

Optimizing UIs using Constraint Layout

**Google I/O Extended Yangon
8th July 2018**



Photo by Dmitri Popov on Unsp



Lin Min Phyo
Android Developer @ nexlabs



Constraint Layout

ViewGroup which children views have abilities to add constraints

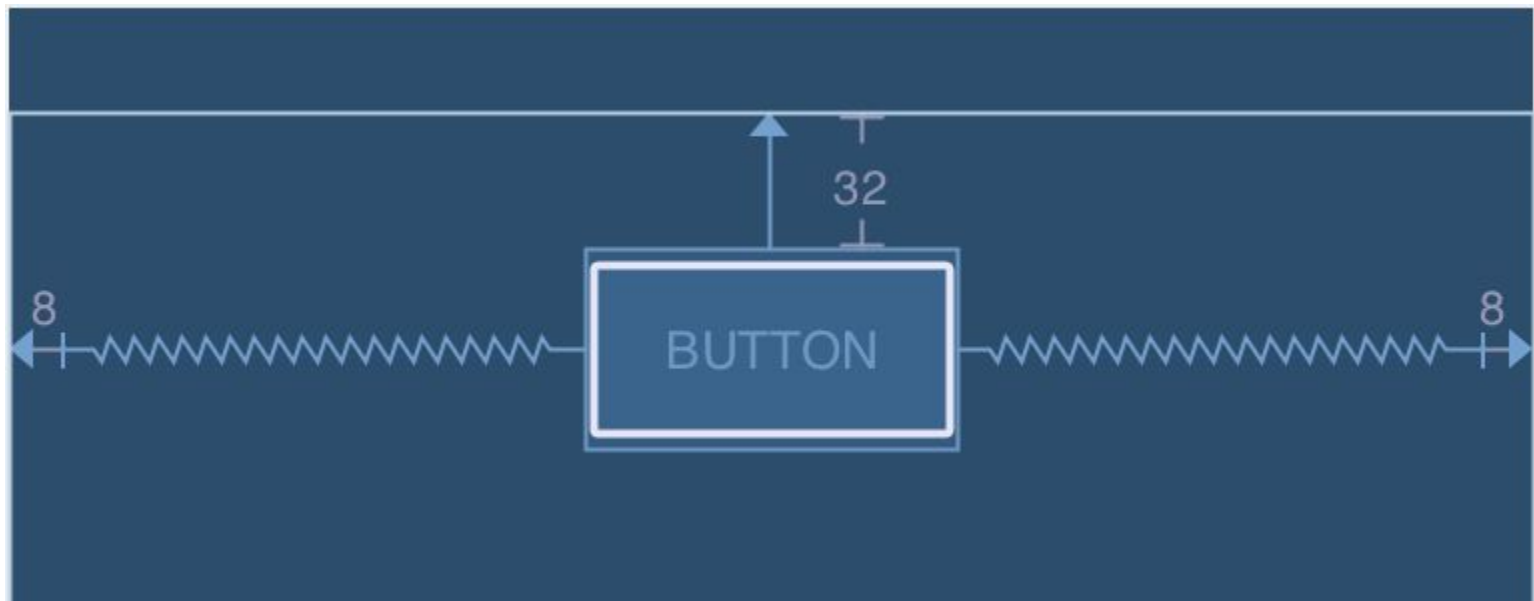




What are constraints ?

- Connections between views
- Connections between view and parent viewgroup
- Helpers that allowed connections

Constraints in Constraint layout



Constraint Layout

=

**All about of connections
between views**

—



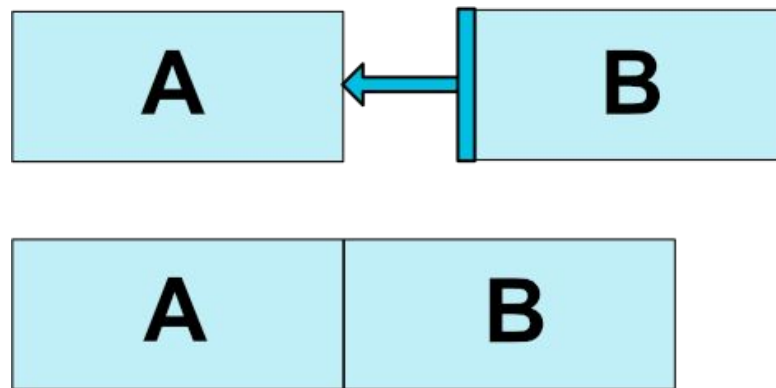
Why Constraint layout?

- Easier dimensions and positioning
- Minimize nested view groups
- Groups, guidelines, placeholders
- Auto transitions using Constraint Sets

Constraint Layout Positioning

—

Positioning Attributes



```
<Button android:id="@+id/buttonA" ... />
```

```
<Button android:id="@+id/buttonB" ...  
app:layout_constraintStart_toEndOf="@+id/buttonA" />
```

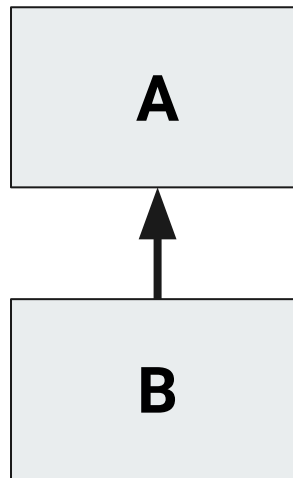
Positioning Attributes

```
<Button  
    android:id="@+id/buttonB" ...  
    app:layout_constraint[ ]_to[ ]Of="@+id/buttonA"  
/>
```

buttonB's side
(top,bottom,start,end)

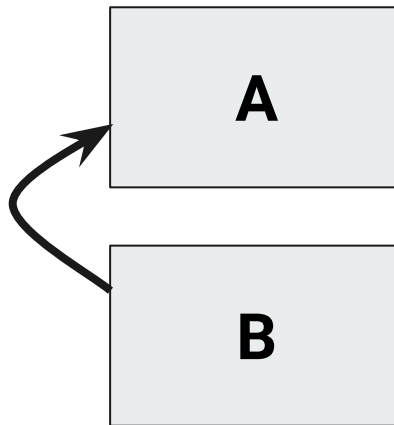
buttonA's side
(top,bottom,start,end)

