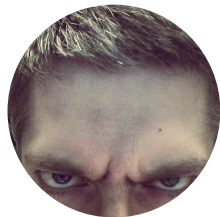


ConstraintLayout

all the things!



Aleksander Piotrowski
@pelotasplus



Why new layout?

Why new layout?

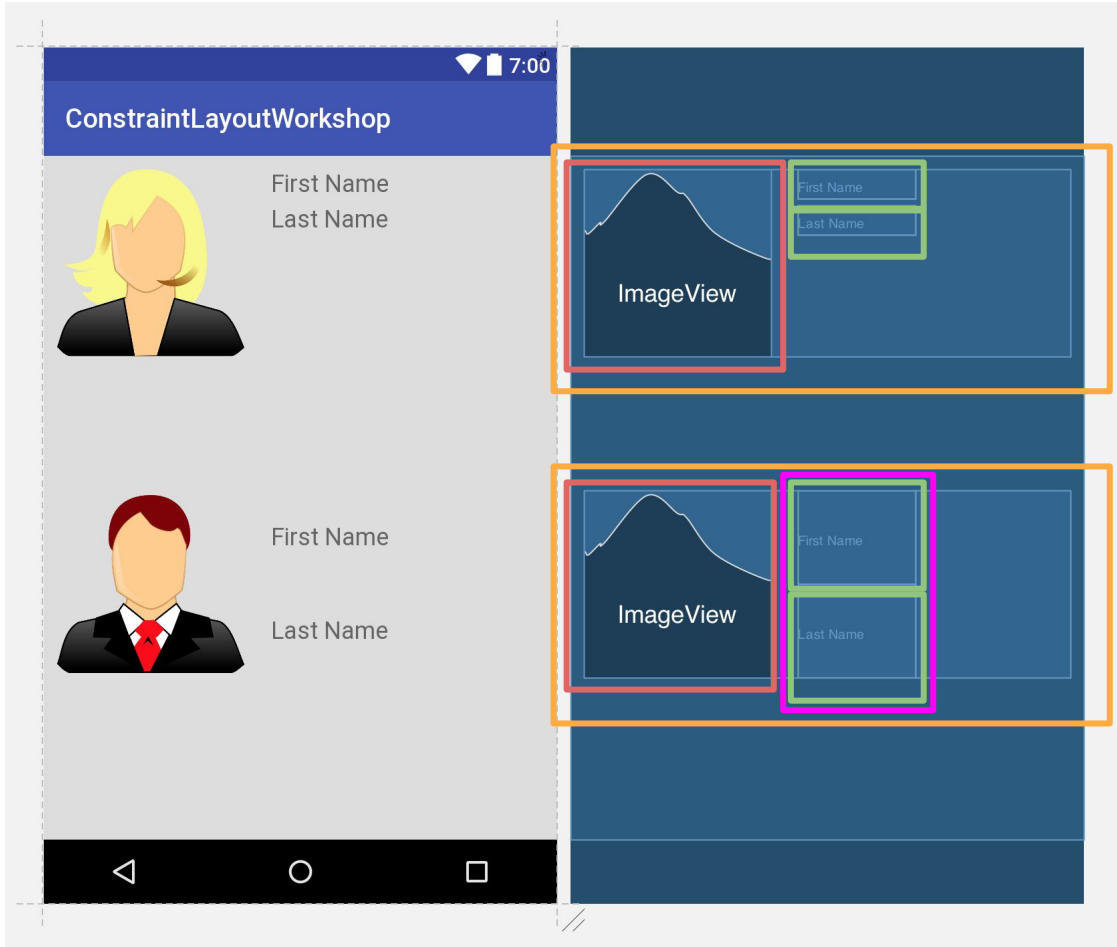
1. Unbundle from Android framework
2. Reduce layout nesting
3. First class support within Android Studio

1. Unbundled library

- Faster development and distribution
 - up to developers not phone manufacturers
 - distributed just like other *support* libraries
- `constraint-layout-1.0.2.aar`
 - ~150 referenced methods, < 40 KB disk size
- `constraint-layout-solver-1.0.2.jar`
 - ~500 referenced methods, < 100 KB disk size
- compatible with API 9 and up
 - 99.9% of Android devices

2. Reduce nesting

- traditional layouts ***are simple***
 - LinearLayout is either *horizontal* or *vertical*
 - cannot spread views equally with RelativeLayout
- simple layouts ***require nesting***
- nesting is ***bad for performance***
 - the flatter hierarchy the better



RelativeLayout

ImageView

TextView

TextView

RelativeLayout

ImageView

LinearLayout

TextView

TextView

ConstraintLayout Examples



Singapore

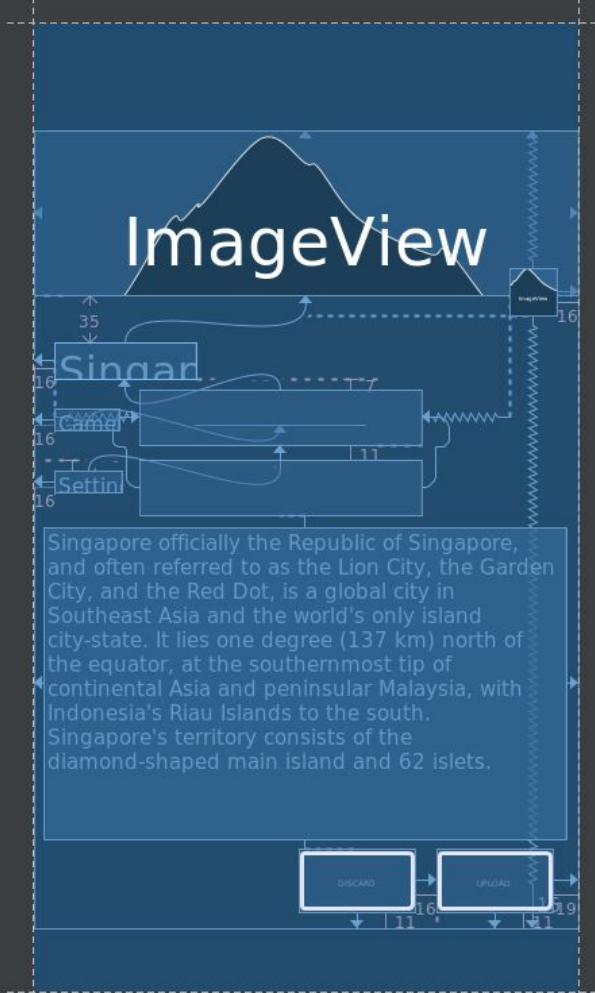
Camera Leica M Typ 240

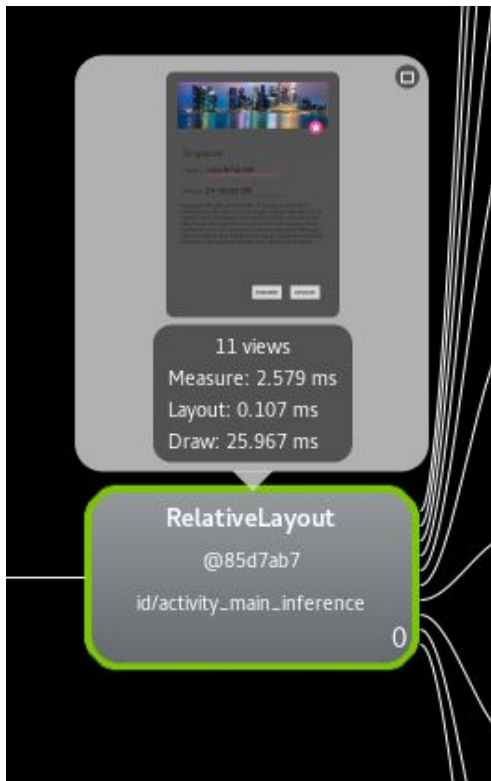
Settings f/4 16s ISO 200

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





2.579

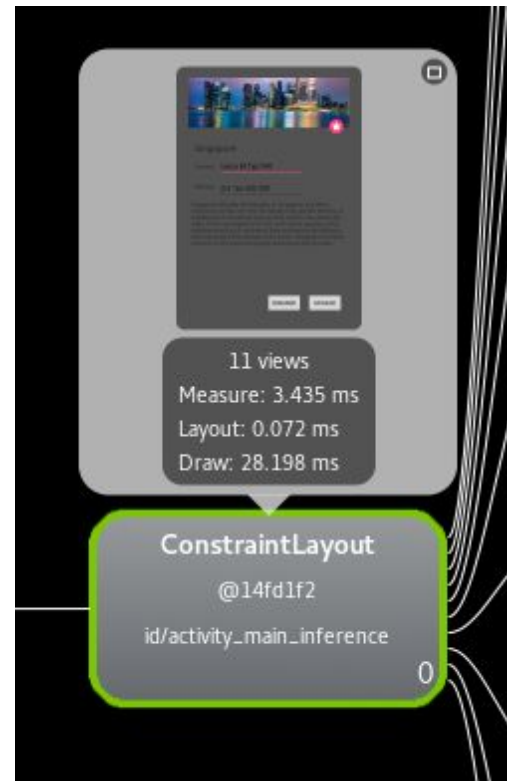
0.107

25.967

Measure

Layout

Draw



3.435

0.072

28.198

3. Better tooling

- have superb support within Android Studio
- new ***blueprint mode*** since AS 2.2 Preview
 - for both *old* layouts and new ConstraintLayout
- first time when UI editor actually is usable

activity_chain.xml - ConstraintLayoutWorkshop - [~/Workspace/pelotasplus/ConstraintLayoutWorkshop]

Project Packages Scratches trainLayout.class RecyclerView.java activity_relative_layout.xml activity_chain.xml Preview

ConstraintLayoutWorkshop ~/Workspace/pelotasplus/Co

- gradle
- idea
- build
- gradle
- mobile
 - build
 - libs
 - src
 - androidTest
 - main
 - java
 - pl.pelotasplus.constraintlayoutworkshop
 - ChainActivity
 - MainActivity
 - RelativeLayoutActivity
 - res
 - drawable
 - drawable-xxxhdpi
 - avatar.png
 - avatar2.png
 - layout
 - activity_chain.xml
 - activity_chain_constraint.xml
 - activity_main.xml
 - activity_relative_layout.xml
 - miplmap-hdpi
 - miplmap-mdpi
 - miplmap-xhdpi
 - miplmap-xxhdpi
 - miplmap-xxxhdpi
 - values
 - AndroidManifest.xml
 - test
 - .gitignore
 - build.gradle
 - mobile.iml
 - proguard-rules.pro
- tv
 - .gitignore
 - build.gradle
 - ConstraintLayoutWorkshop.iml
 - gradle.properties
 - gradlew
 - gradlew.bat
 - local.properties
 - settings.gradle
- External Libraries

