

Flag Quiz App

Fragments, Menus, Preferences, Explicit Intents, Handler, AssetManager, Tweened Animations, Animators, Toasts, Color State Lists, Layouts for Multiple Device Orientations, Logging Error Messages for Debugging



New Project

Android Studio

Configure your new project

Application name:

Company Domain:

Package name: [Edit](#)

Project location: ...



Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.

By targeting API 23 and later, your app will run on < 1% of the devices that are active on the Google Play Store.

[Help me choose](#)

Wear

Minimum SDK

TV

Minimum SDK

Android Auto

Glass

Minimum SDK

Cancel

Previous

Next

Finish

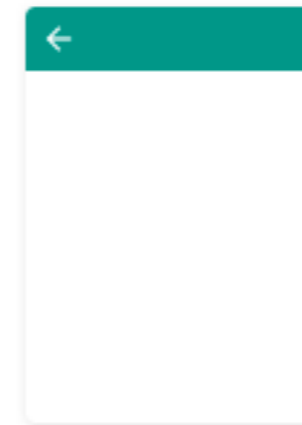


Add an Activity to Mobile



Add No Activity

Basic Activity



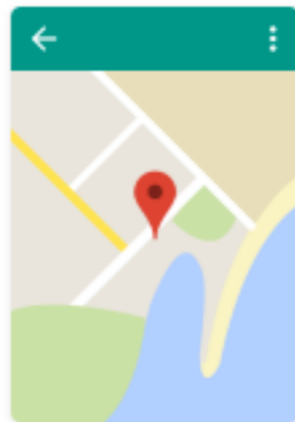
Empty Activity



Fullscreen Activity



Google AdMob Ads Activity



Google Maps Activity



Login Activity



Master/Detail Flow

Cancel

Previous

Next

Finish



Customize the Activity



Creates a new basic activity with an app bar.



Basic Activity

Activity Name:

Layout Name:

Title:

Menu Resource Name:

Use a Fragment

If true, the content will be a fragment

Project Structure

- app
 - manifests
 - java
 - res
 - drawable
 - layout
 - activity_main.xml
 - content_main.xml
 - fragment_main.xml
 - menu
 - mipmap
 - values
 - Gradle Scripts

Palette

- Layouts
 - FrameLayout
 - LinearLayout (Horizontal)
 - LinearLayout (Vertical)
 - TableLayout
 - TableRow
 - GridLayout
 - RelativeLayout
- Widgets
 - Plain TextView
 - Large Text
 - Medium Text
 - Small Text
 - Button
 - Small Button
 - RadioButton
 - CheckBox
 - Switch
 - ToggleButton
 - ImageButton
 - ImageView
 - ProgressBar (Large)
 - ProgressBar (Normal)
 - ProgressBar (Small)
 - ProgressBar (Horizontal)
 - SeekBar
 - RatingBar
 - Spinner
 - WebView
- Text Fields
 - Plain Text
 - Person Name



Component Tree

- Device Screen
 - fragment - at.htl.flagquiz.Mai

Properties

layout:width	match_parent
layout:height	match_parent
name	at.htl.flagquiz.M...
class	
id	fragment
tag	

Füge ein App Icon hinzu (New - Image Asset)



activity_main.xml

- enthält ein CoordinatorLayout
- Dieses CoordinatorLayout enthält eine app bar, welche als Toolbar (package android.support.v7.widget) definiert ist
- Dieses Template ist rückwärtskompatibel mit Android-Versionen, welche noch keine App Bar unterstützen
- Das Layout unterstützt auch material-design basierte Interaktionen mit verschachtelten Views. zB bei Animationen

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    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"
    android:fitsSystemWindows="true"
    tools:context="at.htl.flagquiz.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

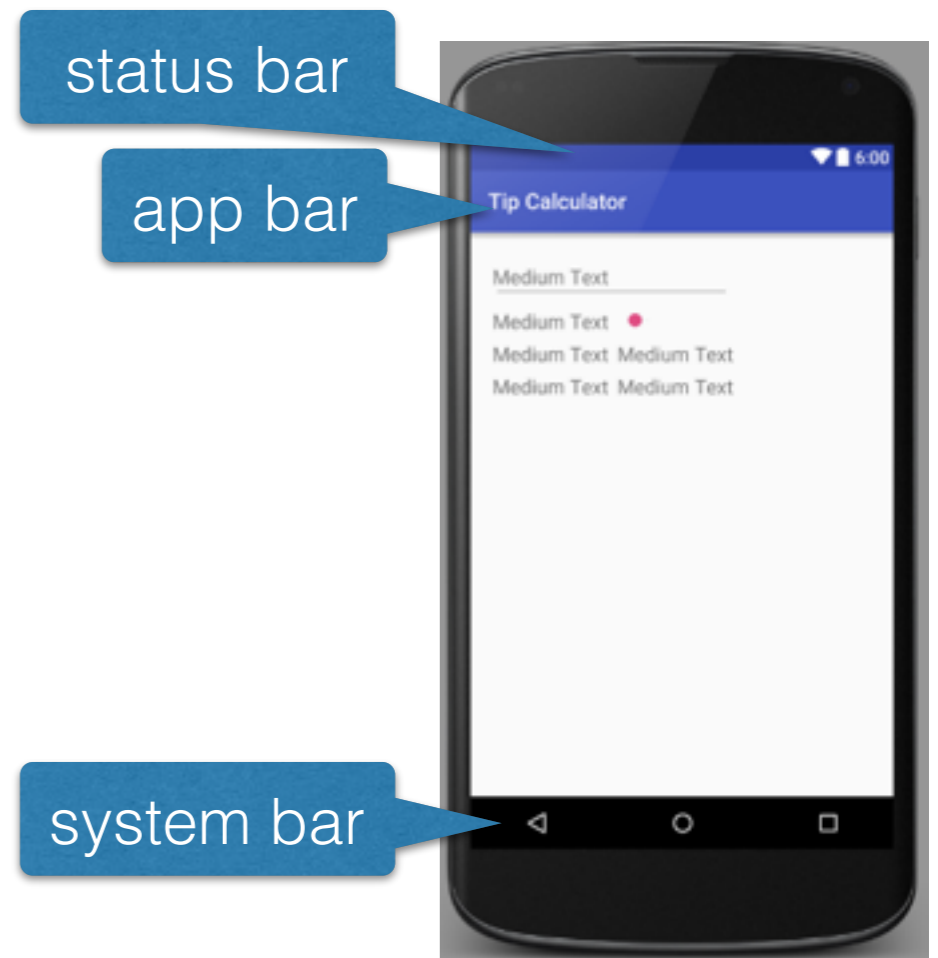
    <include layout="@layout/content_main" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        android:src="@android:drawable/ic_dialog_email" />

</android.support.design.widget.CoordinatorLayout>
```


content_main.xml

- content_main.xml definiert den Bereich der MainActivity-GUI unter der app bar und über der system bar
- enthält nur ein <fragment>-Element (bei Verwendung von Fragments)



```
<fragment 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:id="@+id/fragment"
  android:name="at.htl.flagquiz.MainActivityFragment"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  app:layout_behavior="android.support.design.widget.AppBarLayout$Sc
  tools:layout="@layout/fragment_main" />
```

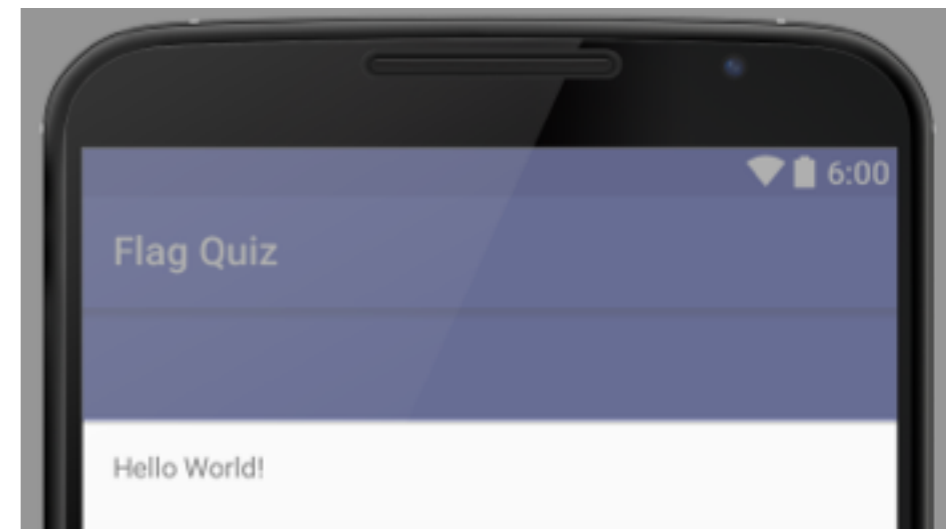
fragment_main.xml

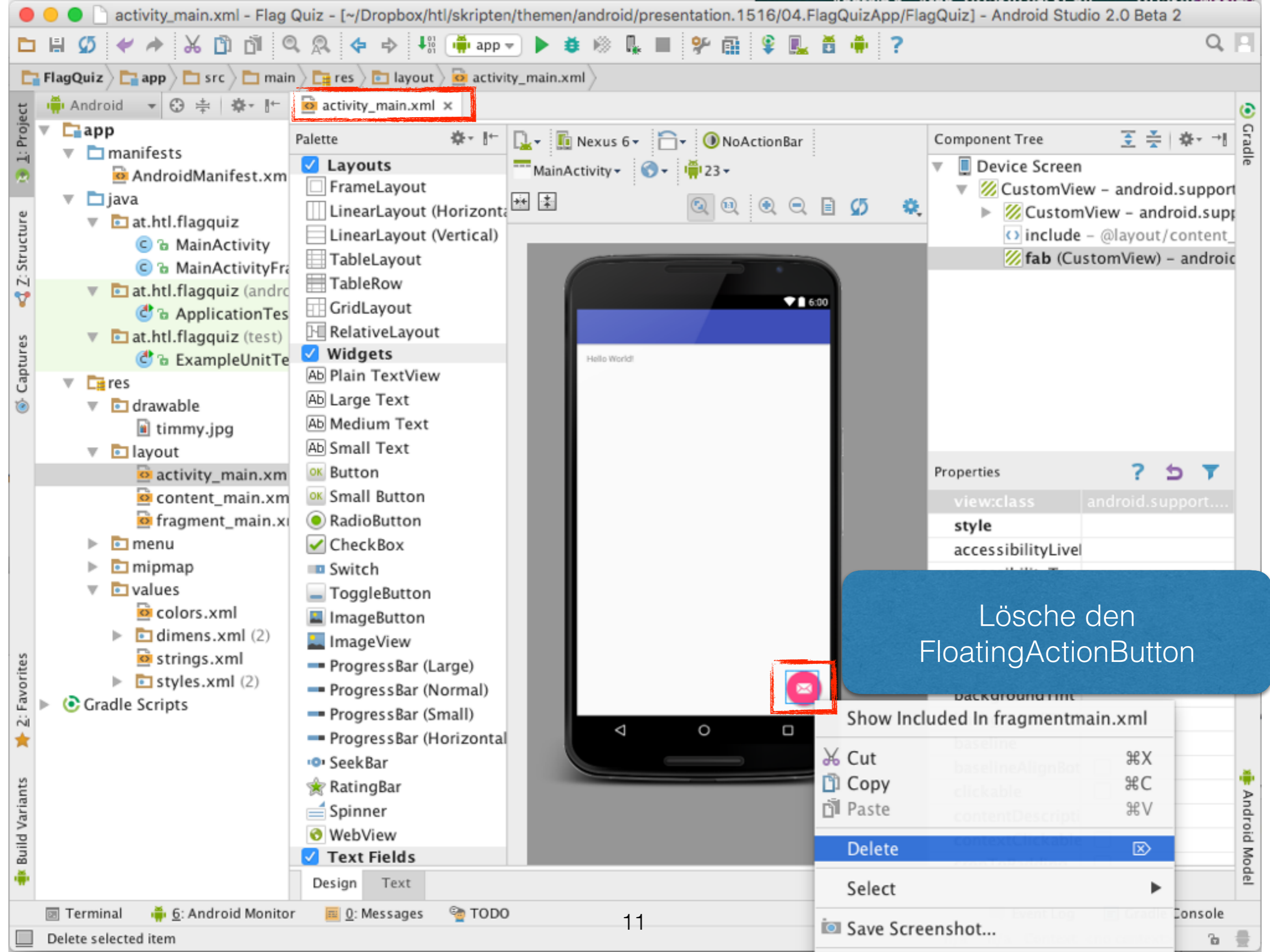
- Hier befindet sich die „eigentlichen“ Steuerelemente (Views)

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="at.htl.flagquiz.MainActivityFragment"
    tools:showIn="@layout/activity_main">

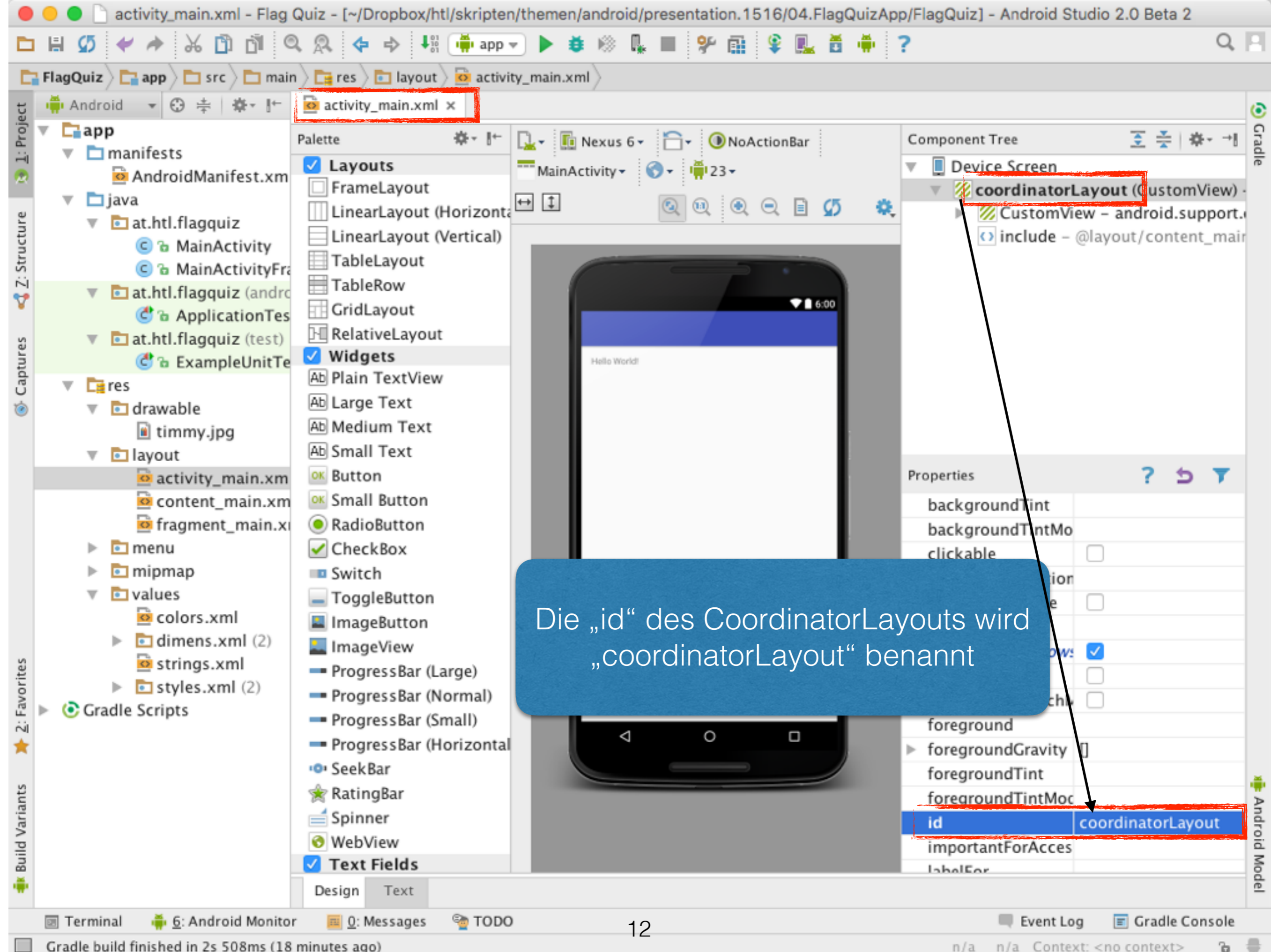
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</RelativeLayout>
```