Positioning Attributes



```
<Button  
    android:id="@+id/buttonB" ...  
    app:layout_constraintTop_toBottomOf="@+id/buttonA"  
>
```

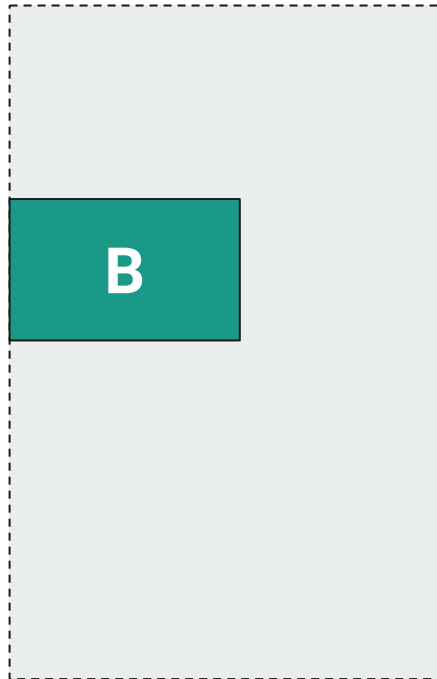
Positioning Attributes



```
<Button  
    android:id="@+id/buttonB" ...  
    app:layout_constraintStart_toStartOf="@+id/buttonA"  
/>
```

Constraint Layout

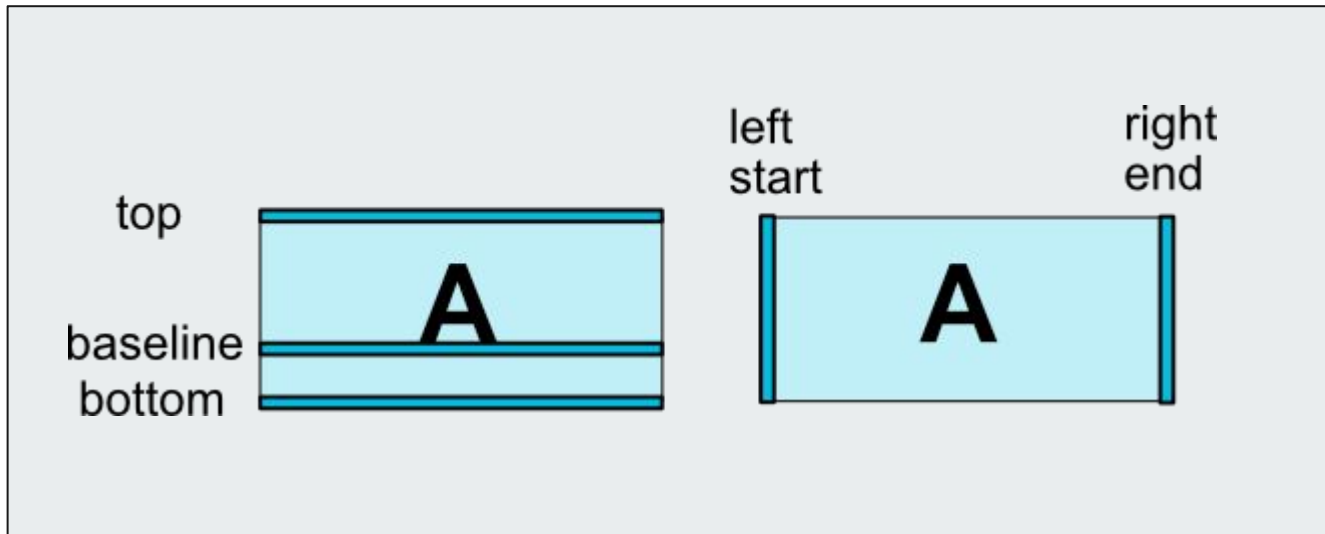
Positioning Attributes



```
<Button  
    android:id="@+id/buttonB" ...  
    app:layout_constraintStart_toStartOf="parent"  
/>
```

Constraint Layout

Positioning Attributes



Constraint Layout

Positioning Attributes



<code>layout_constraintLeft_toLeftOf</code>	<code>layout_constraintBottom_toTopOf</code>
<code>layout_constraintLeft_toRightOf</code>	<code>layout_constraintBottom_toBottomOf</code>
<code>layout_constraintRight_toLeftOf</code>	<code>layout_constraintStart_toEndOf</code>
<code>layout_constraintRight_toRightOf</code>	<code>layout_constraintStart_toStartOf</code>
<code>layout_constraintTop_toTopOf</code>	<code>layout_constraintEnd_toStartOf</code>
<code>layout_constraintTop_toBottomOf</code>	<code>layout_constraintEnd_toEndOf</code>
<code>layout_constraintBaseline_toBaselineOf</code>	

Positioning Attributes



Things to note

- One vertical constraint and one horizontal constraint should be added
- If not, Top left is default
- Only 1 constraint for each side
- No **match parent** (use 0 dp instead)

