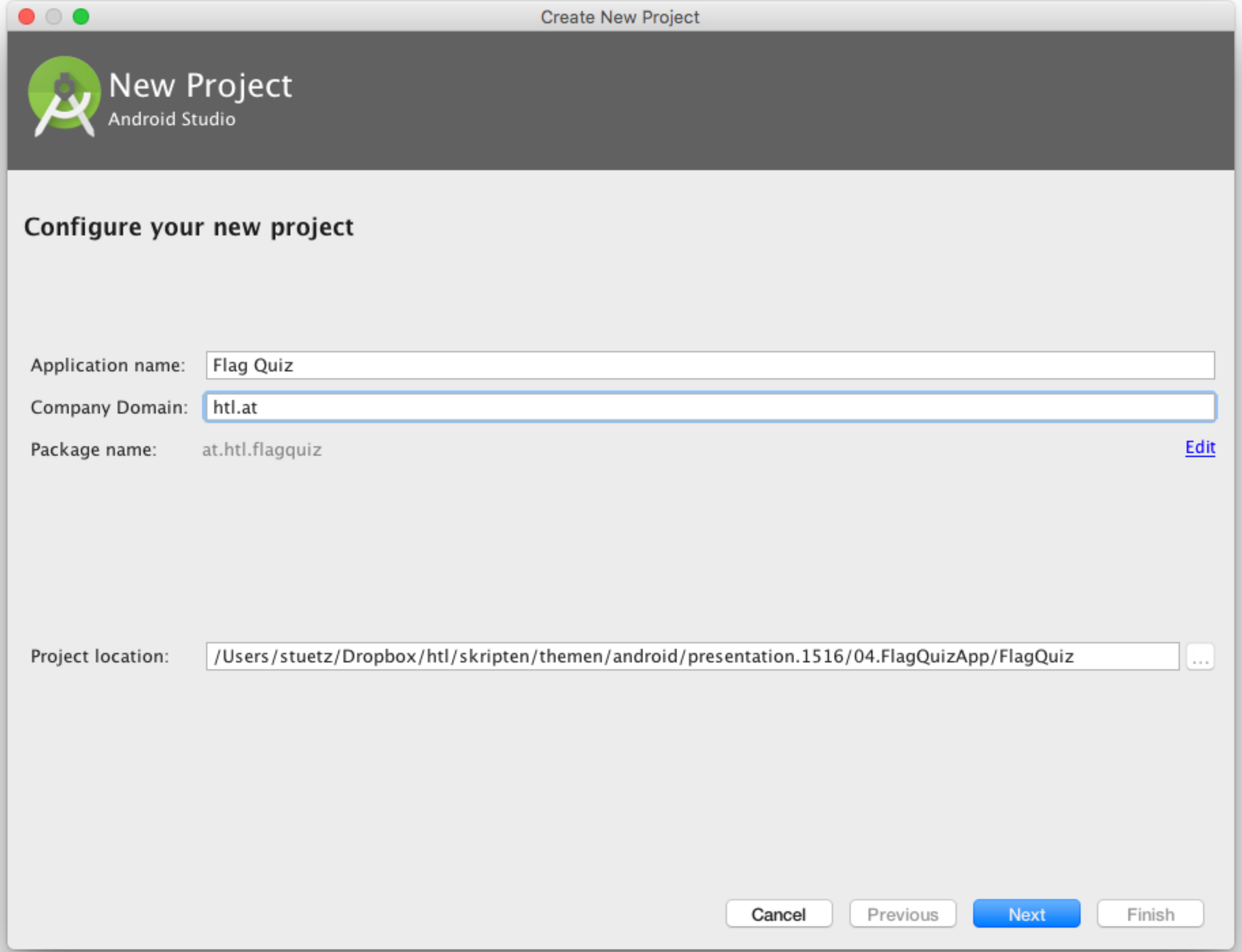
Flag Quiz App

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



Create New Project
New Project
Android Studio

Configure your new project

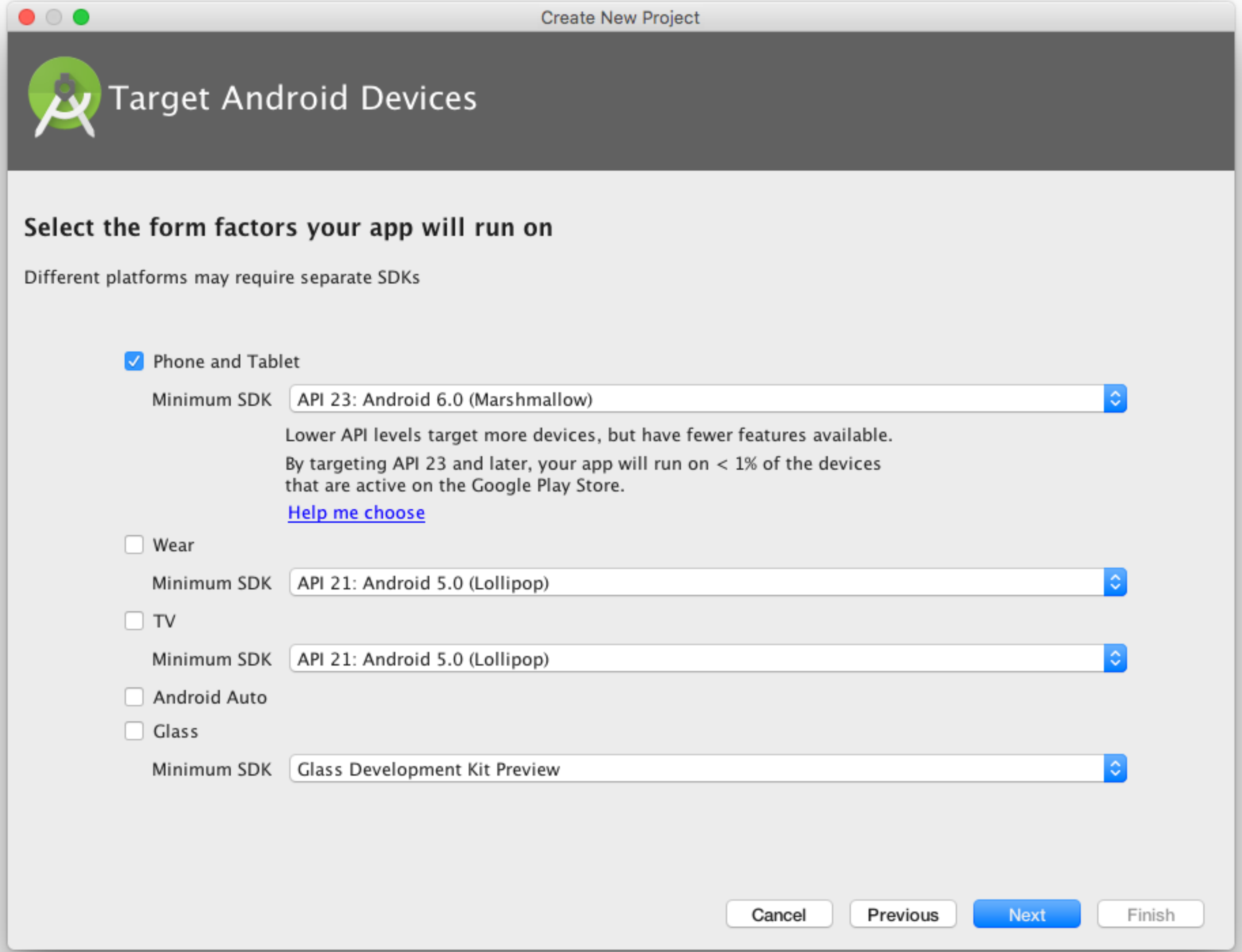
Application name: Flag Quiz

Company Domain: htl.at

Package name: at.htl.flagquiz

Project location: /Users/stuetz/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz

Edit
...
Cancel
Previous
Next
Finish



Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 23: Android 6.0 (Marshmallow)

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: API 21: Android 5.0 (Lollipop)

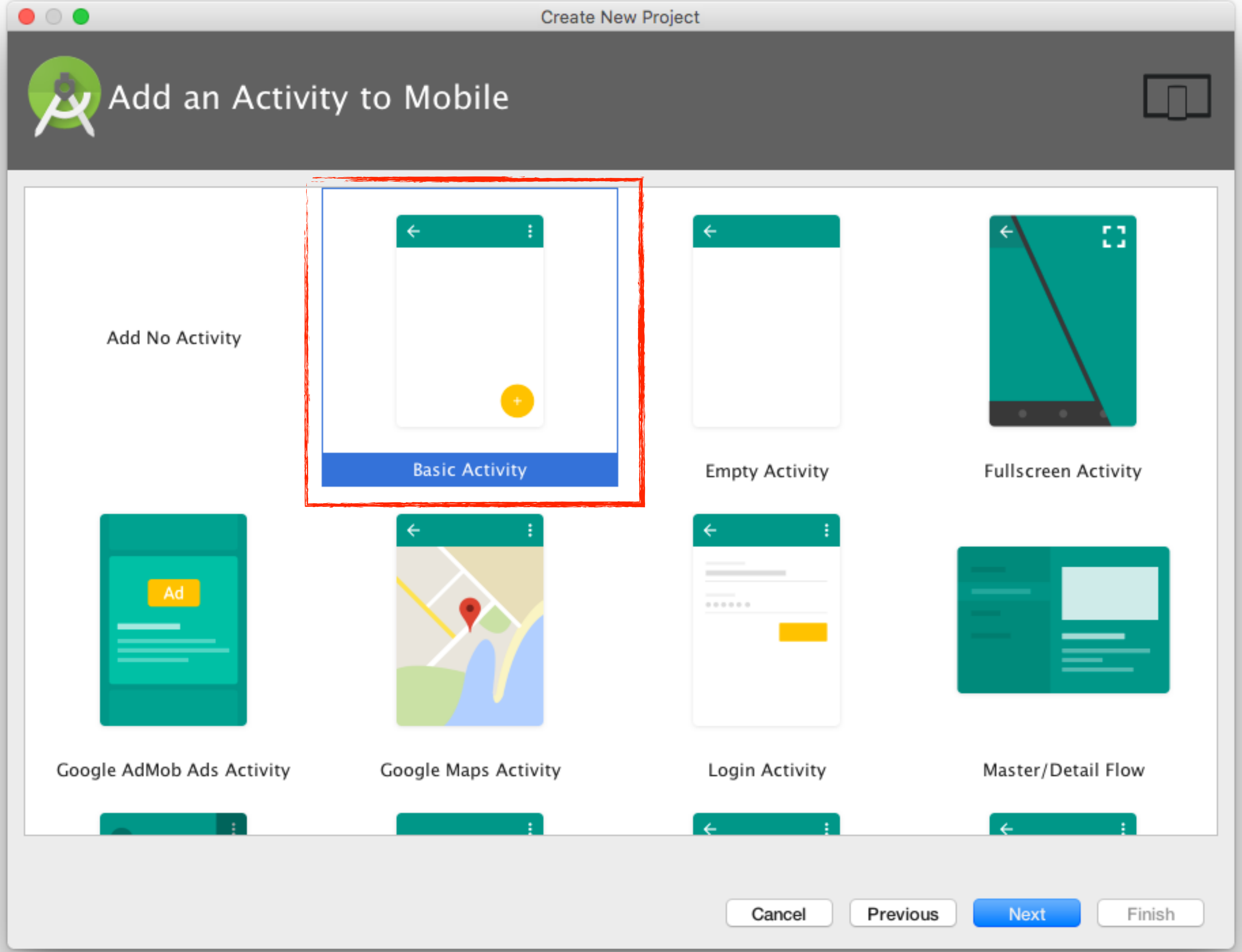
☐ TV

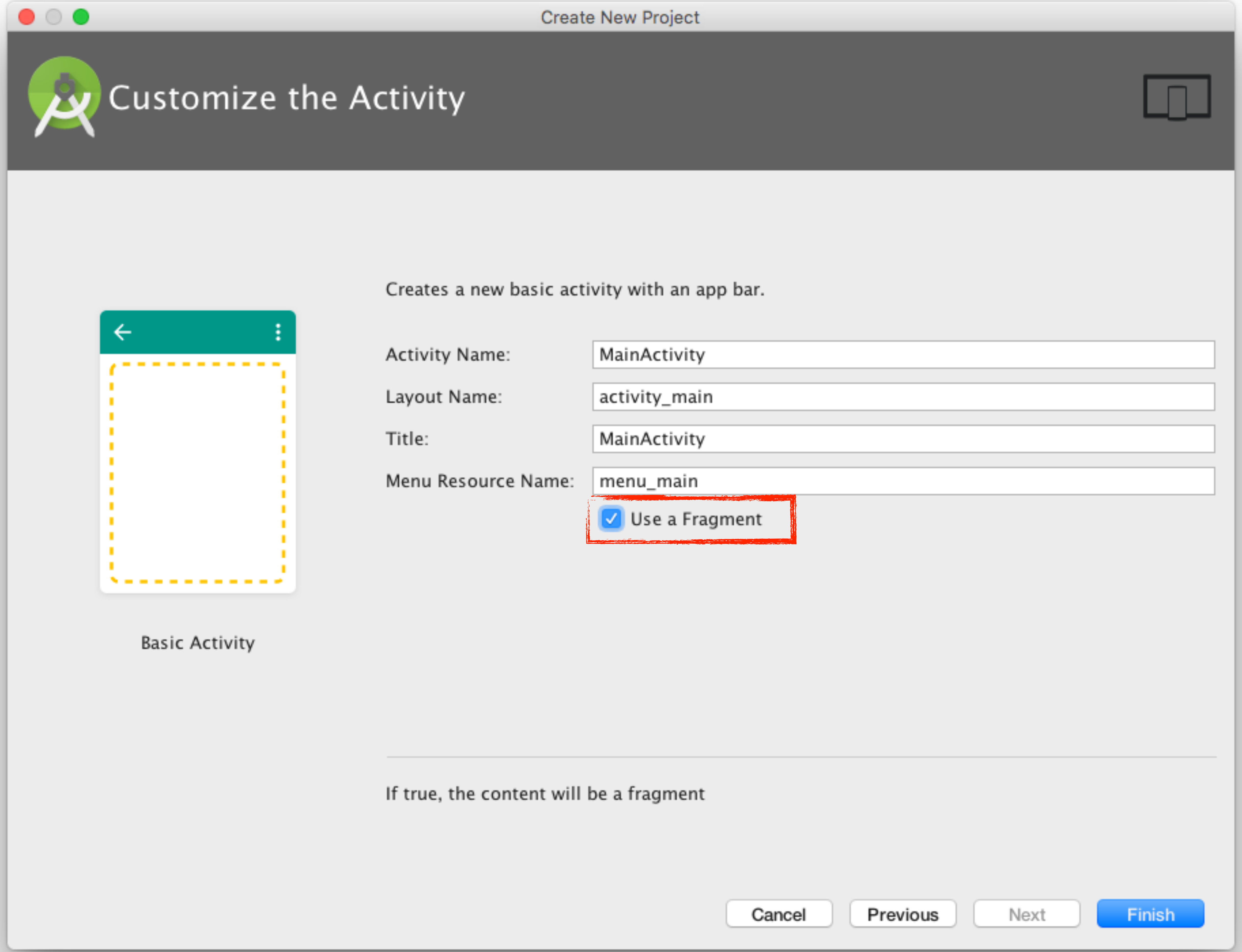
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass

Minimum SDK: Glass Development Kit Preview


Create New Project
Add an Activity to Mobile
Add No Activity
Basic Activity
Empty Activity
Fullscreen Activity
Ad
Google AdMob Ads Activity
Google Maps Activity
Login Activity
Master/Detail Flow
Cancel
Previous
Next
Finish



Create New Project

Customize the Activity

Creates a new basic activity with an app bar.