Lösche den
FloatingActionButton

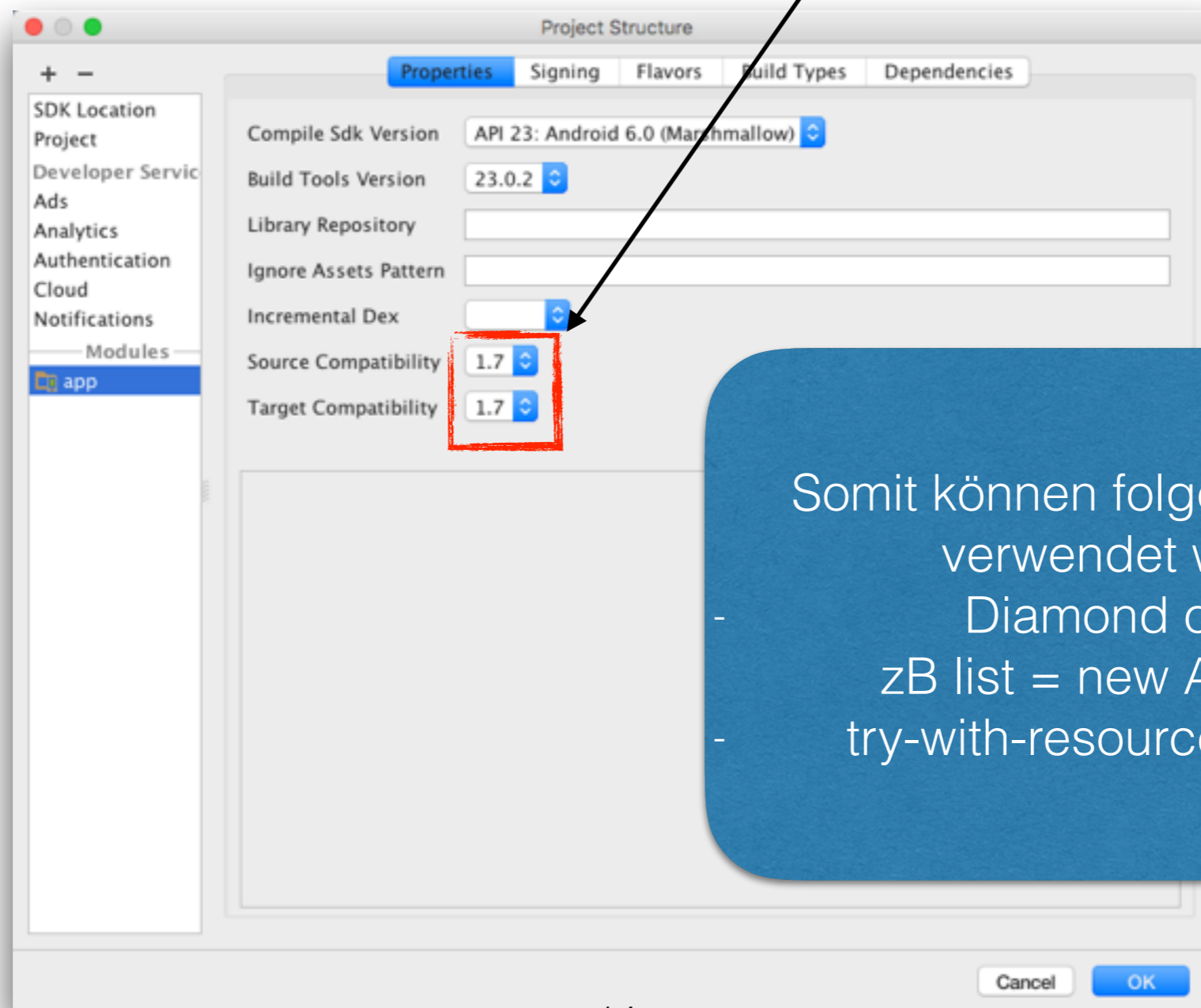
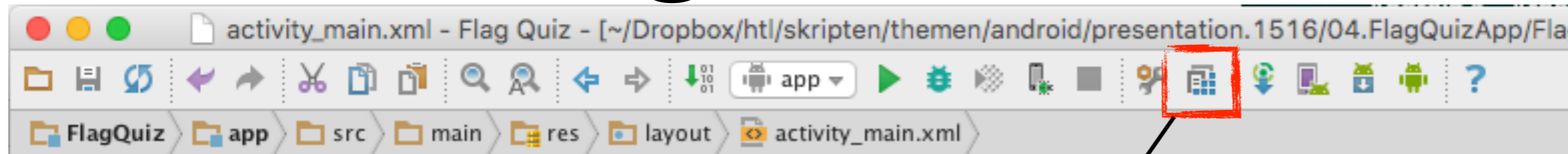


Die „id“ des CoordinatorLayouts wird „coordinatorLayout“ benannt

Properties	
backgroundTint	
backgroundTintMo	
clickable	<input type="checkbox"/>
foreground	
foregroundGravity	
foregroundTint	
foregroundTintMoc	
id	coordinatorLayout
importantForAcces	
labelFor	

Vorbereitende Arbeiten

Konfiguriere Java 7

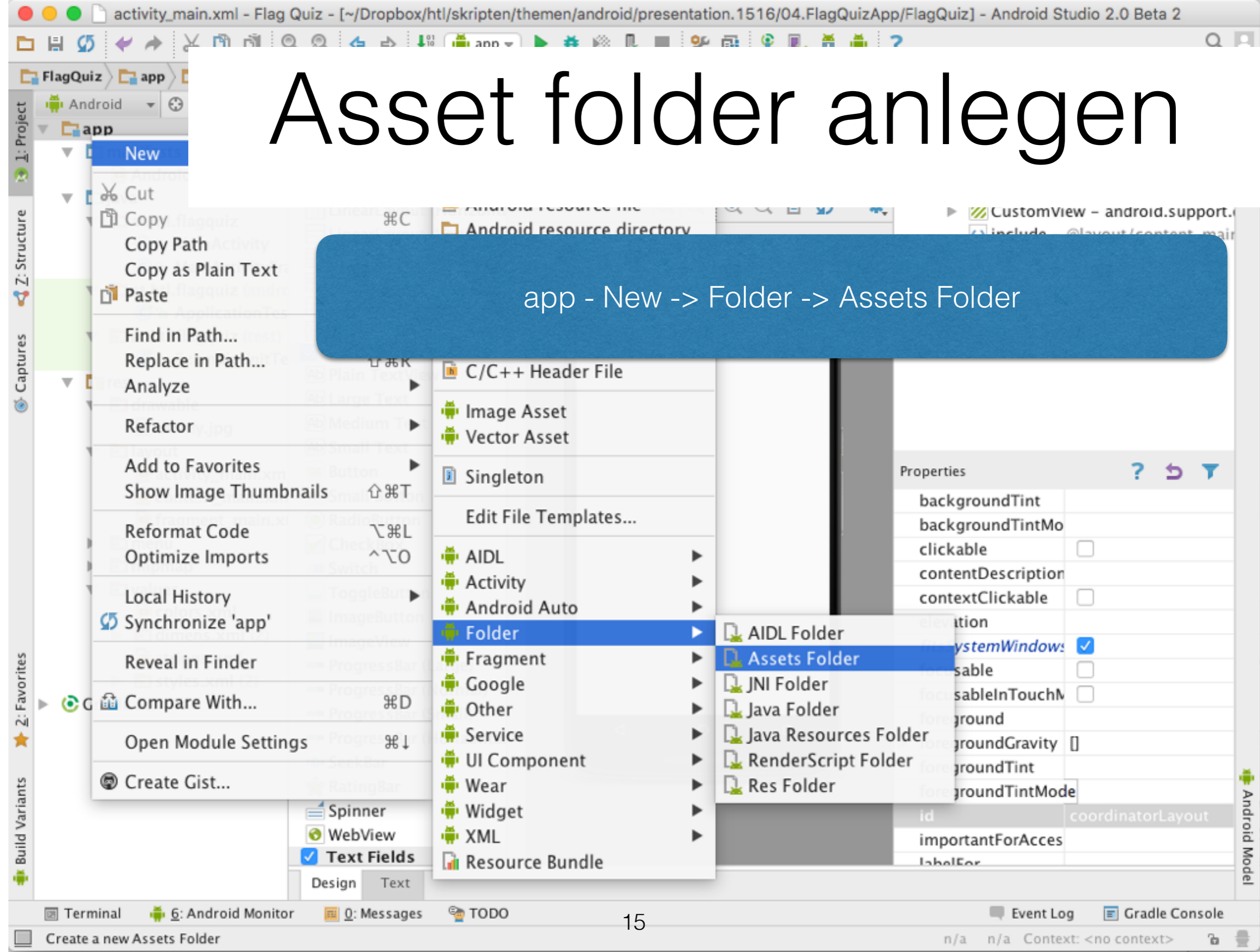


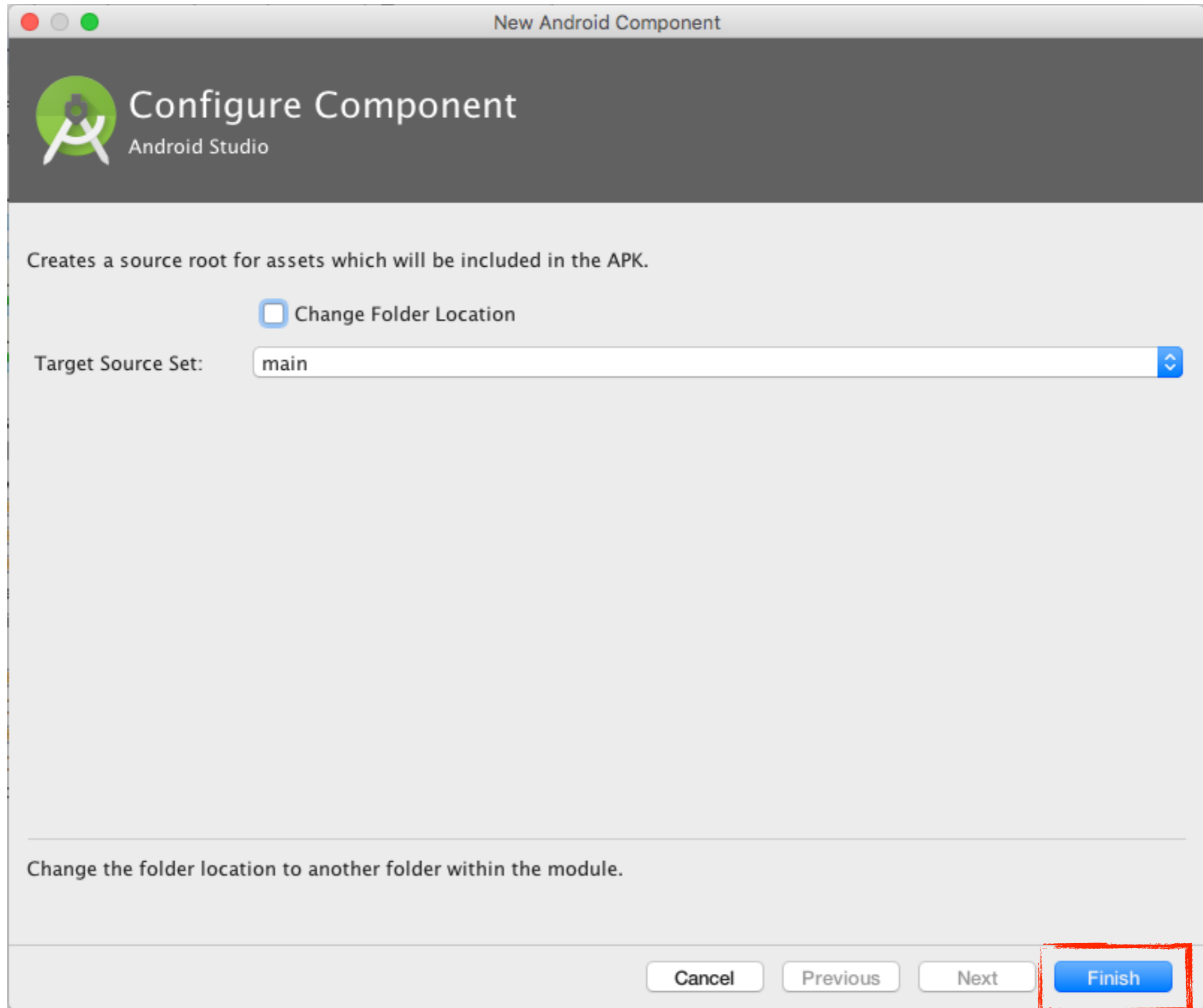
Somit können folgende Features verwendet werden:

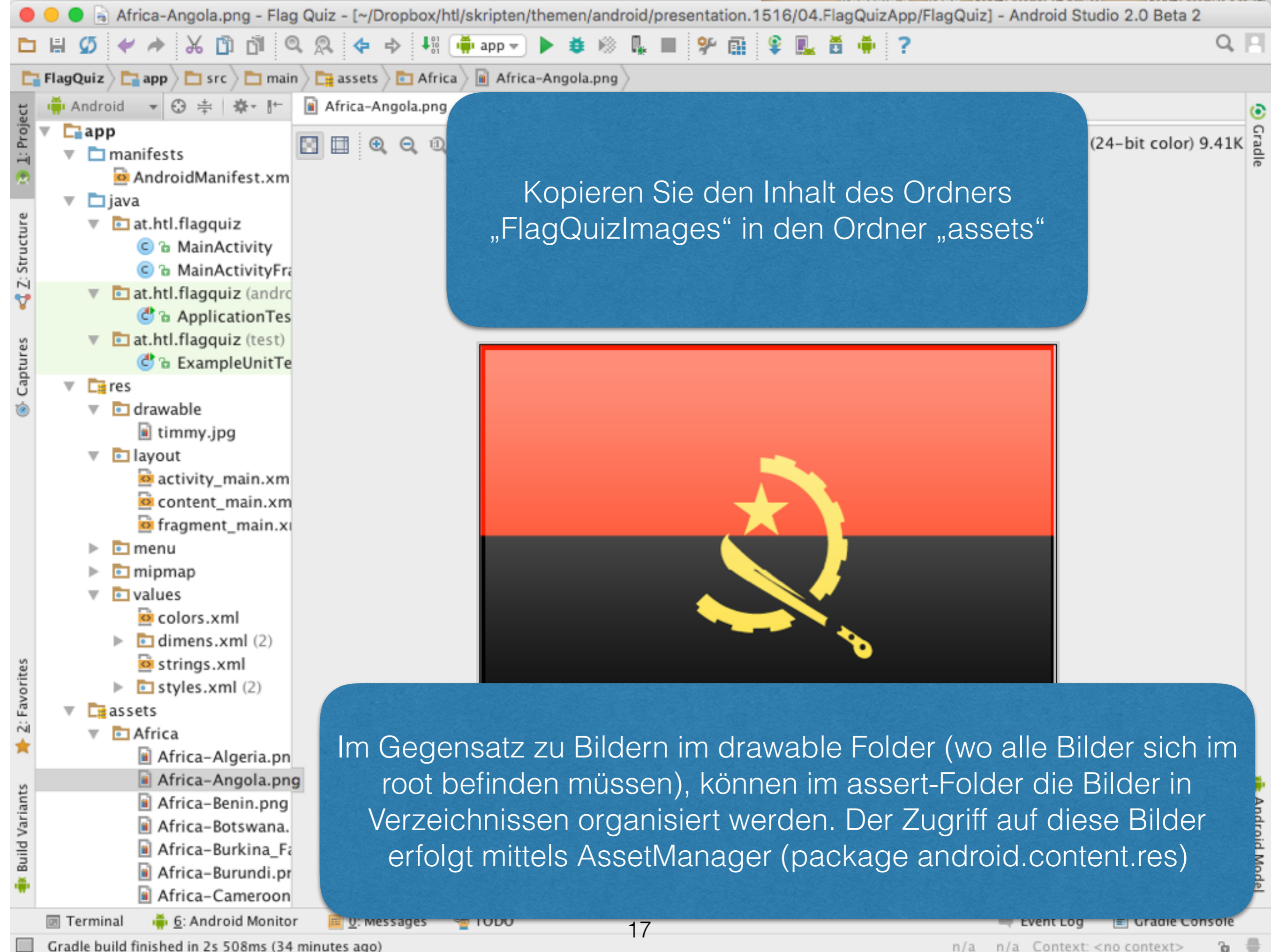
- Diamond operators
zB `list = new ArrayList<>()`
- try-with-resources Statements

Asset folder anlegen

app - New -> Folder -> Assets Folder





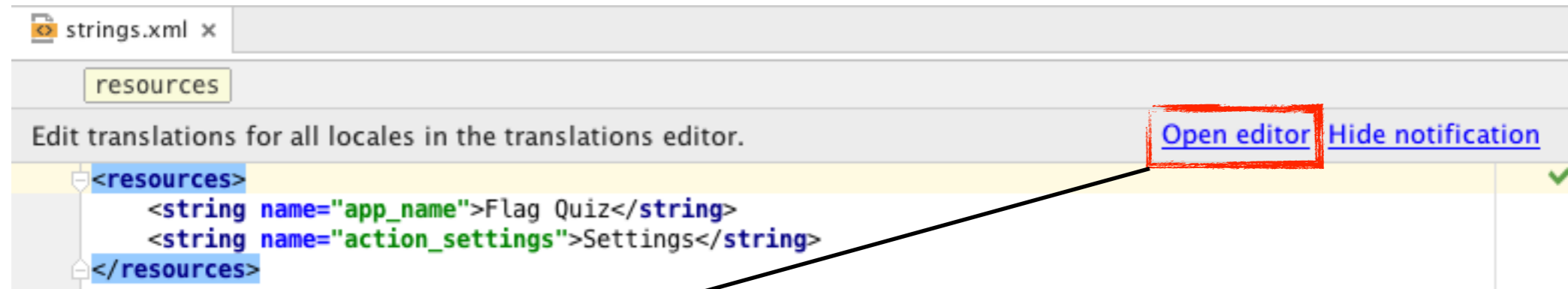


Kopieren Sie den Inhalt des Ordners „FlagQuizImages“ in den Ordner „assets“



Im Gegensatz zu Bildern im drawable Folder (wo alle Bilder sich im root befinden müssen), können im assert-Folder die Bilder in Verzeichnissen organisiert werden. Der Zugriff auf diese Bilder erfolgt mittels AssetManager (package android.content.res)

Erstellen der string- Ressourcen



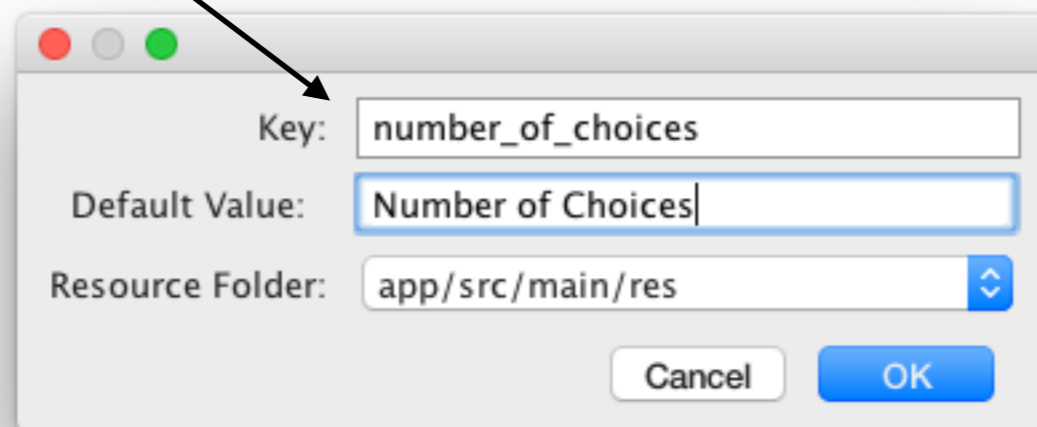
```
strings.xml x
resources
Edit translations for all locales in the translations editor.
Open editor Hide notification
<resources>
  <string name="app_name">Flag Quiz</string>
  <string name="action_settings">Settings</string>
</resources>
```



Translations Editor x

+ Show only keys needing translations Order a translation...