Code sample

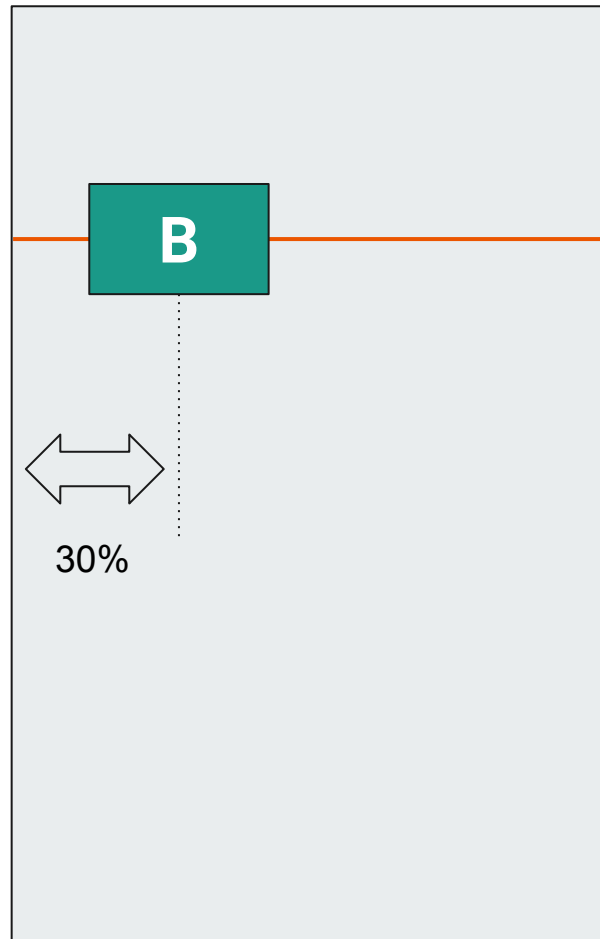
Constraint Layout Bias

Determining which side to bias

—

Constraint Layout

Bias Attributes



Bias Attributes



<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintHorizontal_bias="0.3"

/>

Constraint Layout

Bias Attributes



- Vertical Bias
- Horizontal bias
- Mixing vertical and horizontal

Code Sample

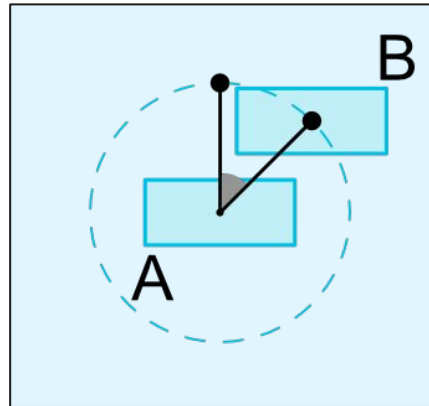
Constraint Layout

Circular Positioning

Position with angle



Circular Positioning



```
<Button android:id="@+id/buttonA" ... />
```

```
<Button android:id="@+id/buttonB" ...
```

```
    app:layout_constraintCircle="@+id/buttonA"
```

```
    app:layout_constraintCircleRadius="100dp"
```

```
    app:layout_constraintCircleAngle="45"
```

```
/>
```

Constraint Layout Group

Grouping views using id

—

Constraint Layout

Group

```
<android.support.constraint.Group  
    android:id="@+id/group"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:visibility="visible" // gone,invisible  
    app:constraint_referenced_ids="button4,button9"  
>
```


Constraint Layout

Circular Positioning + Group



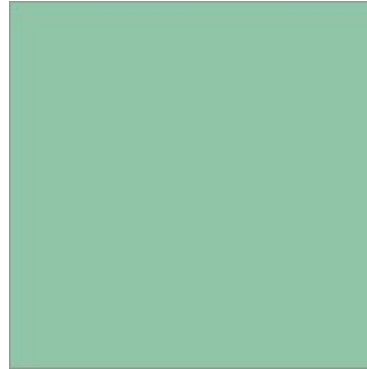
Constraint Layout Ratio

Determining ratio of views



Constraint Layout

Ratio Attributes



<ImageView

`android:layout_width="0dp"`

`android:layout_height="0dp"`

`app:layout_constraintStart_toStartOf="parent"`

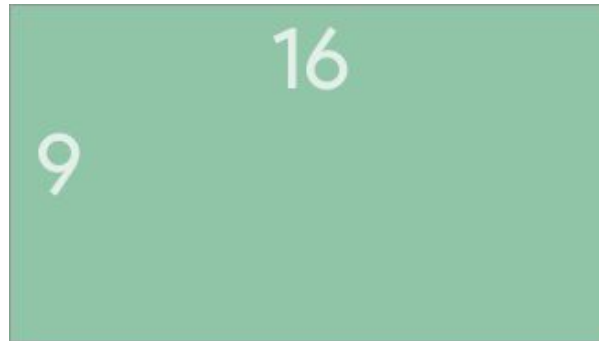
`app:layout_constraintEnd_toEndOf="parent"`

`app:layout_constraintDimensionRatio="1:1"`

/>

Constraint Layout

Ratio Attributes



<Button

```
android:layout_width="0dp"
```

```
android:layout_height="0dp"
```

```
app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
```

```
app:layout_constraintDimensionRatio="16:9"
```