Activity Name: MainActivity

Layout Name: activity_main

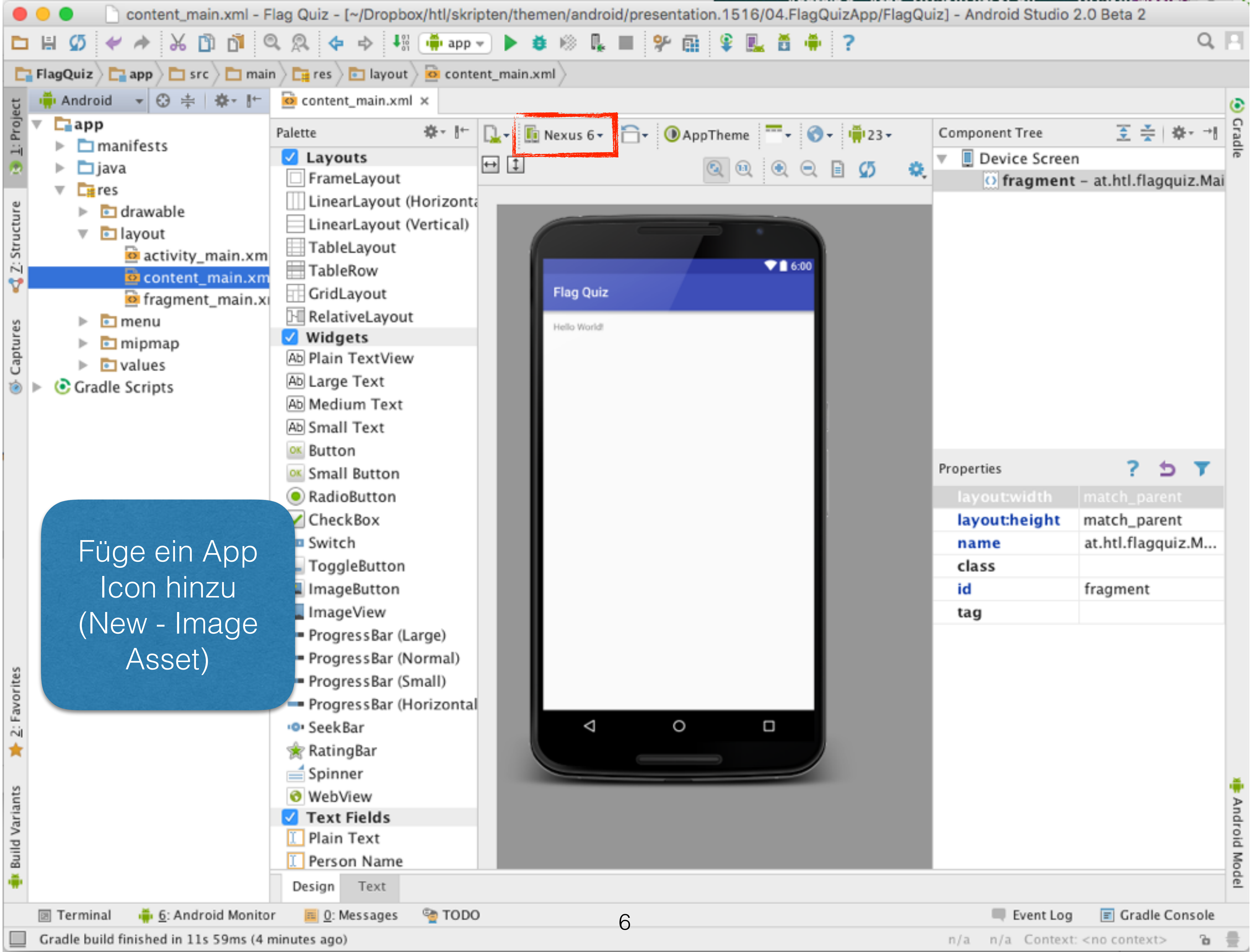
Title: MainActivity

Menu Resource Name: menu_main

☑ Use a Fragment

If true, the content will be a fragment

Basic Activity
Cancel
Previous
Next
Finish


content_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
FlagQuiz
app
src
main
res
layout
content_main.xml
Android
app
manifests
java
res
drawable
layout
activity_main.xm
content_main.xm
fragment_main.xı
menu
mipmap
values
Gradle Scripts
Palette
Layouts
FrameLayout
LinearLayout (Horizonta
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
Design
Text
Nexus 6
AppTheme
23
6:00
Flag Quiz
Hello World!
Component Tree
Device Screen
fragment – at.htl.flagquiz.Mai
Properties
Terminal
6: Android Monitor
0: Messages
TODO
Event Log
Gradle Console
1: Project
7: Structure
Captures
2: Favorites
Build Variants
Gradle
Android Model
n/a
n/a
Context: <no context>

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

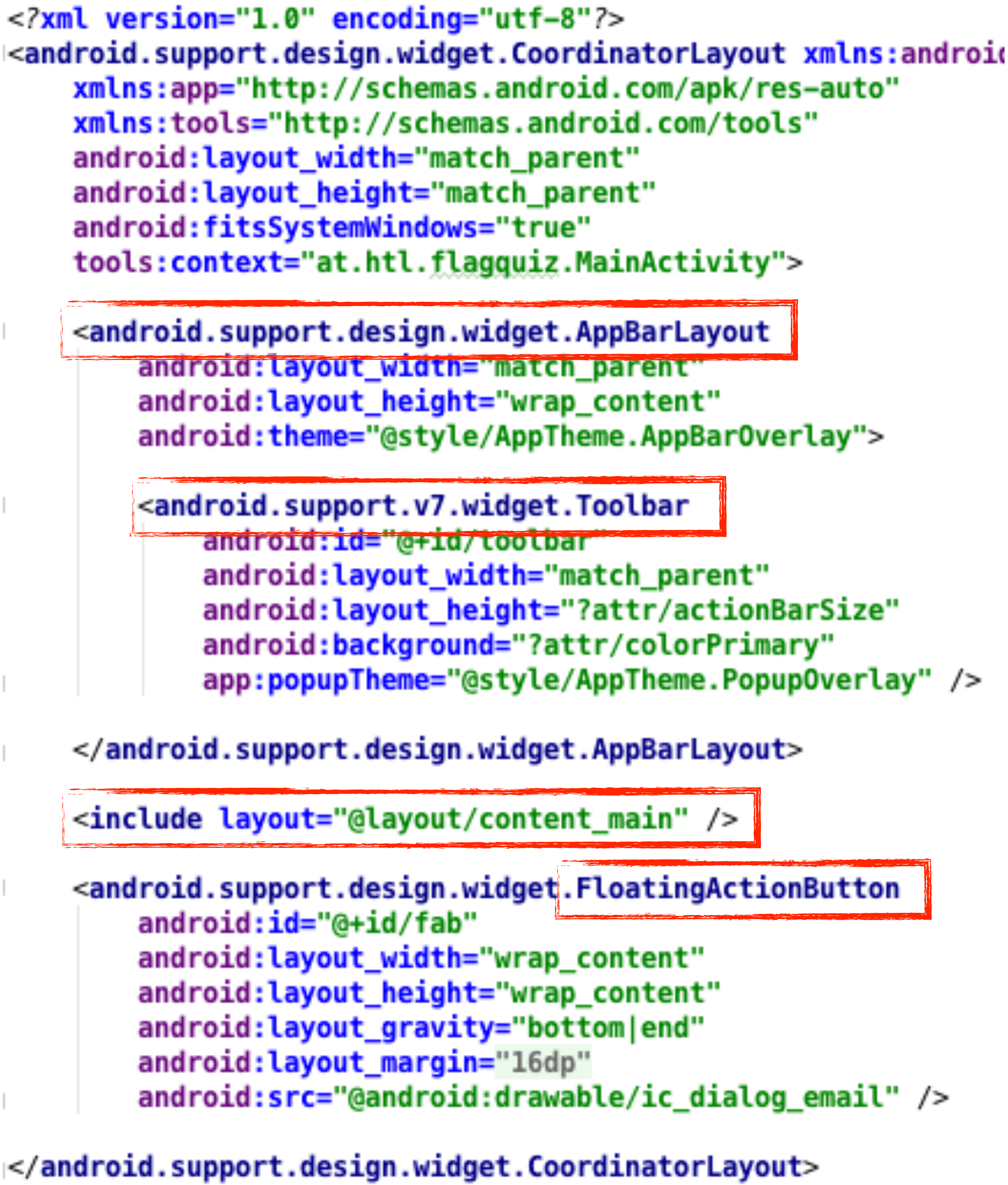
Gradle build finished in 11s 59ms (4 minutes ago)

Füge ein App Icon hinzu (New - Image Asset)

Schedule
TOMORROW
TODAY
Breakf
Ended

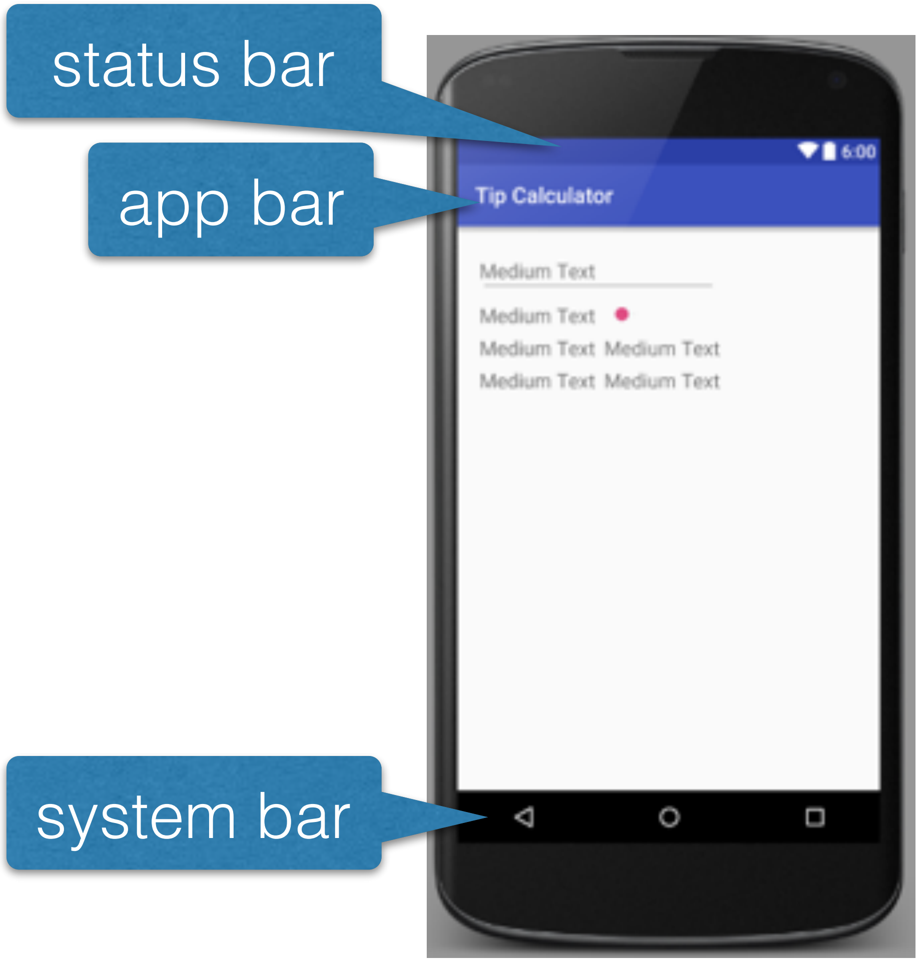
activity_main.xml

- 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



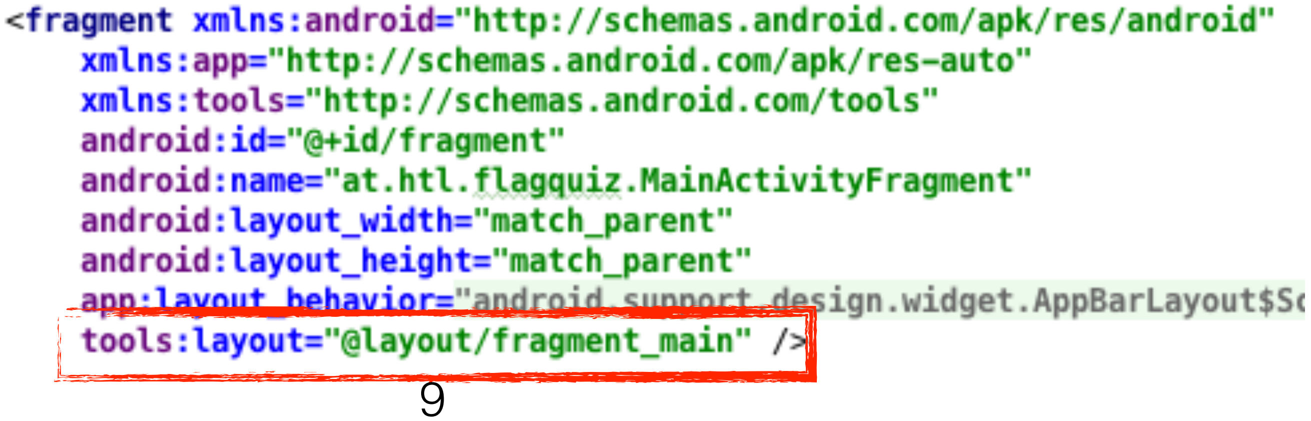
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    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


status bar
app bar
system bar
6:00
Tip Calculator
Medium Text
Medium Text
Medium Text Medium Text
Medium Text Medium Text

- content_main.xml definiert den Bereich der MainActivity-GUI unter der app bar und über der system bar

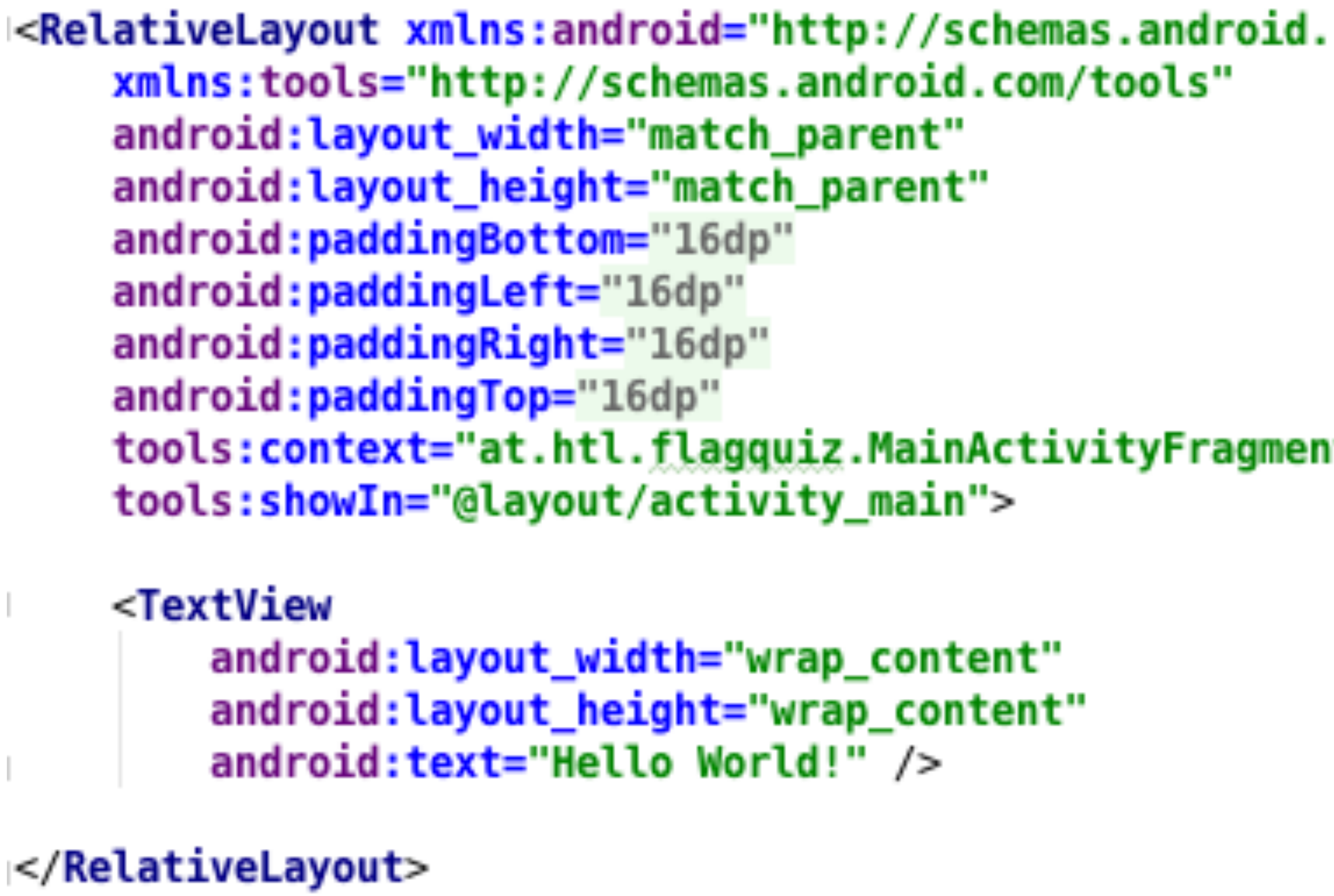
- enthält nur ein <fragment>-Element (bei Verwendung von Fragments)



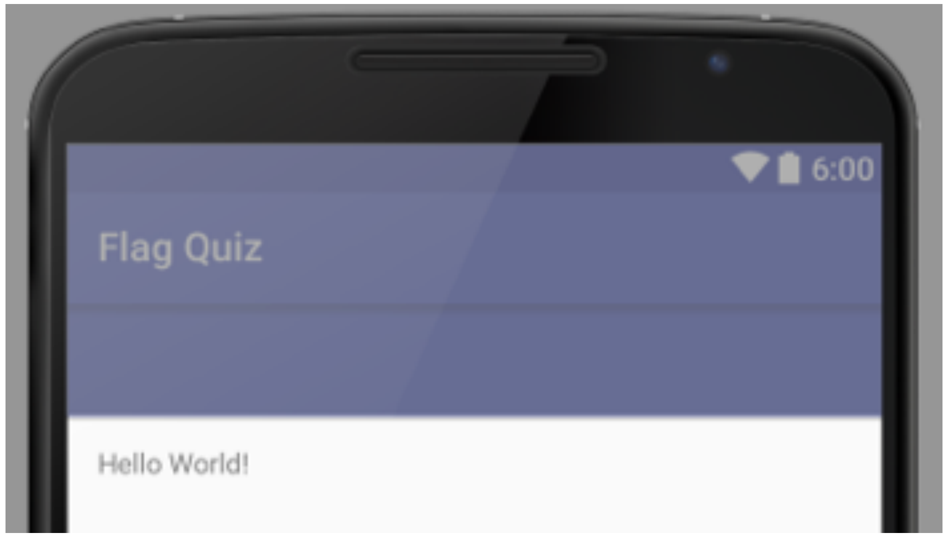
```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragment"
    android:name="at.htl.flagquiz.MainActivityFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="android.support.design.widget.AppBarLayout$S
    tools:layout="@layout/fragment_main" />
```