Key	Default Value	Untra...
action_settings	Settings	<input type="checkbox"/>
app_name	Flag Quiz	<input type="checkbox"/>



Key: number_of_choices

Default Value: Number of Choices

Resource Folder: app/src/main/res

Cancel OK

strings.xml

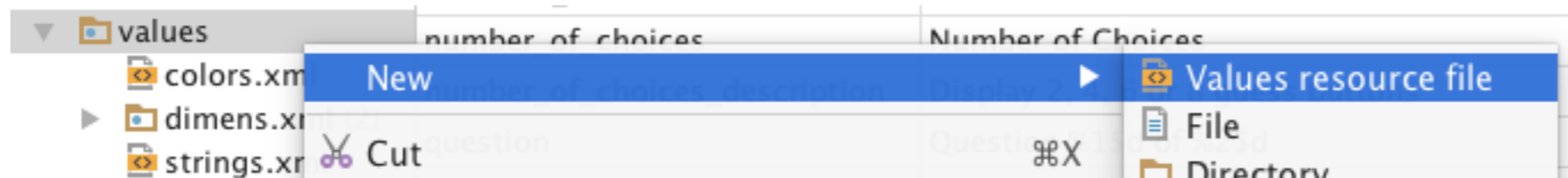
Key	Default Value	Untranslat...
action_settings	Settings	<input type="checkbox"/>
app_name	Flag Quiz	<input type="checkbox"/>
default_region	North_America	<input type="checkbox"/>
default_region_message	One region must be selected. Setting North America as the default region.	<input type="checkbox"/>
guess_country	Guess the country	<input type="checkbox"/>
image_description	Image of the current flag in the quiz	<input type="checkbox"/>
incorrect_answer	Incorrect!	<input type="checkbox"/>
number_of_choices	Number of Choices	<input type="checkbox"/>
number_of_choices_description	Display 2, 4, 6 or 8 guess buttons	<input type="checkbox"/>
question	Question %1\$d of %2\$d	<input type="checkbox"/>
reset_quiz	Reset Quiz	<input type="checkbox"/>
restarting_quiz	Quiz will restart with your new settings	<input type="checkbox"/>
results	%1\$d guesses, %2\$.02f%% correct	<input type="checkbox"/>
world_regions	Regions	<input type="checkbox"/>
world_regions_description	Regions to include in the quiz	<input type="checkbox"/>

format Strings

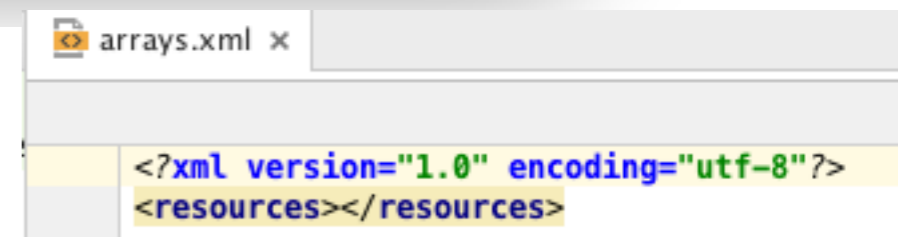
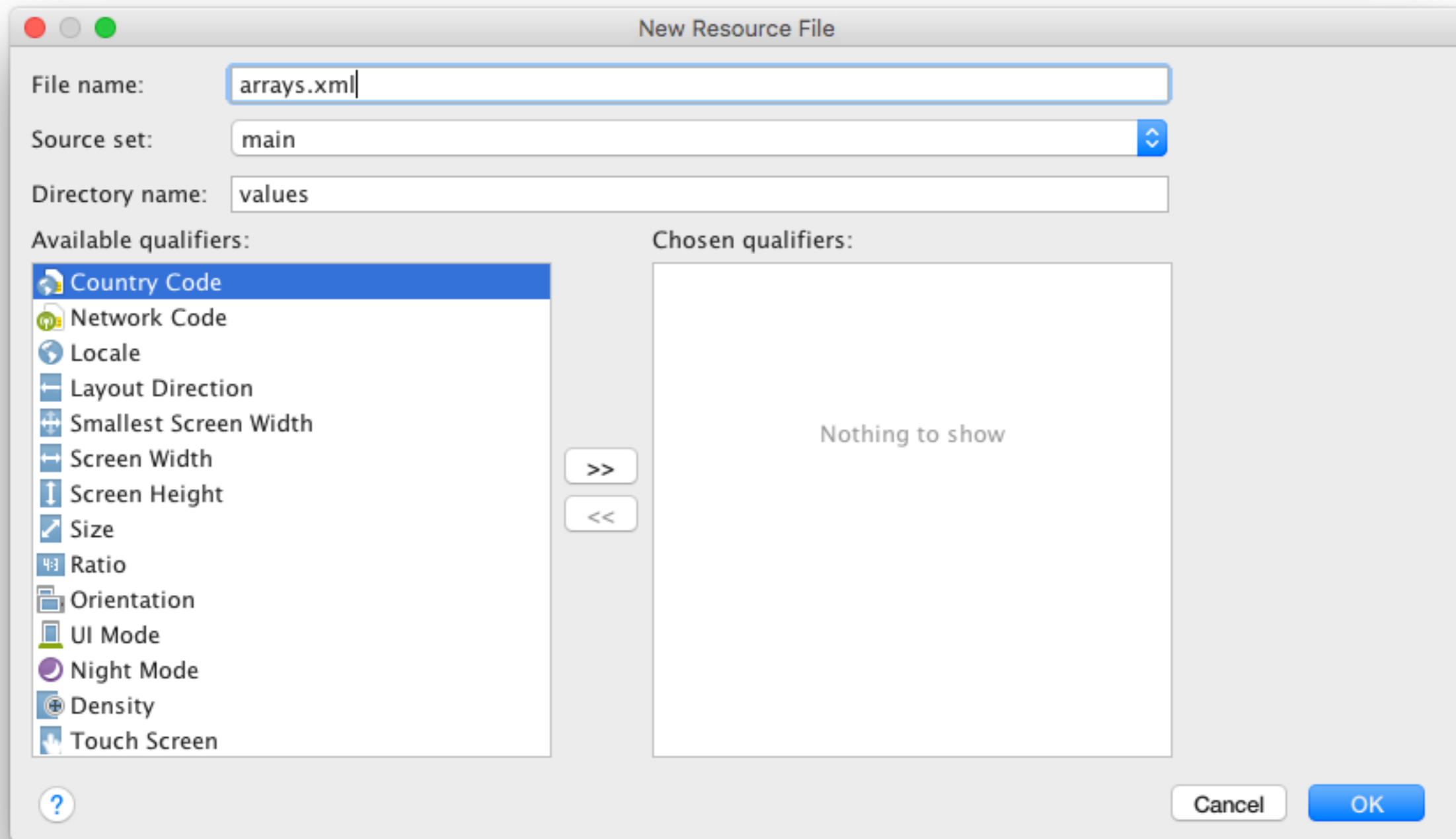
- `%1$d guesses, %2$.02f%%` correct
- `1$` ... erster Wert, der `%1$d` ersetzt
- `2$` ... zweiter Wert, der `%2$.02f` ersetzt
- `d` ... Integer
- `f` ... Floating-point Number
- Die Position von `1$` und `2$` im Text ist unerheblich. somit ist man sprachspezifisch flexibel

arrays.xml

- Technisch gesehen, könnten sämtliche Einträge im res/value-Folder in einem File stehen
- Dies wird jedoch vermieden
- Anlegen eines neuen Resource-Files: arrays.xml



arrays.xml



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="regions_list">
    <item>Africa</item>
    <item>Asia</item>
    <item>Europe</item>
    <item>North_America</item>
    <item>Oceania</item>
    <item>South_America</item>
  </string-array>

  <string-array name="regions_list_for_settings">
    <item>Africa</item>
    <item>Asia</item>
    <item>Europe</item>
    <item>North America</item>
    <item>Oceania</item>
    <item>South America</item>
  </string-array>

  <string-array name="guesses_list">
    <item>2</item>
    <item>4</item>
    <item>6</item>
    <item>8</item>
  </string-array>
</resources>
```

colors.xml x

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

```
<resources>
```

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

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

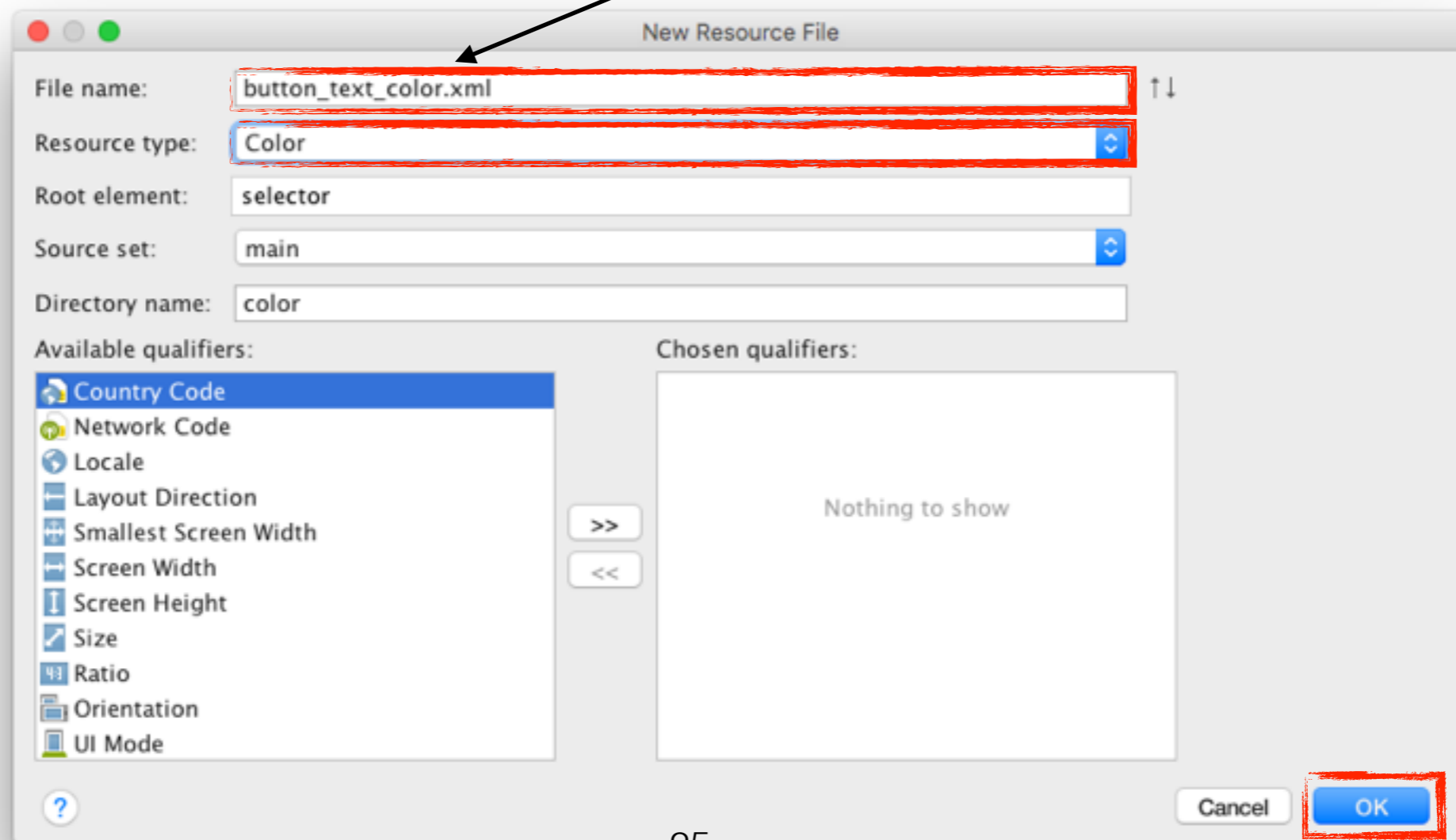
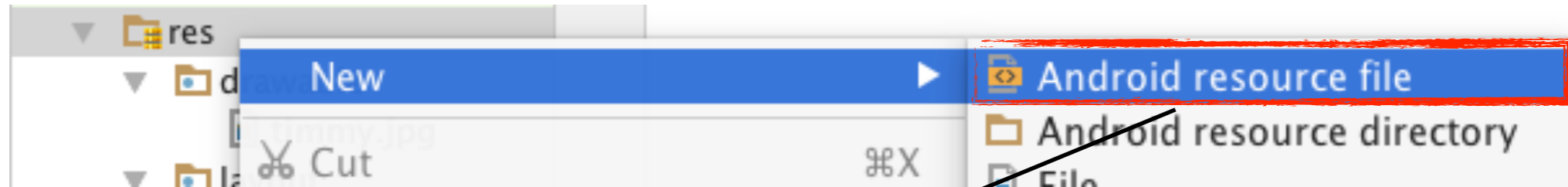
```
    <color name="colorAccent">#448AFF</color>
```

```
    <color name="correct_answer">#00CC00</color>
```

```
    <color name="incorrect_answer">#FF0000</color>
```

```
</resources>
```


button_text_color.xml



button_text_color.xml x

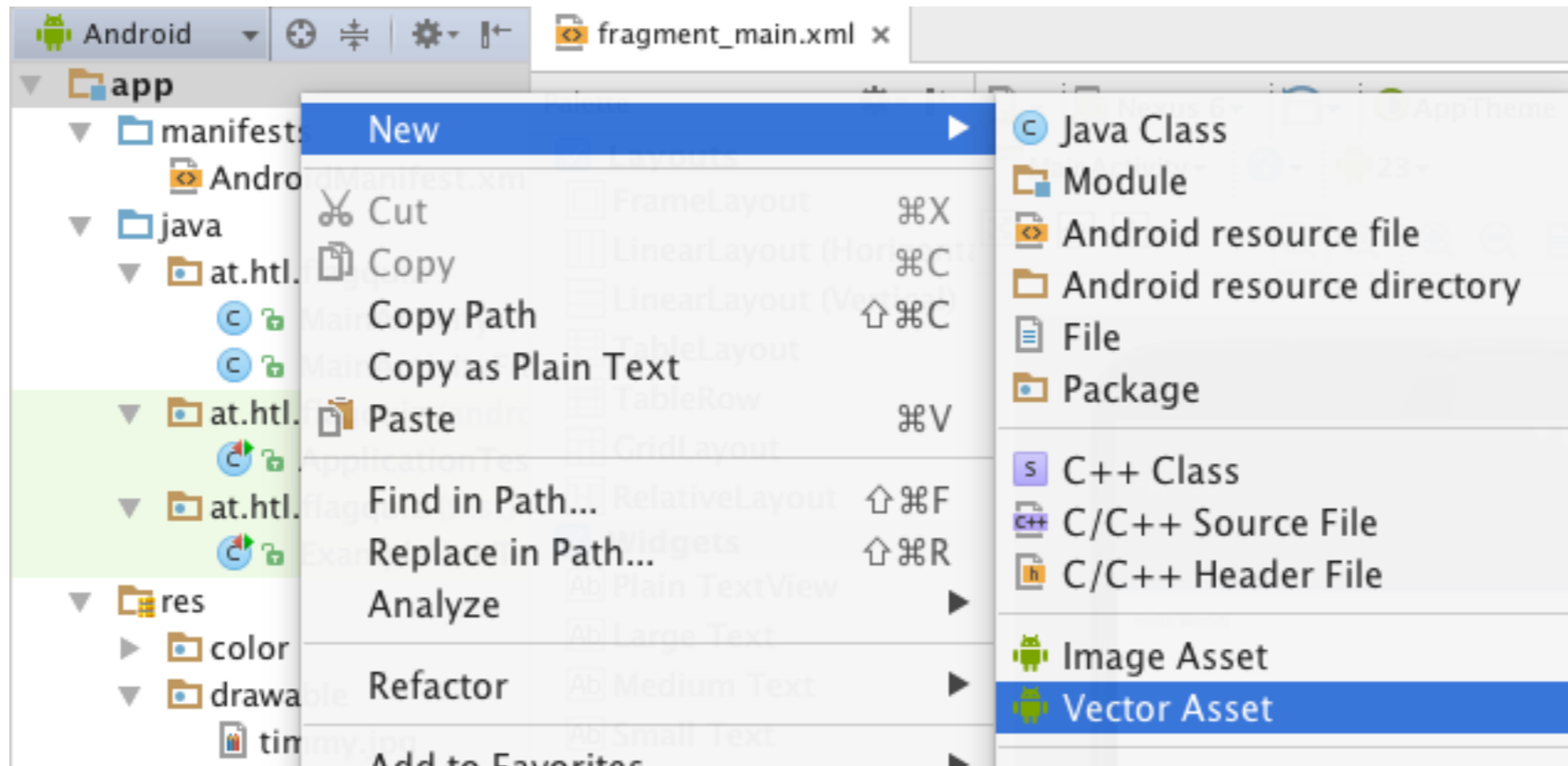
```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:color="@android:color/primary_text_dark"
    android:state_enabled="true" />