/>

Constraint Layout

Ratio Attributes



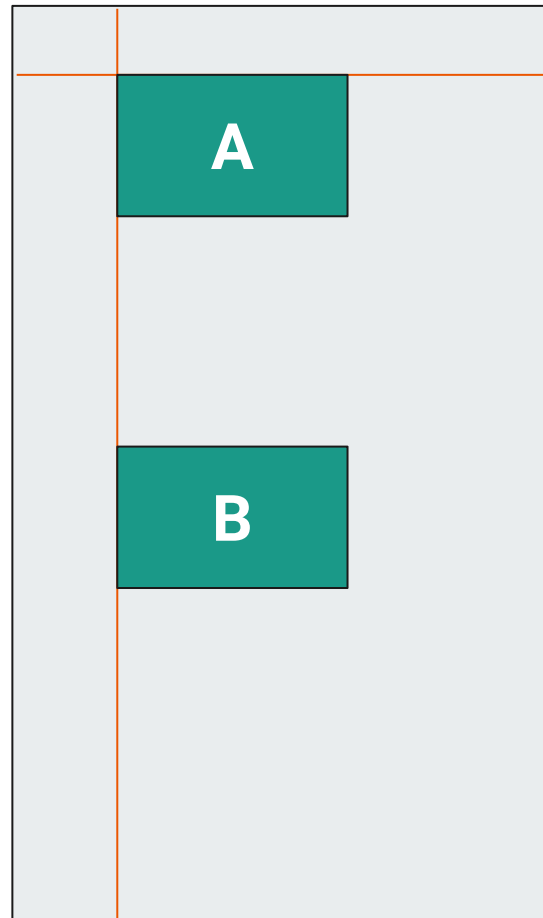
Code sample

Constraint Layout Guidelines

Virtual, invisible helper views

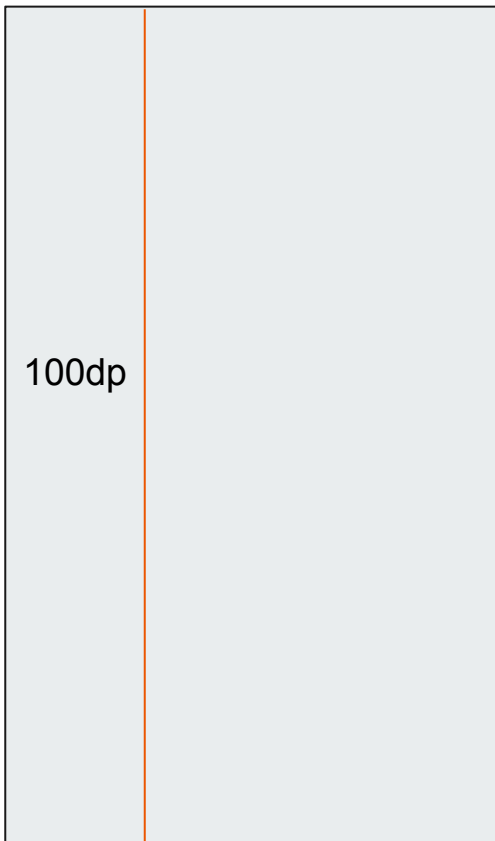


Constraint Layout **Guideline**



Constraint Layout

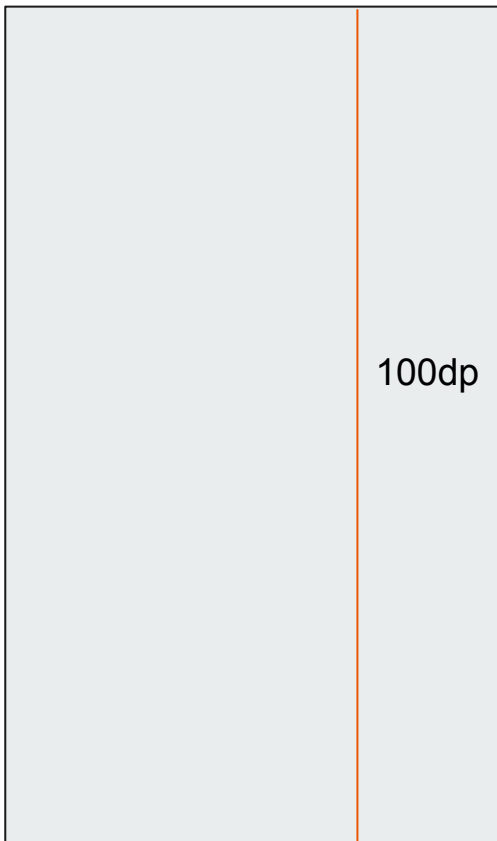
Guideline



```
<android.support.constraint.Guideline
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/guideline"
    app:layout_constraintGuide_begin="100dp"
    android:orientation="vertical"
/>
```


Constraint Layout

Guideline



```
<android.support.constraint.Guideline
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/guideline"
    app:layout_constraintGuide_end="100dp"
    android:orientation="vertical"
/>
```

Constraint Layout

Guideline



```
<android.support.constraint.Guideline
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/guideline"
    app:layout_constraintGuide_percent="0.5"
    android:orientation="vertical"
/>
```