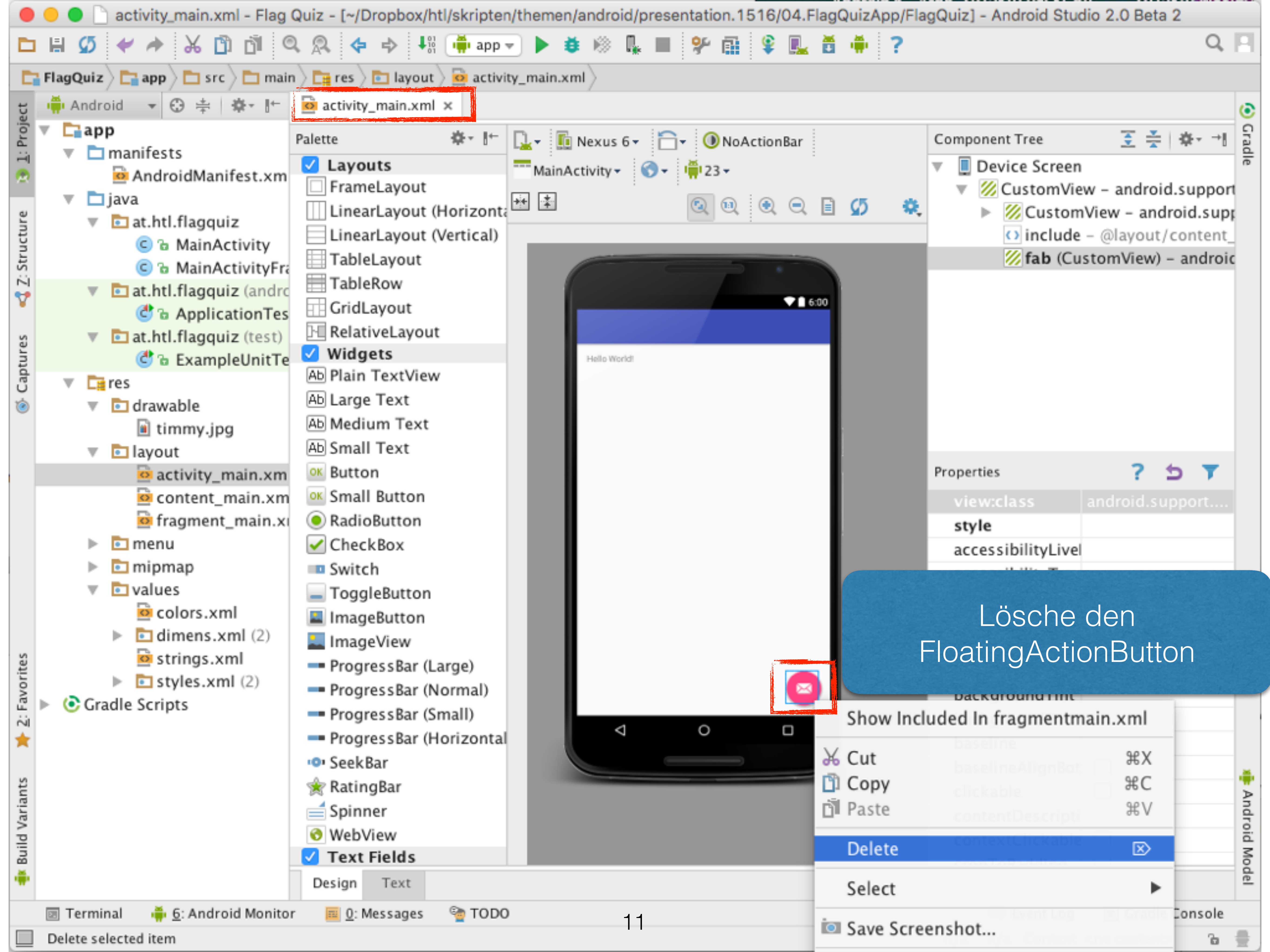
fragment_main.xml

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

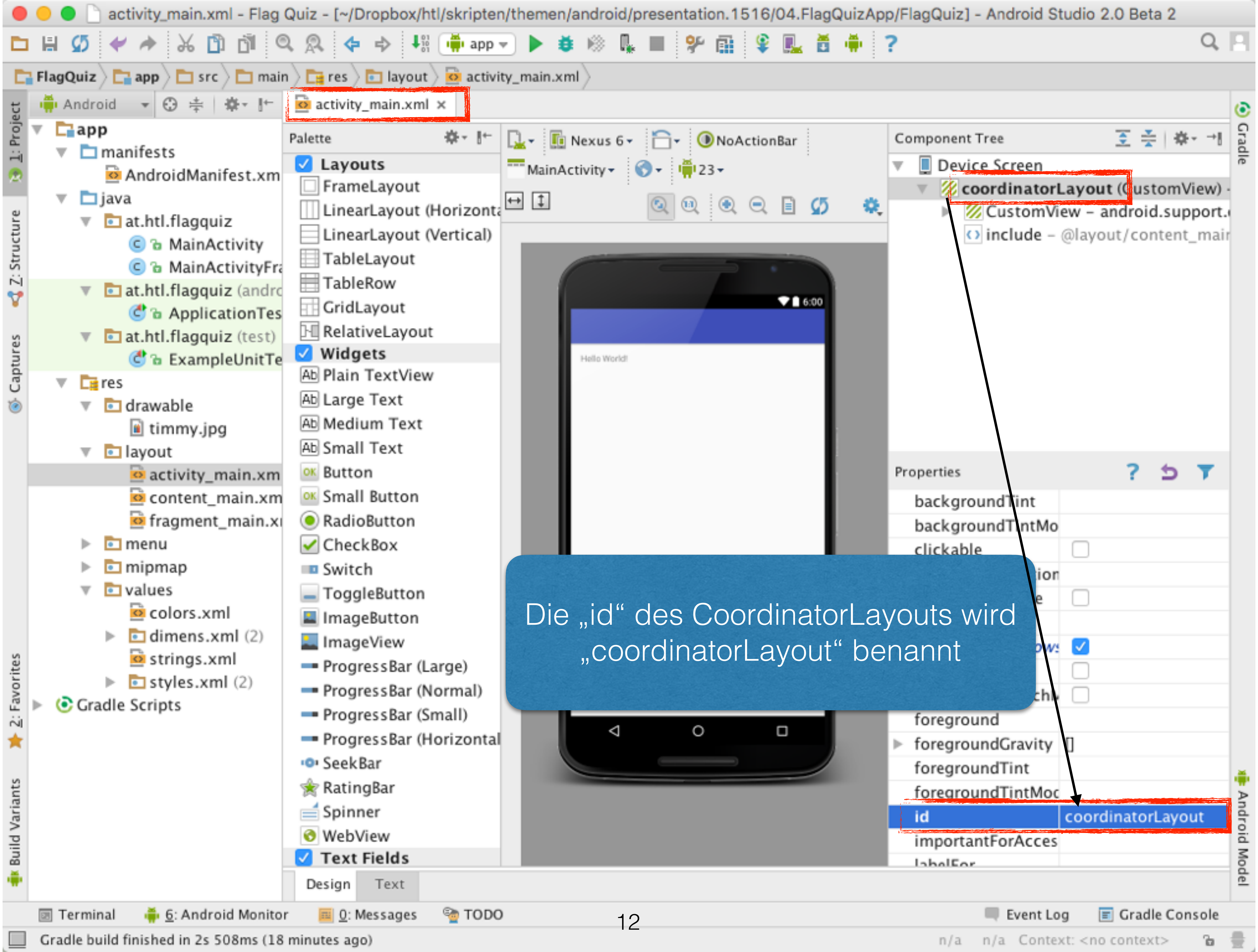


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    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>
```


6:00
Flag Quiz
Hello World!


activity_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
FlagQuiz
app
src
main
res
layout
activity_main.xml
Palette
Layouts
Widgets
Text Fields
Component Tree
Device Screen
CustomView – android.support
include – @layout/content_
fab (CustomView) – androic
Properties
Nexus 6
NoActionBar
MainActivity
23
Hello World!
Design
Text
Show Included In fragmentmain.xml
Cut ⌘X
Copy ⌘C
Paste ⌘V
Delete
Select
Save Screenshot...
Terminal
6: Android Monitor
0: Messages
TODO
Delete selected item

Lösche den FloatingActionButton


activity_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2

Component Tree:
- Device Screen
  - coordinatorLayout (CustomView) -
    - CustomView - android.support.
    - include - @layout/content_mair

Properties:

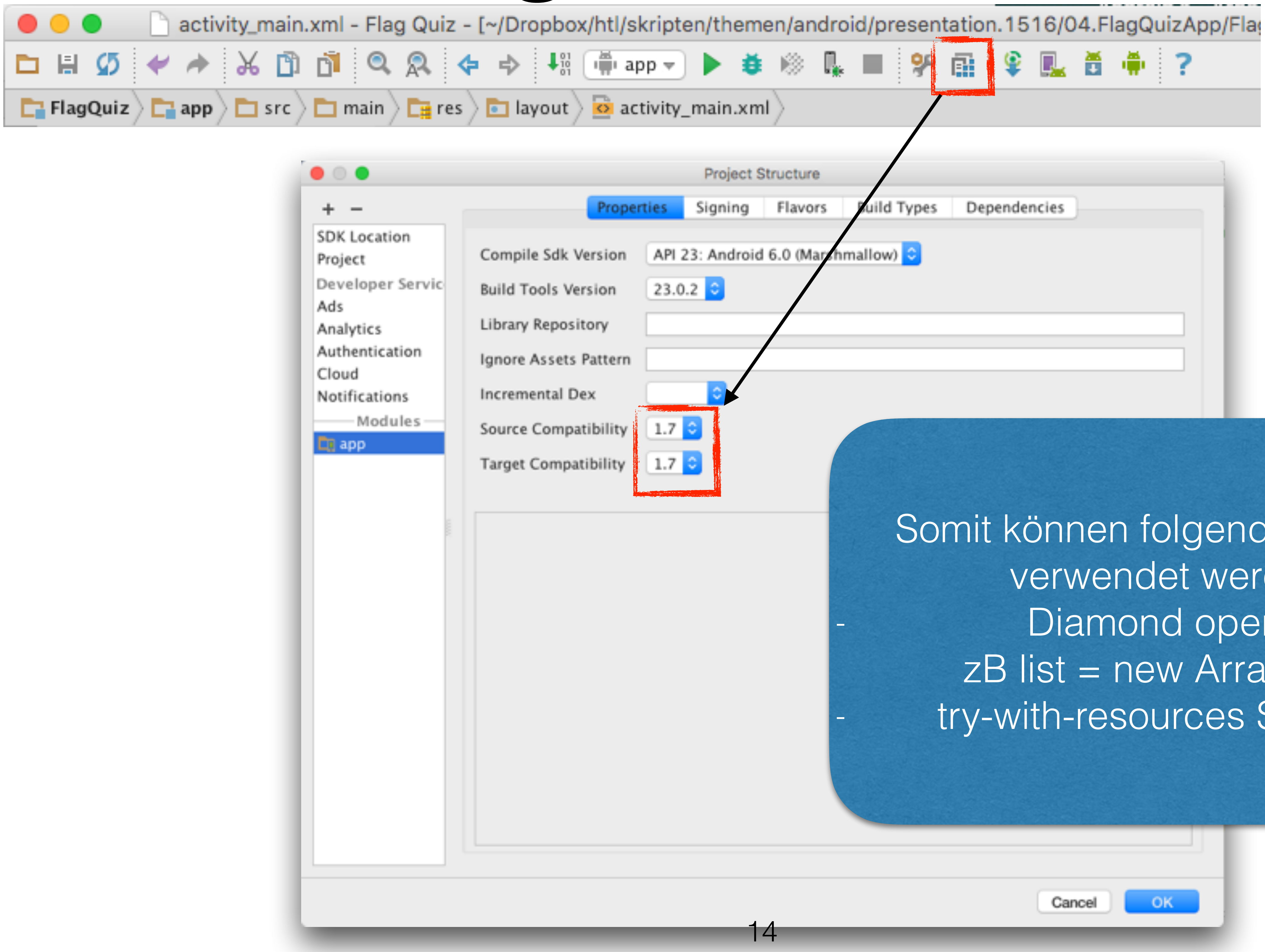
id	coordinatorLayout

FlagQuiz
app
src
main
res
layout
activity_main.xml
Palette
Layouts
Widgets
Text Fields
Nexus 6
NoActionBar
MainActivity
23
Hello World!
Design
Text
Terminal
6: Android Monitor
0: Messages
TODO
Event Log
Gradle Console
Gradle build finished in 2s 508ms (18 minutes ago)

Die „id“ des CoordinatorLayouts wird „coordinatorLayout“ benannt

Vorbereitende Arbeiten

Konfiguriere Java 7


activity_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/Fla

FlagQuiz 〉 app 〉 src 〉 main 〉 res 〉 layout 〉 activity_main.xml

Project Structure

app
+ −
SDK Location
Project
Developer Servic
Ads
Analytics
Authentication
Cloud
Notifications
Modules
app
Properties
Signing
Flavors
Build Types
Dependencies
Cancel
OK

Compile Sdk Version: API 23: Android 6.0 (Marshmallow)

Build Tools Version: 23.0.2

Library Repository:

Ignore Assets Pattern:

Incremental Dex:

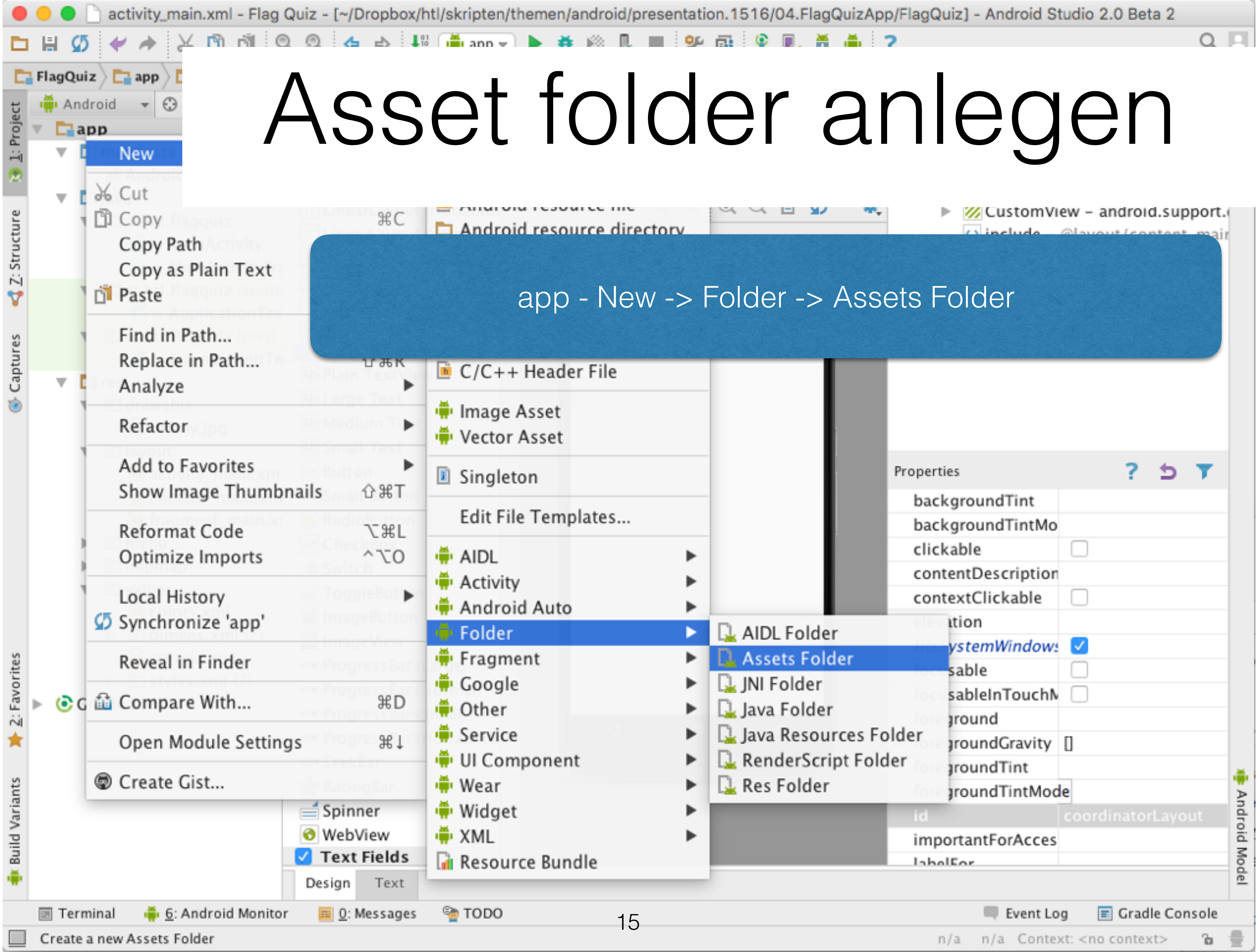
Source Compatibility: 1.7

Target Compatibility: 1.7

Somit können folgende Features verwendet werden:

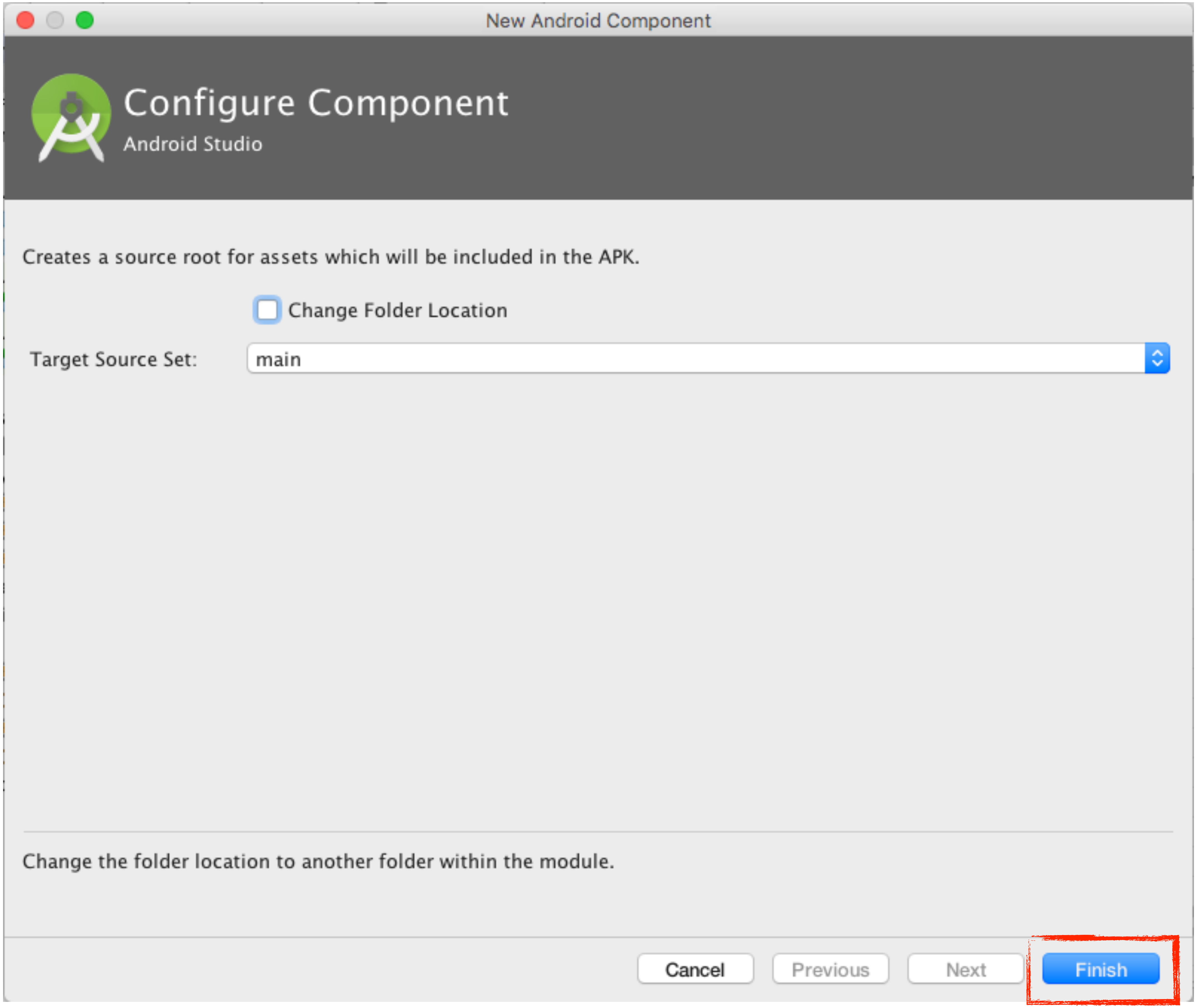
- Diamond operators
  zB list = new ArrayList<>()
- try-with-resources Statements

Asset folder anlegen



app - New -> Folder -> Assets Folder

activity_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
FlagQuiz
app
Android
New
Cut
Copy ⌘C
Copy Path
Copy as Plain Text
Paste