LinearLayout RelativeLayout TextView

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:background="@drawable/avator"
7   android:orientation="vertical"
8   android:padding="10dp"
9   tools:context="pl.pelotasplus.constraintlayoutworkshop.ChainActivity">
10
11
12   <RelativeLayout
13     android:layout_width="match_parent"
14     android:layout_height="wrap_content">
15
16     <ImageView
17       android:id="@+id/avatar"
18       android:layout_width="140dp"
19       android:layout_height="140dp"
20       android:src="@drawable/avator" />
21
22     <TextView
23       android:id="@+id/first_name"
24       android:layout_width="wrap_content"
25       android:layout_height="wrap_content"
26       android:layout_marginStart="20dp"
27       android:layout_toRightOf="@+id/avatar"
28       android:text="First Name"
29       android:textAppearance="@style/TextAppearance.AppCompat.Medium" />
30
31     <TextView
32       android:id="@+id/last_name"
33       android:layout_width="wrap_content"
34       android:layout_height="wrap_content"
35       android:layout_alignStart="@+id/first_name"
36       android:layout_below="@+id/first_name"
37       android:layout_marginTop="5dp"
38       android:layout_toRightOf="@+id/avatar"
39       android:text="Last Name"
40       android:textAppearance="@style/TextAppearance.AppCompat.Medium" />
41   </RelativeLayout>
42
43   <RelativeLayout
44     android:layout_width="match_parent"
45     android:layout_height="wrap_content"
46     android:layout_marginTop="100dp">
47
48     <ImageView
49       android:id="@+id/avatar2"
50       android:layout_width="140dp"
51       android:layout_height="140dp"
52       android:src="@drawable/avatar2" />
53
54     <LinearLayout
55       android:layout_width="match_parent"
56       android:layout_height="match_parent"
57       android:layout_alignBottom="@+id/avatar2"
58       android:layout_marginStart="20dp"
59       android:layout_toRightOf="@+id/avatar2"
60       android:orientation="vertical">
61
62       <TextView
63         android:id="@+id/first_name2"
64         android:layout_width="wrap_content"
65         android:layout_height="80dp"

```

Design Text

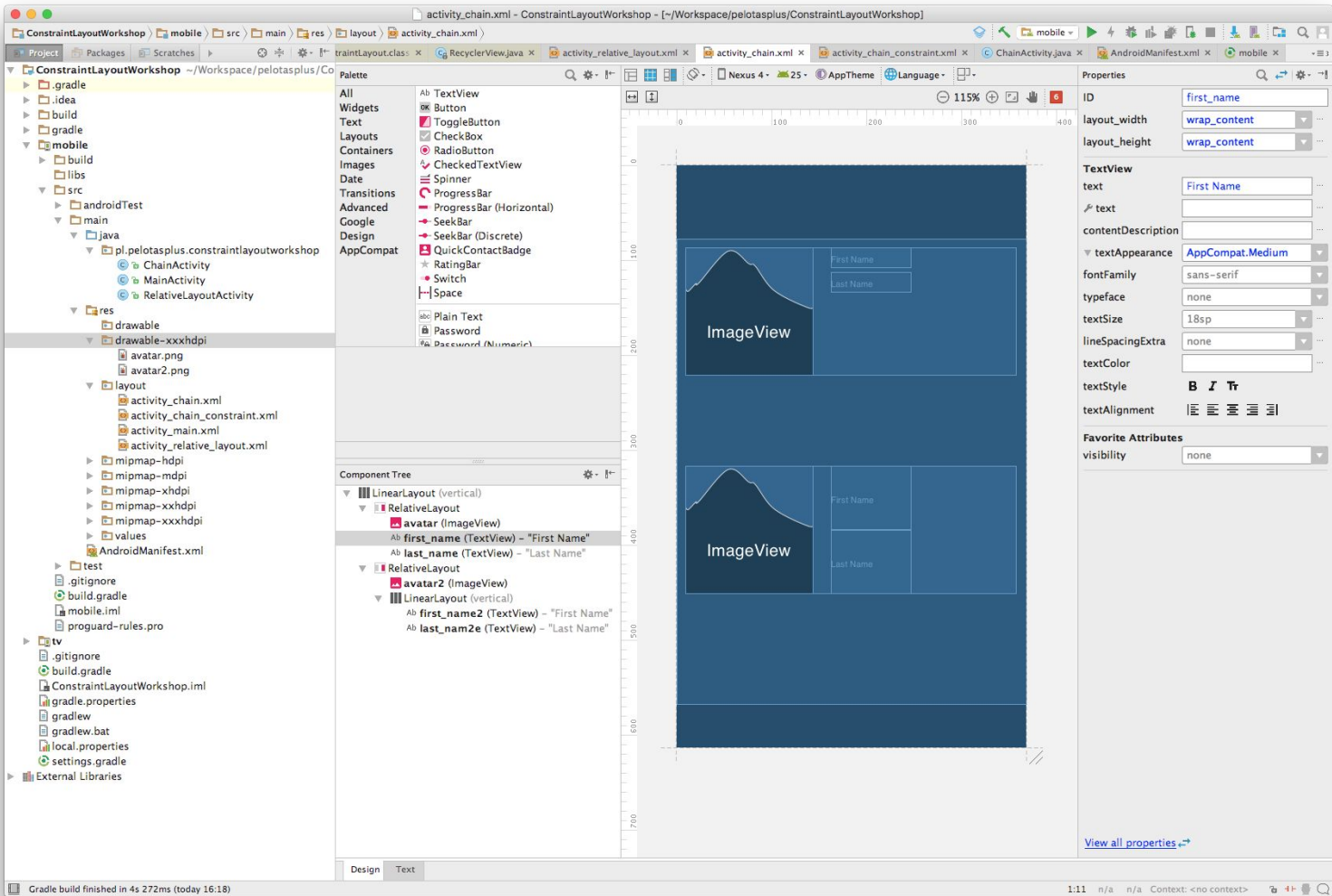
ConstraintLayoutWorkshop

First Name
Last Name

First Name
Last Name

7:00

30:18 LF UTF-8 Context: <no context>



activity_chain.xml - ConstraintLayoutWorkshop - [~/Workspace/pelotasplus/ConstraintLayoutWorkshop]

ConstraintLayoutWorkshop | mobile | src | main | res | layout | activity_chain.xml

activity_main.xml x activity_dimension.xml x DimensionActivity.java x BiasActivity.java x mobile/.../MainActivity.java x activity_chain.xml x activity_chain_constraint.xml x ChainActivity.java x tv/.../MainActivity.java x

Palette

- All
- Widgets
- Text
 - Button
 - ToggleButton
 - CheckBox
 - RadioButton
 - CheckedTextView
- Containers
- Images
- Date
- Transitions
- Advanced
- Google
- Design
- AppCompat

Component Tree

- LinearLayout (vertical)
 - RelativeLayout
 - avatar (ImageView)
 - first_name (TextView) - "First Name"
 - last_name (TextView) - "Last Name"
 - RelativeLayout
 - avatar2 (ImageView)
 - LinearLayout (vertical)
 - first_name2 (TextView) - "First Name"
 - last_name2e (TextView) - "Last Name"
 - ConstraintLayout
 - avatar3 (ImageView)
 - first_name3 (TextView) - "First Name"
 - last_name3 (TextView) - "Last Name"

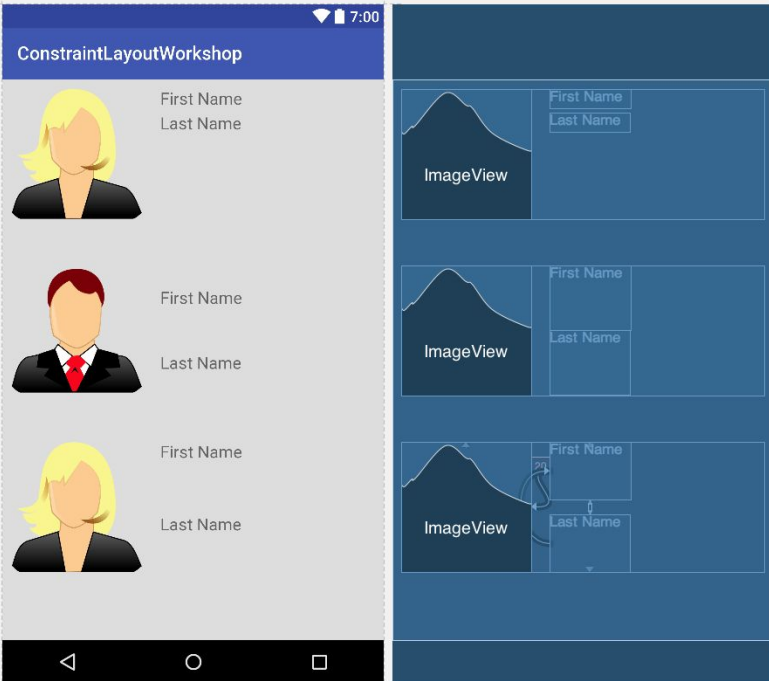
Properties

 - ID
 - layout_width: match_parent
 - layout_height: match_parent
 - LinearLayout orientation: vertical
 - Favorite Attributes
 - layout_constraint...
 - style
 - visibility: none

Design | Text

Gradle build finished in 1s 172ms (yesterday 18:35)

n/a n/a Context: <no context>



Constraints

Constraint

- a connection
- between *View* and another *View*
- or to *View* and *parent*
(ConstraintLayout)
- optional margin to create a gap
- ... can be done with
RelativeLayout



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="pl.pelotasplus.constraintlayoutworkshop.MainActivity"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="81dp">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="50dp"
        android:text="First Name"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textSize="40sp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="0dp"
        android:layout_marginTop="50dp"
        android:text="LastName"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textSize="40sp"
        app:layout_constraintLeft_toLeftOf="@+id/textView2"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />

</android.support.constraint.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="pl.pelotasplus.constraintlayoutworkshop.MainActivity"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="81dp">
```

<TextView

```
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="50dp"
    android:text="First Name"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="40sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginTop="50dp"
    android:text="LastName"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="40sp"
    app:layout_constraintLeft_toLeftOf="@+id/textView2"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

</android.support.constraint.ConstraintLayout>


```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="pl.pelotasplus.constraintlayoutworkshop.MainActivity"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="81dp">
```

```
<TextView
```

```
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="50dp"
    android:text="First Name"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="40sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginTop="50dp"
    android:text="LastName"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="40sp"
    app:layout_constraintLeft_toLeftOf="@+id/textView2"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

```
</android.support.constraint.ConstraintLayout>
```

How to create constraints?

- manually
- in the XML or even in the Java code
- auto infer
 - works most of the time
 - don't break already existing constraints
 - don't move widgets around
- autoconnect
 - per widget basis
- select few widgets and group them

Sizing

Size types

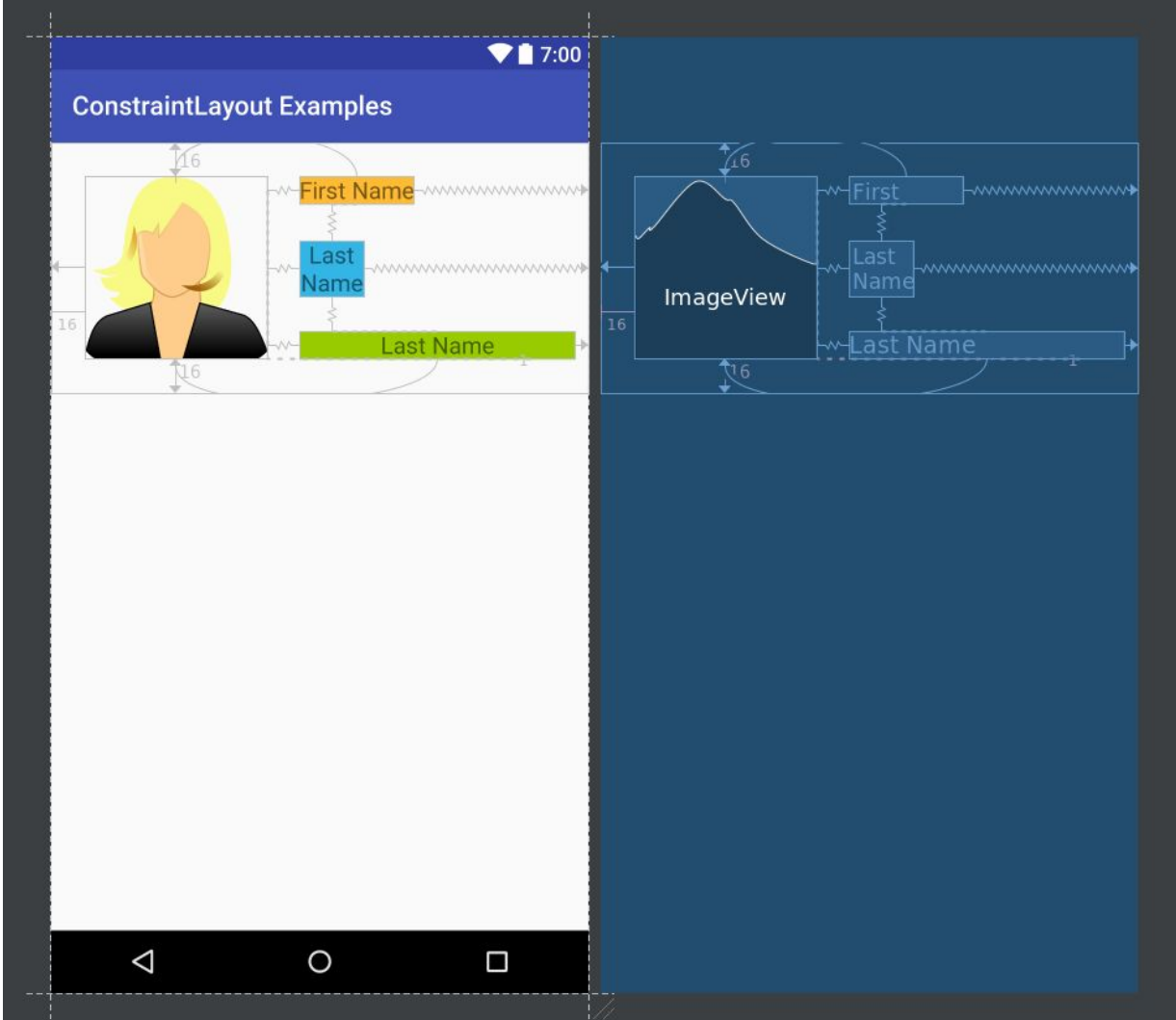
classic layouts

- wrap_content
- fixed size 72dp
- match_parent

ConstraintLayout

- wrap_content
- fixed size 72dp
- 0dp *aka* "match_constraint"