Constraint Layout

Guideline



Orientation

- Vertical
- Horizontal

Use cases

- Keylines
- Reduce multiple margin on views

Code Sample

Constraint Layout Chain

Linking multiple views in horizontal
or vertical

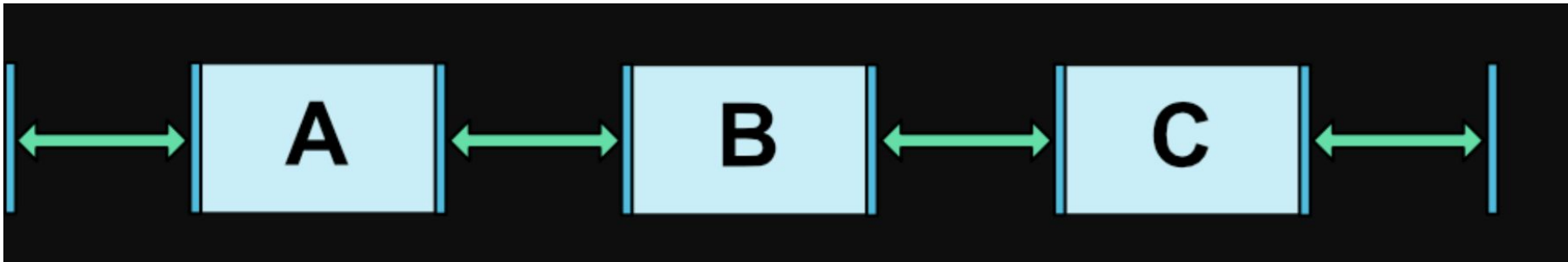


Constraint Layout

Chain



CHAIN_SPREAD

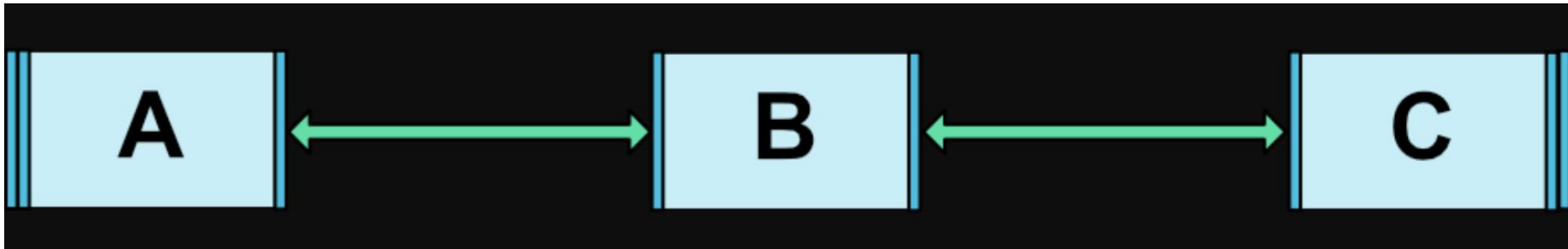


Constraint Layout

Chain



CHAIN_SPREAD_INSIDE

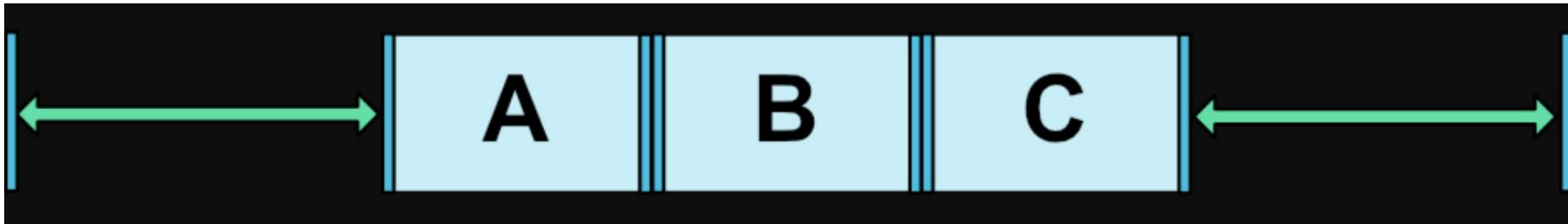


Constraint Layout

Chain



CHAIN_PACKED

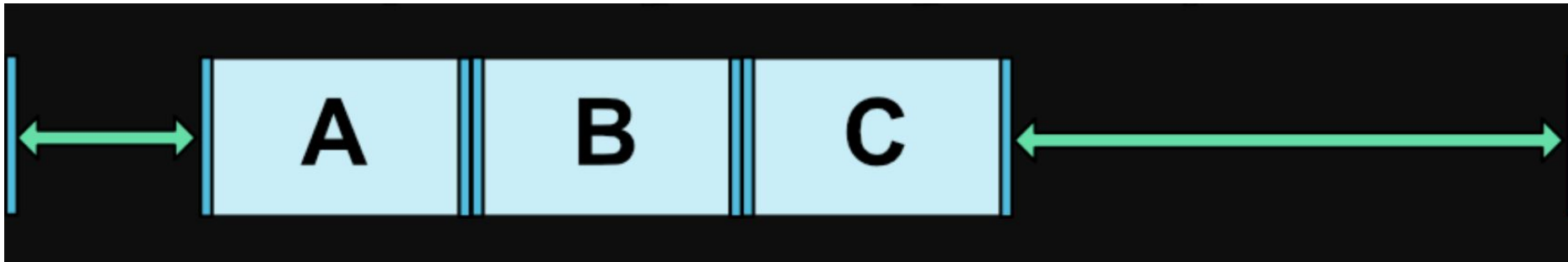


Constraint Layout

Chain



CHAIN_PACKED + Biasing



CHAIN_SPREAD + 0dp (match parent)



Constraint Layout

Chain



Orientation

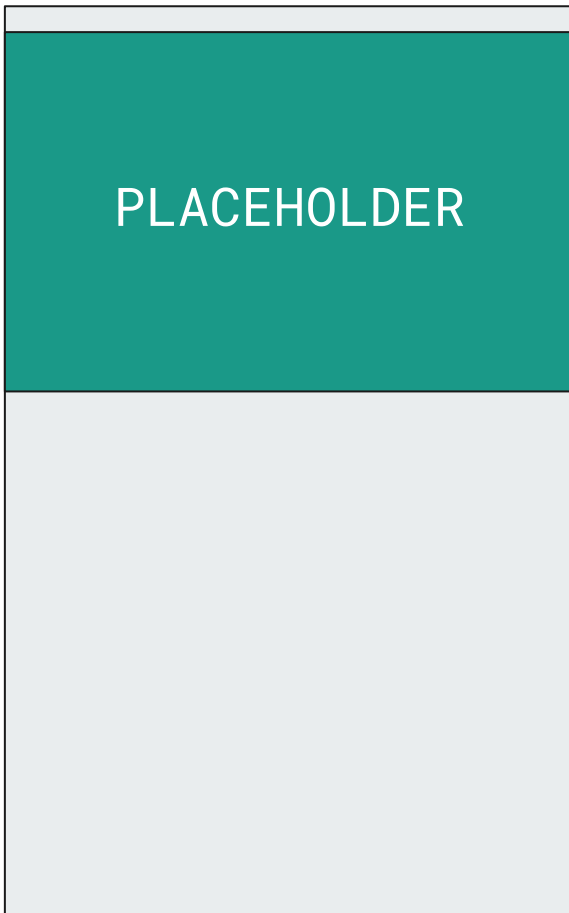
- Vertical
- Horizontal

Code Sample

Placeholders

Placeholder view which is ready for other views to place

Placeholder



- Empty View (Placeholder) with constraints
- Any view can be swapped into it dynamically or directly in XML
- TransitionManager helps auto transitions