Find in Path...
Replace in Path...
Analyze
Refactor
Add to Favorites
Show Image Thumbnails ⇧⌘T
Reformat Code ⌥⌘L
Optimize Imports ^⌥O
Local History
Synchronize 'app'
Reveal in Finder
Compare With... ⌘D
Open Module Settings ⌘↓
Create Gist...
Android resource directory
C/C++ Header File
Image Asset
Vector Asset
Singleton
Edit File Templates...
AIDL
Activity
Android Auto
Folder
Fragment
Google
Other
Service
UI Component
Wear
Widget
XML
Resource Bundle
AIDL Folder
Assets Folder
JNI Folder
Java Folder
Java Resources Folder
RenderScript Folder
Res Folder
Spinner
WebView
Text Fields
Design
Text
CustomView - android.support.
Properties
backgroundTint
backgroundTintMo
clickable
contentDescription
contextClickable
id
coordinatorLayout
importantForAcces
1: Project
Z: Structure
Captures
2: Favorites
Build Variants
Android Model
Terminal
6: Android Monitor
0: Messages
TODO
Event Log
Gradle Console
Create a new Assets Folder
n/a
n/a
Context: <no context>



New Android Component
Configure Component
Android Studio

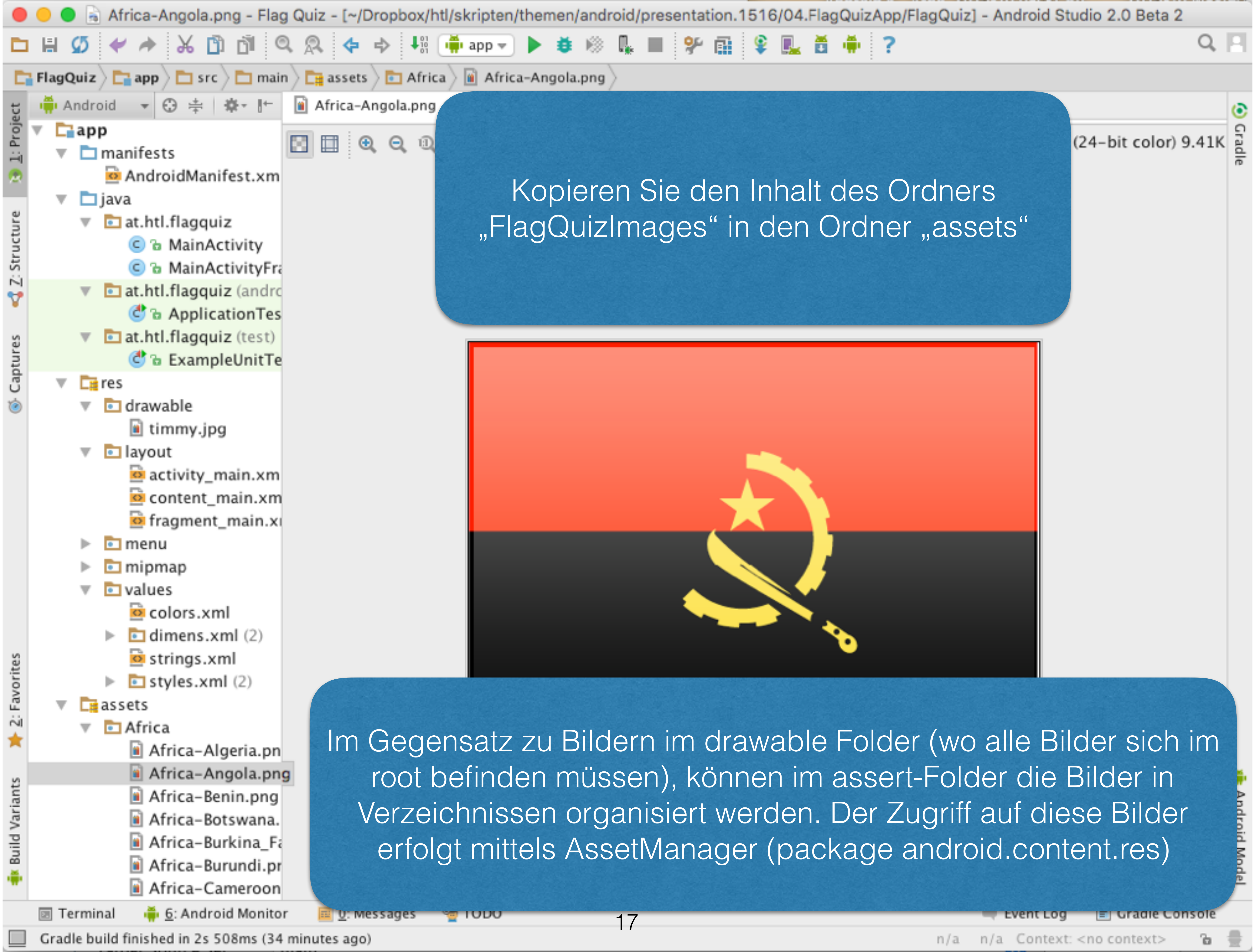
Creates a source root for assets which will be included in the APK.

☐ Change Folder Location

Target Source Set: main

Change the folder location to another folder within the module.

Cancel
Previous
Next
Finish

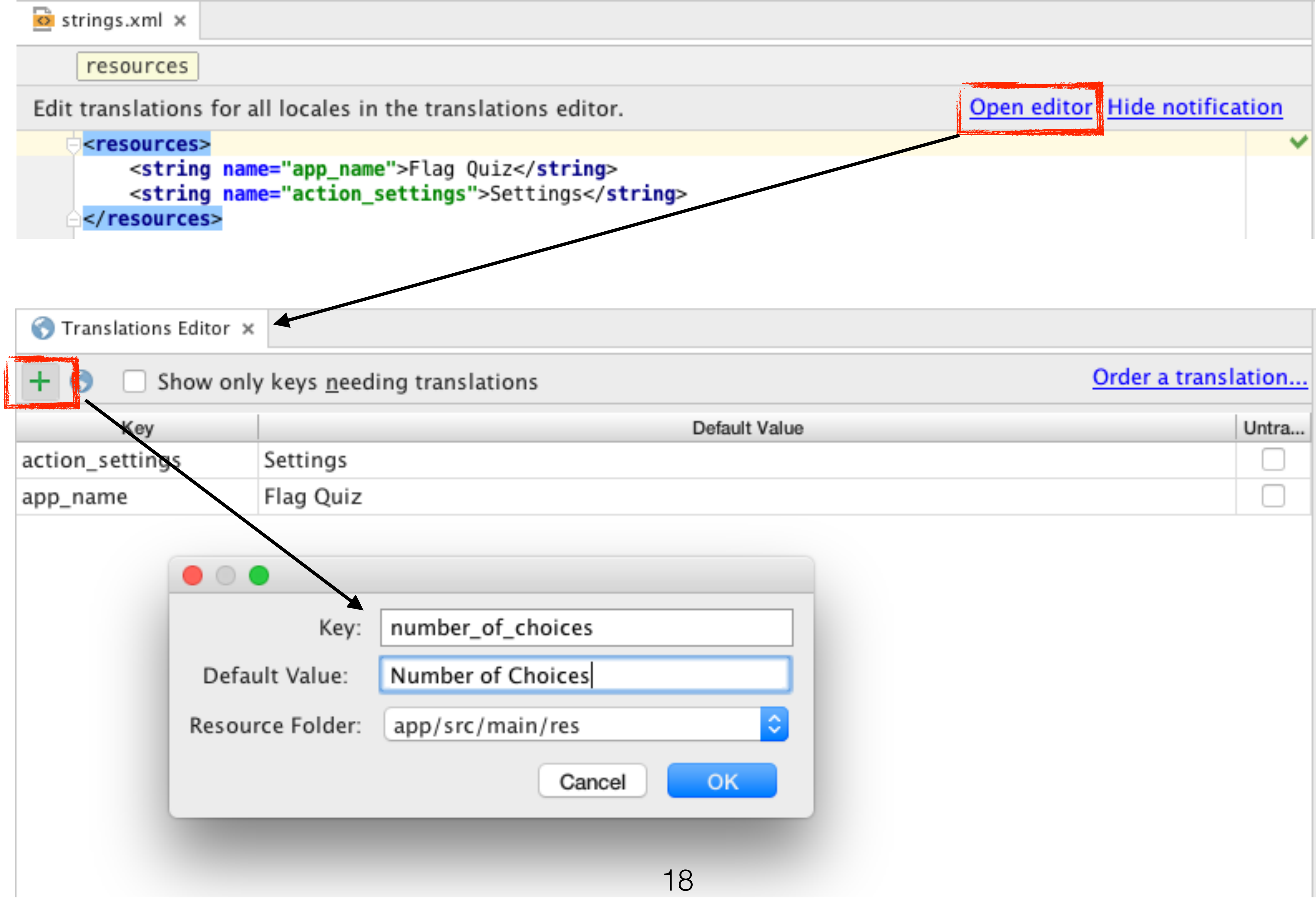

Africa-Angola.png - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
app
FlagQuiz
app
src
main
assets
Africa
Africa-Angola.png
Android
Africa-Angola.png
(24-bit color) 9.41K
1: Project
Z: Structure
Captures
2: Favorites
Build Variants
Gradle
Android Model
app
manifests
AndroidManifest.xm
java
at.htl.flagquiz
MainActivity
MainActivityFra
at.htl.flagquiz (andro
ApplicationTes
at.htl.flagquiz (test)
ExampleUnitTe
res
drawable
timmy.jpg
layout
activity_main.xm
content_main.xm
fragment_main.xı
menu
mipmap
values
colors.xml
dimens.xml (2)
strings.xml
styles.xml (2)
assets
Africa
Africa-Algeria.pn
Africa-Angola.png
Africa-Benin.png
Africa-Botswana.
Africa-Burkina_Fa
Africa-Burundi.pr
Africa-Cameroon
Terminal
6: Android Monitor
0: Messages
TODO
Event Log
Gradle Console
n/a
n/a
Context: <no context>

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)

Gradle build finished in 2s 508ms (34 minutes ago)

Erstellen der string-Resourcen


strings.xml ×
resources
Open editor
Hide notification
Translations Editor ×
+
Order a translation...
Cancel
OK

Edit translations for all locales in the translations editor.