  <item
    android:color="@android:color/darker_gray"
    android:state_enabled="false" />
</selector>
```

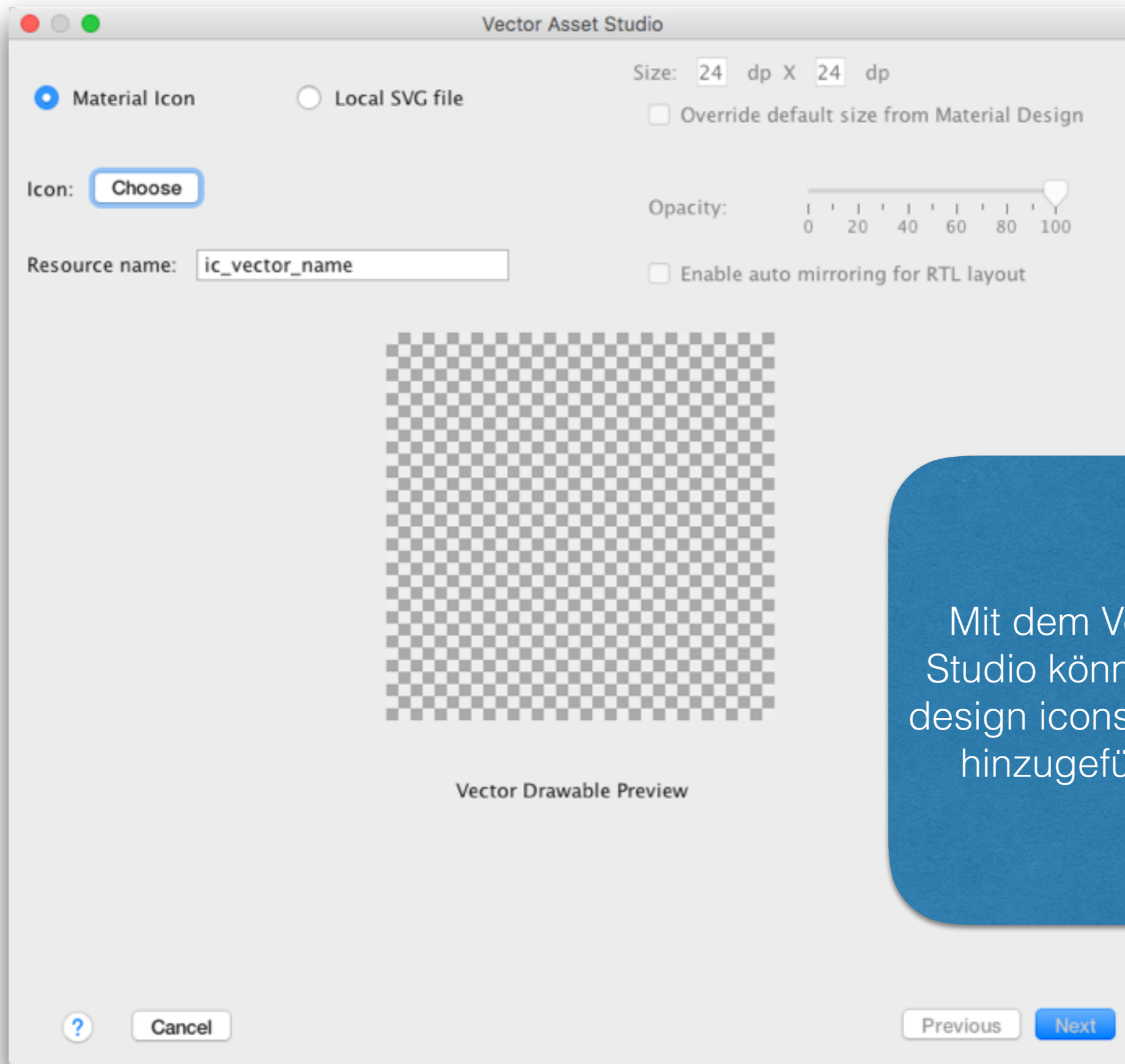
Nun brauchen wir ein
Icon für das Main Menu



Vector Asset Studio

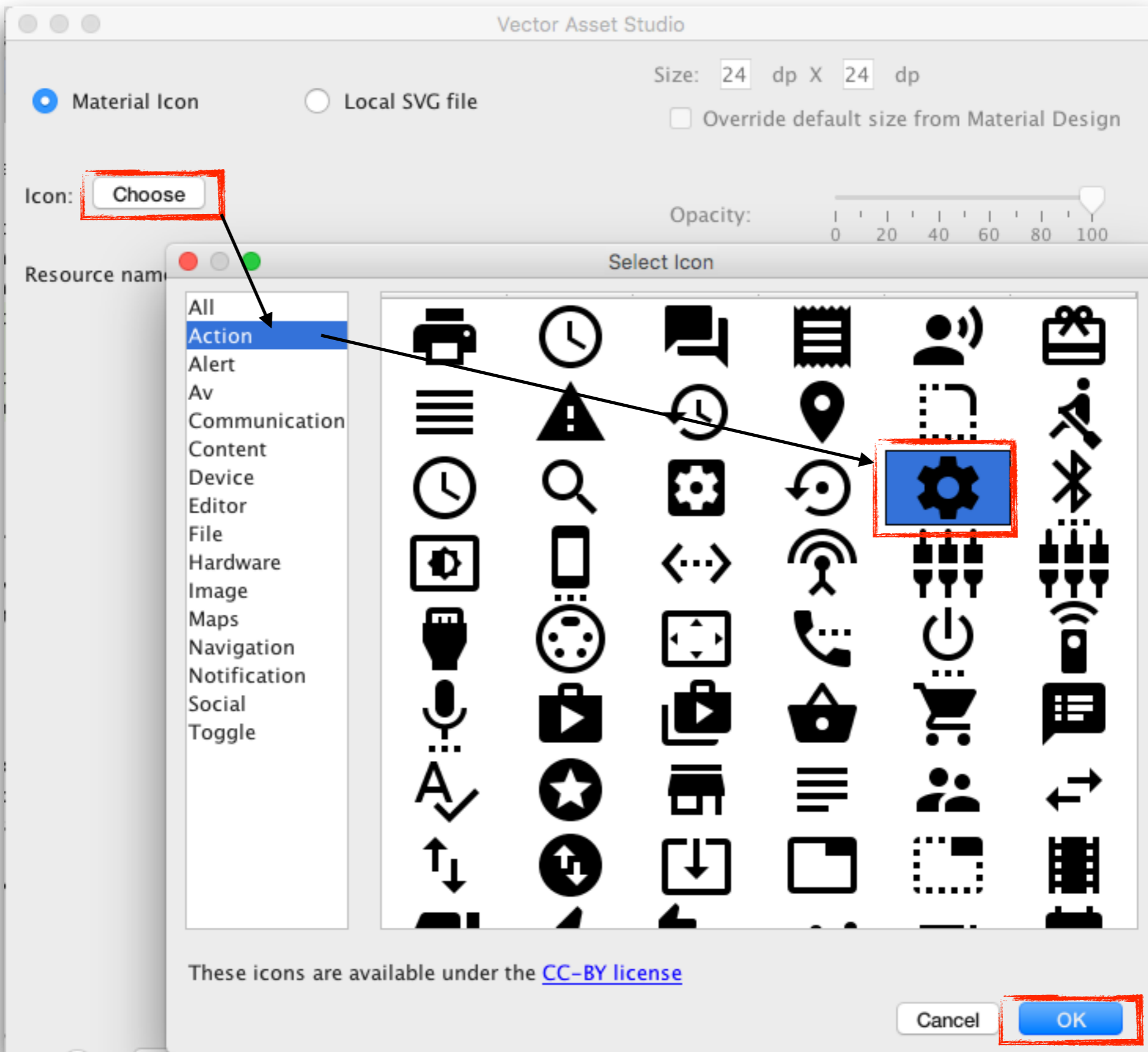


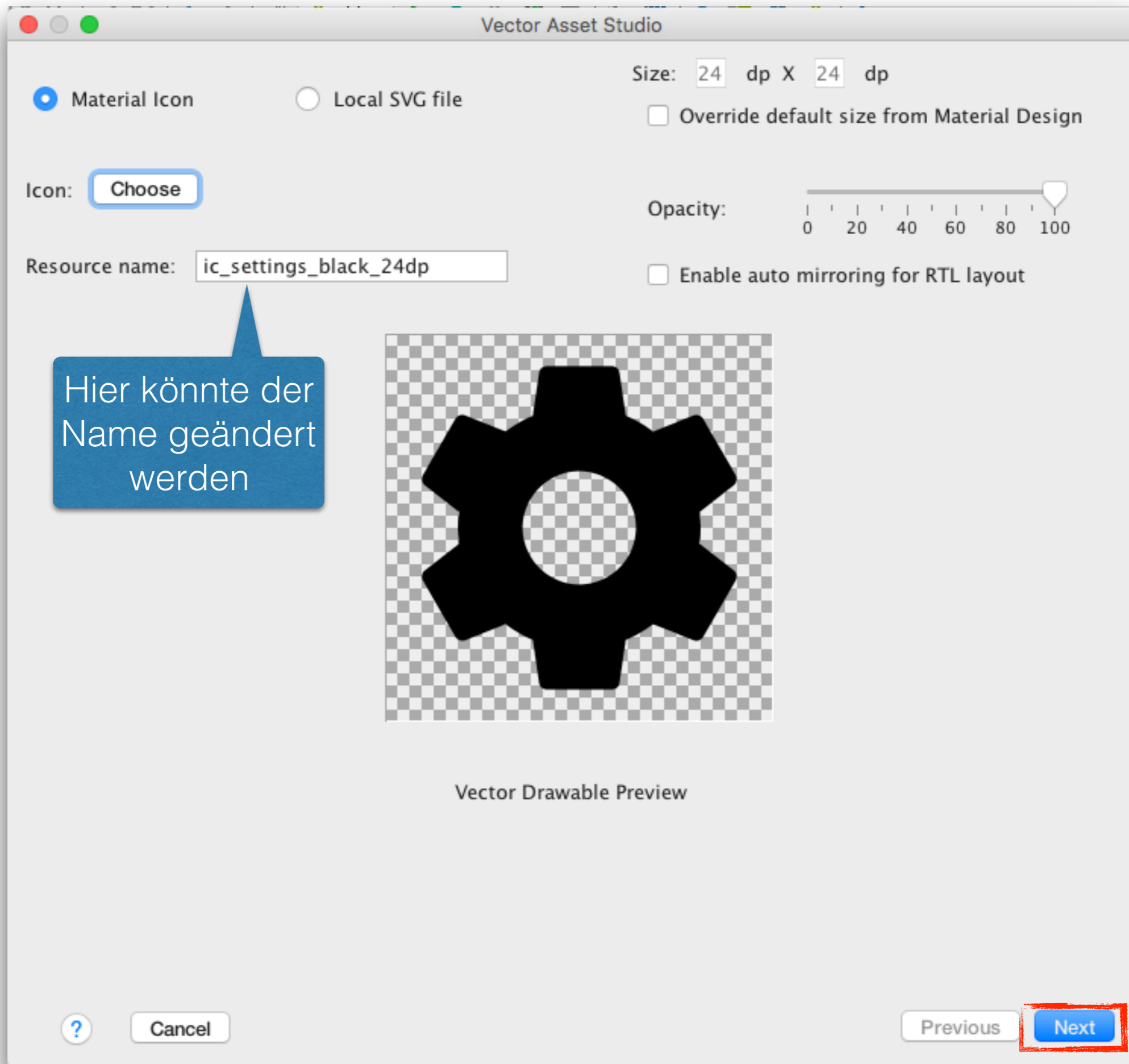
app - New -> Vector Asset

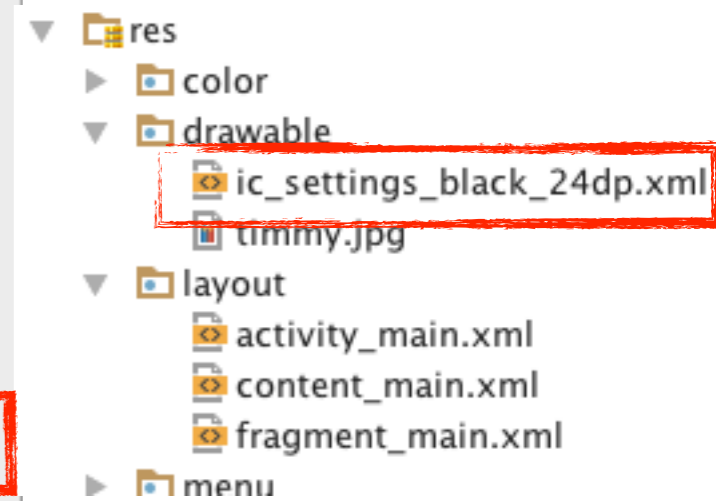
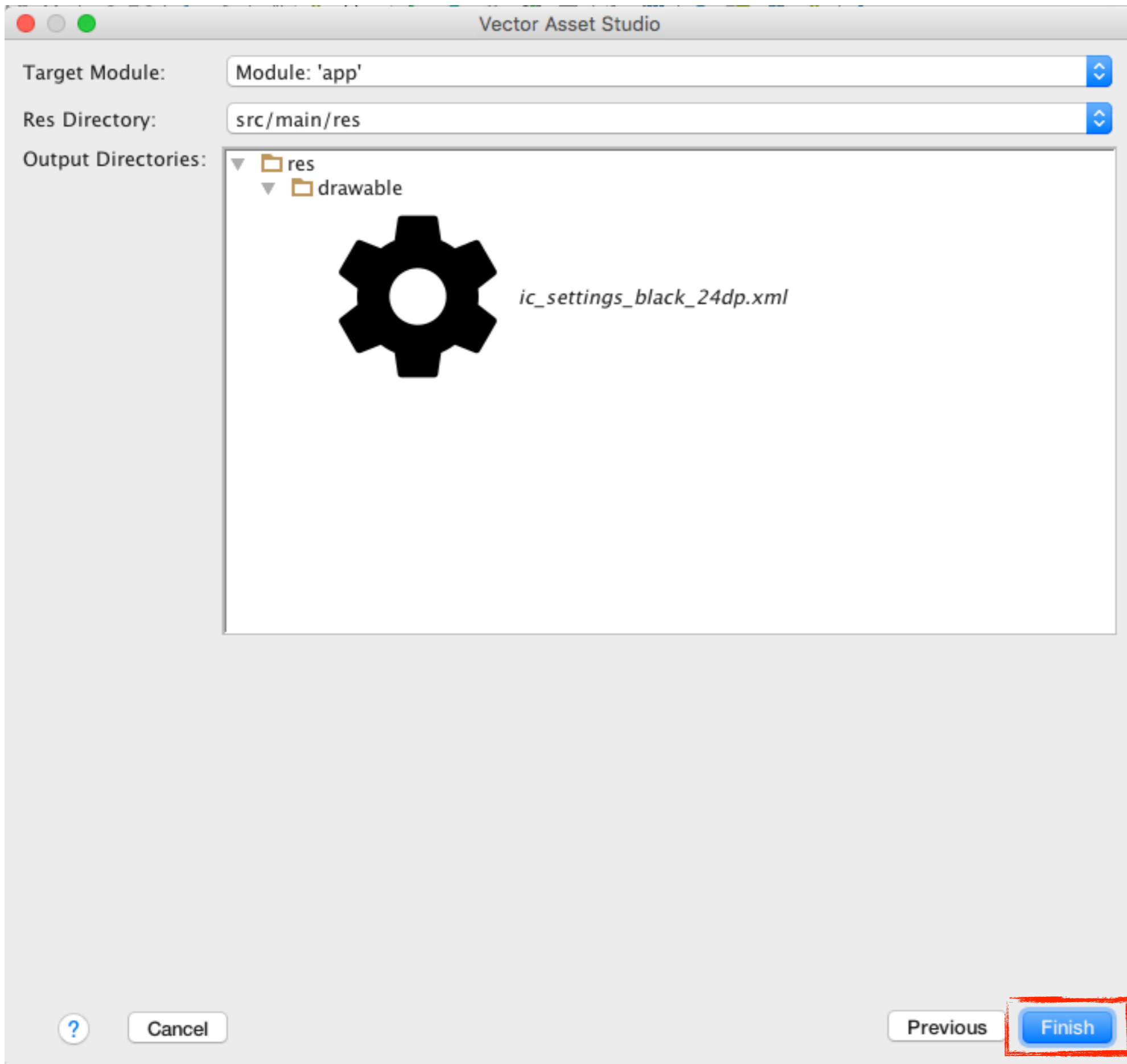


Mit dem Vector Asset Studio können „material design icons“ zum Projekt hinzugefügt werden

<https://design.google.com/icons/>





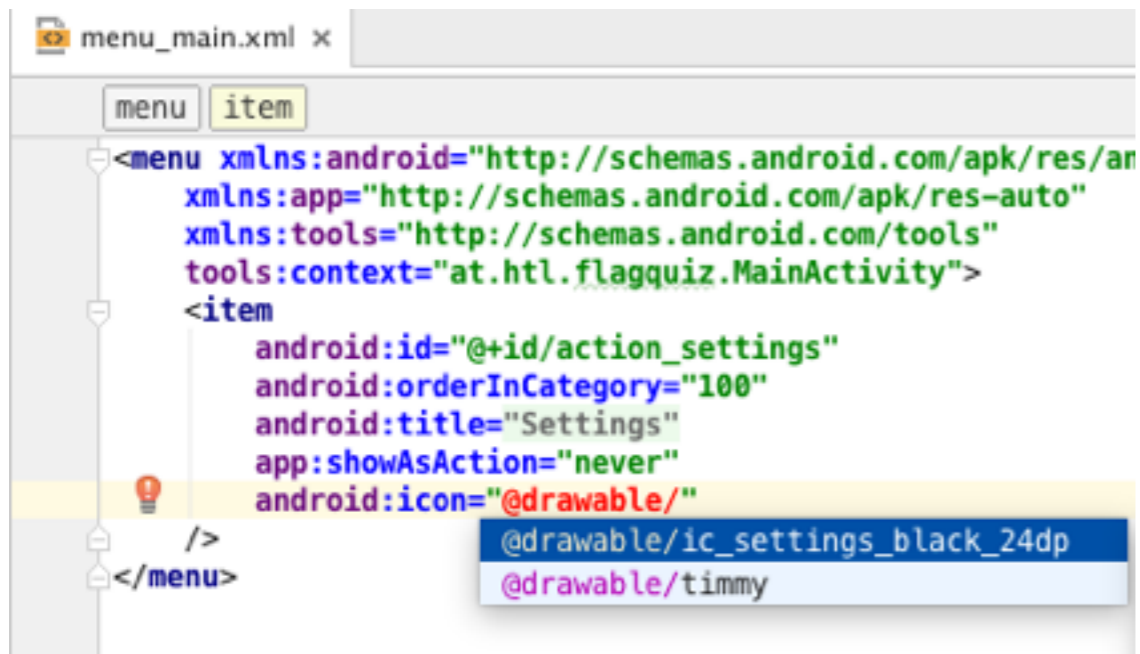


ic_settings_black_24dp.xml