Constraint Layout

Placeholder

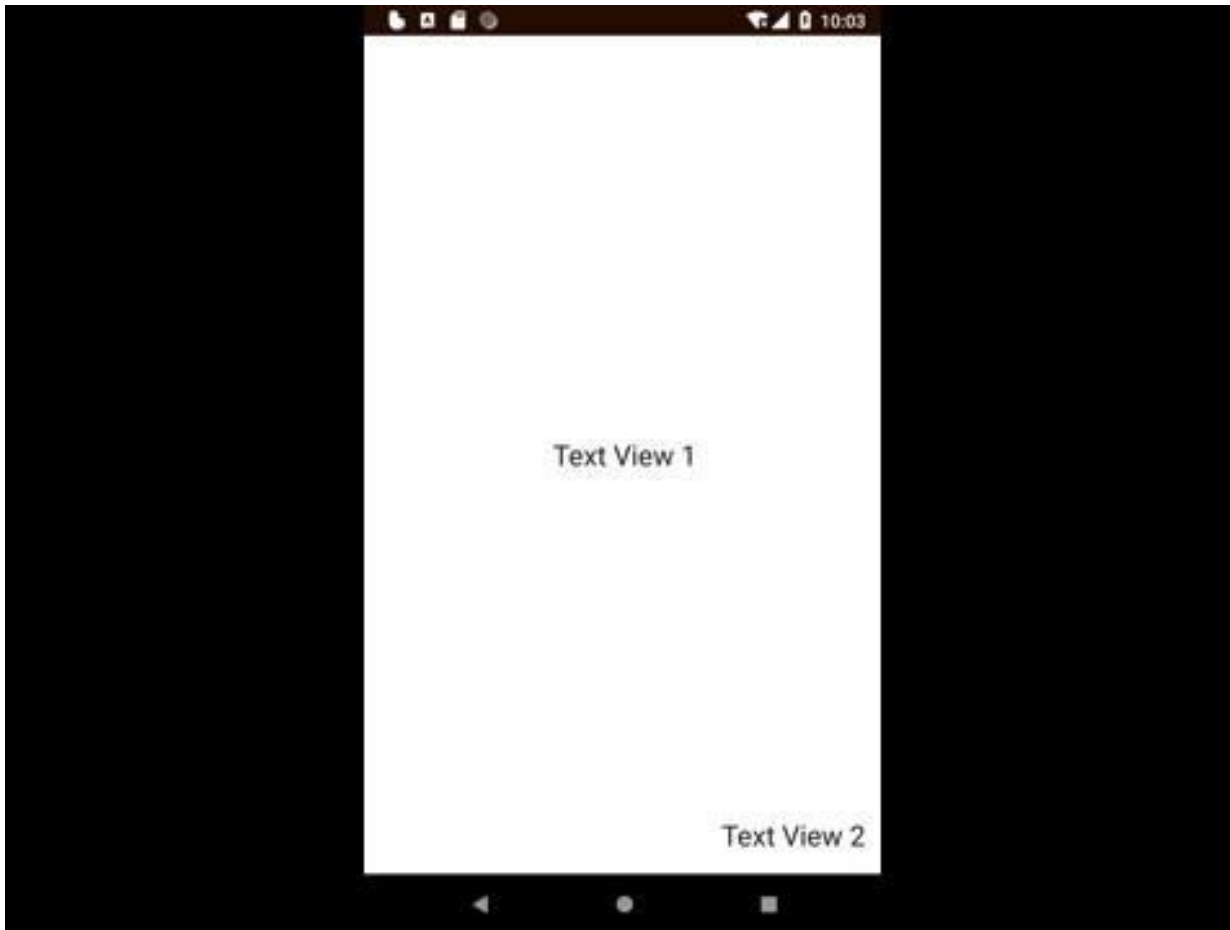


Two use cases

- Transition by placing dynamically
- Template of multi screen support

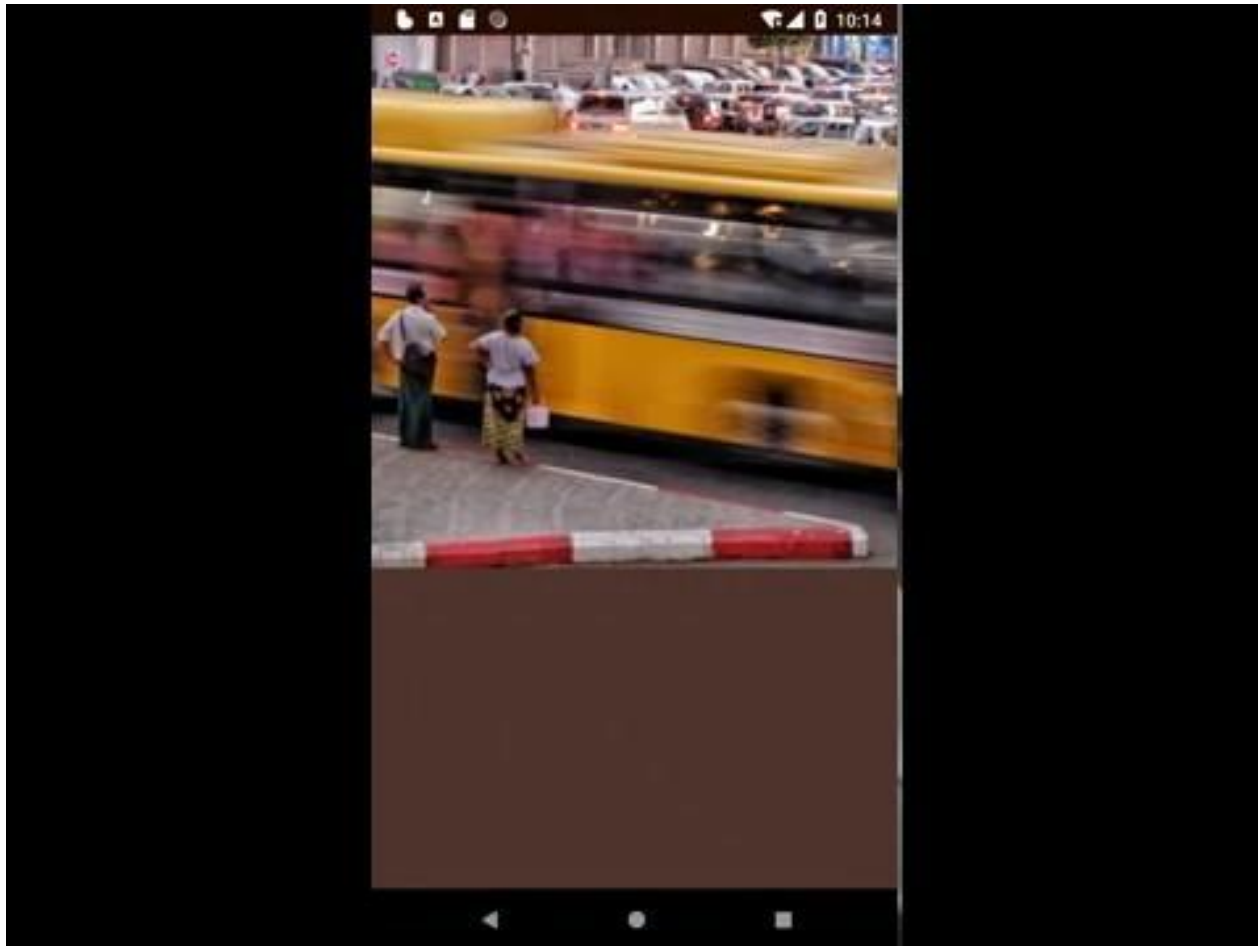
Constraint Layout

Placeholder



Constraint Layout

Placeholder



Constraint Layout

Placeholder



```
<ImageView android:id="@+id/image" .... />
```

```
<android.support.constraint.Placeholder
```

```
.....
```

```
    android:id="@+id/placeholder_image"
```

```
    app:content="@+id/image"
```

```
/>
```

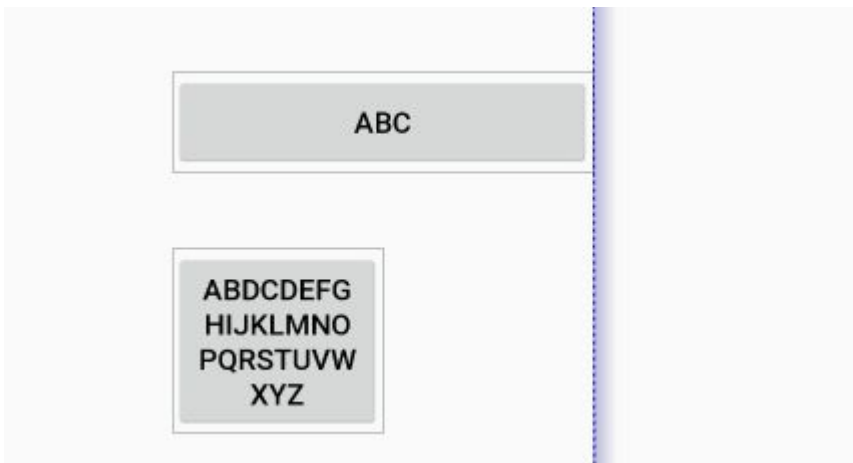


Barriers

Barrier between views

Constraint Layout

Barrier