`match_parent`

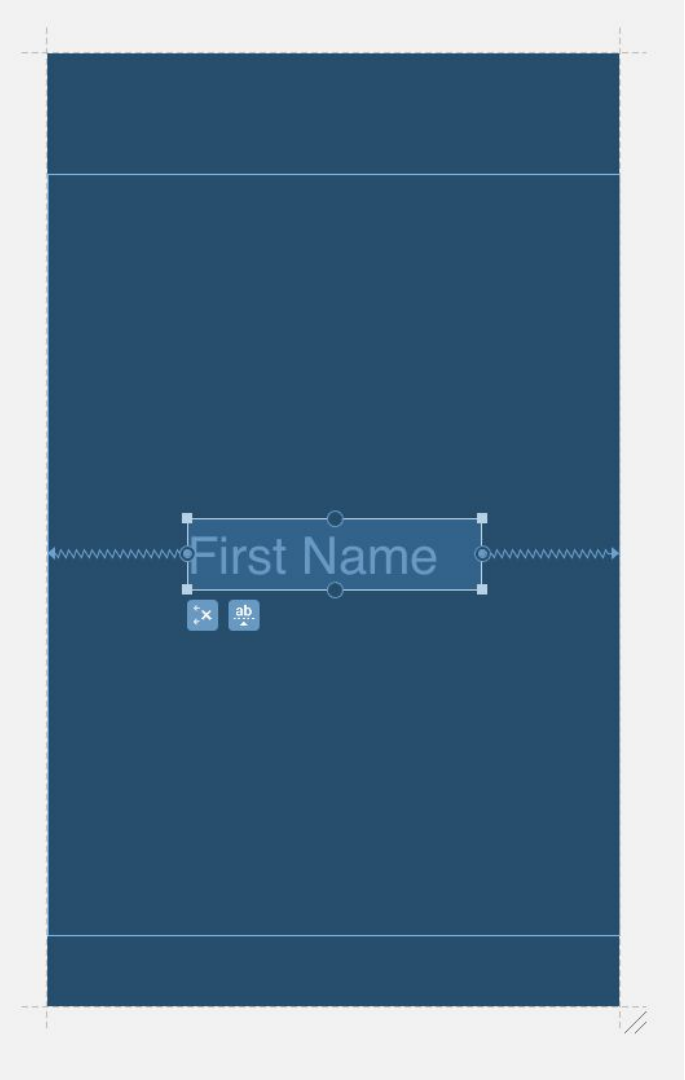


`0dp`
aka
match constraint

Fully constraint w/ bias

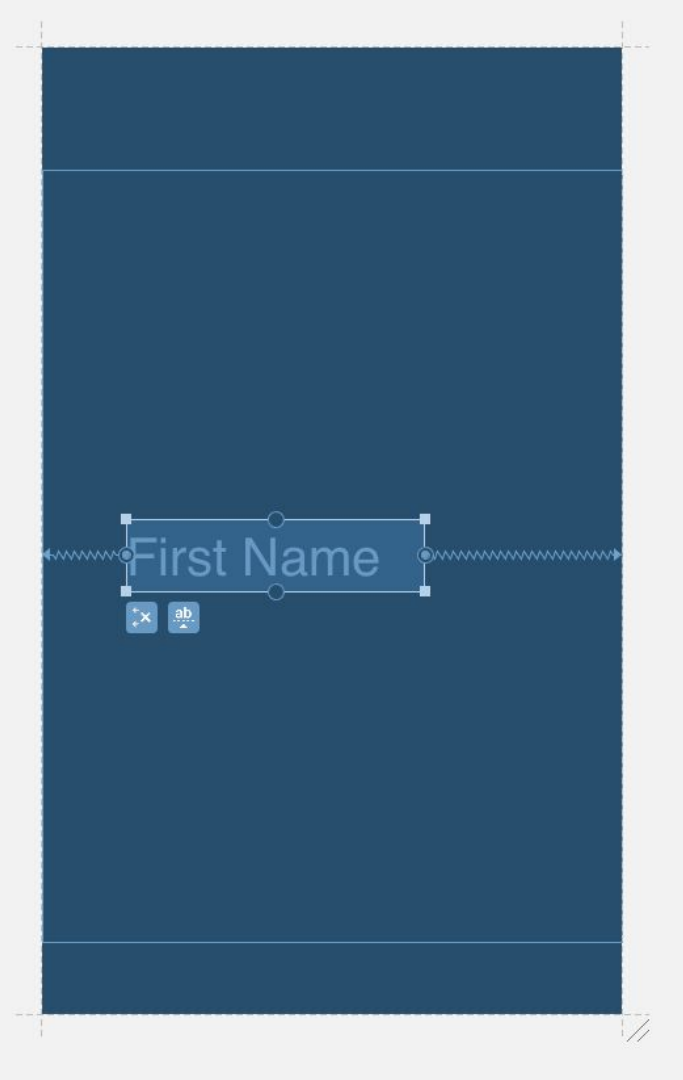
Fully constrained

- has constraints from both sides
- centered by default



Fully constrained

- has constraints from both sides
- centered by default
- drag view or bias slider to change bias
- ... actually can be done with `PercentRelativeLayout`




```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="pl.pelotasplus.constraintlayoutworkshop.BiasActivity">
```

<TextView

```
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:text="First Name"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="40sp"
    app:layout_constraintHorizontal_bias="0.301"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    tools:layout_editor_absoluteY="231dp" />
```

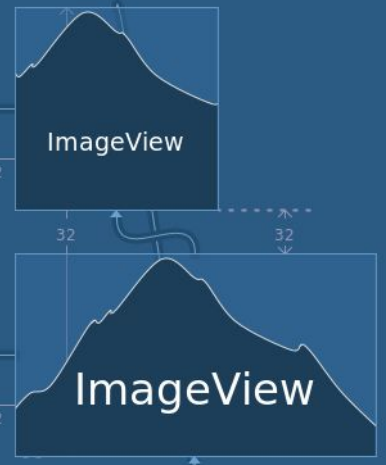
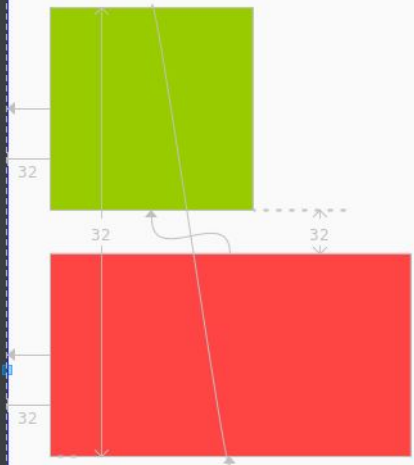
```
</android.support.constraint.ConstraintLayout>
```

Dimension ratio

Dimension ratio

- useful for `ImageView` but can be used with all kinds of `Views`
- one size locked other flexible,
or both can be flexible
- ratio provided as W:H (16:9)
- ... actually can be done with `PercentRelativeLayout`

ConstraintLayout Examples



```
private static class Image {
    String url;
    String dimension;

    Image(String url, String dimension) {
        this.url = url;
        this.dimension = dimension;
    }
}

public static List<Image> getImages() {
    List<Image> ret = new ArrayList<>();

    ret.add(new Image(
        "https://example.com/file0.png",
        "1:1"
    ));
    ret.add(new Image(
        "https://example.com/file1.png",
        "16:9"
    ));
    return ret;
}
```

```
private static class Image {
    String url;
    String dimension;

    Image(String url, String dimension) {
        this.url = url;
        this.dimension = dimension;
    }
}

public static List<Image> getImages() {
    List<Image> ret = new ArrayList<>();

    ret.add(new Image(
        "https://example.com/file0.png",
        "1:1"
    ));
    ret.add(new Image(
        "https://example.com/file1.png",
        "16:9"
    ));
    return ret;
}
```

`@Override`

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    LayoutDimensions2Binding binding =  
        DataBindingUtil.setContentView(this, R.layout.layout_dimensions2);  
  
    loadImage(binding.image1, getImages().get(1));  
  
    loadImage(binding.image2, getImages().get(0));  
}
```

```
private void loadImage(ImageView imageView, Image image) {  
    ConstraintLayout.LayoutParams layoutParams =  
        (ConstraintLayout.LayoutParams) imageView.getLayoutParams();  
  
    layoutParams.dimensionRatio = image.dimension;  
  
    Glide.with(this).load(image.url).into(imageView);  
}
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    LayoutDimensions2Binding binding =
        DataBindingUtil.setContentView(this, R.layout.Layout_dimensions2);

    loadImage(binding.image1, getImages().get(1));

    loadImage(binding.image2, getImages().get(0));
}
```

```
private void loadImage(ImageView imageView, Image image) {
    ConstraintLayout.LayoutParams layoutParams =
        (ConstraintLayout.LayoutParams) imageView.getLayoutParams();

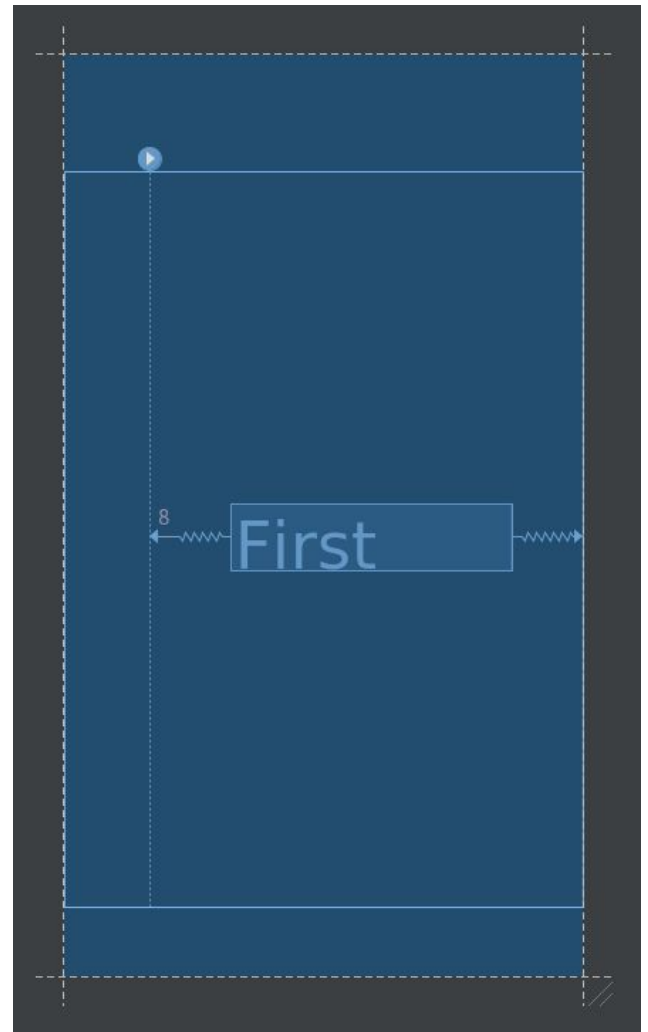
    layoutParams.dimensionRatio = image.dimension; // "16:9" or "1:1"