```
ic_settings_black_24dp.xml x
vector
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24.0"
    android:viewportHeight="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M19.43,12.98c0.04,-0.32 0.07,-0.64 0.07,-0.96"
    />
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24.0"
    android:viewportHeight="24.0">
    <path
        android:fillColor="@android:color/white"
        android:pathData="M19.43,12.98c0.04,-0.32 0.07,-0.64 0.07,-0.96"
    />
</vector>
```

Hinzufügen des Icon zu menu_main.xml

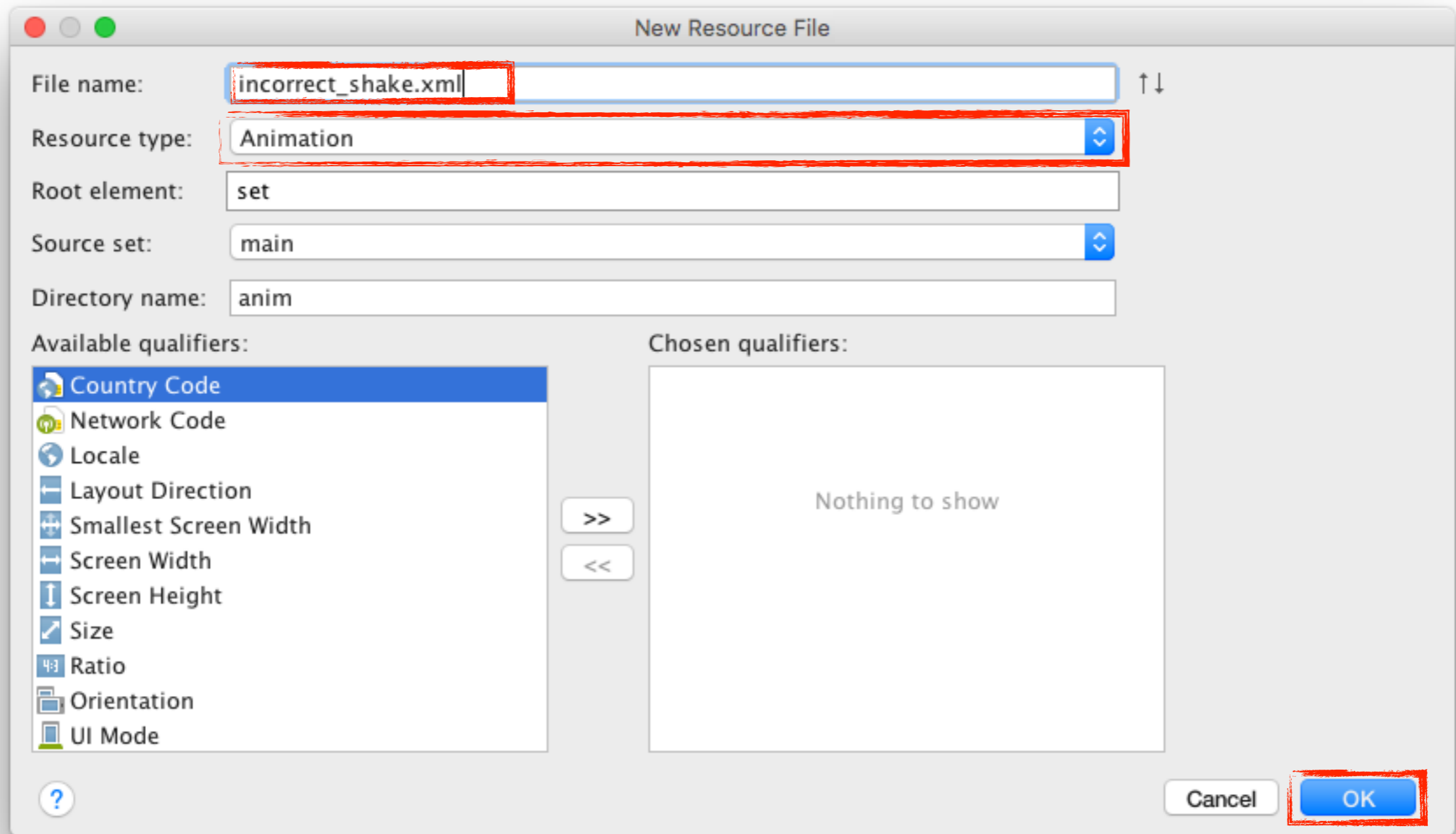


```
<menu xmlns:android="http://schemas.android.com/apk/res/an
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
tools:context="at.htl.flagquiz.MainActivity">
  <item
    android:id="@+id/action_settings"
    android:orderInCategory="100"
    android:title="Settings"
    app:showAsAction="never"
    android:icon="@drawable/"
  />
</menu>
```



```
<item
  android:id="@+id/action_settings"
  android:orderInCategory="100"
  android:title="Settings"
  app:showAsAction="always"
  android:icon="@drawable/ic_settings_black_24dp"
/>
```

Unsere erste Animation





shake-Effekt

```
incorrect_shake.xml x
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/decelerate_interpolator">

    <translate android:duration="100" android:fromXDelta="0"
        android:toXDelta="-5%p" />

    <translate android:duration="100" android:fromXDelta="-5%p"
        android:toXDelta="5%p" android:startOffset="100"/>

    <translate android:duration="100" android:fromXDelta="5%p"
        android:toXDelta="-5%p" android:startOffset="200"/>

</set>
```

Tweened Animations

- Tweening ... Prozess der Interpolation von Zwischenbilder und Schlüsselbilder beim Produktionsprozess einer Animation
- alpha ... transparency
- scale ... resize
- translate ... move
- rotate

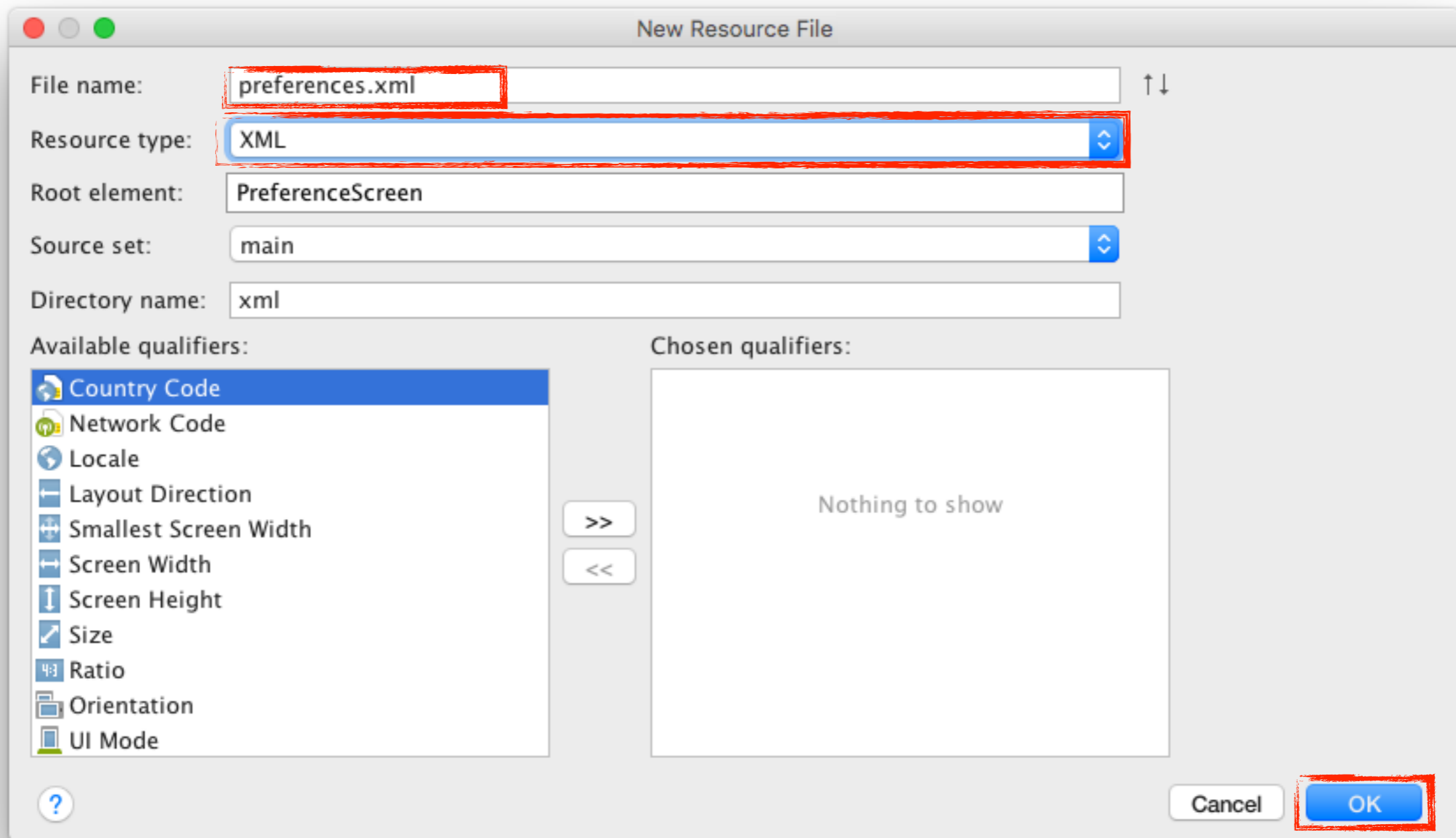
Animations

- Unsere Animation besteht aus einer Serie von drei translate animations
- Eine **translate animation** bewegt eine View innerhalb seiner Parents
- Android unterstützt auch die **property animation**, bei der die Eigenschaften eines Objekts animiert werden

fromXDelta / toXDelta

- Diese Attribute können angeben
 - absolute Werte (in Pixel)
 - einen Prozentsatz der Größe der animierten View
 - einen Prozentsatz der Größe der View Parents
- 5%p gibt 5% der Größe der Parents an
- 5% bezieht sich auf die Größe der View selbst

Erstellen der preferences.xml



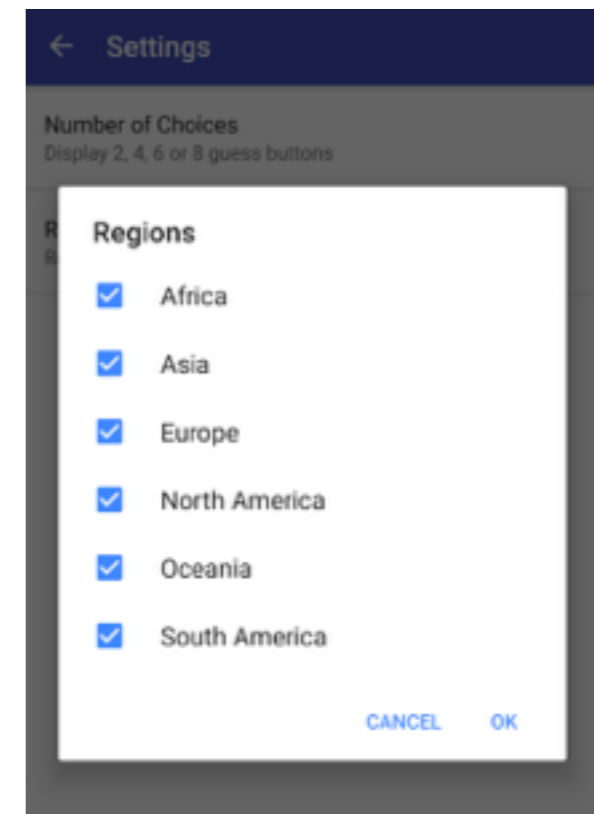
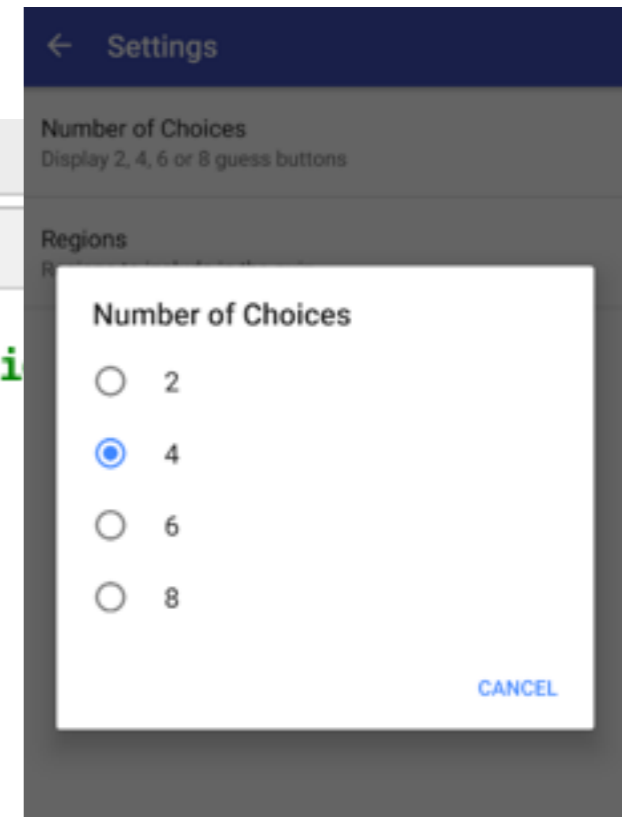
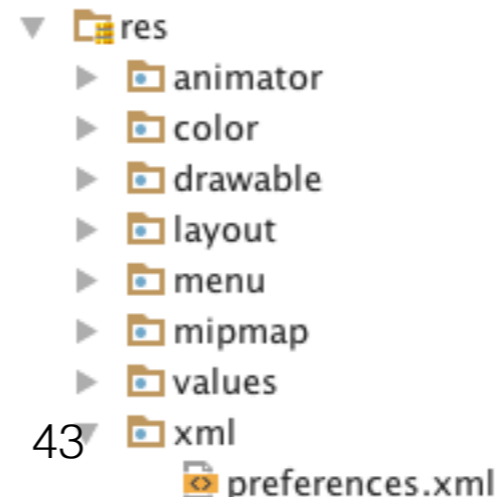
preferences.xml x

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

    <ListPreference
        android:entries="@array/guesses_list"
        android:entryValues="@array/guesses_list"
        android:key="pref_numberOfChoices"
        android:title="@string/number_of_choices"
        android:summary="@string/number_of_choices_description"
        android:persistent="true"
        android:defaultValue="4" />

    <MultiSelectListPreference
        android:entries="@array/regions_list_for_settings"
        android:entryValues="@array/regions_list"
        android:key="pref_regionsToInclude"
        android:title="@string/world_regions"
        android:summary="@string/world_regions_description"
        android:persistent="true"
        android:defaultValue="@array/regions_list" />

</PreferenceScreen>
```

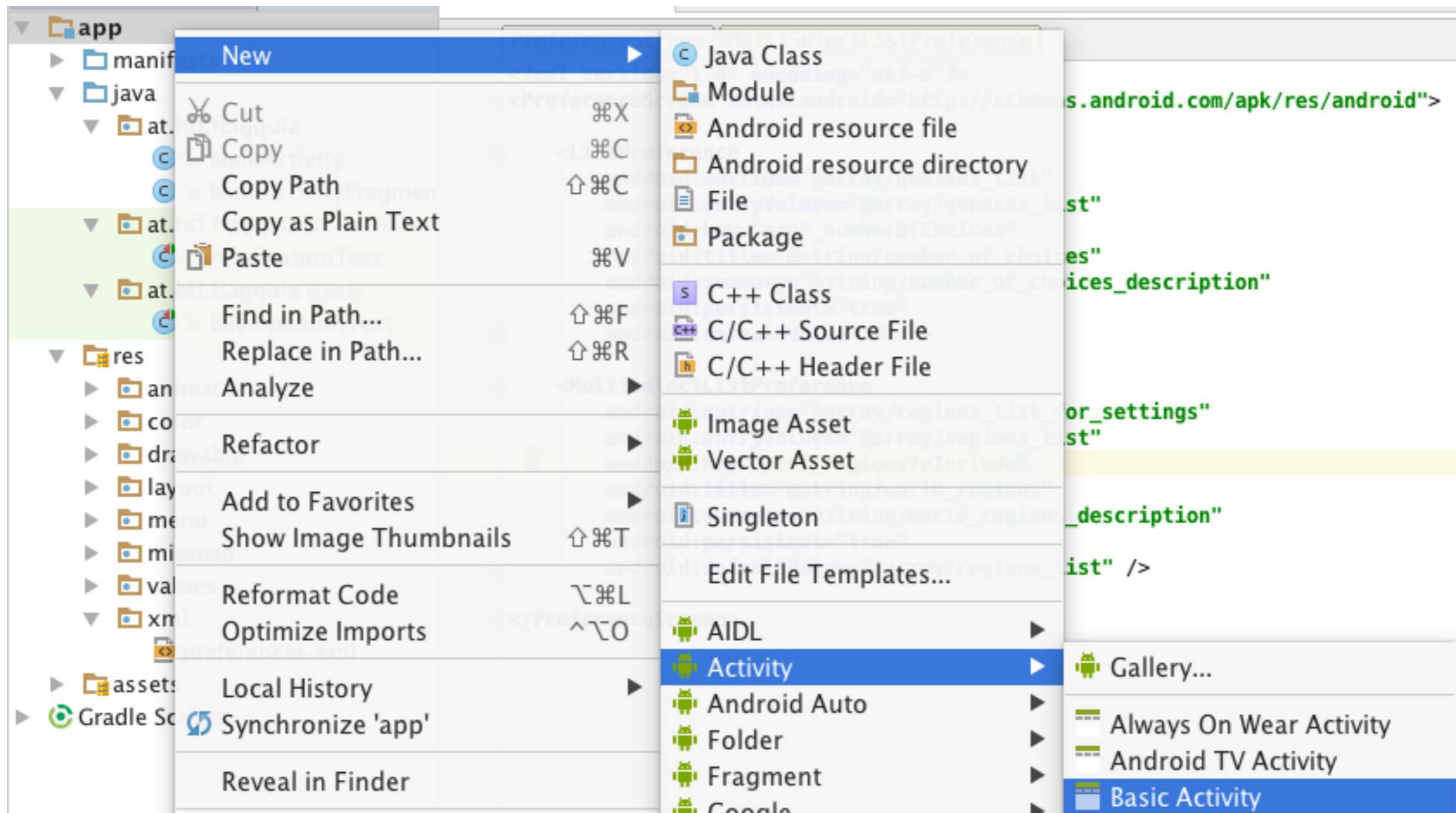


Die nebenstehenden Setting-Dialoge werden später automatisch erstellt

Property	Value	Description
<code>entries</code>	<code>@array/guesses_list</code>	An array of <code>Strings</code> that will be displayed in the list of options.
<code>entryValues</code>	<code>@array/guesses_list</code>	An array of values associated with the options in the <code>Entries</code> property. The selected entry's value will be stored in the app's <code>SharedPreferences</code> .
<code>key</code>	<code>pref_numberOfChoices</code>	The name of the preference stored in the app's <code>SharedPreferences</code> .
<code>title</code>	<code>@string/number_of_choices</code>	The title of the preference displayed in the GUI.
<code>summary</code>	<code>@string/number_of_choices_description</code>	A summary description of the preference that's displayed below its title.
<code>persistent</code>	<code>true</code>	Whether the preference should persist after the app terminates—if <code>true</code> , class <code>PreferenceFragment</code> immediately persists the preference value each time it changes.
<code>defaultValue</code>	<code>4</code>	The item in the <code>Entries</code> property that's selected by default.