Want to show a view at the end of two buttons

Constraint Layout

Placeholder

```
<TextView android:id="@+id/textView1" ... />
<TextView android:id="@+id/textView2" ... />
<android.support.constraint.Barrier
    android:id="@+id/barrier"
    app:barrierDirection="right"
    app:constraint_referenced_ids="textView1, textView2" />
<TextView
    android:text="Text right of barrier"
    app:layout_constraintStart_toStartOf="@+id/barrier" />
```

Constraint Layout

Placeholder



Code Sample

Constraint Sets

Animate between two layouts

Constraint Layout

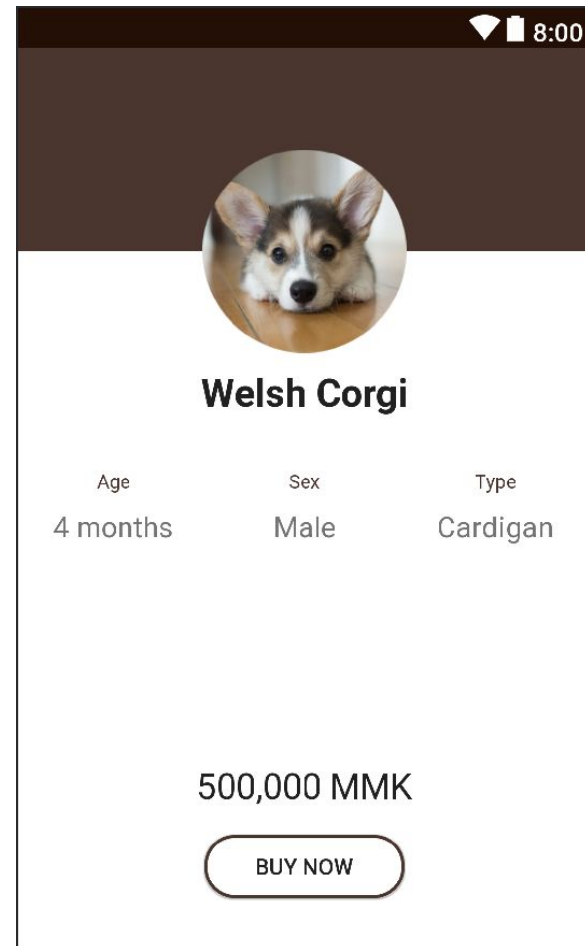
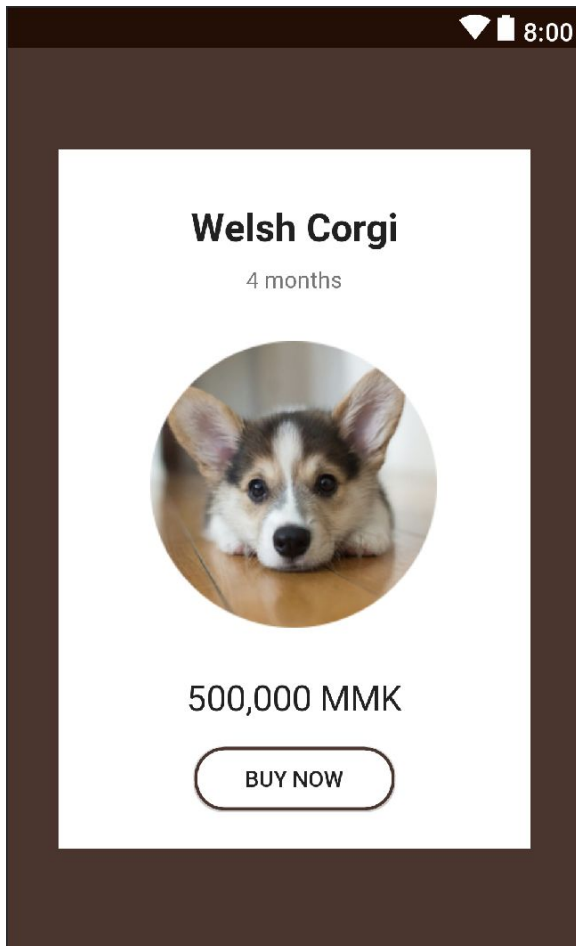
Constraint Set