```xml
<resources>
    <string name="app_name">Flag Quiz</string>
    <string name="action_settings">Settings</string>
</resources>
```

☐ Show only keys needing translations

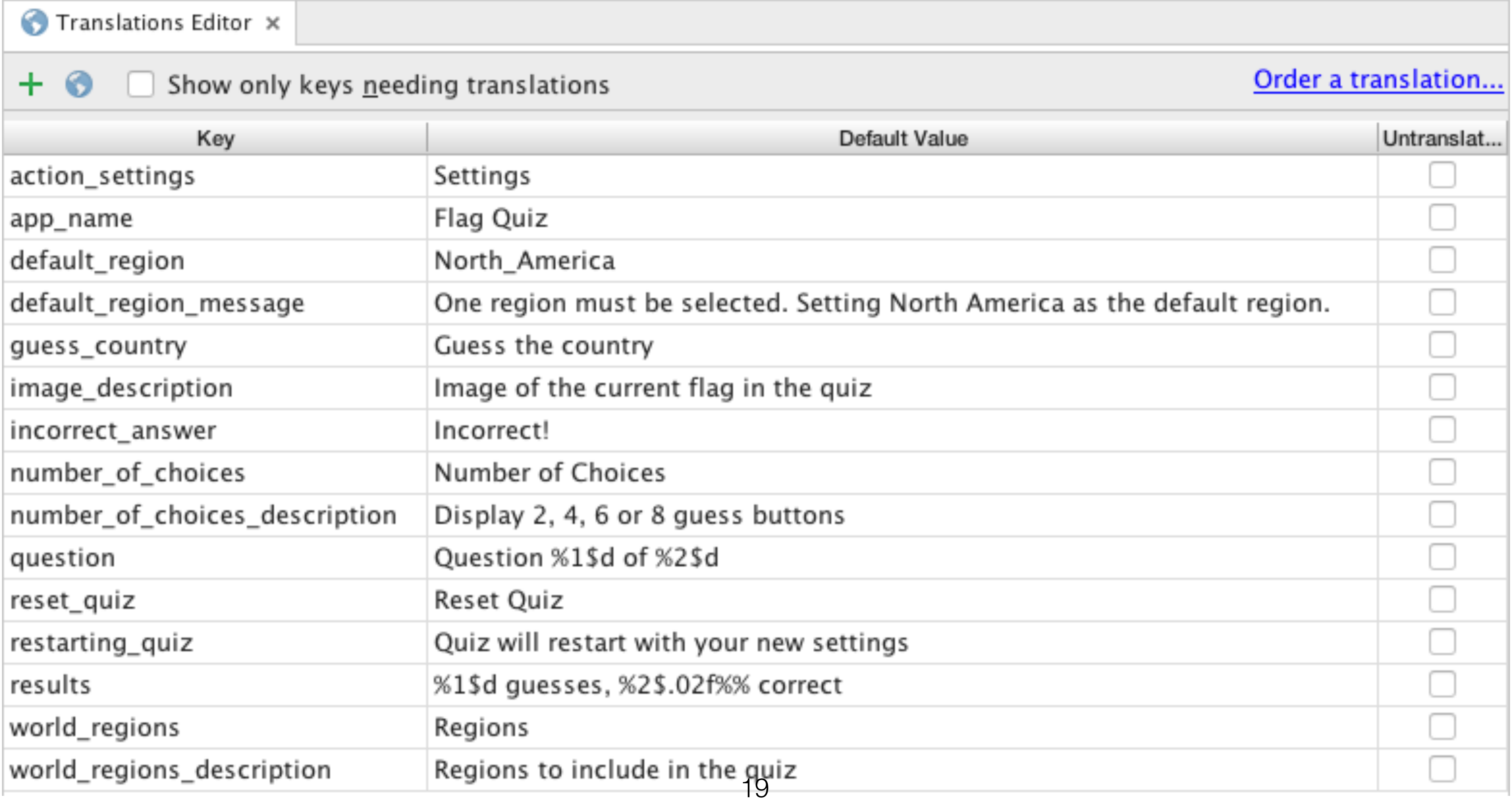
Key	Default Value	Untra...
action_settings	Settings	☐
app_name	Flag Quiz	☐

Key: number_of_choices

Default Value: Number of Choices

Resource Folder: app/src/main/res

strings.xml


Translations Editor ×
☐ Show only keys needing translations
Order a translation...

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

format Strings

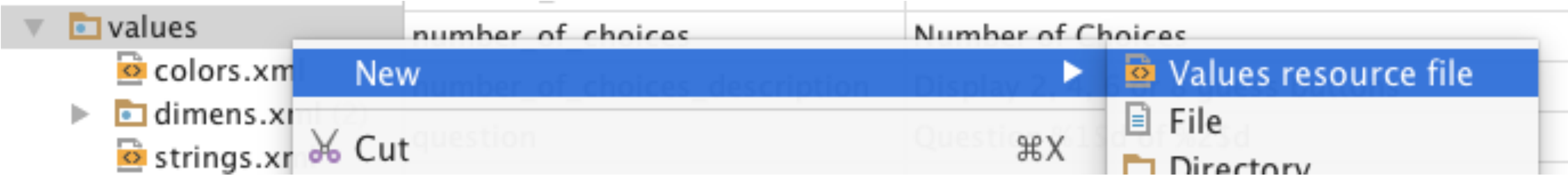
- %1$d guesses, %2$.02f%% correct
- 1$ … erster Wert, der %1$d ersetzt
- 2$ … zweiter Wert, der %2$.02f ersetzt
- d … Integer
- f … Floating-point Number
- Die Position von 1$ und 2$ im Text ist unerheblich. somit ist man sprachspezifisch flexibel

arrays.xml

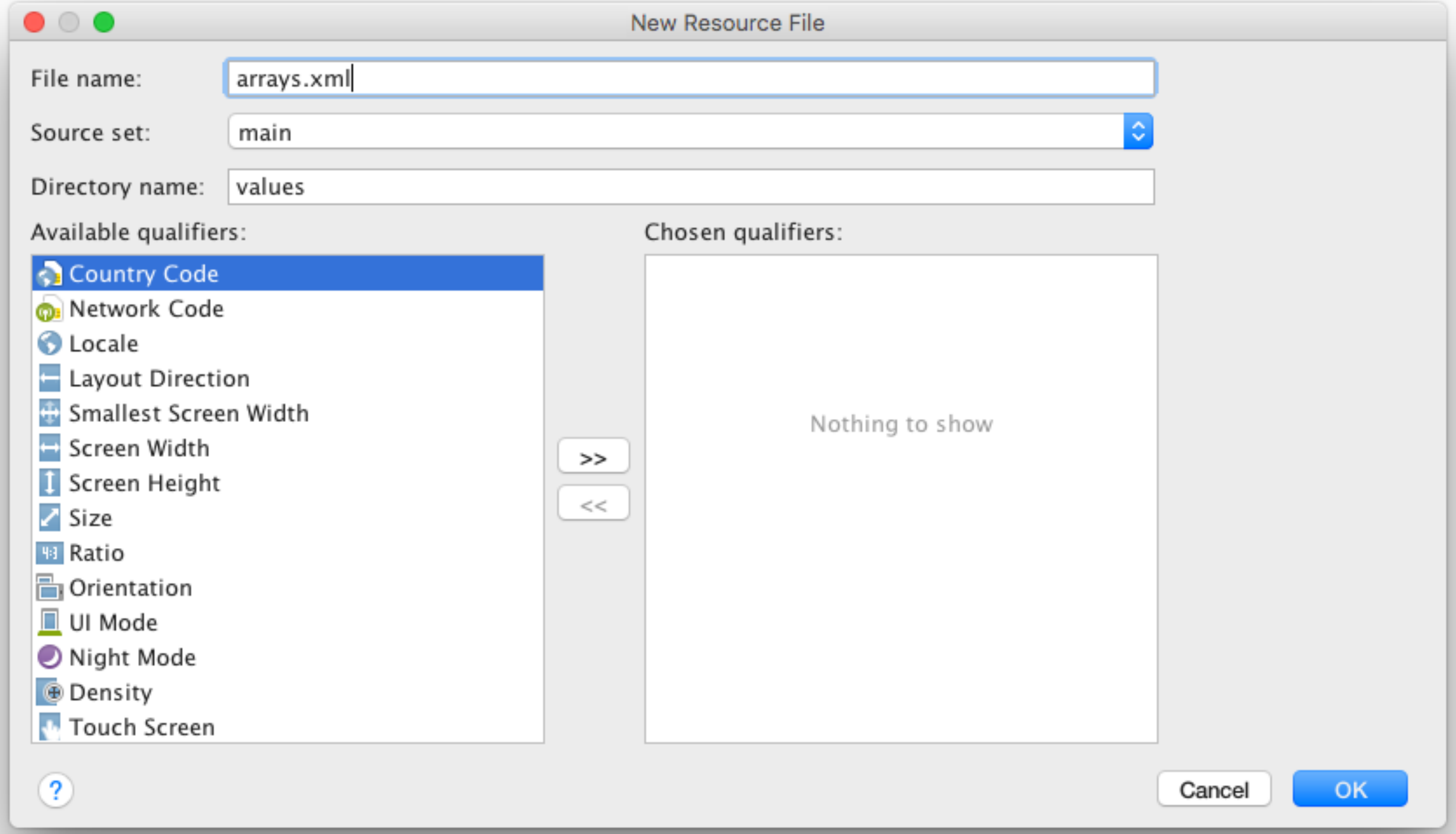
- Technisch gesehen, könnten sämtliche Einträge im res/value-Folder in einem File stehen

- Dies wird jedoch vermieden

- Anlegen eines neuen Resource-Files: arrays.xml


values
colors.xml
dimens.xml
strings.xml
number_of_choices
Number of Choices
New
Cut
⌘X
Values resource file
File
Directory

arrays.xml



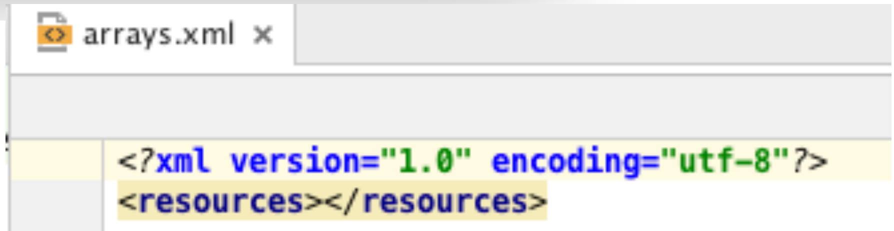
New Resource File

File name: arrays.xml

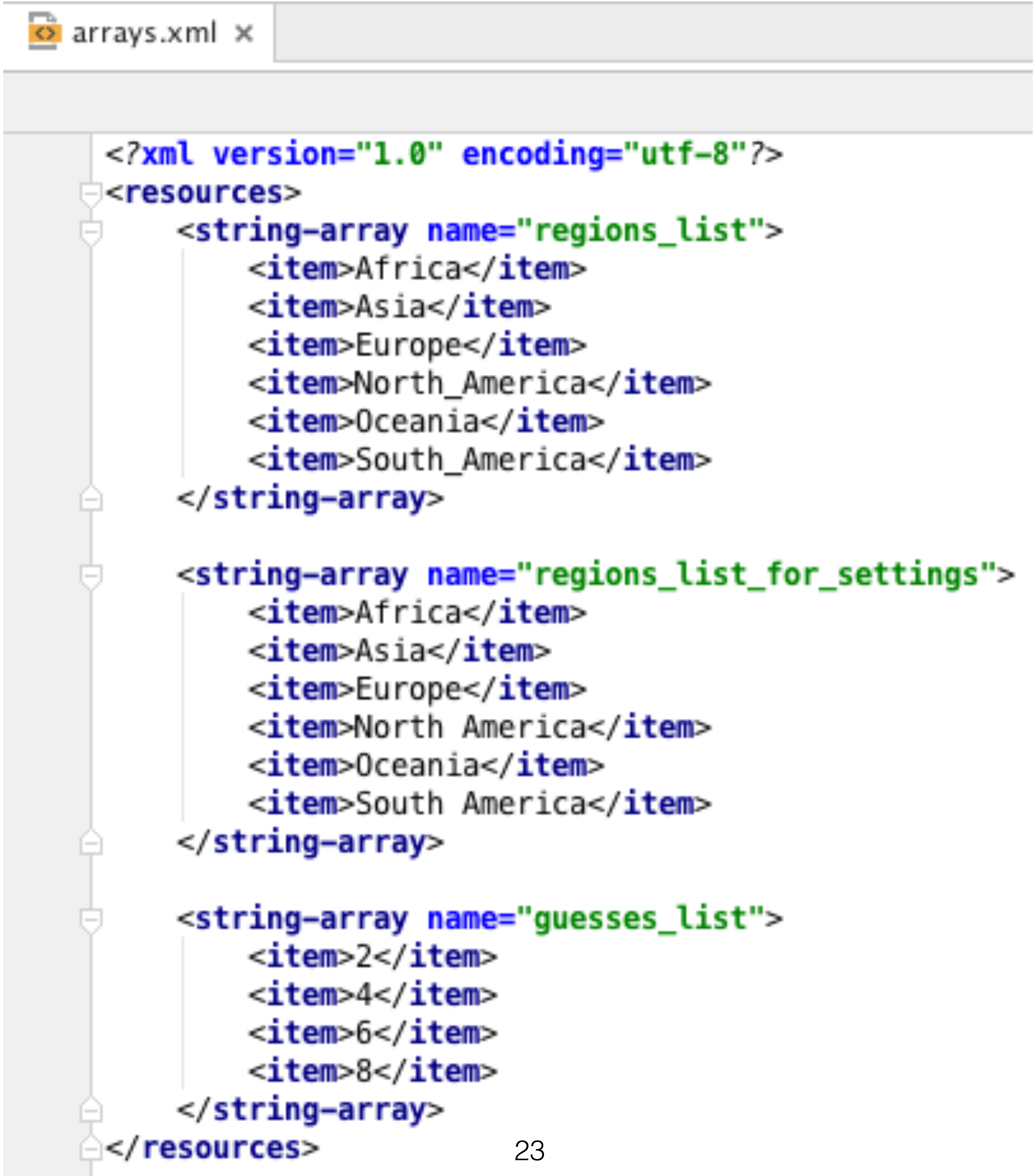
Source set: main

Directory name: values

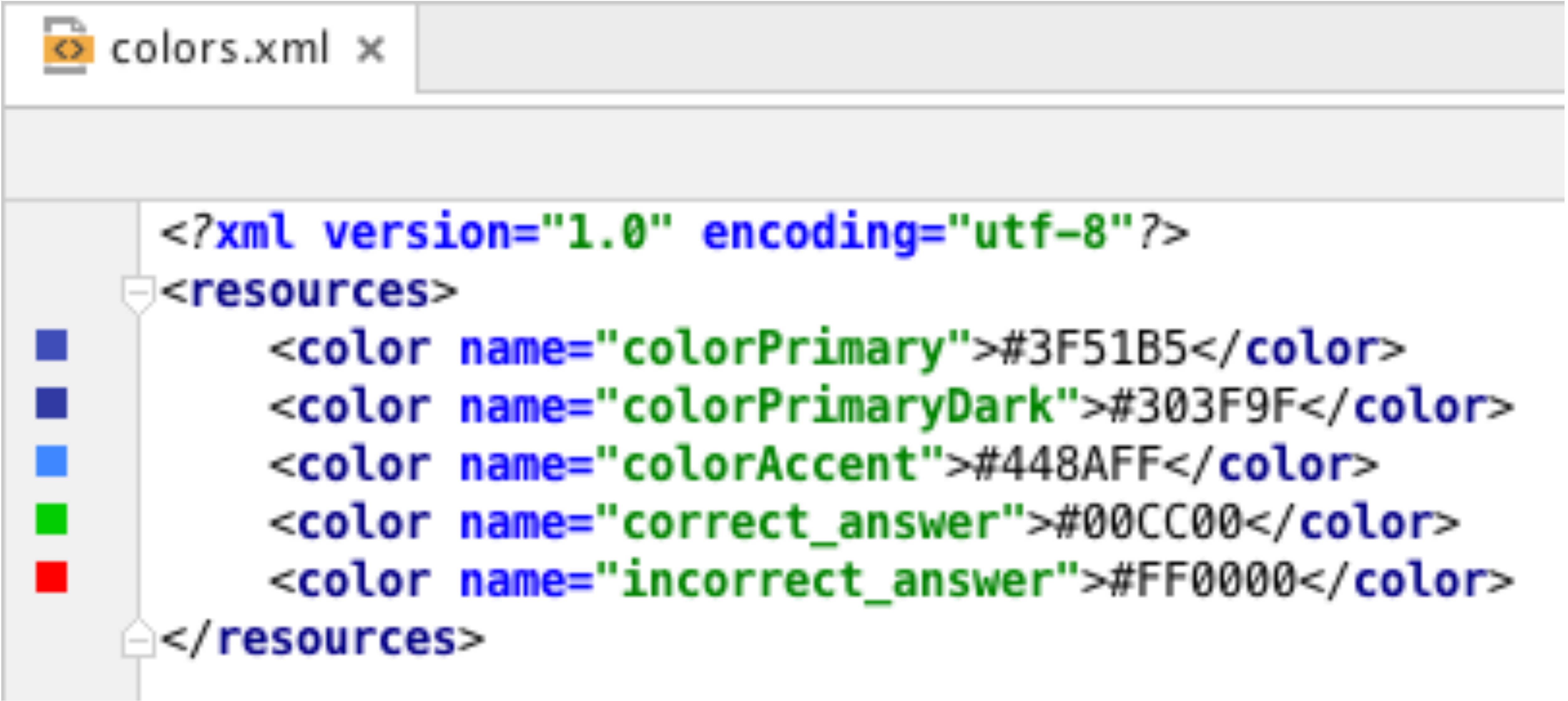
Available qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
Night Mode
Density
Touch Screen
>>
<<
Chosen qualifiers:
Nothing to show
?
Cancel
OK



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

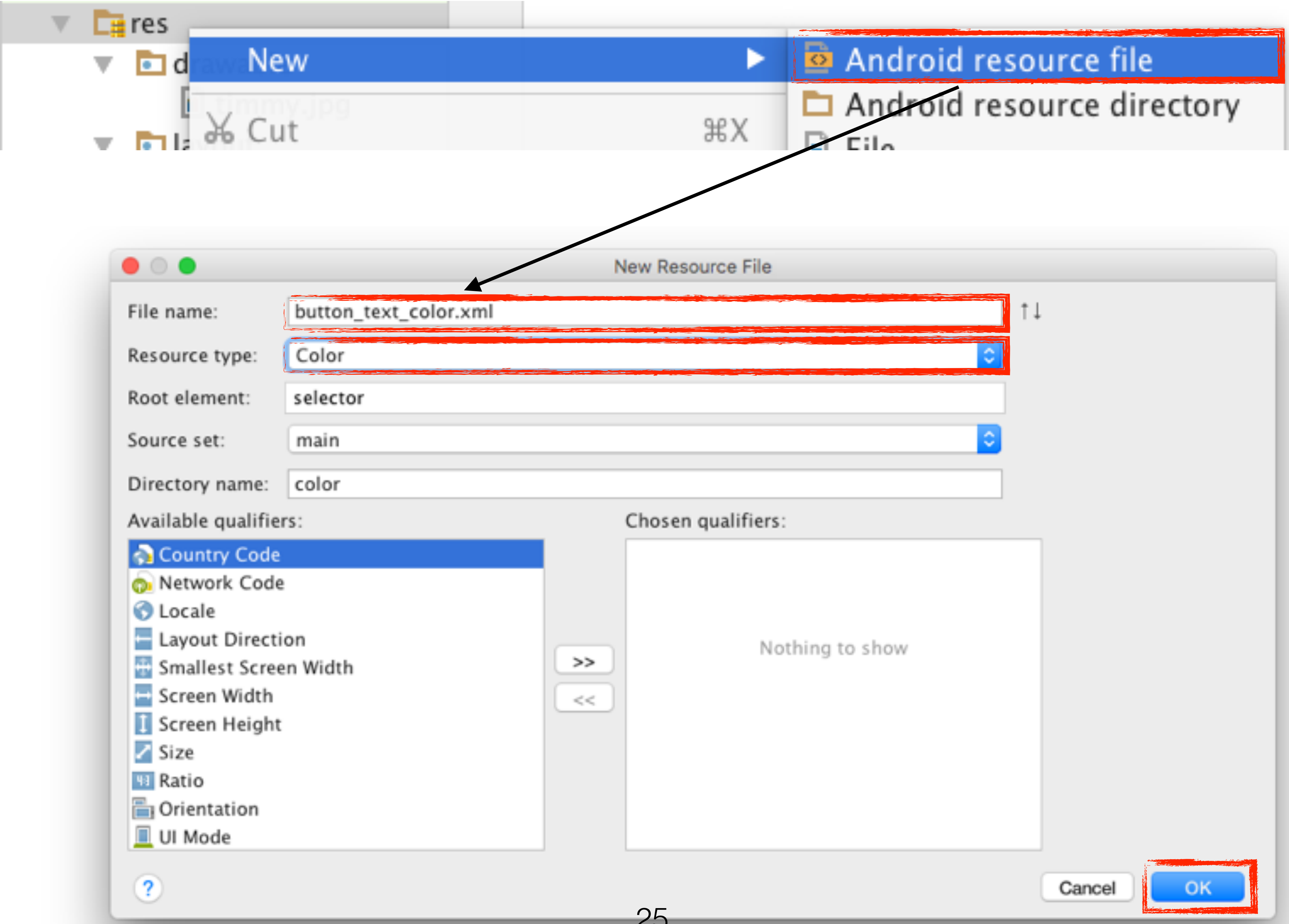


```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="regions_list">
        <item>Africa</item>
        <item>Asia</item>
        <item>Europe</item>
        <item>North_America</item>
        <item>Oceania</item>
        <item>South_America</item>
    </string-array>

    <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>
```



```xml
<?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


res
New
Cut
⌘X
Android resource file
Android resource directory
File
New Resource File