Property	Value	Description
<code>entries</code>	<code>@array/regions_list_for_settings</code>	An array of Strings that will be displayed in the list of options.
<code>entryValues</code>	<code>@array/regions_list</code>	An array of the values associated with the options in the <code>Entries</code> property. The selected entries' values will <i>all</i> be stored in the app's SharedPreferences.
<code>key</code>	<code>pref_regionsToInclude</code>	The name of the preference stored in the app's SharedPreferences.
<code>title</code>	<code>@string/world_regions</code>	The title of the preference displayed in the GUI.
<code>summary</code>	<code>@string/world_regions_description</code>	A summary description of the preference that's displayed below its title.
<code>persistent</code>	<code>true</code>	Whether the preference should persist after the app terminates.
<code>defaultValue</code>	<code>@array/regions_list</code>	An array of the default values for this preference—in this case, all of the regions will be selected by default.

Erstellen von SettingsActivity und SettingsActivityFragment



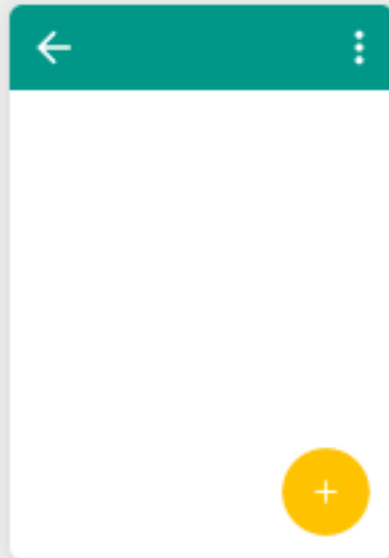
app - New -> Activity -> Basic Activity



Configure Activity

Android Studio

Creates a new basic activity with an app bar.



Activity Name:

Layout Name:

Title:

Launcher Activity

Use a Fragment

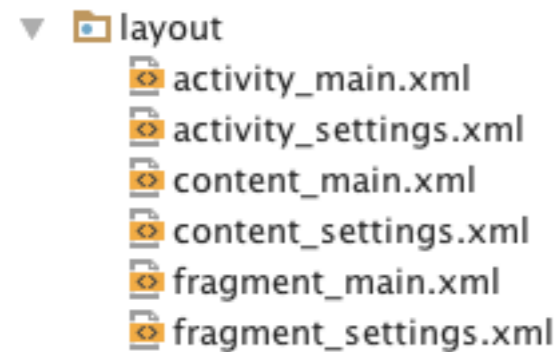
Hierarchical Parent:

Package name:

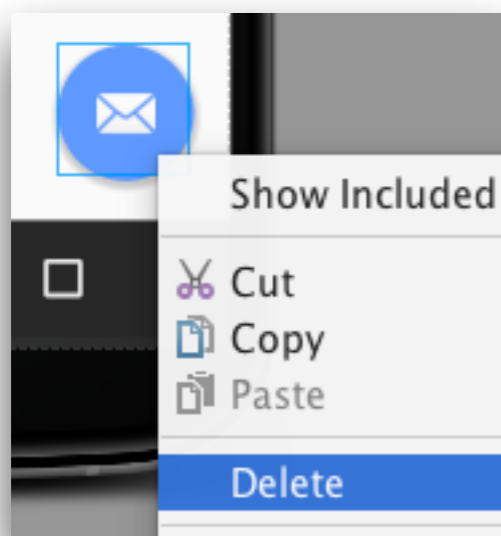
Target Source Set:

The hierarchical parent activity, used to provide a default implementation for the 'Up' button

Settings



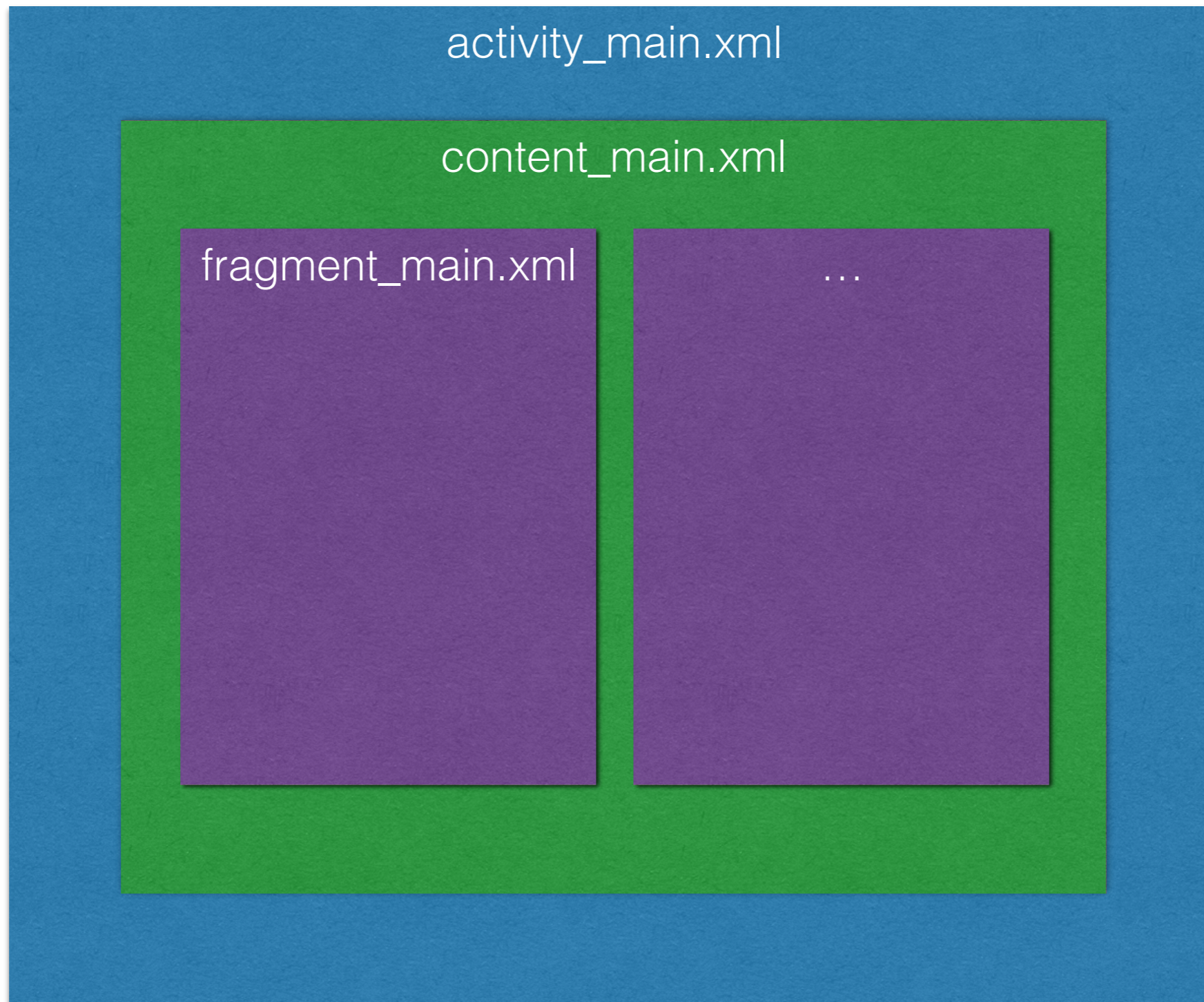
Die settings Files wurden in layout erstellt

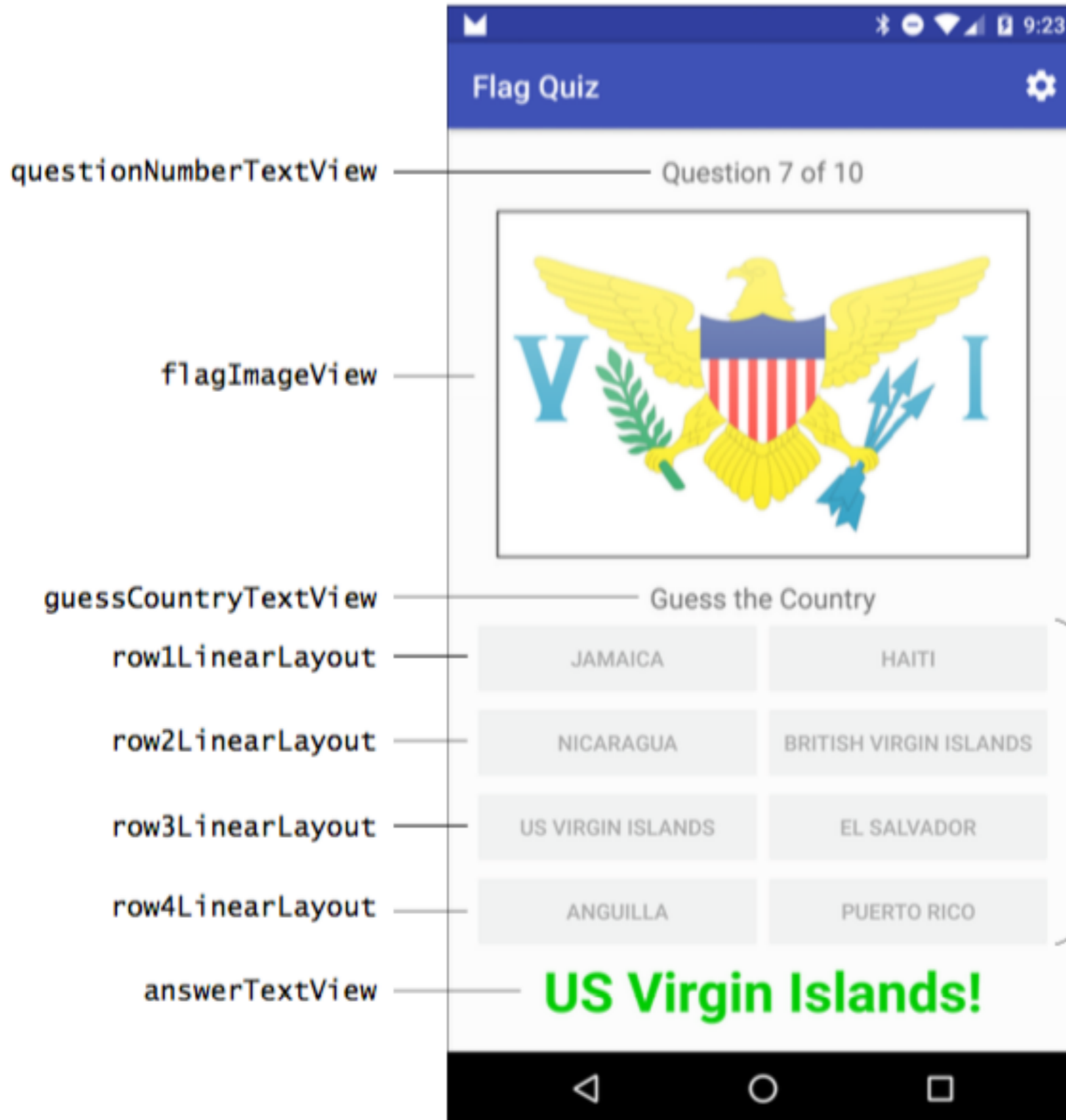


Delete the FloatingActionButton

Erstellen der GUI

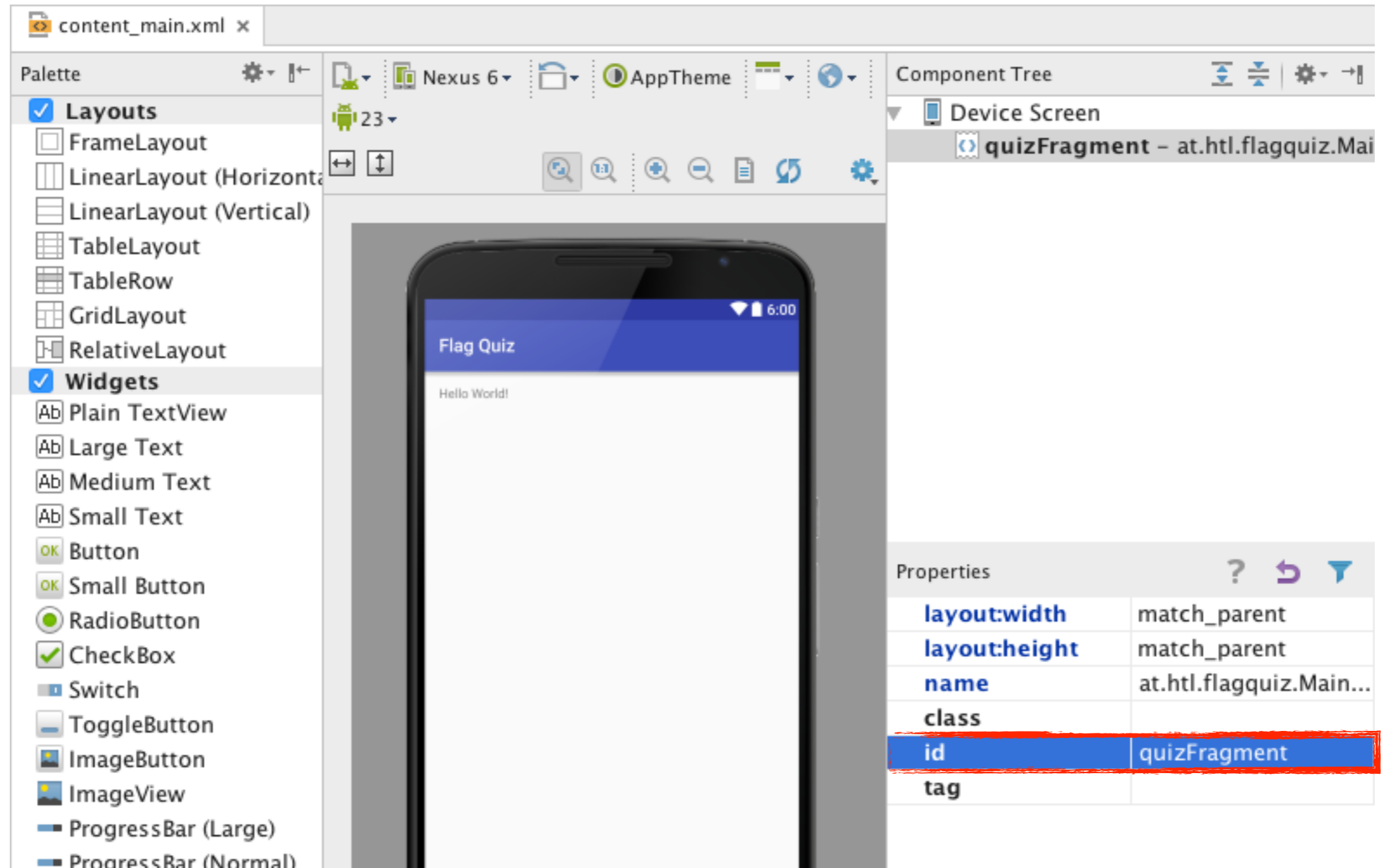
Derzeitige GUI





The **ids** for the **Buttons** in these **LinearLayouts** are not used in this app, so they're not specified in this figure

Fragment umbenennen



Das RelativeLayout von fragment_main.xml in ein LinearLayout ändern

The screenshot displays the Android Studio interface. The central design view shows a mobile device with a purple header labeled "Flag Quiz" and a white content area with the text "Hello World!".

On the left, the "Palette" is visible, showing a list of widgets under the "Widgets" and "Text Fields" categories. The "Widgets" category is expanded, and "CheckBox" is checked. The "Text Fields" category is also expanded, and "Plain Text" is selected.