    Glide.with(this).load(image.url).into(imageView);
}
```


Guidelines

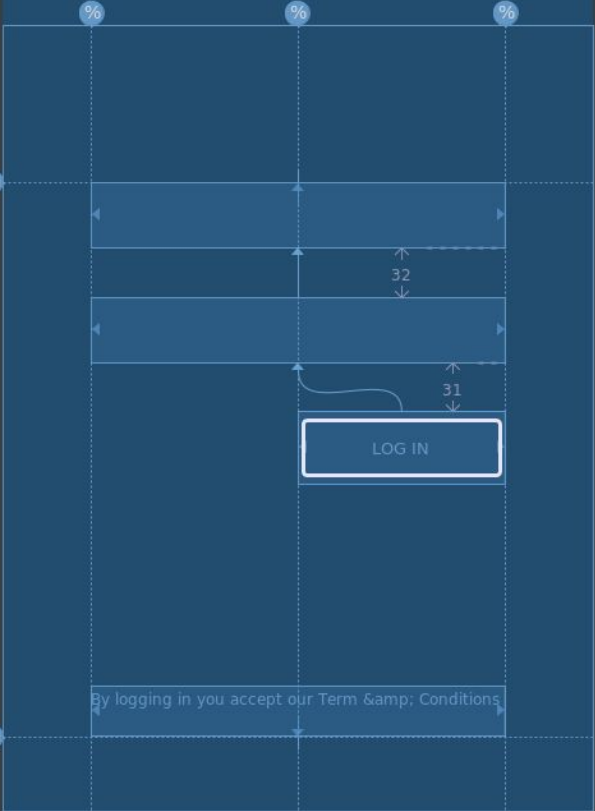
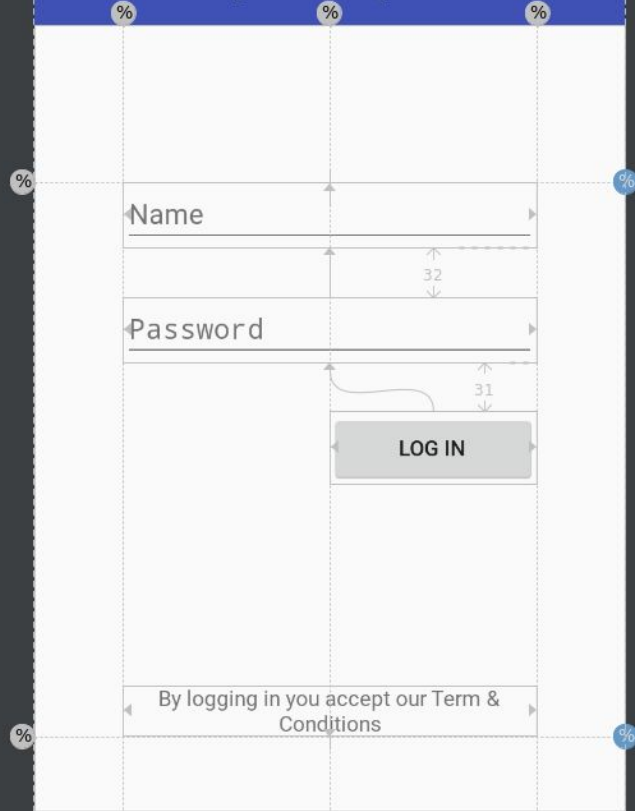
Guidelines

- a new View
- used to generate new equations
- ... but not when layouting or drawing
- have a relation to parent
- can be in 40dp or 40%



ConstraintLayout Examples

7:00



```
<android.support.constraint.ConstraintLayout
```

```
...>
```

```
<android.support.constraint.Guideline
```

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

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:orientation="vertical"
```

```
    app:layout_constraintGuide_percent="0.2" />
```

```
<ImageView
```

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

```
    android:layout_width="219dp"
```

```
    android:layout_height="183dp"
```

```
    android:layout_marginLeft="64dp"
```

```
    android:src="@drawable/avatar"
```

```
    app:layout_constraintLeft_toLeftOf="@+id/guideline"
```

```
    tools:layout_editor_absoluteY="76dp" />
```

```
...
```

```
</android.support.constraint.ConstraintLayout>
```

```
<android.support.constraint.ConstraintLayout
    ...>

    <android.support.constraint.Guideline
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.2" />

    <ImageView
        android:id="@+id/avatar"
        android:layout_width="219dp"
        android:layout_height="183dp"
        android:layout_marginLeft="64dp"
        android:src="@drawable/avatar"
        app:layout_constraintLeft_toLeftOf="@+id/guideline"
        tools:layout_editor_absoluteY="76dp" />

    ...

</android.support.constraint.ConstraintLayout>
```

```
<android.support.constraint.ConstraintLayout
    ...>

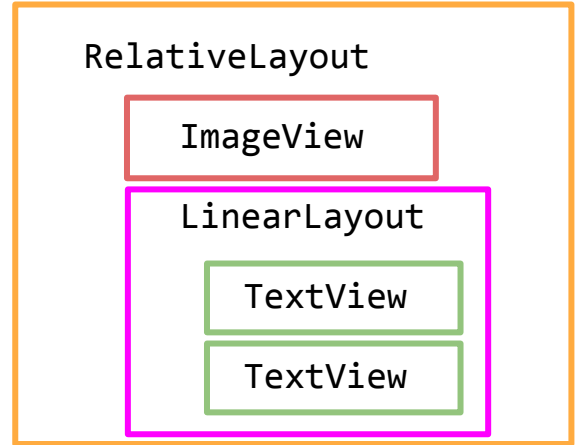
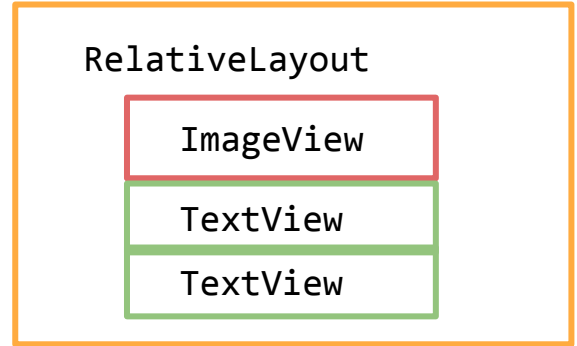
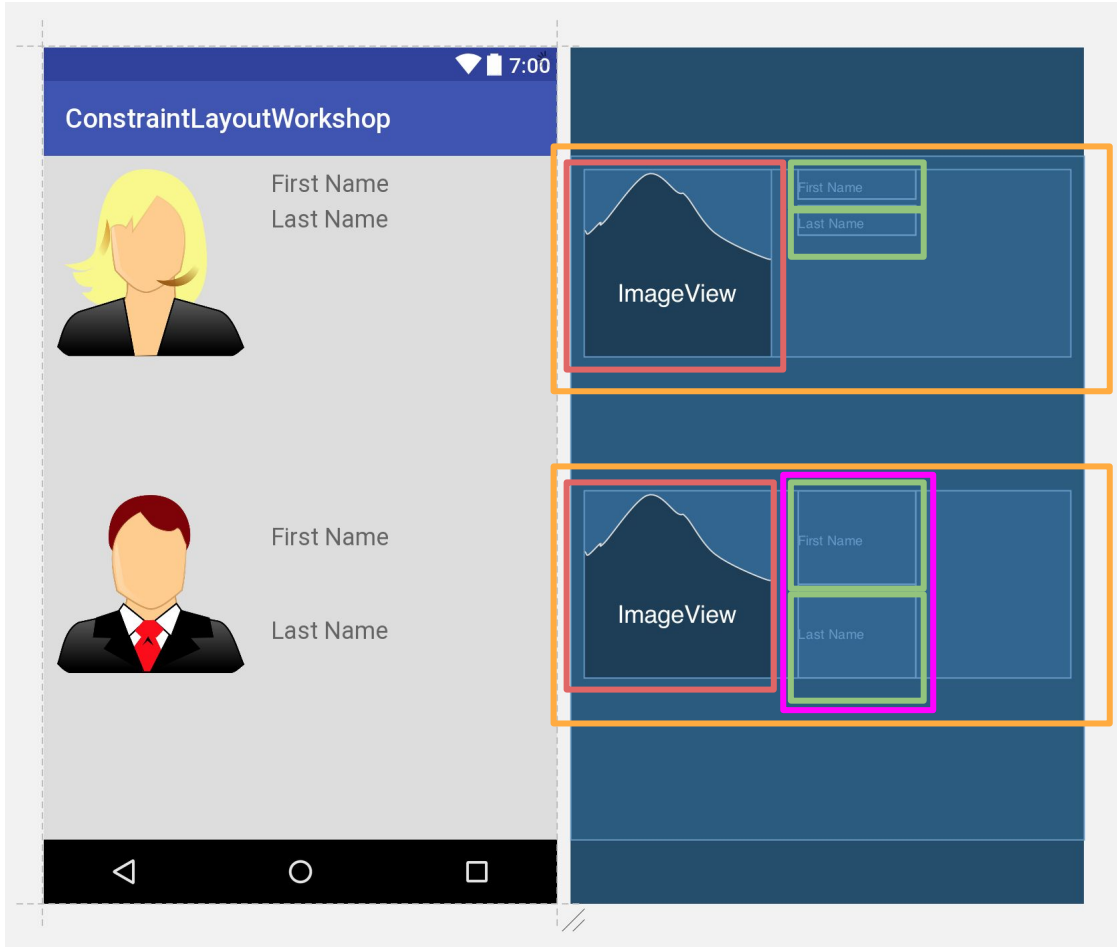
    <android.support.constraint.Guideline
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.2" />

    <ImageView
        android:id="@+id/avatar"
        android:layout_width="219dp"
        android:layout_height="183dp"
        android:layout_marginLeft="64dp"
        android:src="@drawable/avatar"
        app:layout_constraintLeft_toLeftOf="@+id/guideline"
        tools:layout_editor_absoluteY="76dp" />

    ...

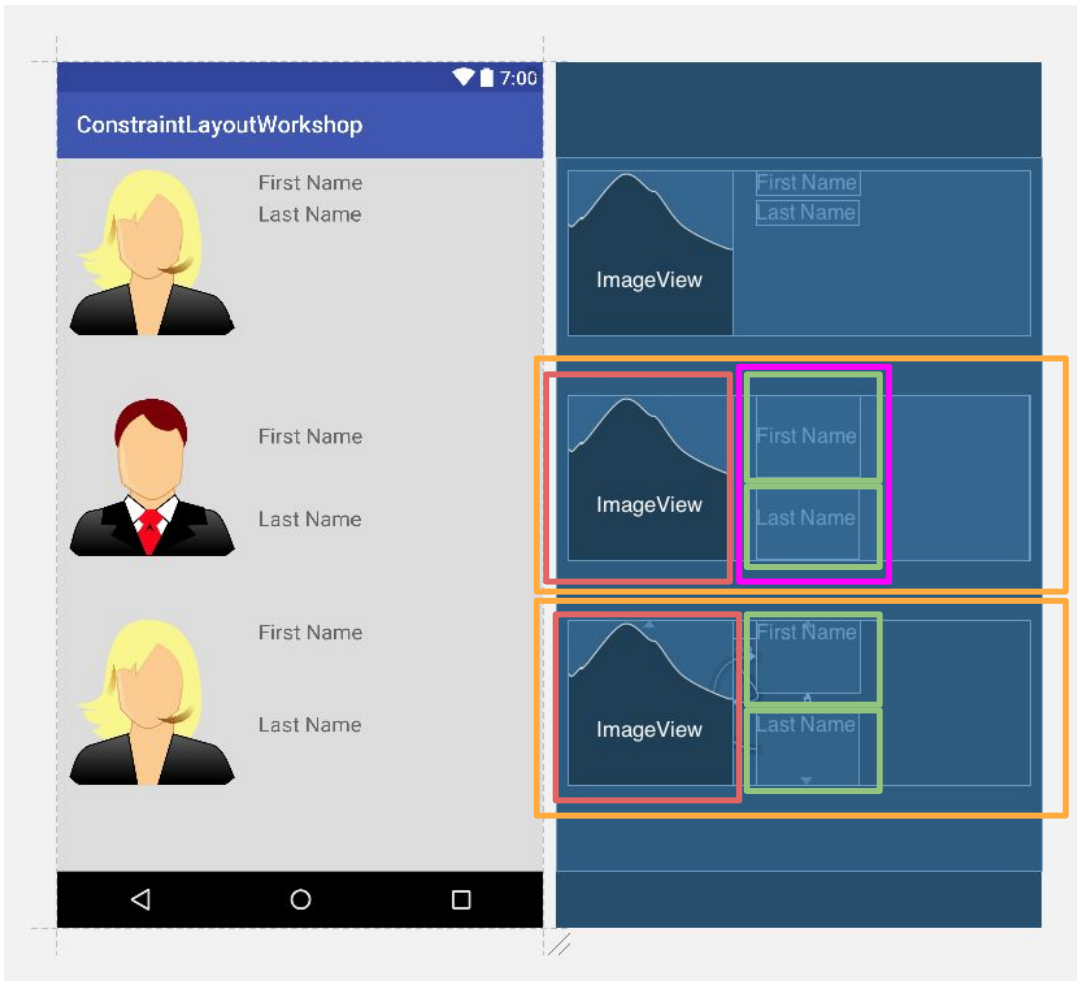
</android.support.constraint.ConstraintLayout>
```