File name: button_text_color.xml

Resource type: Color

Root element: selector

Source set: main

Directory name: color

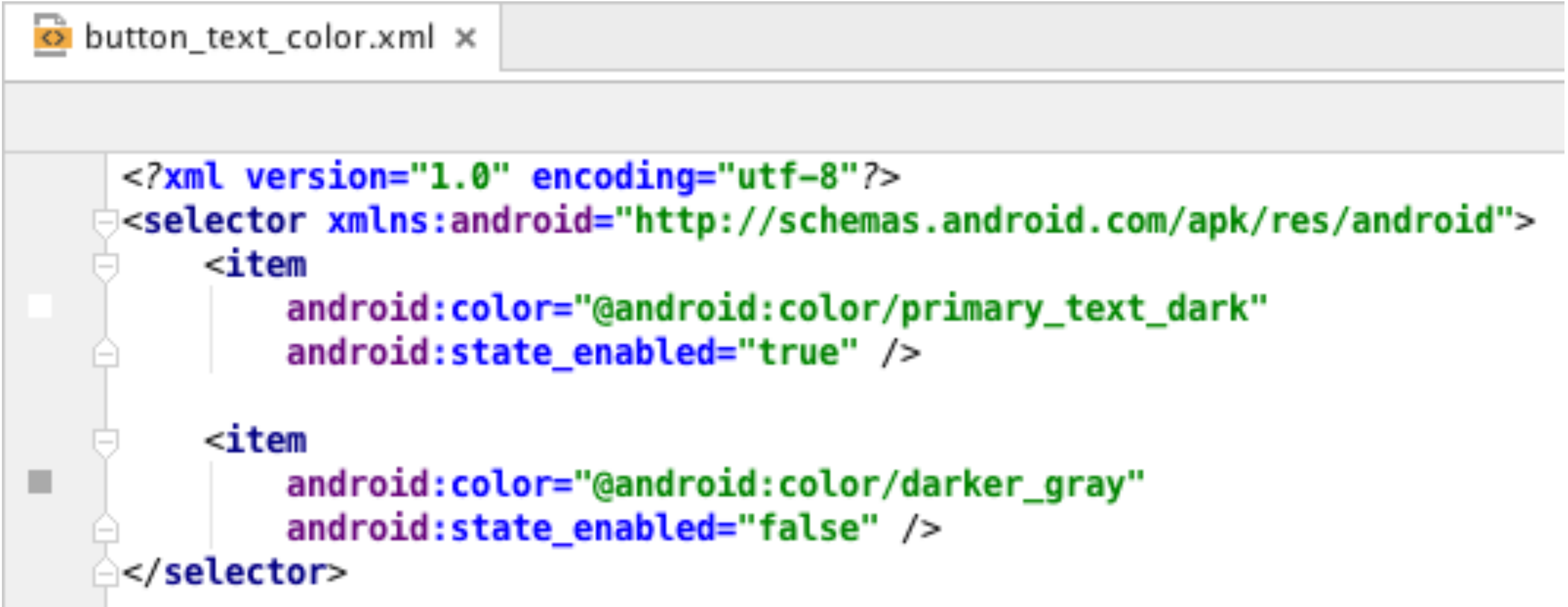
Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode

Chosen qualifiers:

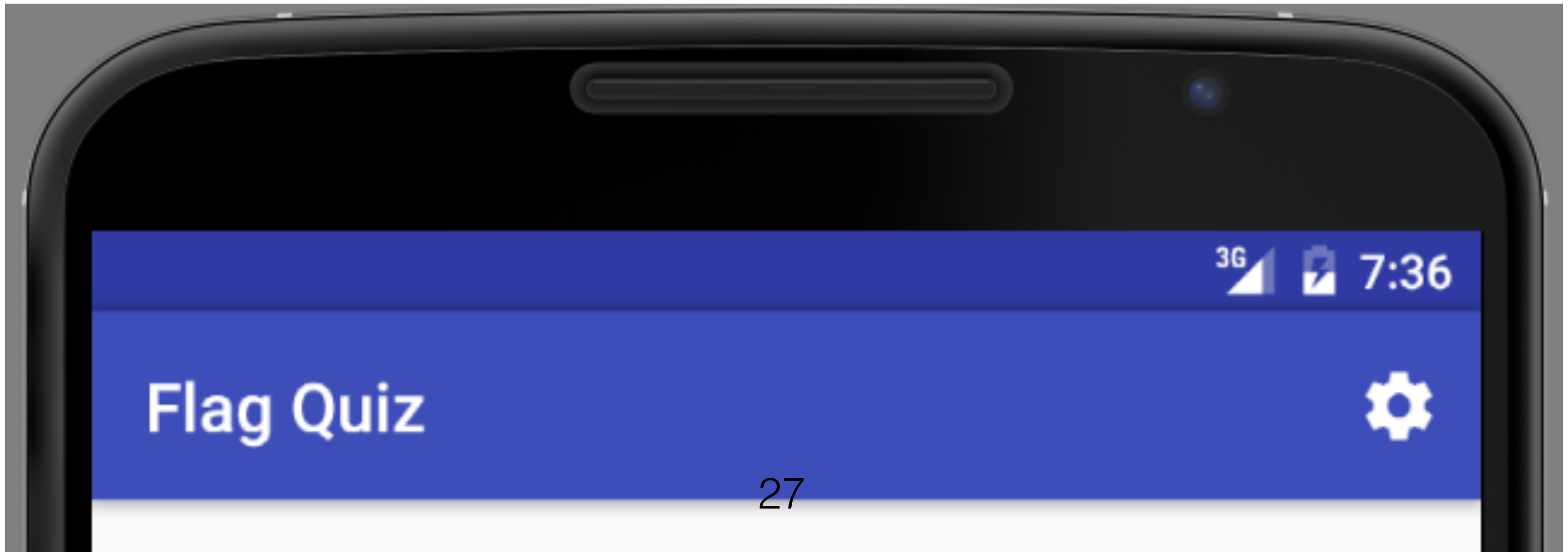
Nothing to show

>>
<<
?
Cancel
OK

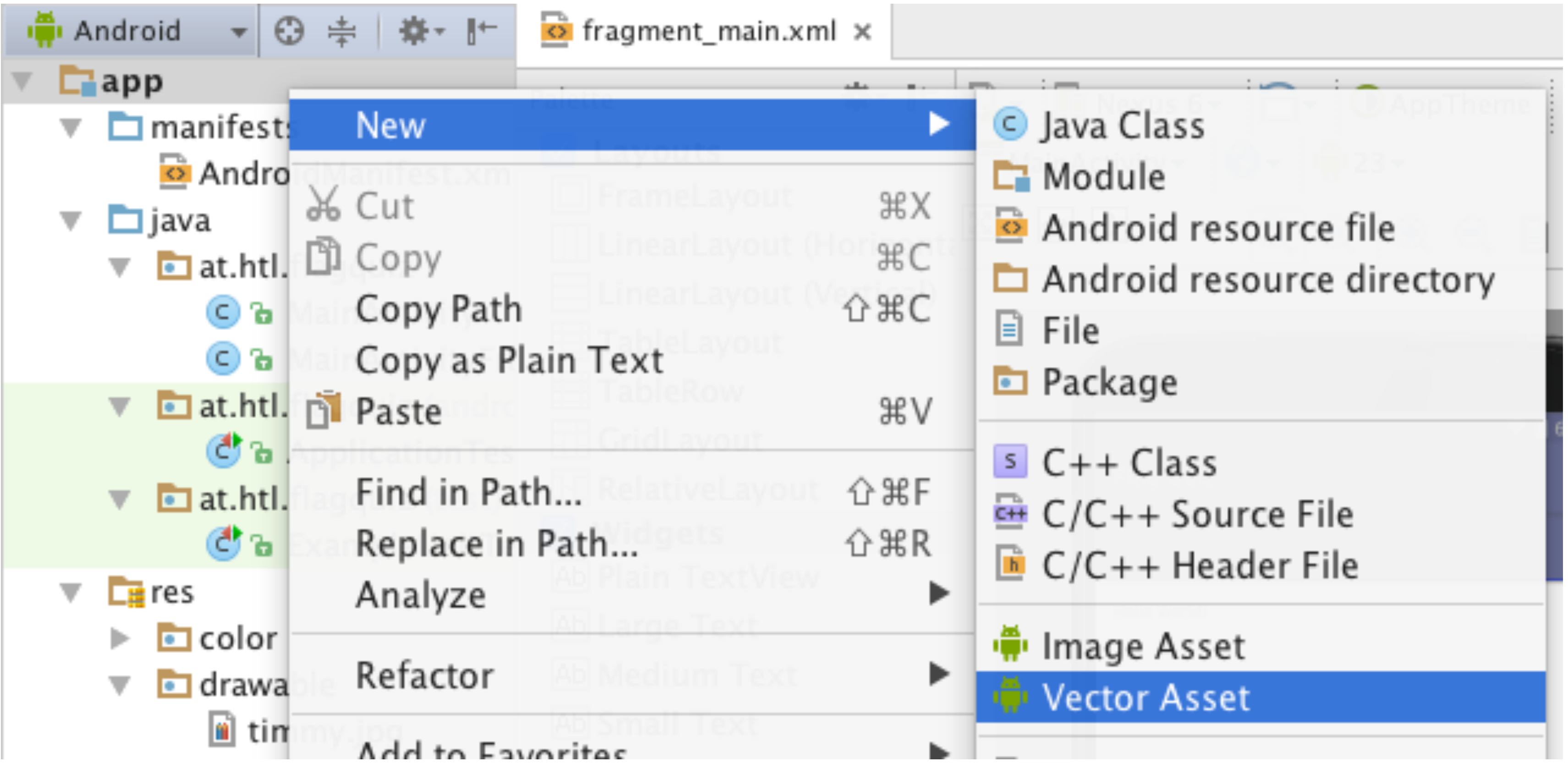

button_text_color.xml

```
<?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>
```

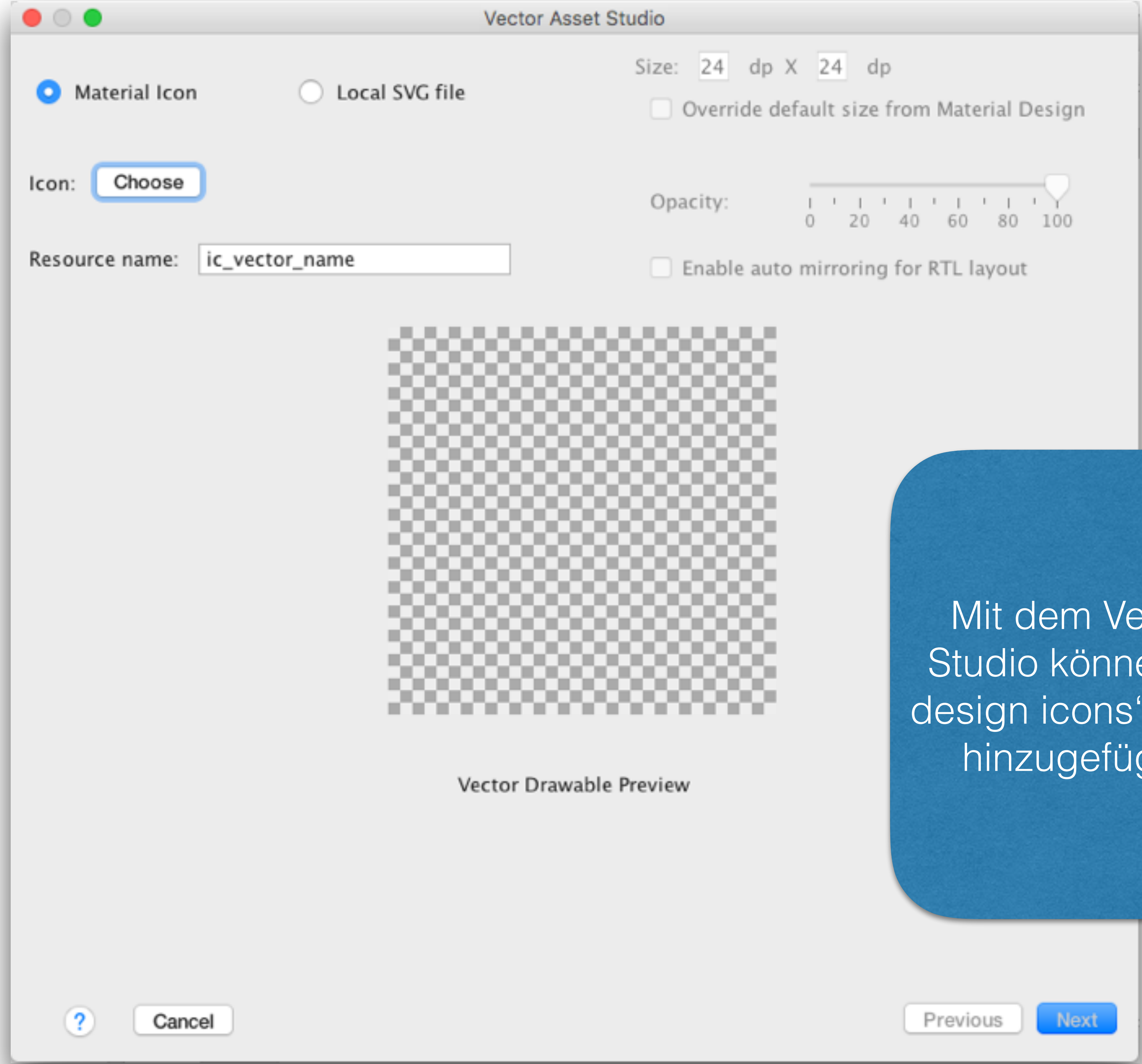
Nun brauchen wir ein Icon für das Main Menu


3G
7:36
Flag Quiz

Vector Asset Studio


Android
fragment_main.xml
app
manifests
AndroidManifest.xml
java
at.htl.
at.htl.
at.htl.
res
color
drawable
New
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy as Plain Text
Paste ⌘V
Find in Path... ⇧⌘F
Replace in Path... ⇧⌘R
Analyze
Refactor
Add to Favorites
Java Class
Module
Android resource file
Android resource directory
File
Package
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset

app - New -> Vector Asset


Vector Asset Studio

● Material Icon ○ Local SVG file

Size: 24 dp X 24 dp

☐ Override default size from Material Design

Icon: Choose

Opacity: 100
0
20
40
60
80
100

Resource name: ic_vector_name

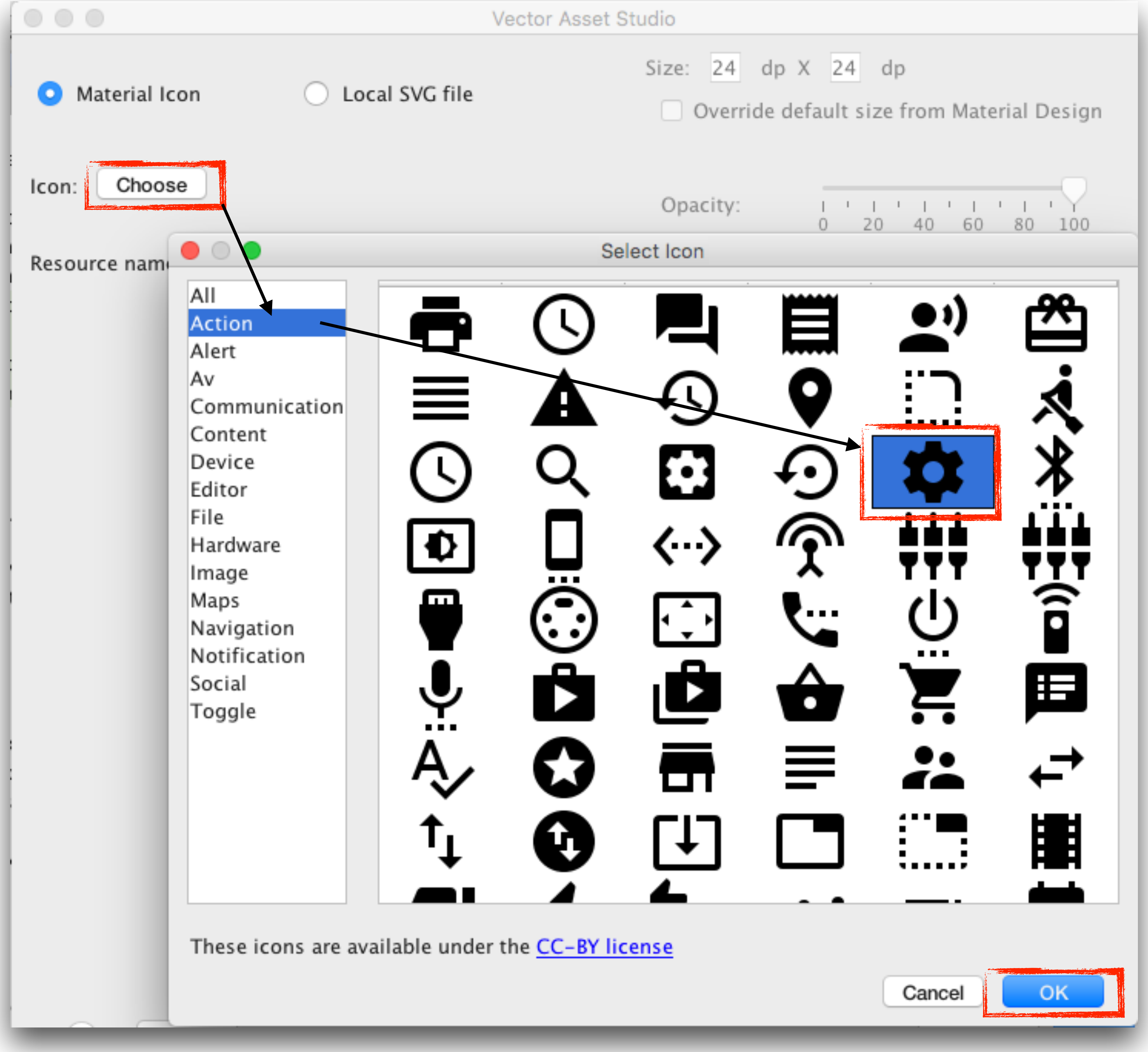
☐ Enable auto mirroring for RTL layout

Vector Drawable Preview

?
Cancel
Previous
Next

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

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

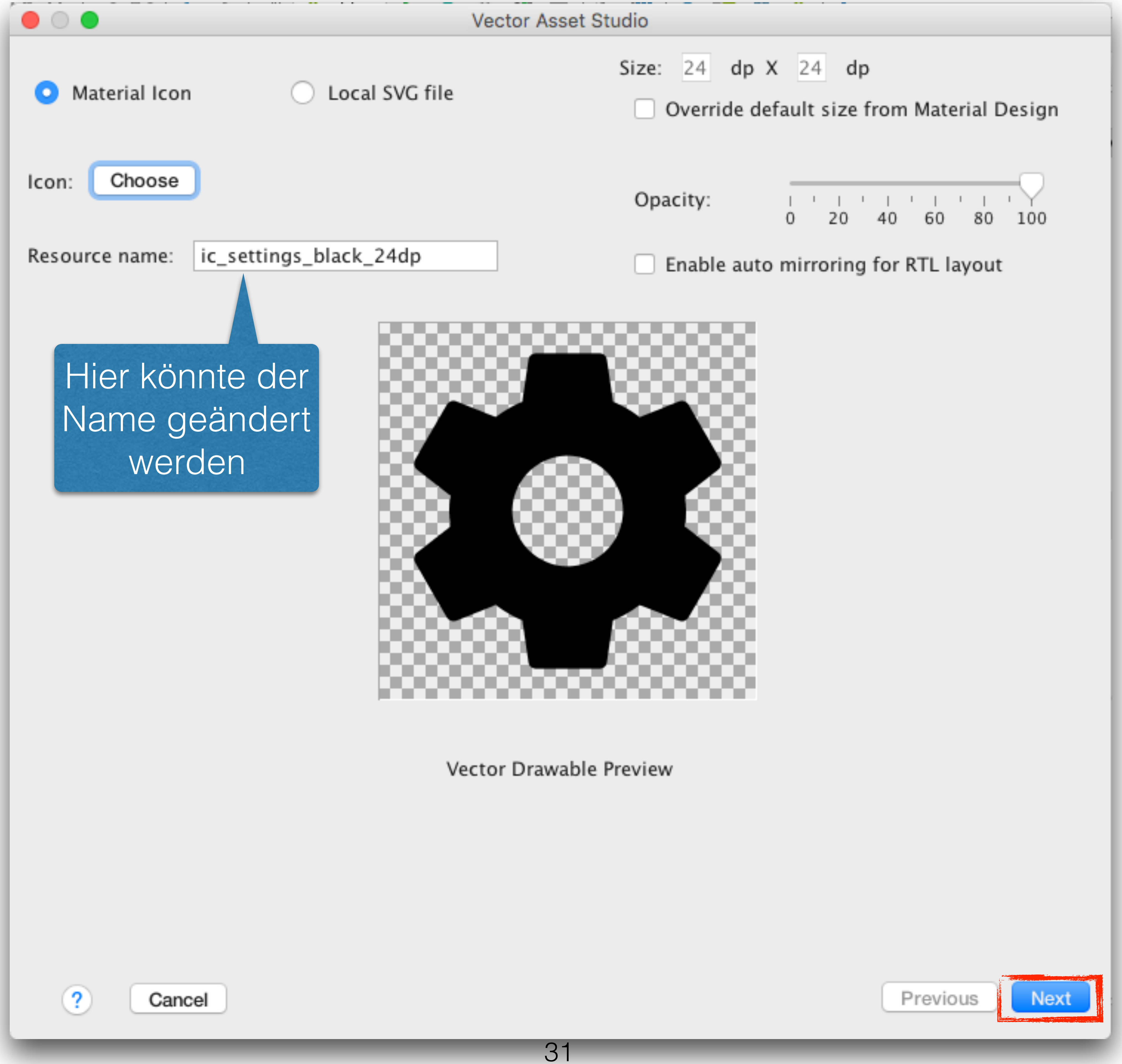

Vector Asset Studio
Icon:
Choose
Opacity:
0
20
40
60
80
100
Resource name
Select Icon
All
Action
Alert
Av
Communication
Content
Device
Editor
File
Hardware
Image
Maps
Navigation
Notification
Social
Toggle
Cancel
OK

● Material Icon ○ Local SVG file

Size: 24 dp X 24 dp

☐ Override default size from Material Design

These icons are available under the CC-BY license



Vector Asset Studio

● Material Icon ○ Local SVG file

Size: 24 dp X 24 dp

☐ Override default size from Material Design

Icon: Choose

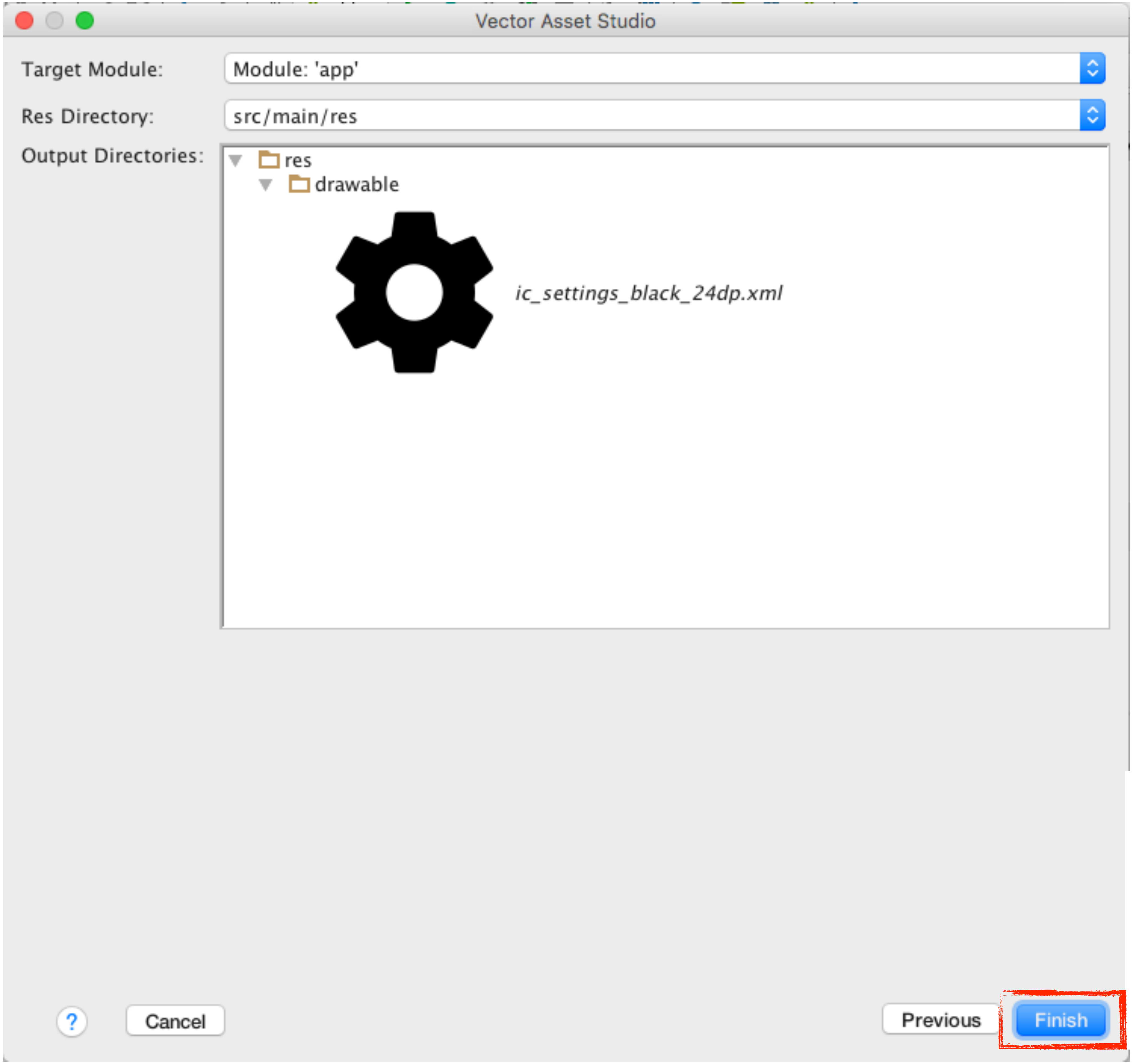
Opacity: 0 20 40 60 80 100

Resource name: ic_settings_black_24dp

☐ Enable auto mirroring for RTL layout

Hier könnte der Name geändert werden

Vector Drawable Preview



Vector Asset Studio

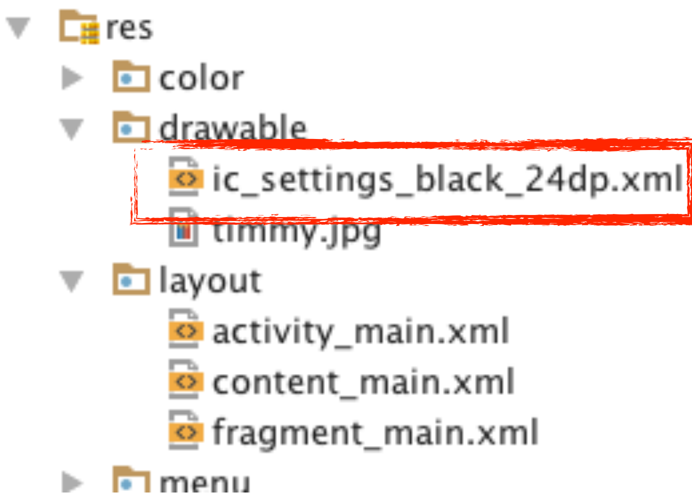
Target Module: Module: 'app'

Res Directory: src/main/res

Output Directories:

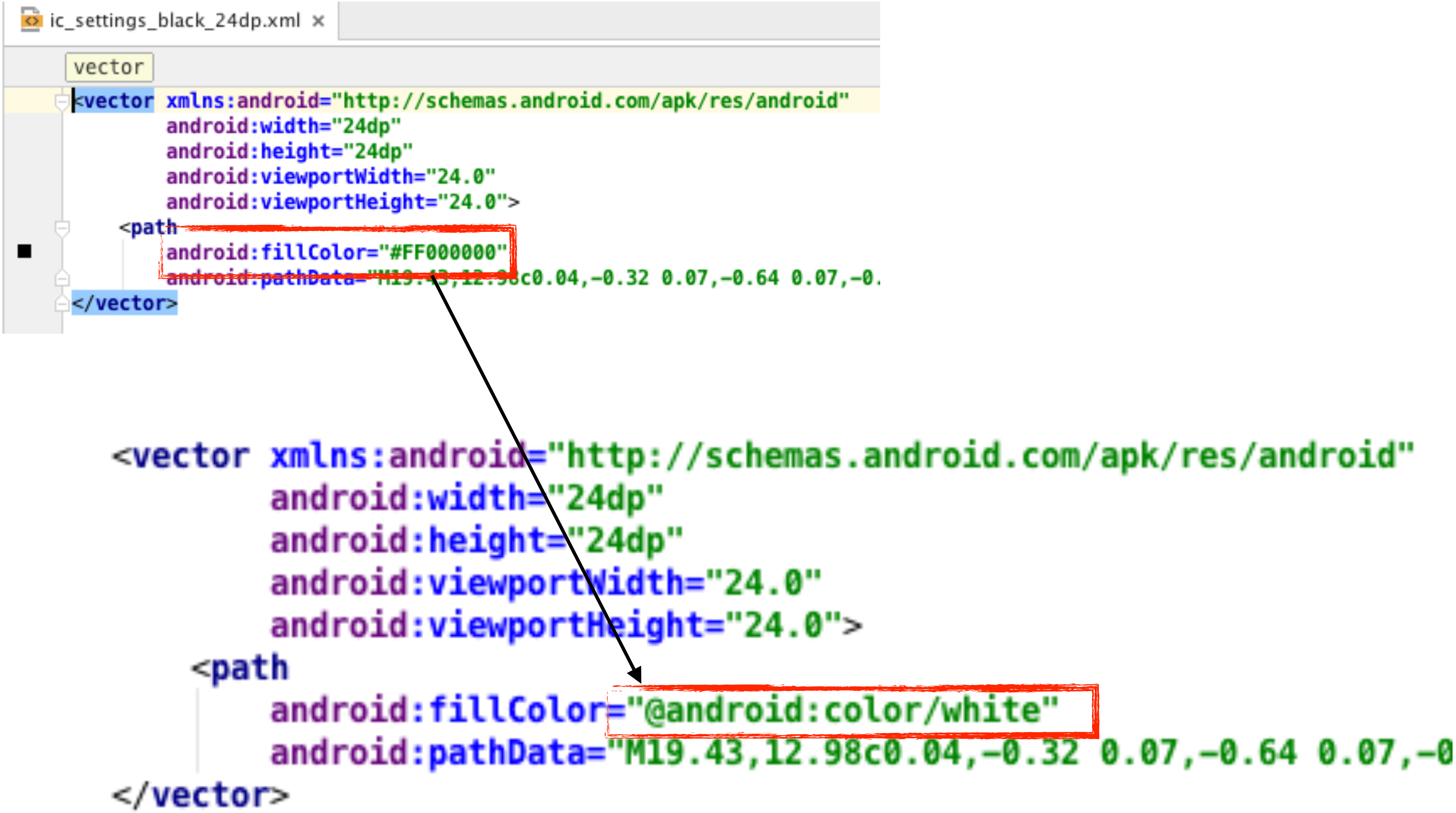
- res
  - drawable
    - ic_settings_black_24dp.xml

?
Cancel
Previous
Finish



- res
  - color
  - drawable
    - ic_settings_black_24dp.xml
    - timmy.jpg
  - layout
    - activity_main.xml
    - content_main.xml
    - fragment_main.xml
  - menu

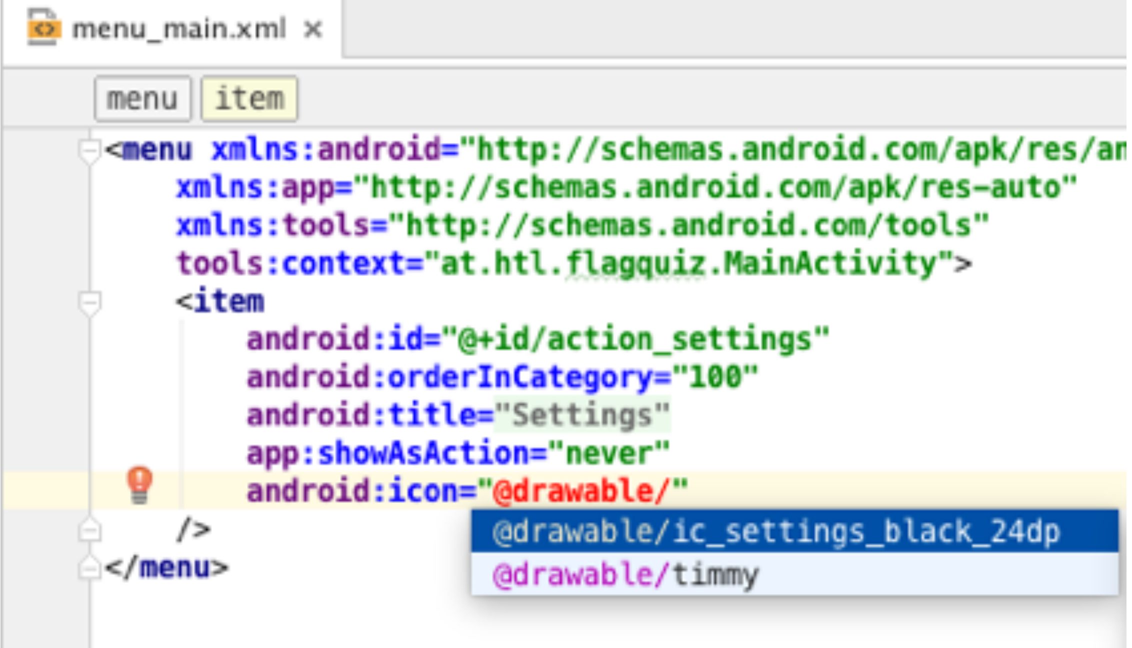
ic_settings_black_24dp.xml


ic_settings_black_24dp.xml ×
vector

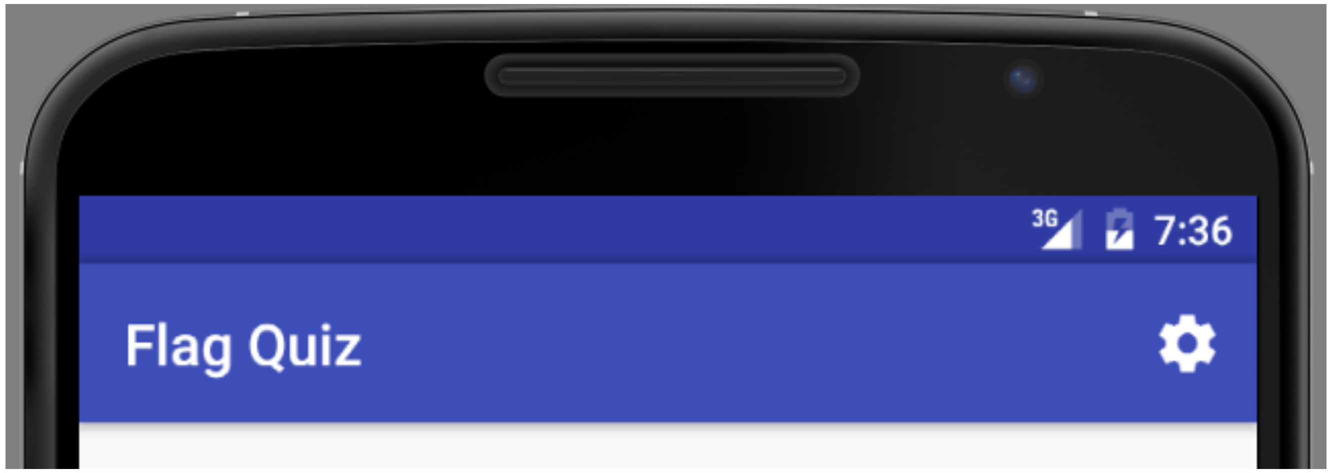
```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M19.43,12.98c0.04,-0.32 0.07,-0.64 0.07,-0.
</vector>
```