On the right, the "Component Tree" shows the layout hierarchy. The selected component is "quizLinearLayout (LinearLayout)", which contains a "TextView - 'Hello World!'".

Below the Component Tree, the "Properties" panel is visible, showing the following attributes for the selected component:

layout:width	match_parent
layout:height	match_parent
style	
orientation	vertical
gravity	[]
accessibilityLiveRe	
accessibilityTraver:	
accessibilityTraver:	
alpha	
background	
backgroundTint	
backgroundTintMo	
baselineAligned	<input type="checkbox"/>
baselineAlignedChi	
clickable	<input type="checkbox"/>
contentDescription	
id	quizLinearLayout



Look-and-Feel Observation 4.2

According to the Android design guidelines, 16dp is the recommended space between the edges of a device's touchable screen area and the app's content; however, many apps (such as games) use the full screen.

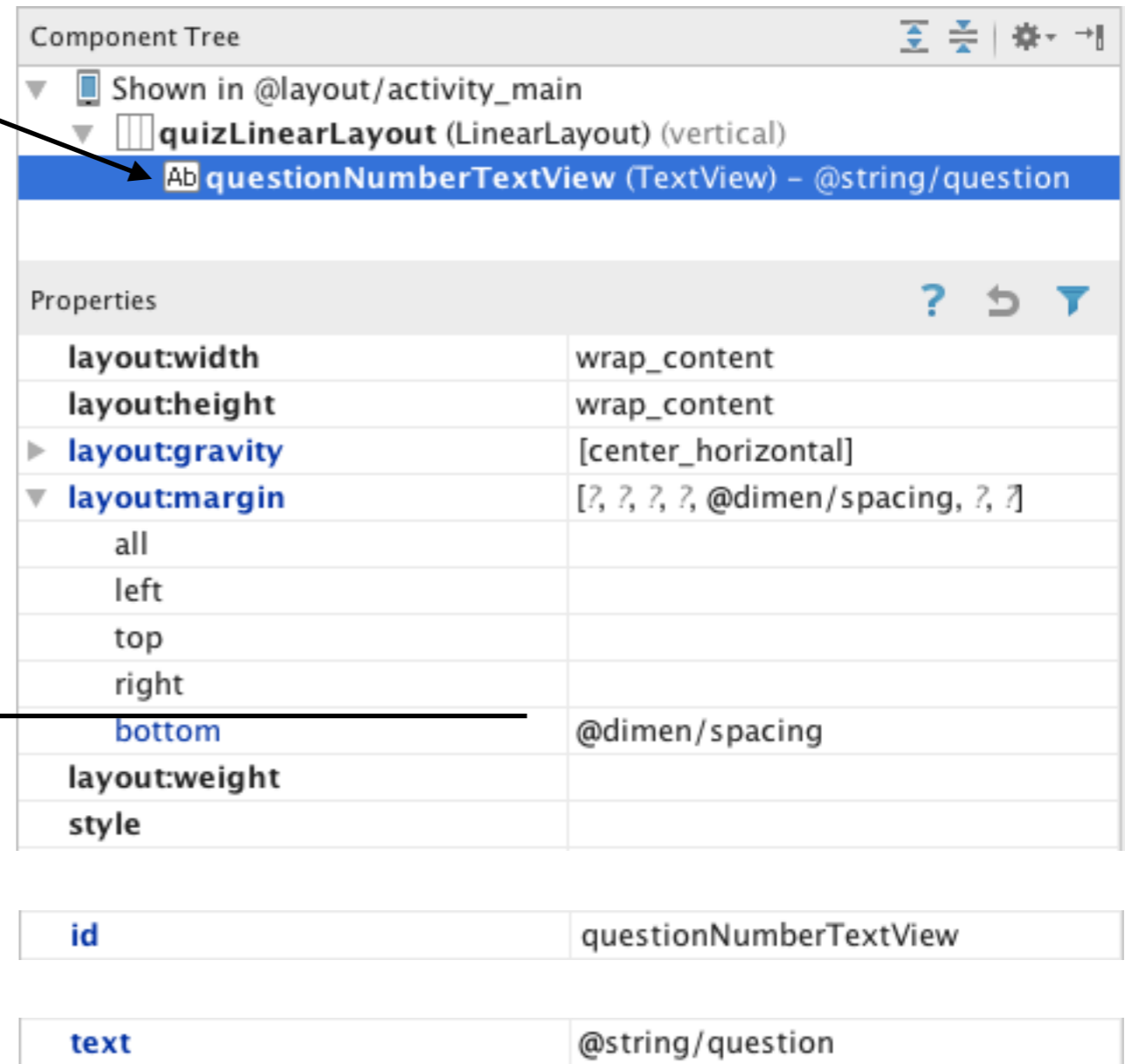
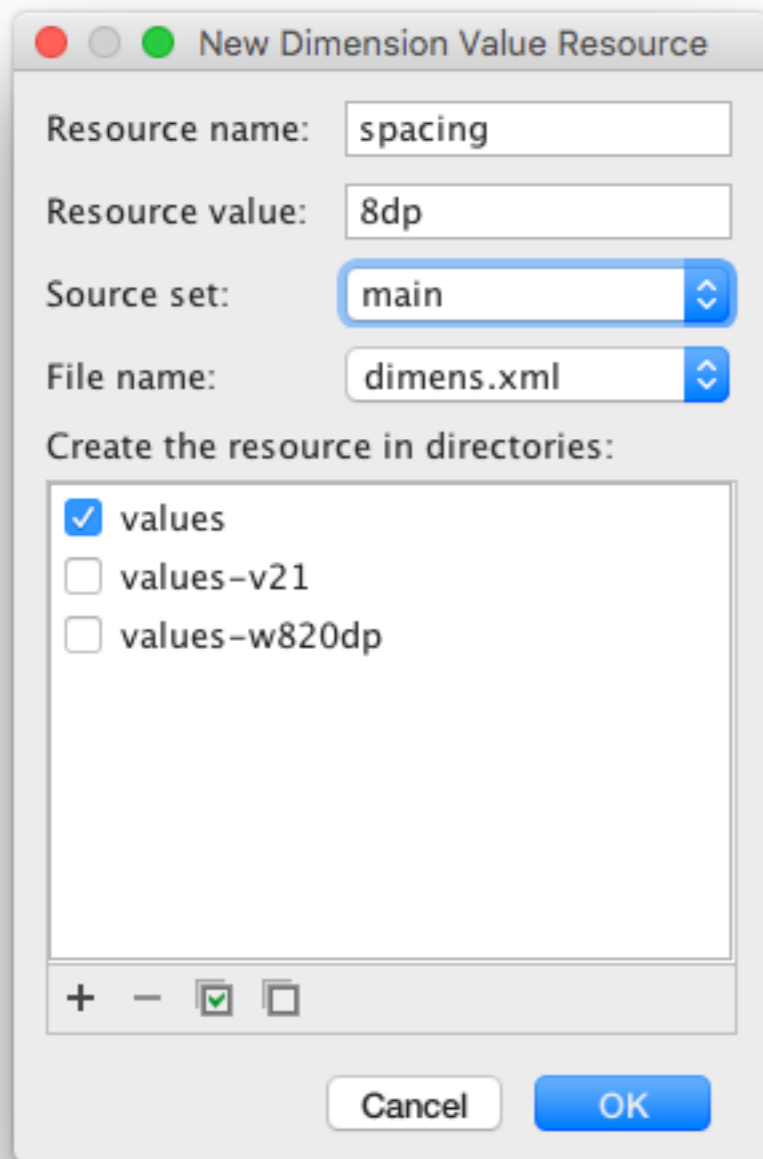
fragment_main.xml ×

LinearLayout


```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="at.htl.flagquiz.MainActivityFragment"
    tools:showIn="@layout/activity_main"
    android:orientation="vertical"
    android:id="@+id/quizLinearLayout"
    android:measureWithLargestChild="true">
```

fragment_main.xml

add a Medium Text



Nexus 6 | AppTheme | MainActivity



Component Tree
 Shown in @layout/activity_main
 quizLinearLayout (LinearLayout) (vertical)
 questionNumberTextView (TextView) - @string/question
flagImageView (ImageView)

Properties

layout:width	match_parent
layout:height	0dp
layout:gravity	[center]
top	<input type="checkbox"/>
bottom	<input type="checkbox"/>
left	<input type="checkbox"/>
right	<input type="checkbox"/>
center	both
layout:margin	[?, @dimen/activity_horizontal_margin
all	
left	@dimen/activity_horizontal_margin
top	
right	@dimen/activity_horizontal_margin
bottom	@dimen/spacing
layout:weight	1
style	
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
adjustViewBounds	<input checked="" type="checkbox"/>
contentDescription	@string/image_description
id	flagImageView
scaleType	fitCenter

- `layout:height: 0dp ...` dadurch wird die Höhe der `ImageView` durch das `layout:weight` - Property bestimmt
- `layout:margin left bzw right ...` diese 16dp Raum links und rechts werden später für die Schüttel-Animation benötigt
- `layout:weight: 1 ...` Da der `weight-default-Wert` für alle Komponenten 0 ist, wird die `flagImageView` wichtiger als die anderen Views und erhält dadurch den gesamten verfügbaren vertikalen Raum
- `adjustViewBounds: true ...` dadurch wird die `aspect ratio` (Seitenverhältnis) der `ImageView` eingehalten
- `scaleType: fitCenter ...` die `ImageView` sollte so skaliert werden, dass entweder die gesamte verfügbare Höhe oder Breite verwendet wird unter Einhaltung der `aspect ratio`. Die jeweils andere Dimension wird zentriert



Look-and-Feel Observation 4.3

Recall that it's considered a best practice in Android to ensure that every GUI component can be used with TalkBack. For components that don't have descriptive text, such as `ImageViews`, set the component's `contentDescription` property.

guessCountryTextView

The screenshot shows the Android Studio interface with the following components:

- Palette:** Under the 'Widgets' section, 'Medium Text' is highlighted with a red box.
- Component Tree:** The hierarchy is shown as: `quizLinearLayout (LinearLayout) (vertical)` containing `questionNumberTextView (TextView) - @string/...`, `flagImageView (ImageView)`, and `guessCountryTextView (TextView) - @string/...`. The `guessCountryTextView` is highlighted with a blue selection bar.
- Properties:** The properties for `guessCountryTextView` are listed below. The `layout:gravity` section is expanded, and the `center` option is selected, with the `horizontal` alignment also highlighted by a red box.

Property	Value
<code>layout:width</code>	<code>wrap_content</code>
<code>layout:height</code>	<code>wrap_content</code>
<code>layout:gravity</code>	<code>[center_horizontal]</code>
top	<input type="checkbox"/>
bottom	<input type="checkbox"/>
left	<input type="checkbox"/>
right	<input type="checkbox"/>
center	<input checked="" type="checkbox"/>
<code>layout:margin</code>	<code>[]</code>
<code>layout:weight</code>	
<code>style</code>	
<code>accessibilityLiveRegion</code>	

<code>id</code>	<code>guessCountryTextView</code>
-----------------	-----------------------------------

<code>text</code>	<code>@string/guess_country</code>
-------------------	------------------------------------

Component Tree

- Shown in @layout/activity_main
 - quizLinearLayout (LinearLayout) (vertical)
 - questionNumberTextView (TextView) - @string/question
 - flagImageView (ImageView)
 - guessCountryTextView (TextView) - @string/guess_country
 - row1LinearLayout (LinearLayout) (horizontal)
 - button - "New Button"
 - button2 - "New Button"

Properties

layout:width	match_parent
layout:height	wrap_content
layout:gravity	☐
layout:margin	☐
layout:weight	
style	
orientation	horizontal
gravity	☐
id	row1LinearLayout

Die Buttons benötigen keine Id, da sie aus dem Code nicht referenziert werden

Es ist zu beachten, dass die Buttons auf das LinearLayout gesetzt werden (eingerückt)

Jedes neue LinearLayout ist auf das quizLinearLayout zu ziehen

Component Tree

- Shown in @layout/activity_main
 - quizLinearLayout (LinearLayout) (vertical)
 - questionNumberTextView (TextView) - @string/question
 - flagImageView (ImageView)
 - guessCountryTextView (TextView) - @string/guess_country
 - row1LinearLayout (LinearLayout) (horizontal)
 - button - "New Button"
 - button2 - "New Button"
 - row2LinearLayout (LinearLayout) (horizontal)
 - button3 - "New Button"
 - button4 - "New Button"
 - row3LinearLayout (LinearLayout) (horizontal)
 - button5 - "New Button"
 - button6 - "New Button"
 - row4LinearLayout (LinearLayout) (horizontal)
 - button7 - "New Button"
 - button8 - "New Button"

Density-independent Pixels

dp ... px * ratio

Scale-independent Pixels

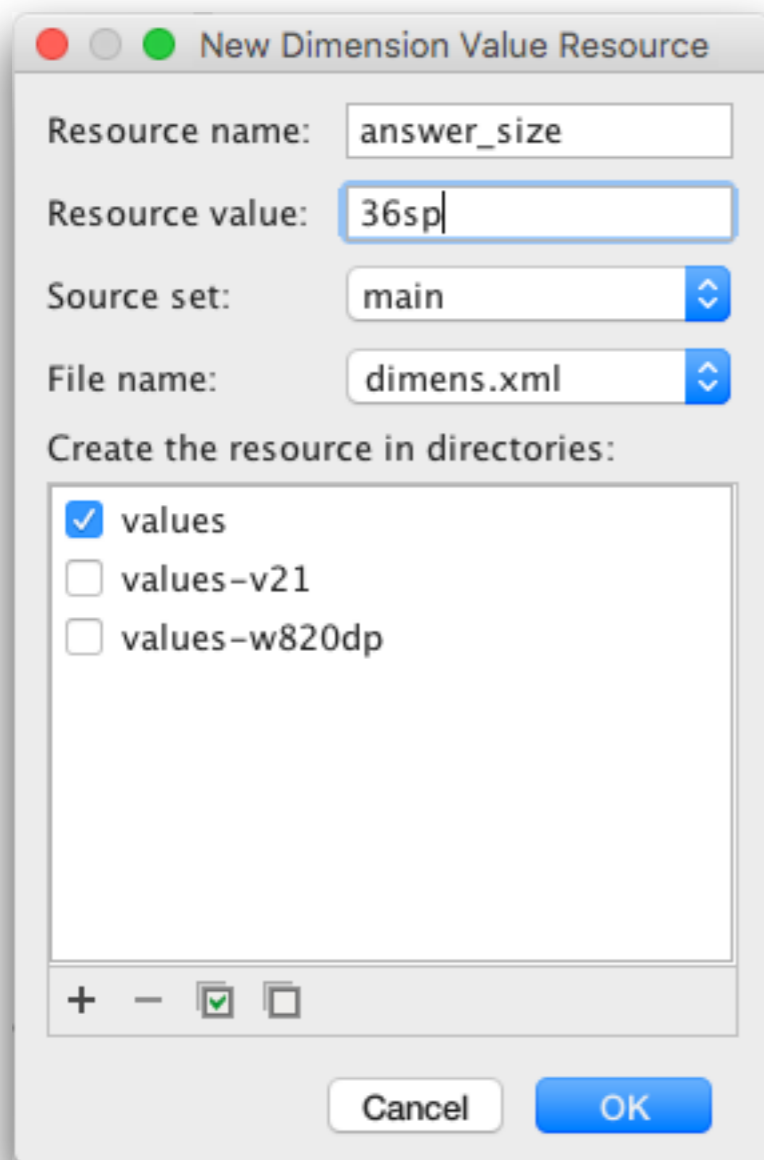
sp ... px * ratio * scale

sp for font sizes

dp for everything else

scale... zB Fontgröße

add a Medium Text

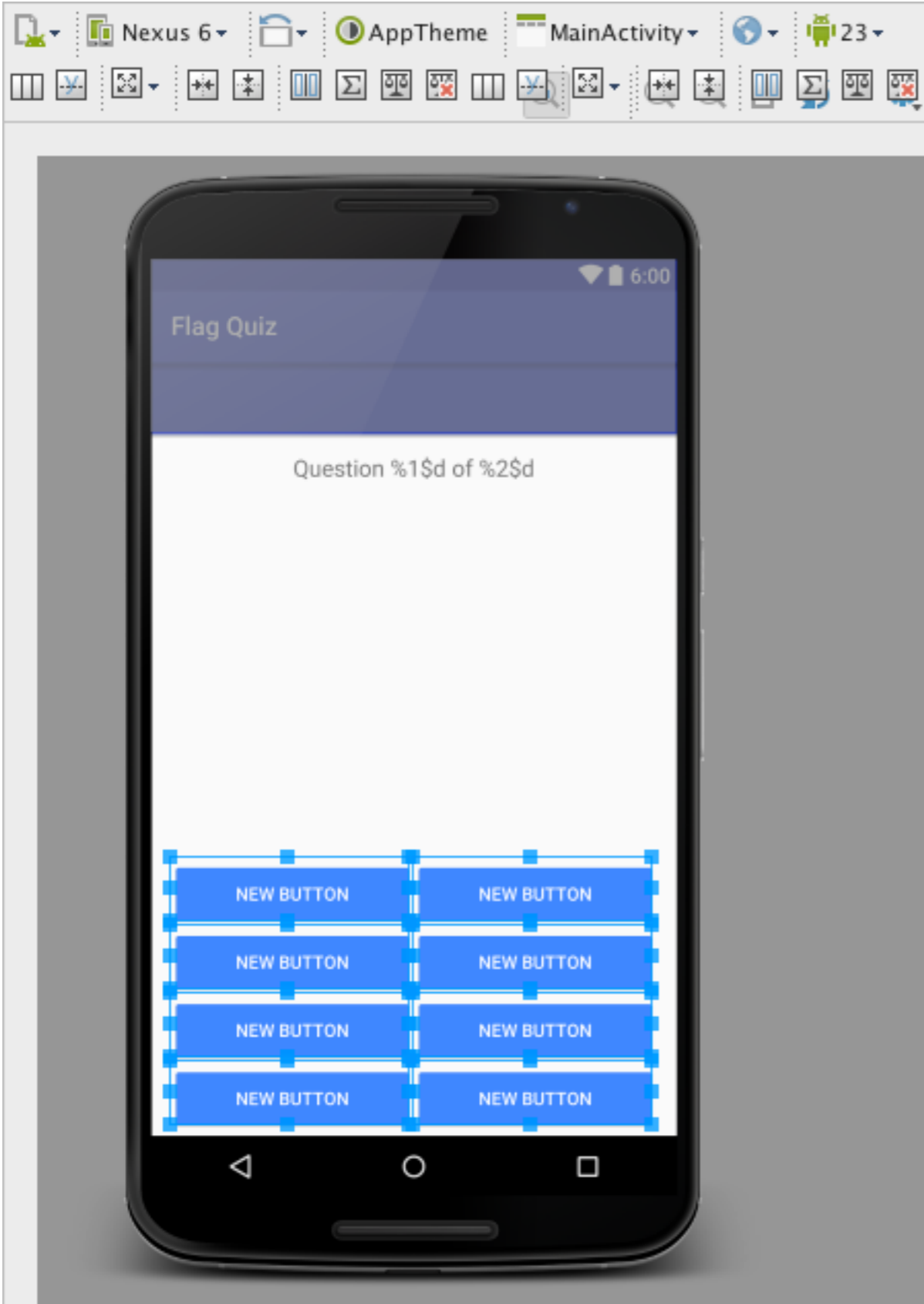


Component Tree

- Shown in @layout/activity_main
 - quizLinearLayout (LinearLayout) (vertical)
 - questionNumberTextView (TextView) - @string/question
 - flagImageView (ImageView)
 - guessCountryTextView (TextView) - @string/guess_country
 - row1LinearLayout (LinearLayout) (horizontal)
 - button - "New Button"
 - button2 - "New Button"
 - row2LinearLayout (LinearLayout) (horizontal)
 - button3 - "New Button"
 - button4 - "New Button"
 - row3LinearLayout (LinearLayout) (horizontal)
 - button5 - "New Button"
 - button6 - "New Button"
 - row4LinearLayout (LinearLayout) (horizontal)
 - button7 - "New Button"
 - button8 - "New Button"
 - answerTextView (TextView) - "Medium Text"