Chains



Chains

- a **way of grouping** of *Views* within *ConstraintLayout*
- defined by **constraining two** or more **views to each other**
 - A constrained with B and B constrained with A
- allows to **apply “some magic” to whole group** of views
 - *aka* create a virtual view group
- ... make *ConstraintLayout* **act as *LinearLayout***



RelativeLayout

ImageView

LinearLayout

TextView

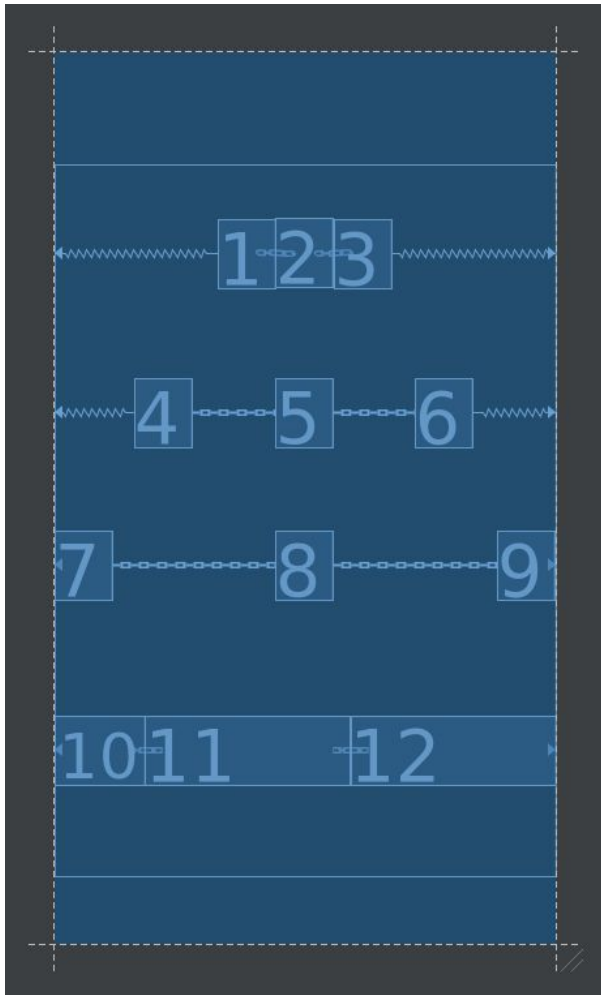
TextView

ConstraintLayout

ImageView

TextView

TextView



packed

spread

spread inside

spread/spread inside
with weights

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout ...>

    <TextView
        android:id="@+id/view1"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:text="1"
        app:layout_constraintHorizontal_chainStyle="packed"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toLeftOf="@+id/view2"
        tools:layout_editor_absoluteY="39dp" />

    <TextView
        android:id="@+id/view2"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:text="2"
        app:layout_constraintLeft_toRightOf="@+id/view1"
        app:layout_constraintRight_toLeftOf="@+id/view3"
        tools:layout_editor_absoluteY="38dp" />

    <TextView
        android:id="@+id/view3"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:text="3"
        app:layout_constraintLeft_toRightOf="@+id/view2"
        app:layout_constraintRight_toRightOf="parent"
        tools:layout_editor_absoluteY="39dp" />

</android.support.constraint.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout ...>

    <TextView
        android:id="@+id/view1"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:text="1"
        app:layout_constraintHorizontal_chainStyle="packed"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toLeftOf="@+id/view2"
        tools:layout_editor_absoluteY="39dp" />

    <TextView
        android:id="@+id/view2"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:text="2"
        app:layout_constraintLeft_toRightOf="@+id/view1"
        app:layout_constraintRight_toLeftOf="@+id/view3"
        tools:layout_editor_absoluteY="38dp" />

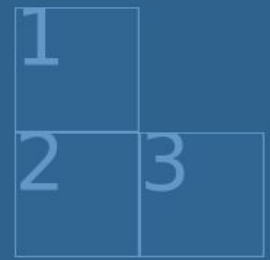
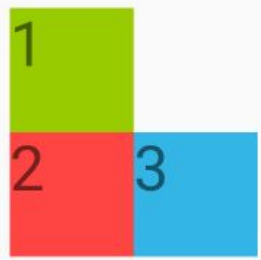
    <TextView
        android:id="@+id/view3"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:text="3"
        app:layout_constraintLeft_toRightOf="@+id/view2"
        app:layout_constraintRight_toRightOf="parent"
        tools:layout_editor_absoluteY="39dp" />

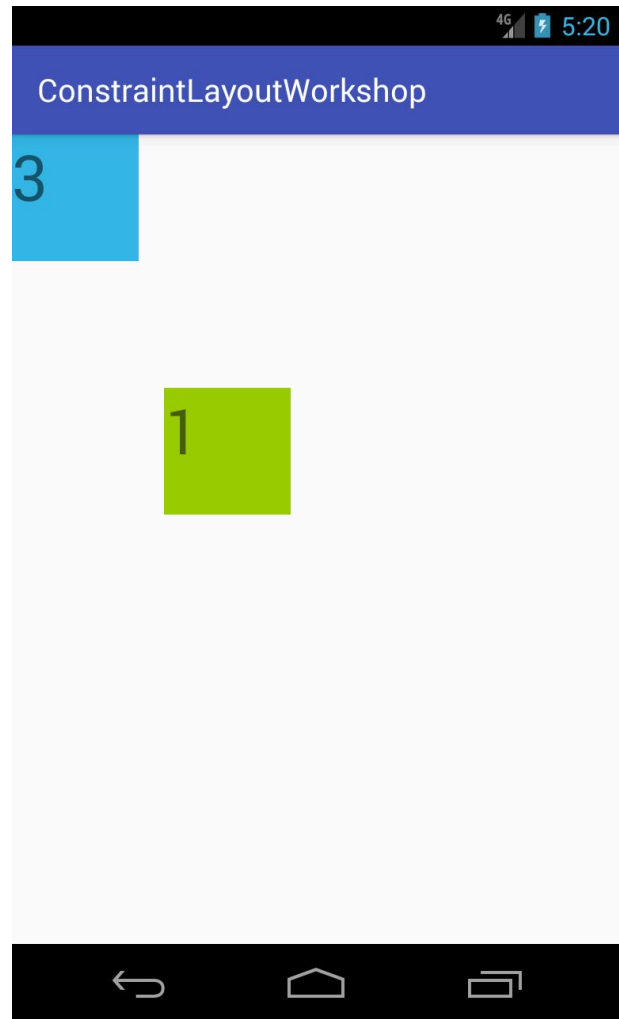
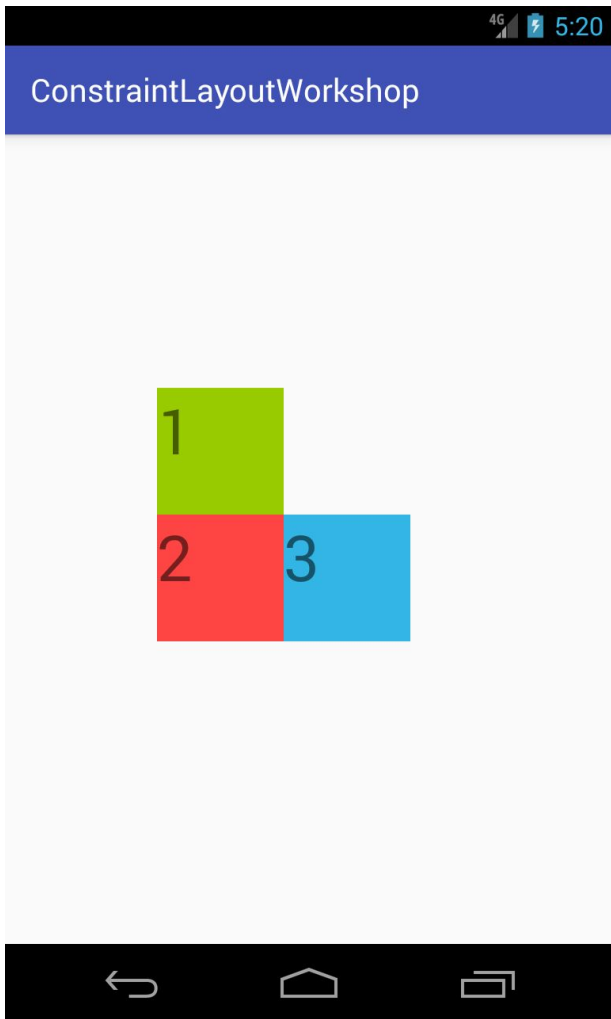
</android.support.constraint.ConstraintLayout>
```



Visibility

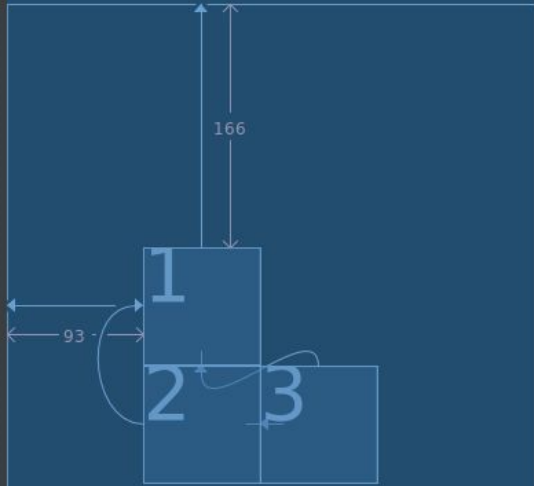
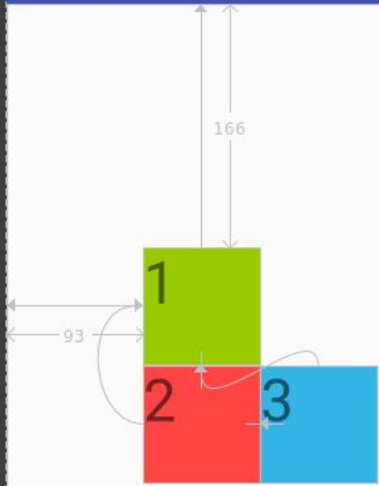
ConstraintLayoutWorkshop

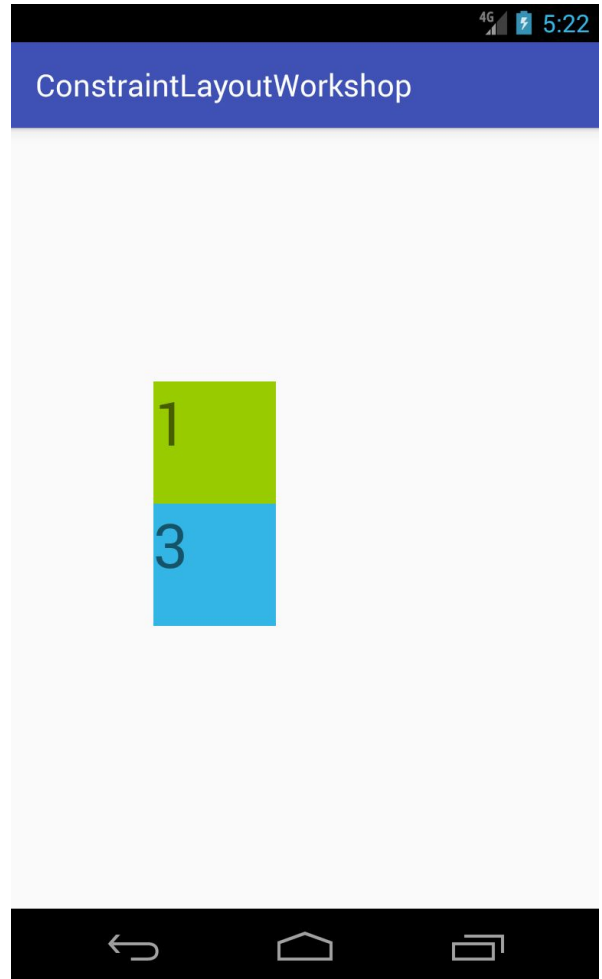
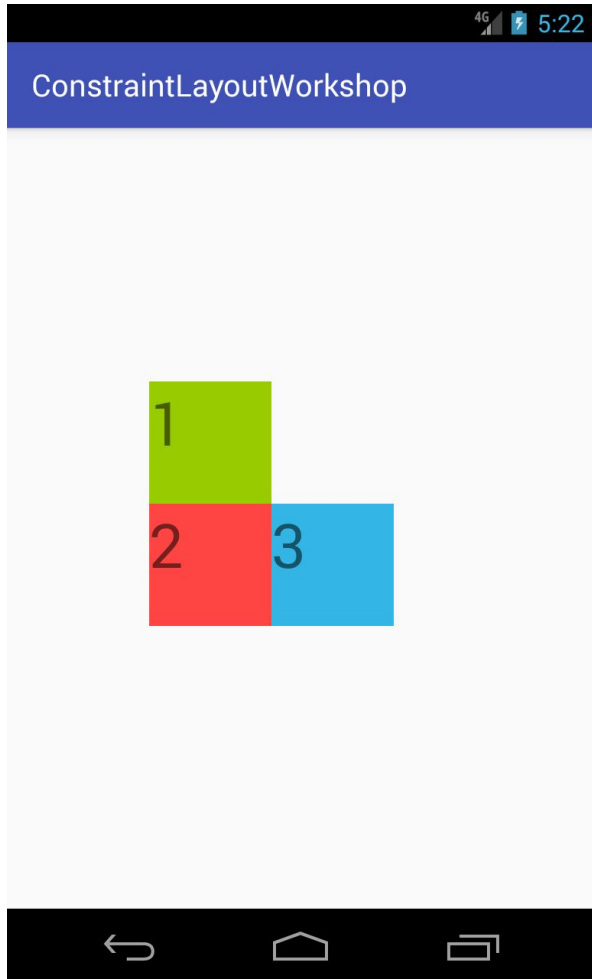




ConstraintLayoutWorkshop

7:00





Visibility

- hiding Views doesn't work well with `RelativeLayout`
- in `ConstraintLayout` a View that is `GONE` is mapped to a single point
- so has no size and no margin
- but still its constraints still apply when doing calculations

ConstraintSet

ConstraintSet

- pack many constraints into a set
 - can later be applied to
ConstraintLayout
 - make fancy animations with
help of TransitionManager
 - from code
 - from XML file
 - from other ConstraintSet
- API 19



Jan III

Sobieski

Lorem ipsum dolor sit amet, consetetur adipiscing elit. Mauris id neque interdum, ultrices mi et, feugiat velit. In at cursus nulla. Nam at faucibus nisi. Morbi purus nisi, venenatis eu arcu vitae.

LESS

Jan III



Jan III Sobieski

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id neque interdum, ultrices mi et, feugiat velit. In at cursus nulla. Nam at faucibus nisi. Morbi purus nisi, venenatis eu arcu vitae,

MORE

ImageView

Jan

Sobieski

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id neque interdum, ultrices mi et, feugiat velit. In at cursus nulla. Nam at faucibus nisi. Morbi

MORE

ConstraintLayoutWorkshop



Jan III
Sobieski

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id neque interdum, ultrices mi et, feugiat velit. In at cursus nulla. Nam at faucibus nisi. Morbi purus nisi, venenatis eu arcu vitae, ullamcorper lacinia felis. Vestibulum dapibus velit quis ornare mattis. Etiam sapien mi, finibus vestibulum massa non, feugiat pulvinar quam. Vestibulum luctus ligula sed massa efficitur, eu auctor lectus venenatis. Vestibulum auctor erat sit amet orci pharetra, eget sodales velit laoreet. In interdum, est sit amet euismod tempus, nulla velit egestas mauris, a lacinia lacus neque sagittis ligula. Sed ac fermentum neque, ut hendrerit massa. Quisque lorem urna, blandit id nisi vitae, facilisis ultricies quam. Aliquam sit amet justo sem. In hac habitasse platea dictumst. Ut vitae enim quis mi aliquam congue. Vestibulum neque lorem, consectetur sit amet ligula id, volutpat fermentum sapien. Donec non consequat justo. Praesent in sapien ultricies, convallis ipsum et, efficitur sapien. Suspendisse bibendum dui lorem, sed elementum dui gravida id. Praesent mattis dui ac venenatis hendrerit. Aliquam ipsum mauris, dictum scelerisque erat nec, dapibus semper nisi. Integer egestas porttitor aliquet. Curabitur hendrerit gravida nisi

LESS



ImageView

Jan
Sobieski

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id neque interdum, ultrices mi et, feugiat velit. In at cursus nulla. Nam at faucibus nisi. Morbi purus nisi, venenatis eu arcu vitae, ullamcorper lacinia felis. Vestibulum dapibus velit quis ornare mattis. Etiam sapien mi, finibus vestibulum massa non, feugiat pulvinar quam. Vestibulum luctus ligula sed massa efficitur, eu auctor lectus venenatis. Vestibulum auctor erat sit amet orci pharetra, eget sodales velit laoreet. In interdum, est sit amet euismod tempus, nulla velit egestas mauris, a lacinia lacus neque sagittis ligula. Sed ac fermentum neque, ut hendrerit massa. Quisque lorem urna, blandit id nisi vitae, facilisis ultricies quam. Aliquam sit amet justo sem. In hac habitasse platea dictumst. Ut vitae enim quis mi aliquam congue. Vestibulum neque lorem, consectetur sit amet ligula id, volutpat fermentum sapien. Donec non consequat justo. Praesent in sapien ultricies, convallis ipsum et, efficitur sapien.

LESS