1. Create two layouts
2. Clone into two ConstraintSets
3. Apply To ConstraintLayout

Constraint Layout

Constraint Set



Constraint Set



Things to note

- Should be inside a single constraint layout
- All children must have id
- Does not animate text size
- Views invisible? Must hide some view on initial layout

Do's



- Try **Layout Editor**
- **Groups** for visibility control
- Play with **Chains** instead of weights
- Using **Guidelines** as keylines
- Basic transitions with **Constraint sets**

Don'ts



- Do not overuse
- Reduce constraints (esp. in RecyclerView)
- Think twice before using barriers in RecyclerView
- Missing IDs
- Nested constraint layout

Constraint Layout 2.0

What's new

Constraint Layout 2.0

Constraint Helpers



Helper Views like Guidelines

- Linear
- Flow (Flexbox style)

Constraint Layout 2,0

Layers



- Hide, show, lock elements in editor
- Support transforms

Constraint Layout 2.0

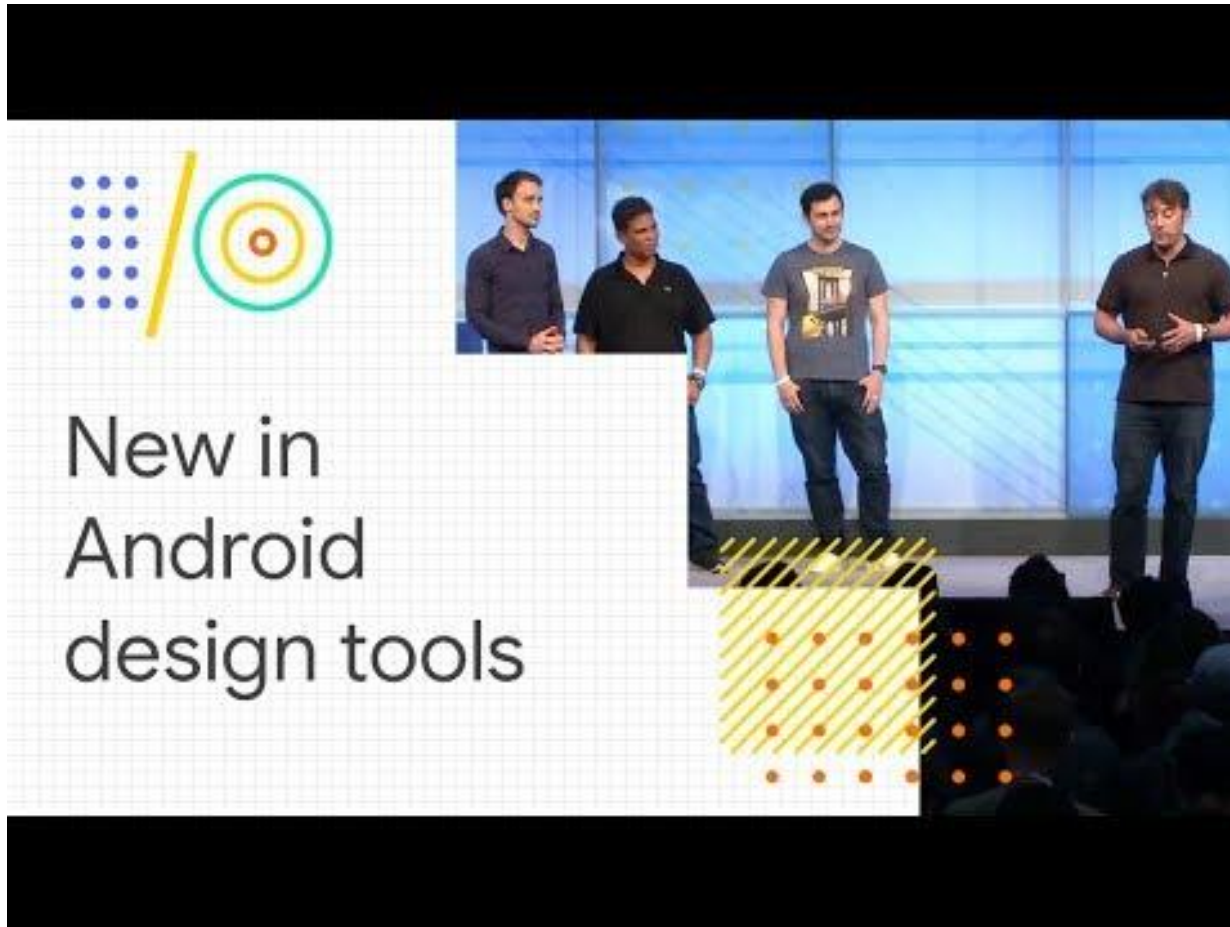
Layout Management



- Improved layout editor
- Constraint sets via xml (Constraint states)
- Motion layout + Motion Scene
(Constraint states + Motion Events + Key Frames)

Constraint Layout 2.0

Motion Layout



Resources



- <https://constraintlayout.com/>
- <https://developer.android.com/reference/android/support/constraint/ConstraintLayout>
- <https://github.com/hashlin/constraint-layout-example>
- <https://www.youtube.com/watch?v=ytZteMo4ETk>
- <https://medium.com/google-developers/introduction-to-motionlayout-part-i-29208674b10d>

Thanks !

Time to start using Constraint Layout