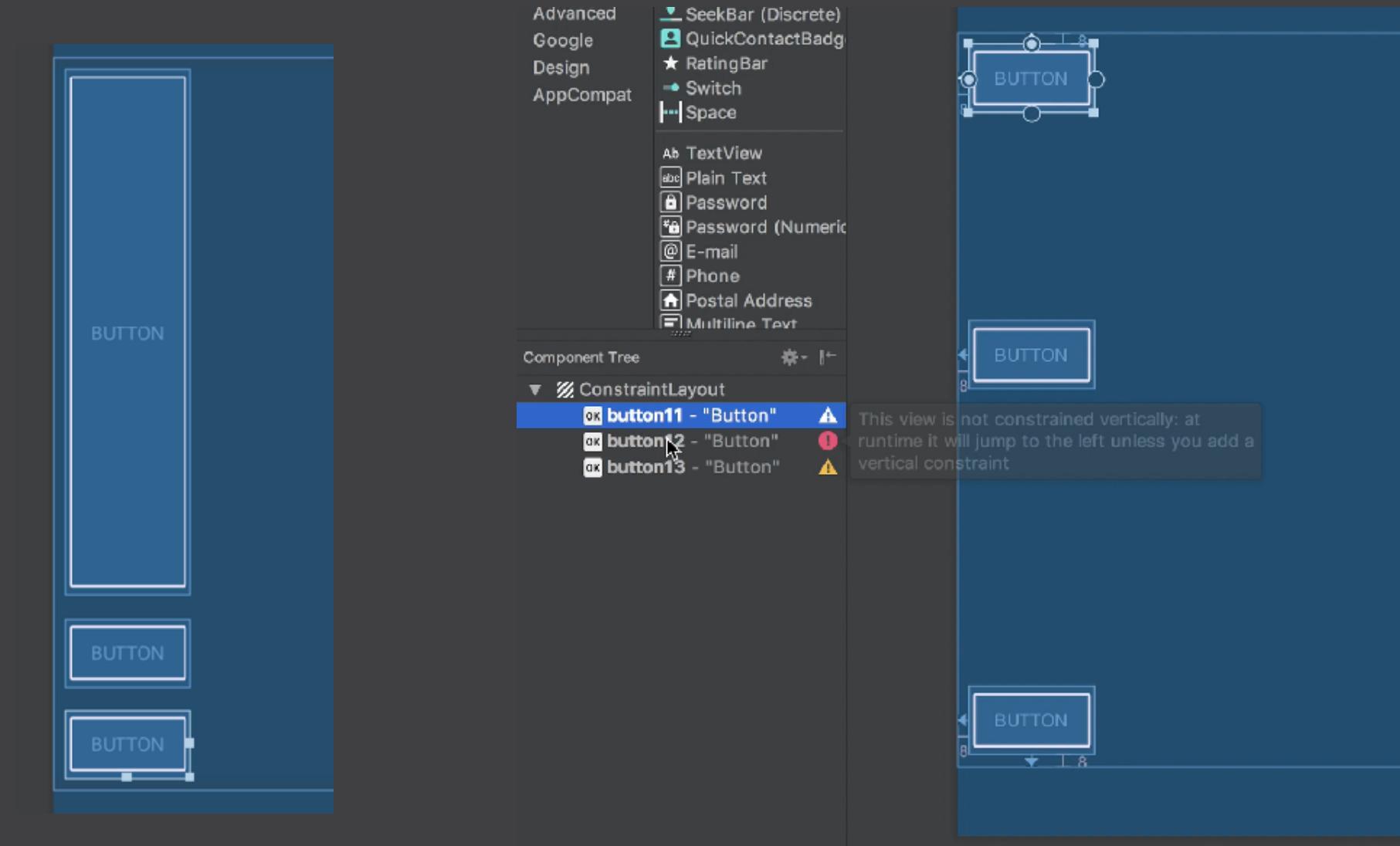
# Converting Existing Layouts

# LinearLayout

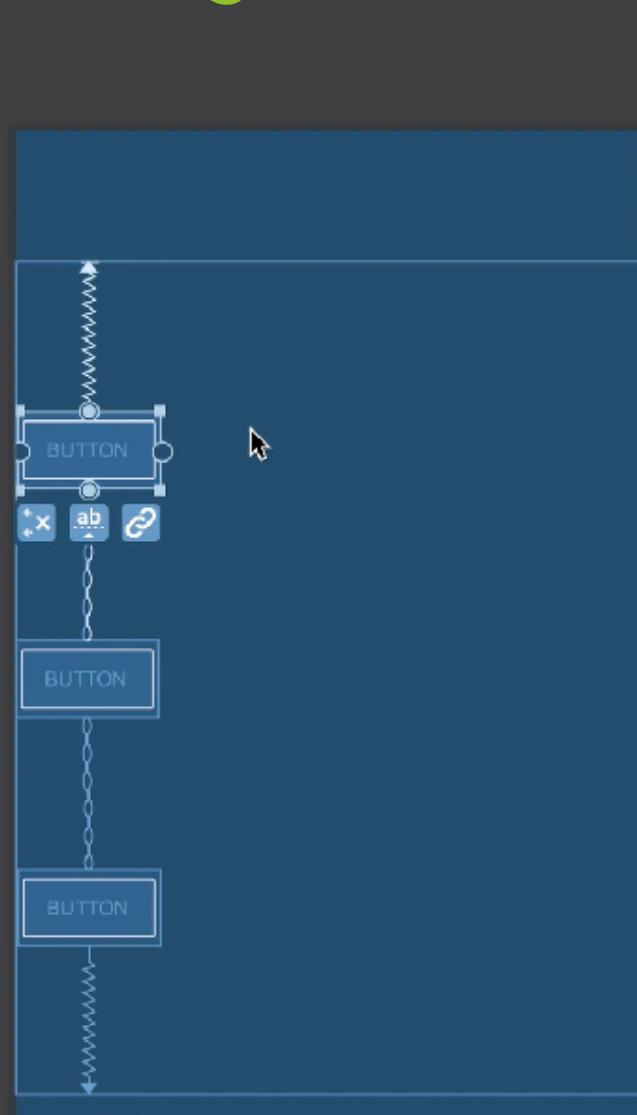
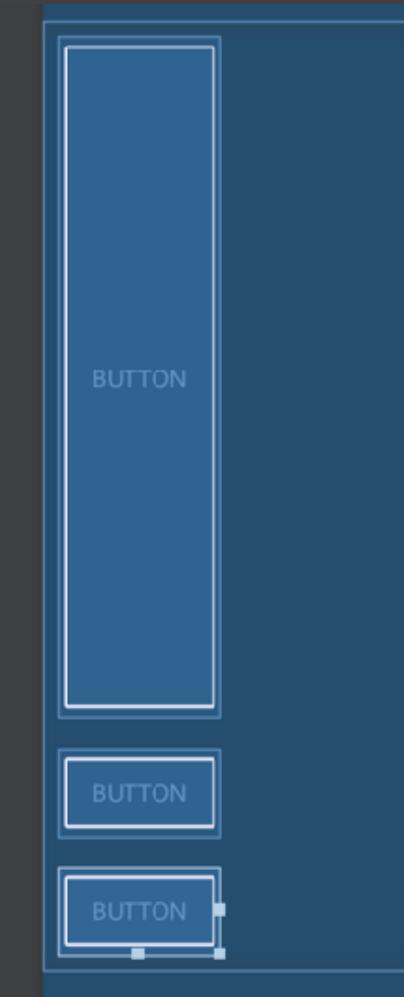
# Flowed behavior



# Weighted behavior



# Weighted behavior

A screenshot of the Android Studio Properties panel showing the properties for a selected button. The properties listed are: layout\_width (wrap\_content), layout\_height (wrap\_content), Button (style: buttonStyle, background: none, backgroundTint: none, stateListAnimator: none, elevation: 0, visibility: none, onClick: none), and TextView (text: Button, text: none, contentDescription: none). The "style" field is currently set to "buttonStyle".

# Weighted behavior