```

public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition);

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}

```

```
public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition);

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}
```

```

public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

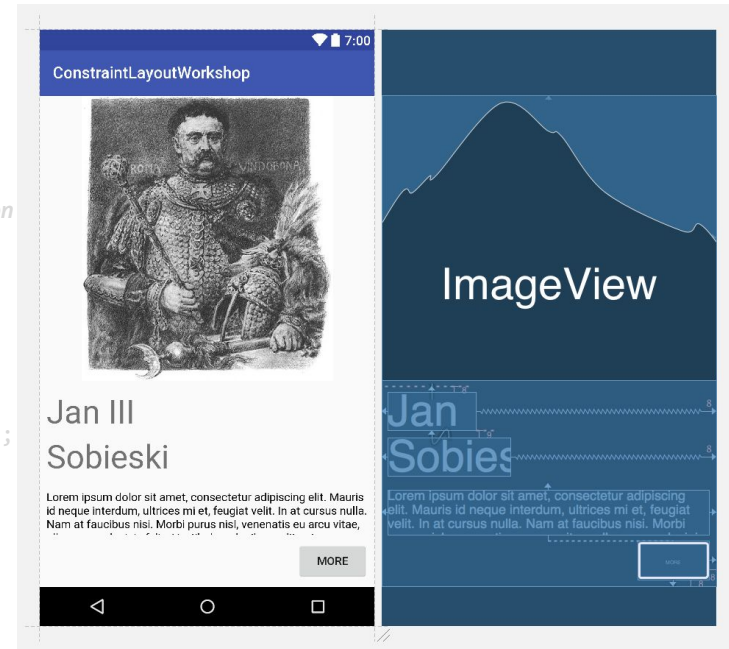
        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}

```



```

public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

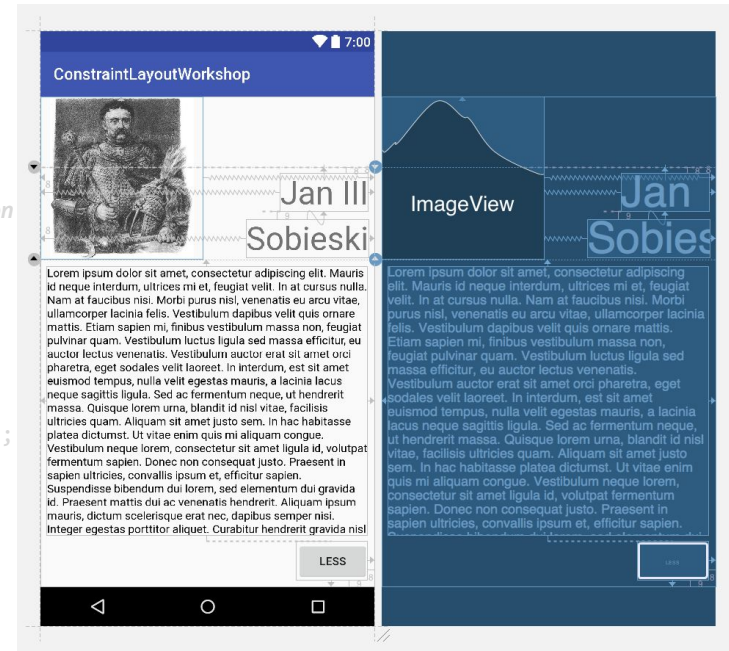
        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}

```



```

public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition);

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}

```

```

public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition);

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}

```

```

public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition);

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}

```

```

public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition);

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}

```



```

public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition);

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}

```

```
public class TransitionActivity extends AppCompatActivity {
    private boolean lessMode = true;

    private ActivityTransitionBinding binding;

    private ConstraintSet moreConstraintSet = new ConstraintSet();
    private ConstraintSet lessConstraintSet = new ConstraintSet();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = DataBindingUtil.setContentView(this, R.layout.activity_transition);

        lessConstraintSet.clone(this, R.layout.activity_transition);
        moreConstraintSet.clone(this, R.layout.activity_transition_2);

        binding.moreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TransitionManager.beginDelayedTransition(binding.constraintLayout);

                if (lessMode) {
                    moreConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = false;
                } else {
                    lessConstraintSet.applyTo(binding.constraintLayout);
                    lessMode = true;
                }
            }
        });
    }
}
```

Internals

Google wasn't first

- <https://github.com/anandsainath/constraint-layout> (Nov 2013)
- <https://github.com/alexbirkett/android-cassowary-layout> (Oct 2014)
- But first to have
 - working production version
 - superb UI editor capabilities
- Apple's AutoLayout

```
public class ConstraintLayout extends ViewGroup {  
  
    ...  
  
    public ConstraintLayout(Context context) {  
        super(context);  
        this.init((AttributeSet)null);  
    }  
  
    public ConstraintLayout(Context context, AttributeSet attrs) {  
        super(context, attrs);  
        this.init(attrs);  
    }  
  
    public ConstraintLayout(Context context, AttributeSet attrs, int defStyleAttr) {  
        super(context, attrs, defStyleAttr);  
        this.init(attrs);  
    }  
  
    ...  
}
```

RelativeLayout

LinearLayout

RecyclerView

```
public class Guideline extends View {
    public Guideline(Context context) {
        super(context);
        super.setVisibility(View.GONE);
    }

    public Guideline(Context context, AttributeSet attrs) {
        super(context, attrs);
        super.setVisibility(View.GONE);
    }

    public Guideline(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        super.setVisibility(View.GONE);
    }

    public void setVisibility(int visibility) {
    }

    public void draw(Canvas canvas) {
    }

    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
        this.setMeasuredDimension(0, 0);
    }
}
```



```
clavicle's remacs: system.scwm.stm
File Edit Apps Options Buffers Tools Help
((motion) (interactive-move))
((double-click) (animated-deiconify)))

(define* (popup-scwm-root-middle-menu)
  "Popup the root middle-button menu."
  (interactive)
  (popup-menu
   (menu
    (cons (menuitem "Scwm" #:extra-label "C-S-M-z"
                  #:image-left pic-lambda-mini #:submenu menu-root-star2
                  #:image-right pic-lambda-mini)
          (cdadr (menu-properties
                  (make-window-list-menu #:by-resource #t))))
    #:popup-delay #f
    #:hover-delay 0)))