Properties

layout:width	wrap_content
layout:height	wrap_content
layout:gravity	[bottom, center_horizontal]
top	<input type="checkbox"/>
bottom	<input checked="" type="checkbox"/>
left	<input type="checkbox"/>
right	<input type="checkbox"/>
center	horizontal
gravity	[center_horizontal]
id	answerTextView
textSize	@dimen/answer_size
textStyle	[bold]



Component Tree

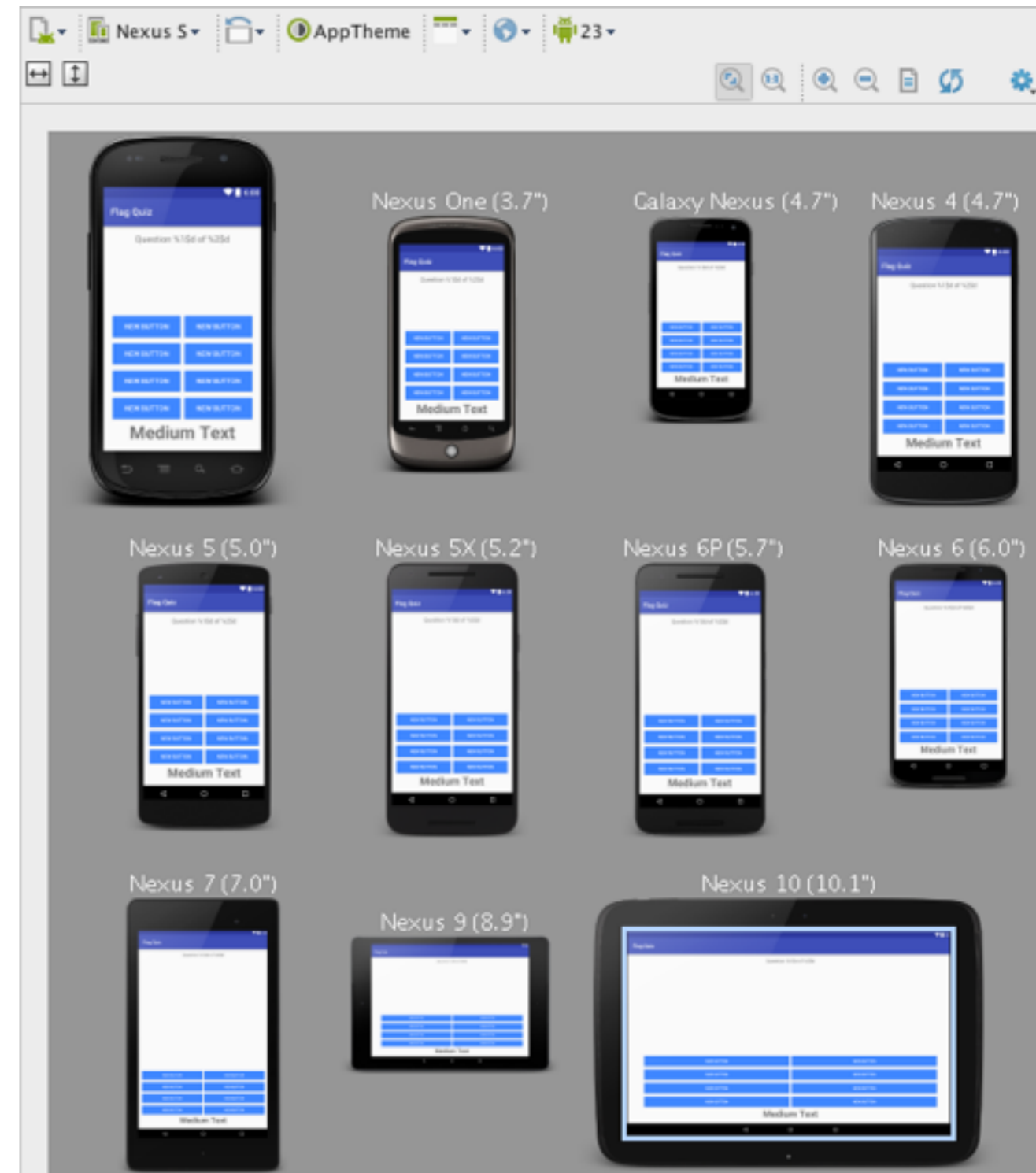
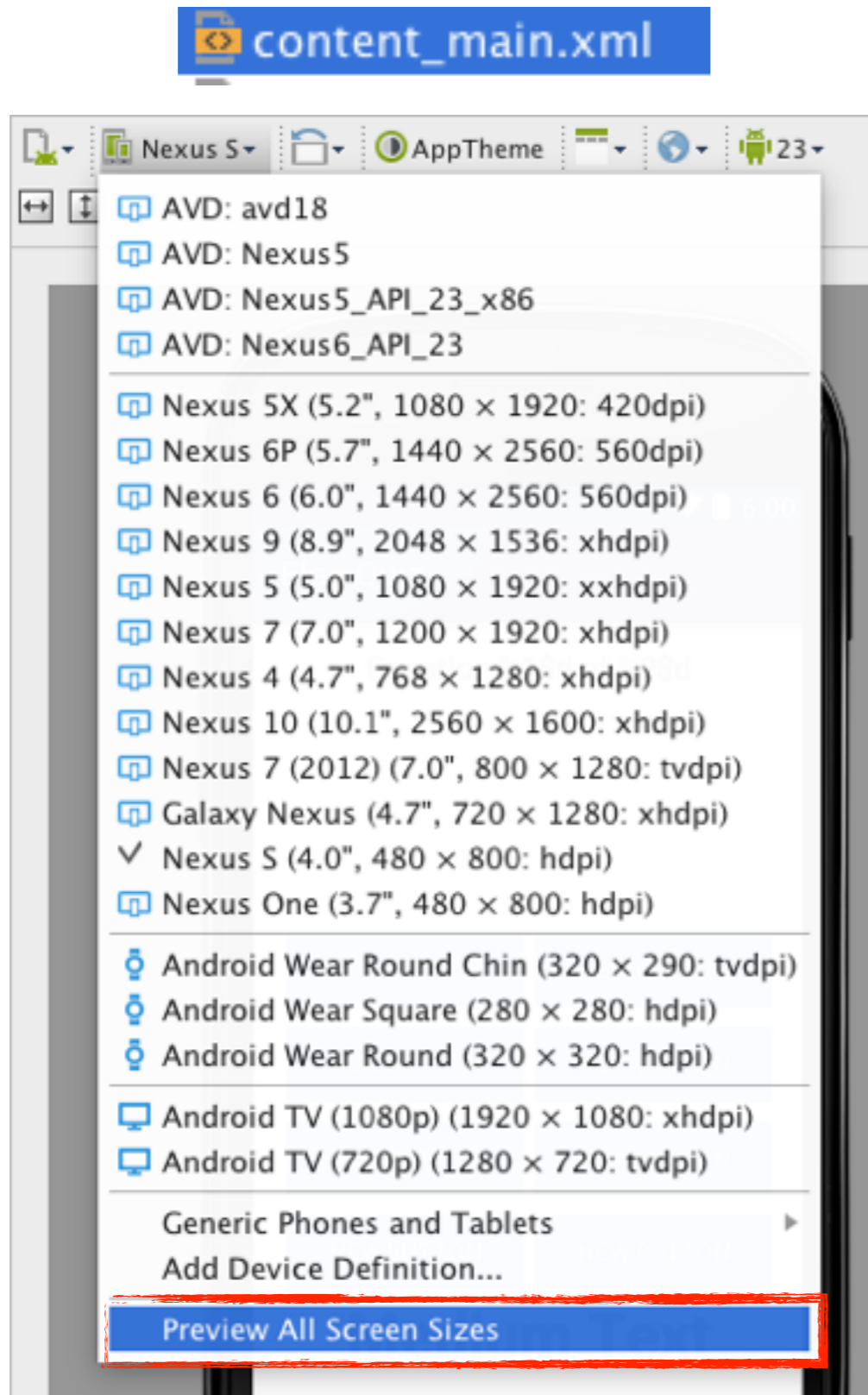
- Shown in @layout/activity_main
 - quizLinearLayout (LinearLayout) (vertical)
 - questionNumberTextView (TextView) - @string/question
 - flagImageView (ImageView)
 - row1LinearLayout (LinearLayout) (horizontal)
 - button - "New Button"
 - button2 - "New Button"
 - row2LinearLayout (LinearLayout) (horizontal)
 - button3 - "New Button"
 - button4 - "New Button"
 - row3LinearLayout (LinearLayout) (horizontal)
 - button5 - "New Button"
 - button6 - "New Button"
 - row4LinearLayout (LinearLayout) (horizontal)
 - button7 - "New Button"
 - button8 - "New Button"
 - answerTextView (TextView) - "Medium Text"

Properties

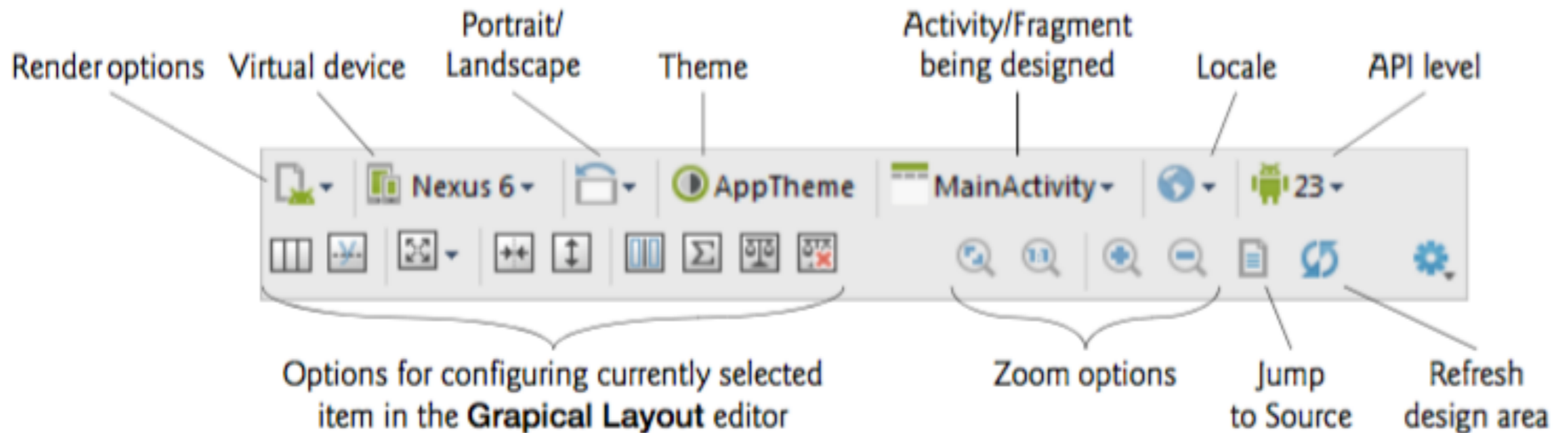
layout:width	0dp
layout:height	match_parent
layout:gravity	[]
layout:margin	[]
layout:weight	1
style	@android:style/Widget.Material.Button.Colored
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
allowUndo	<input type="checkbox"/>
alpha	
lines	2
textColor	@color/button_text_color

- Durch das Setzen von `layout:width` auf `0dp` und `layout:weight` auf `1` wird der verfügbare horizontale Raum auf die beiden Buttons gleich aufgeteilt
- Durch `layout:height` auf `match_parent` nutzen die Buttons die verfügbare Höhe der horizontalen Layouts
- `lines : 2` bewirkt, dass die Buttons die gleiche Höhe haben, auch wenn der Text zwei Zeilen beansprucht. Text, der länger als zwei Zeilen ist, wird abgeschnitten
- `style: @android:style/Widget.Material.Button.Colored` bewirkt farbige Buttons, entsprechend dem Theme
- `textColor : @color/button_text_color` bewirkt, dass sich die Farbe des Buttons ändert, je nachdem ob er `enabled` / `disabled` ist.

Preview All Screen Sizes



Canvas Configuration Options

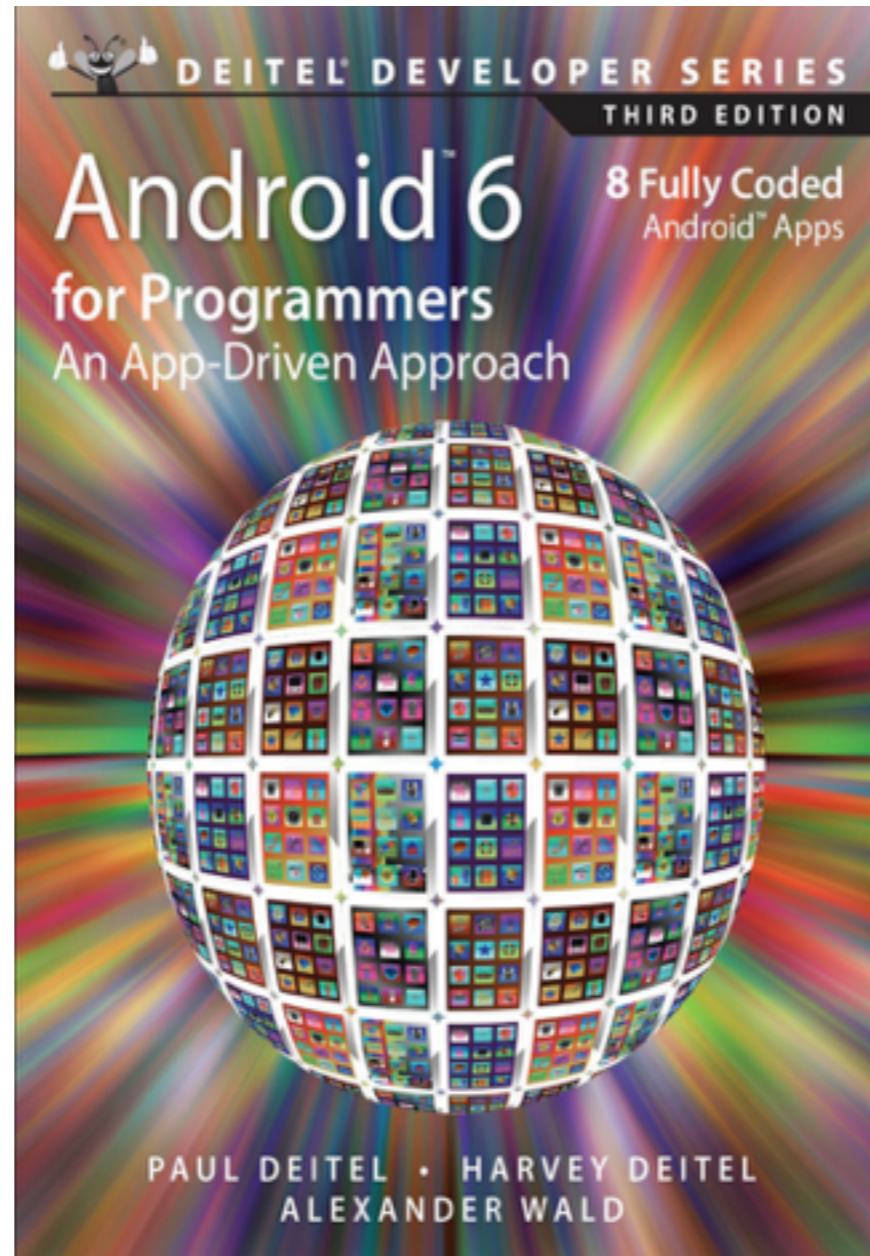


to be continued ...



Noch
Fragen?

Quelle



<http://www.deitel.com/Books/Android/AndroidforProgrammersAnAppDrivenApproach3/tabid/3671/Default.aspx>