(define* (show-window-list-by-focus-with-geometry)
```

Scwm Options

Decorations

File locations

Focus

Fvwm2 Modules

Gnome

Menu

Message window

Preferences

System

Virtual

Window Ops

netscape

Menu

menu text color black

menu bg color gray80

menu hl fg color inherit

Font Selection Dialog

Font

Font Information

Filter

Font:

utopia

nil

open look cursor

open look glyph

song ti

symbol

terminal

times

utopia

Reset Filter

Preview:

Scwm - A Turing Complete Window Manager

GNU/Linux

- Start
- New shell
- Applications
- Utilities
- Multimedia
- Games
- Constraint solver
- Hosts - Servers
- Hosts - Linux
- Hosts - Solaris
- Hosts - Other
- Window Operations
- Windows list
- Windows by resource
- Scwm Options....
- Other Preferences
- About Scwm
- Help Scwm
- Lock Screen
- Exit Scwm

Window Ops

- Move
- Resize
- Raise
- Lower
- Iconify/Restore
- Stick/Unstick
- Focus
- Maximize/Reset
- Maximize Tall/Reset
- Maximize Wide/Reset
- Shade/Reset
- Set gravity
- Shove
- Keep-on-top/Reset
- Group
- Change theme
- Delete
- Close
- Kill
- Switch to...
- Refresh Screen

Move window

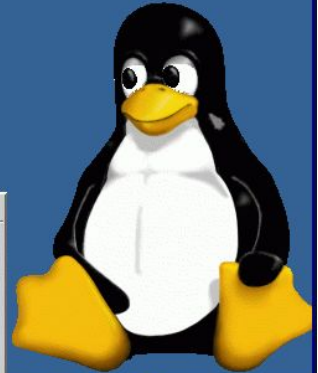
- Center
- North
- East
- South
- West
- Northeast
- Southeast
- Southwest
- Northwest

debug_resize.h	message-window,doc	window,o
decor.c	message-window,h	window,x
decor.doc	message-window,o	window_fud,h
decor.h	message-window,x	winprop,c
decor.o	miscprocs,c	winprop,doc
decor.x	miscprocs,doc	winprop,h
decorations,c	miscprocs,h	winprop,o
decorations,h	miscprocs,o	winprop,x
	miscprocs,x	xmisc,c
	module-interface,c	xmisc,h
	module-interface,doc	xmisc,o
	module-interface,h	xproperty,c
	module-interface,o	xproperty,doc
	module-interface,x	xproperty,h
	module-types,h	xproperty,o
	move.c	xproperty,x

Constraint investigator

Strict relative position: 3 windows Delete

Disable All Enable All Delete All



Taskbar with icons for various applications and system status. System tray shows 'clavicle.cc.washin' and 'Mon Feb 21 3:58'.

Cassowary

- Greg J. Badros, Alan Borning (University of Washington)
- Peter J. Stuckey (University of Melbourne)
- The Cassowary Linear Arithmetic Constraint Solving Algorithm
- <https://constraints.cs.washington.edu/solvers/cassowary-tochi.pdf>
- Cassowary Algorithm

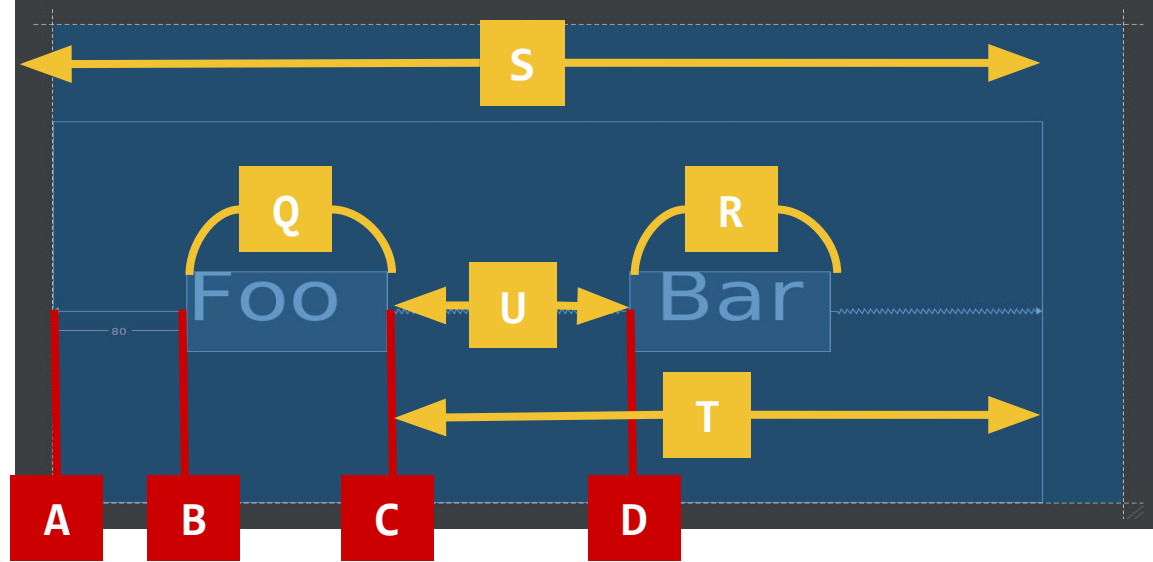
An incremental constraint solving toolkit that efficiently solves systems of linear equalities and inequalities

$$y = 2x + z$$

$$y \leq x + 30$$

$$~~y = x * z~~$$





$$A = ?$$

$$B = A + 80$$

$$C = B + Q$$

$$D = C + U$$

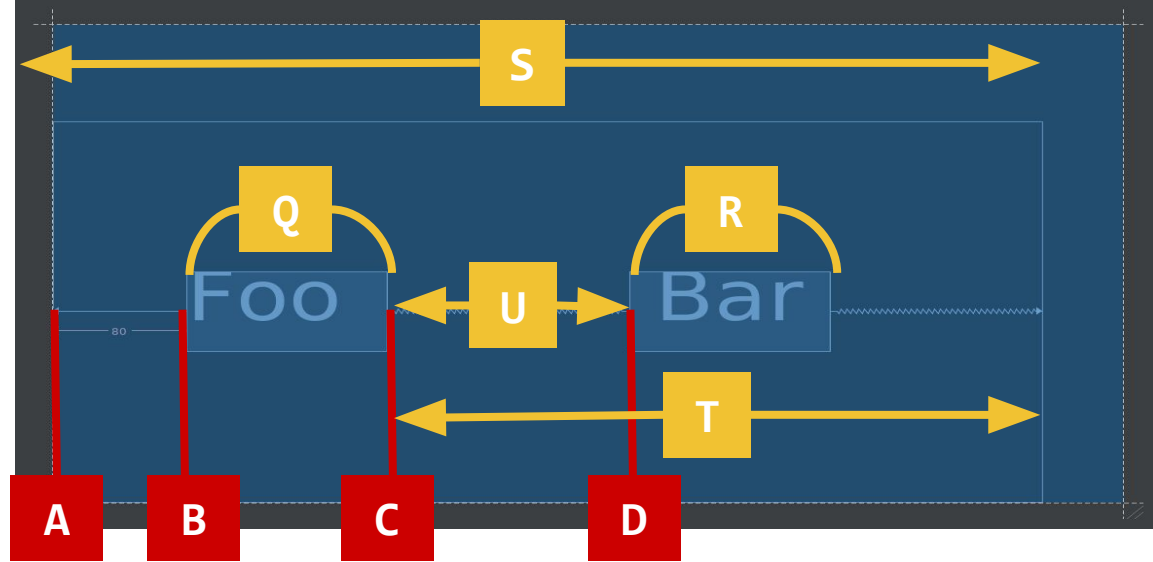
$$Q = 100$$

$$R = 100$$

$$S = 500$$

$$U = (T - R) / 2$$

$$T = S - C$$



$$A = 0$$

$$B = A + 80$$

$$C = B + Q$$

$$D = C + U$$

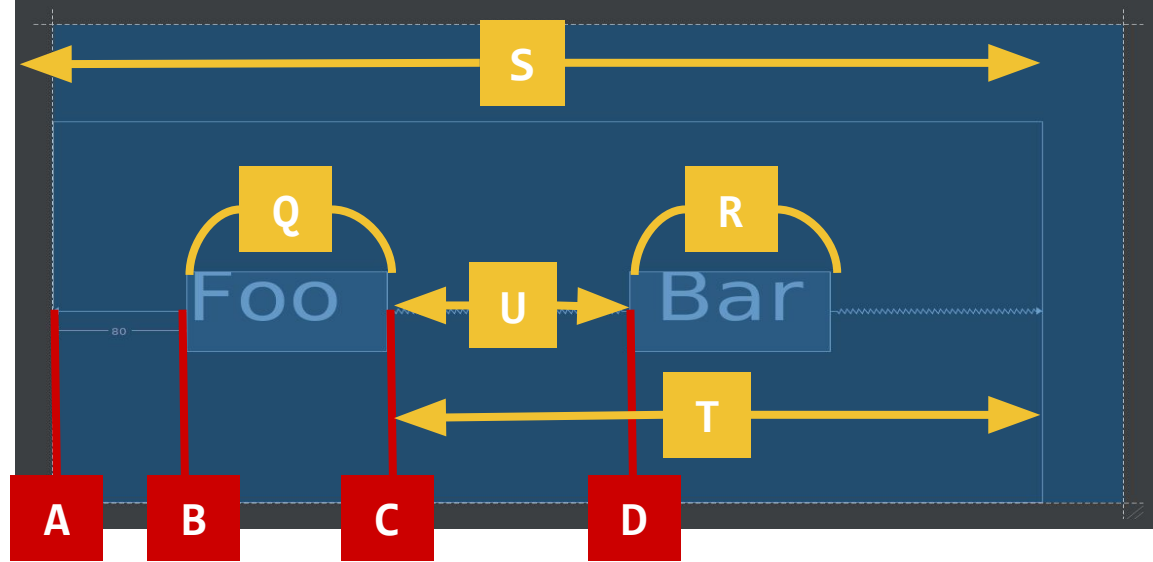
$$Q = 100$$

$$R = 100$$

$$S = 500$$

$$U = (T - R) / 2$$

$$T = S - C$$



$$A = 0$$

$$B = 0 + 80$$

$$C = B + Q$$

$$D = C + U$$

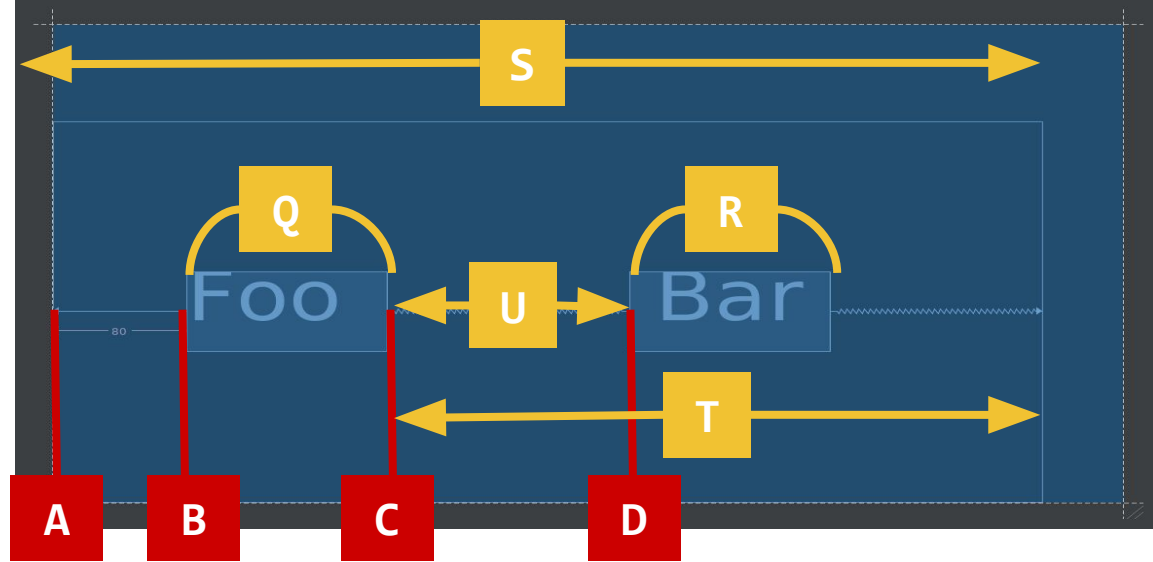
$$Q = 100$$

$$R = 100$$

$$S = 500$$

$$U = (T - R) / 2$$

$$T = S - C$$



$$A = 0$$

$$B = 0 + 80$$

$$C = 80 + 100$$

$$D = C + U$$

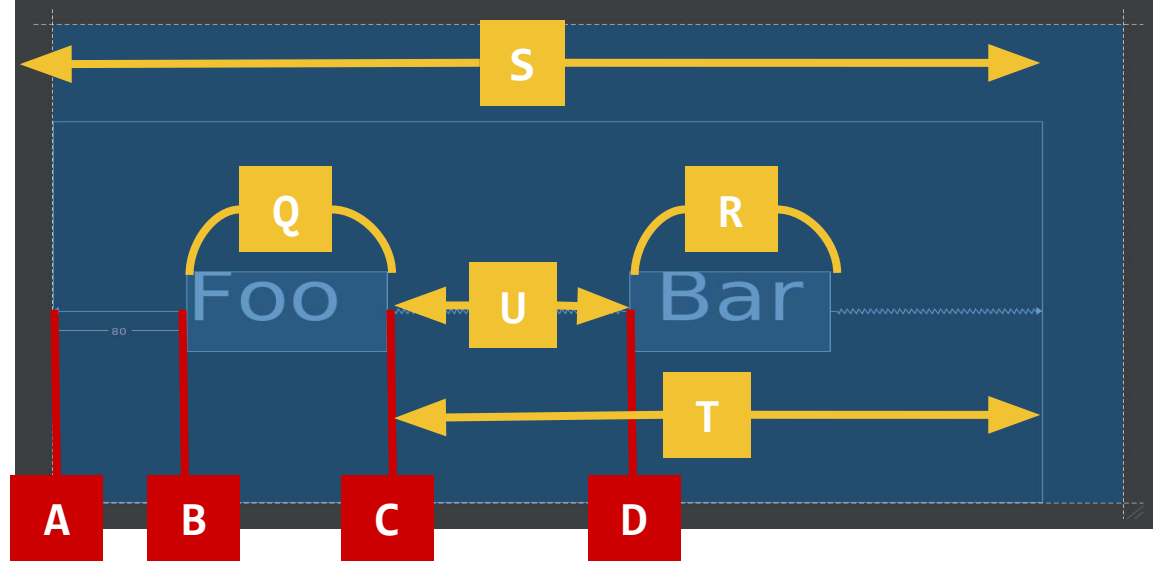
$$Q = 100$$

$$R = 100$$

$$S = 500$$

$$U = (T - R) / 2$$

$$T = S - C$$



$$A = 0$$

$$B = 0 + 80$$

$$C = 80 + 100$$

$$D = C + U$$

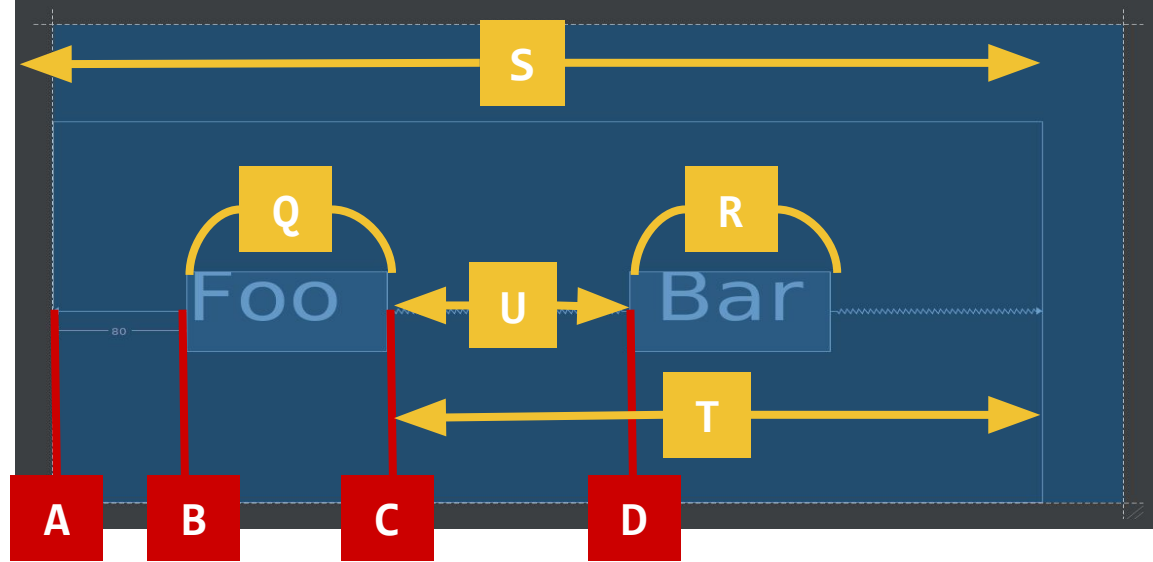
$$Q = 100$$

$$R = 100$$

$$S = 500$$

$$U = (T - R) / 2$$

$$T = 500 - 180$$



$$A = 0$$

$$B = 0 + 80$$

$$C = 80 + 100$$

$$D = C + U$$

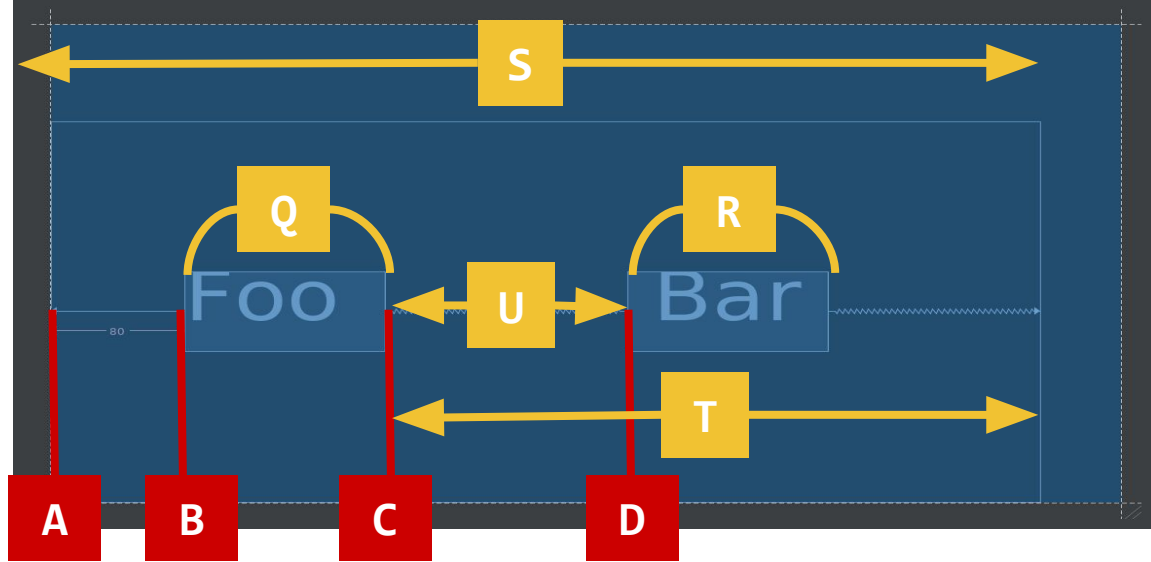
$$Q = 100$$

$$R = 100$$

$$S = 500$$

$$U = (500 - 180 - 100) / 2$$

$$T = 500 - 180$$



$$A = 0$$

$$B = 60$$

$$C = 160$$

$$D = C + U$$

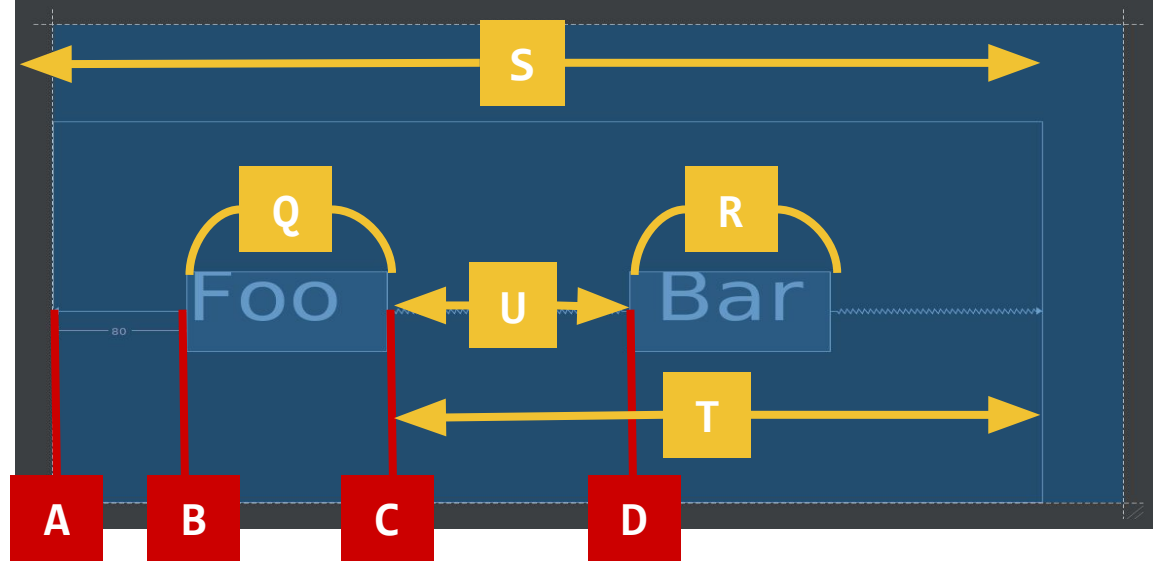
$$Q = 100$$

$$R = 100$$

$$S = 500$$

$$U = 120$$

$$T = 320$$



$$A = 0$$

$$B = 60$$

$$C = 160$$

$$D = 160 + 120$$

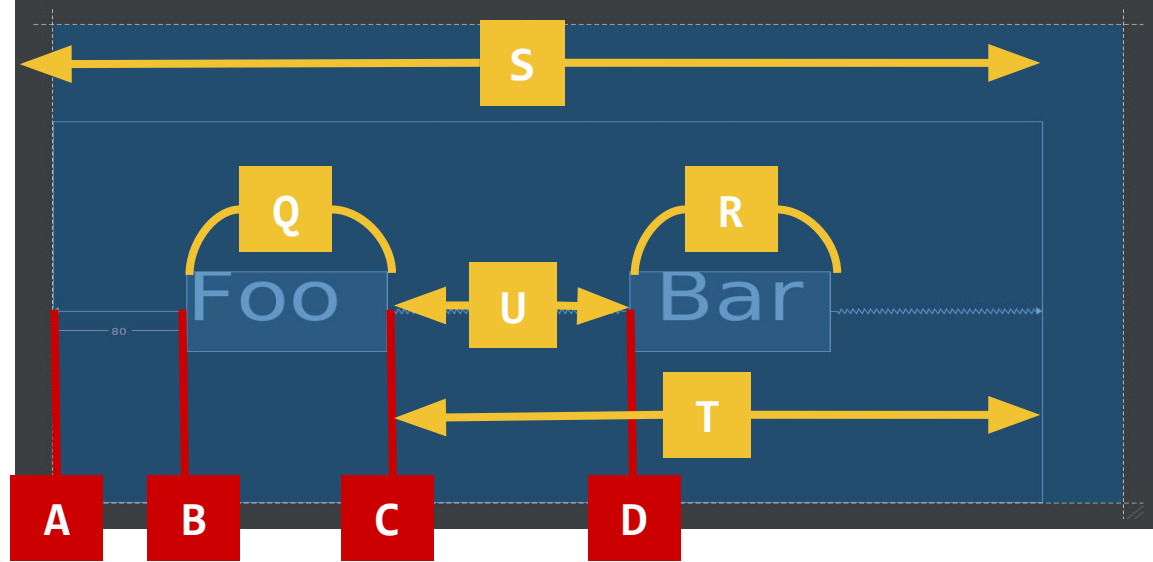
$$Q = 100$$

$$R = 100$$

$$S = 500$$

$$U = 120$$

$$T = 340$$



A = 0

B = 60

C = 160

D = 280

Q = 100

R = 100

S = 500

U = 120

T = 340

Q&A



Aleksander Piotrowski
@pelotasplus