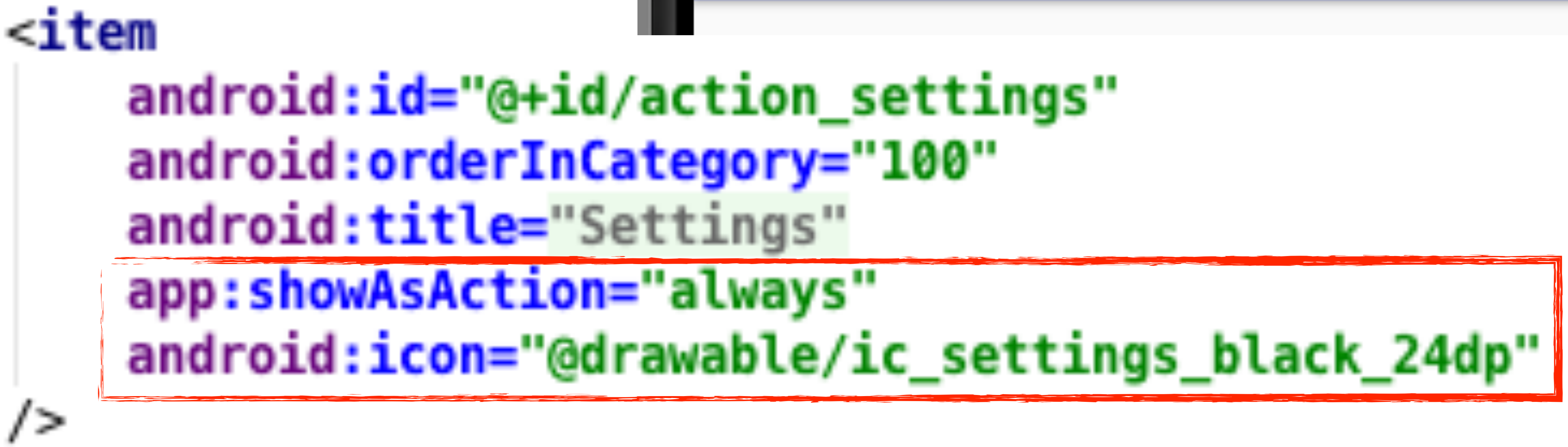
```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0">
    <path
        android:fillColor="@android:color/white"
        android:pathData="M19.43,12.98c0.04,-0.32 0.07,-0.64 0.07,-0
</vector>
```

Hinzufügen des Icon zu menu_main.xml


menu_main.xml ×
menu
item
@drawable/ic_settings_black_24dp
@drawable/timmy

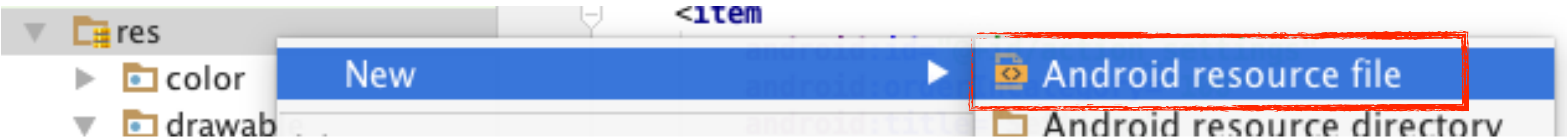
```
<menu xmlns:android="http://schemas.android.com/apk/res/an
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="at.htl.flagquiz.MainActivity">
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"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

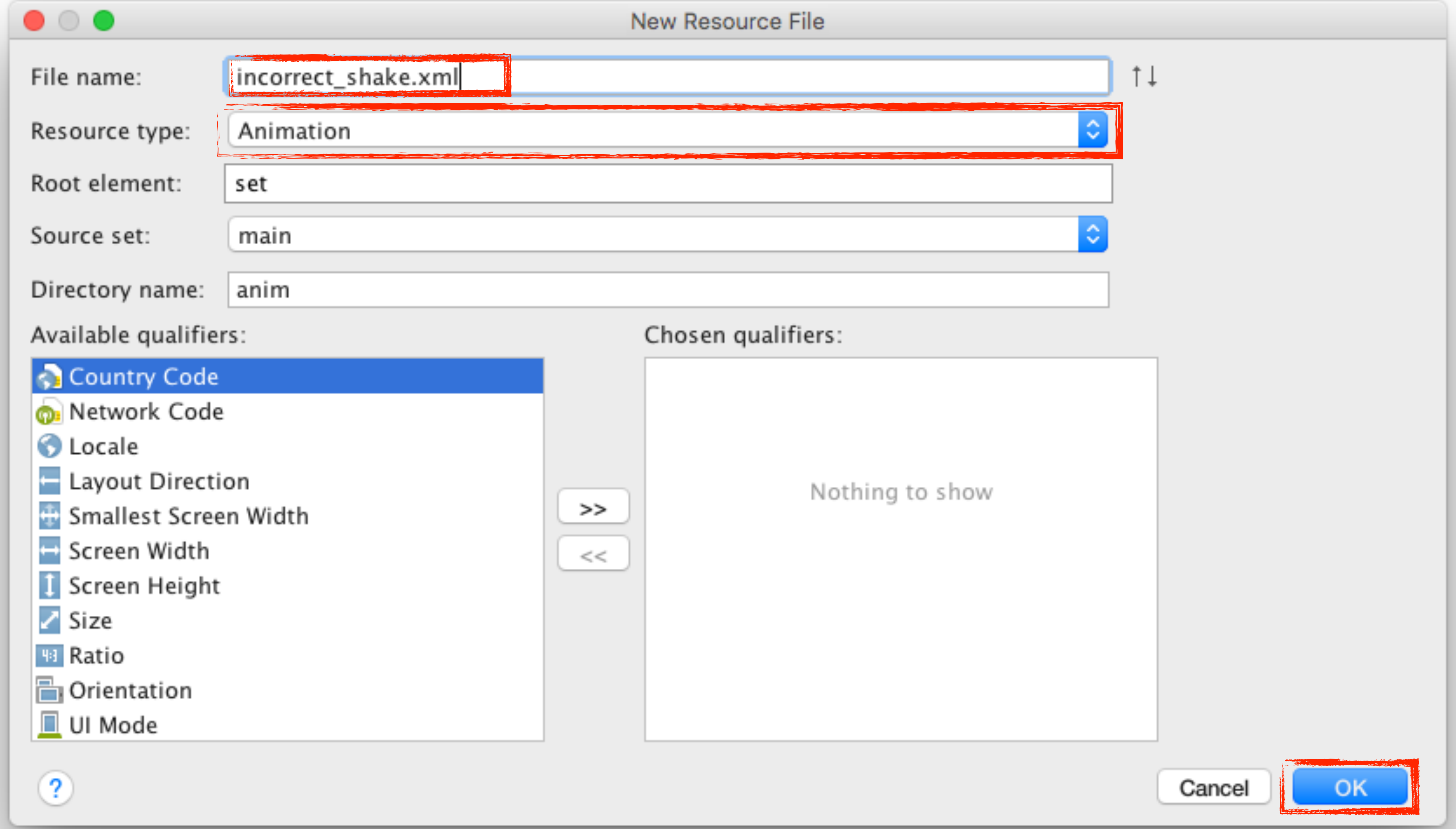

3G
7:36
Flag Quiz



```
<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


res
color
drawab
<item
New ▶
Android resource file
Android resource directory


New Resource File

File name: incorrect_shake.xml

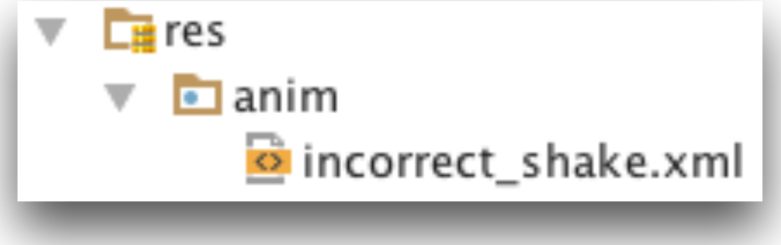
Resource type: Animation

Root element: set

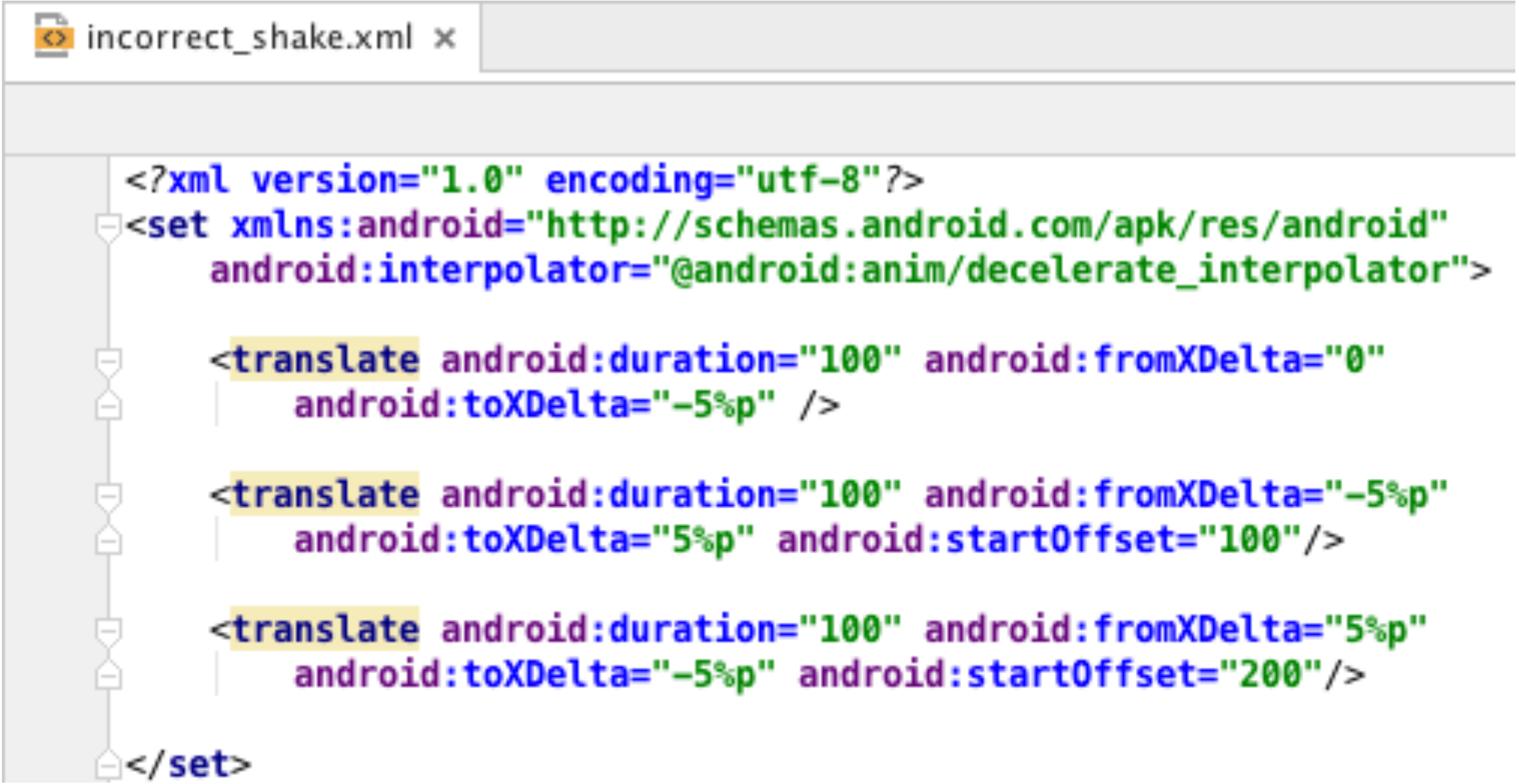
Source set: main

Directory name: anim

Available qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
>>
<<
Chosen qualifiers:
Nothing to show
?
Cancel
OK


res
anim
incorrect_shake.xml

shake-Effekt


incorrect_shake.xml ×

```xml
<?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

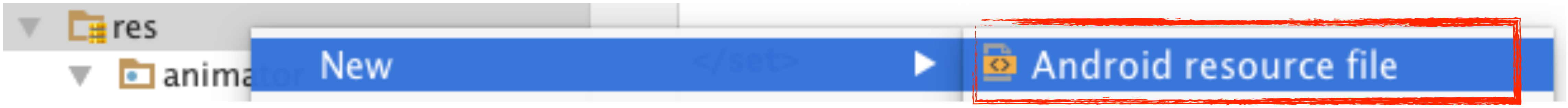
Animations

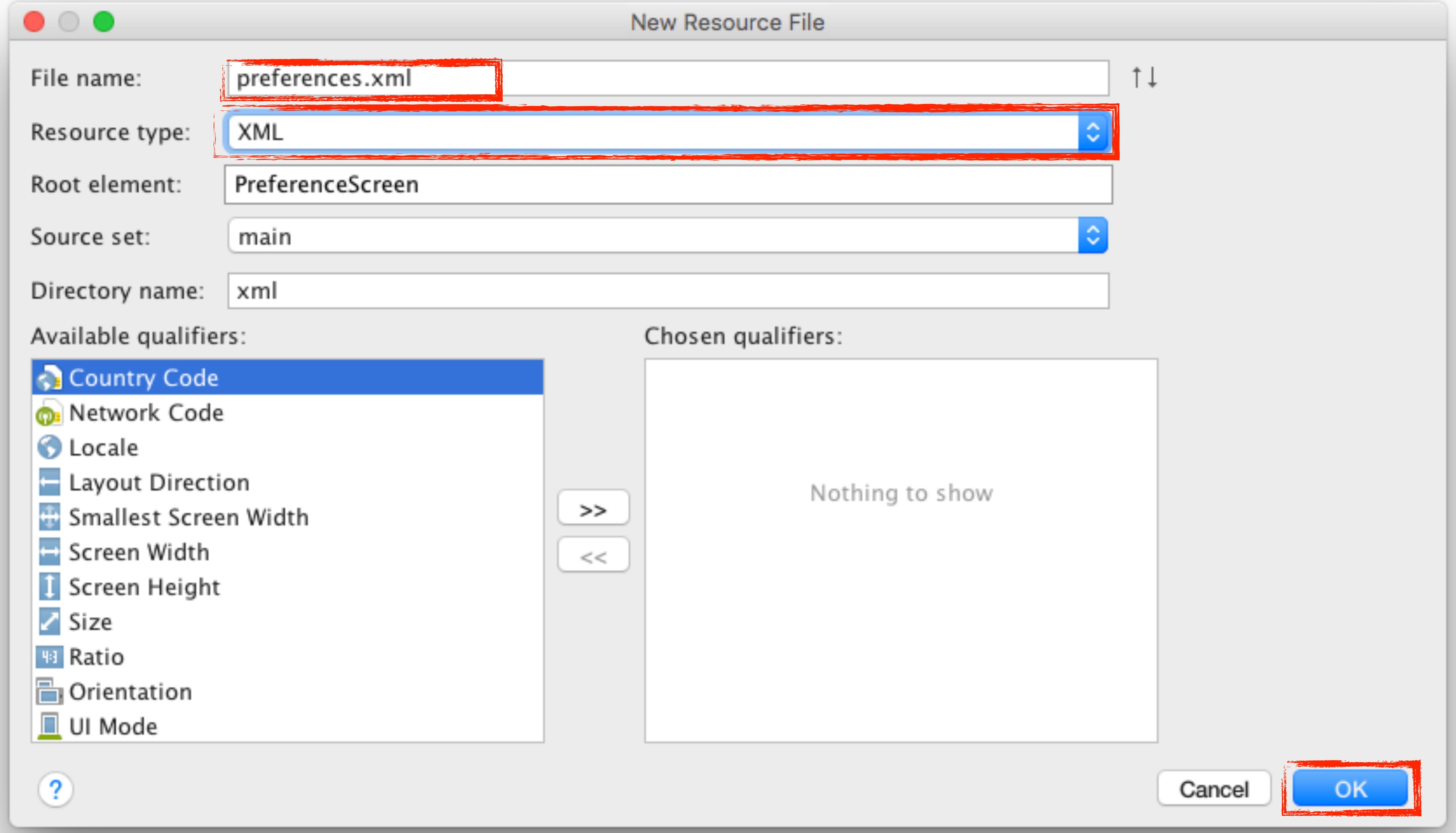
- Unsere Animation besteht aus einer Serie von drei translate animations
- Eine **translate animation** bewegt eine View innerhalb seiner Parents
- Android unterstützt auch die **property animation**, bei der die Eigenschaften eines Objekts animiert werden

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


res
animator
New
Android resource file


New Resource File

File name: preferences.xml

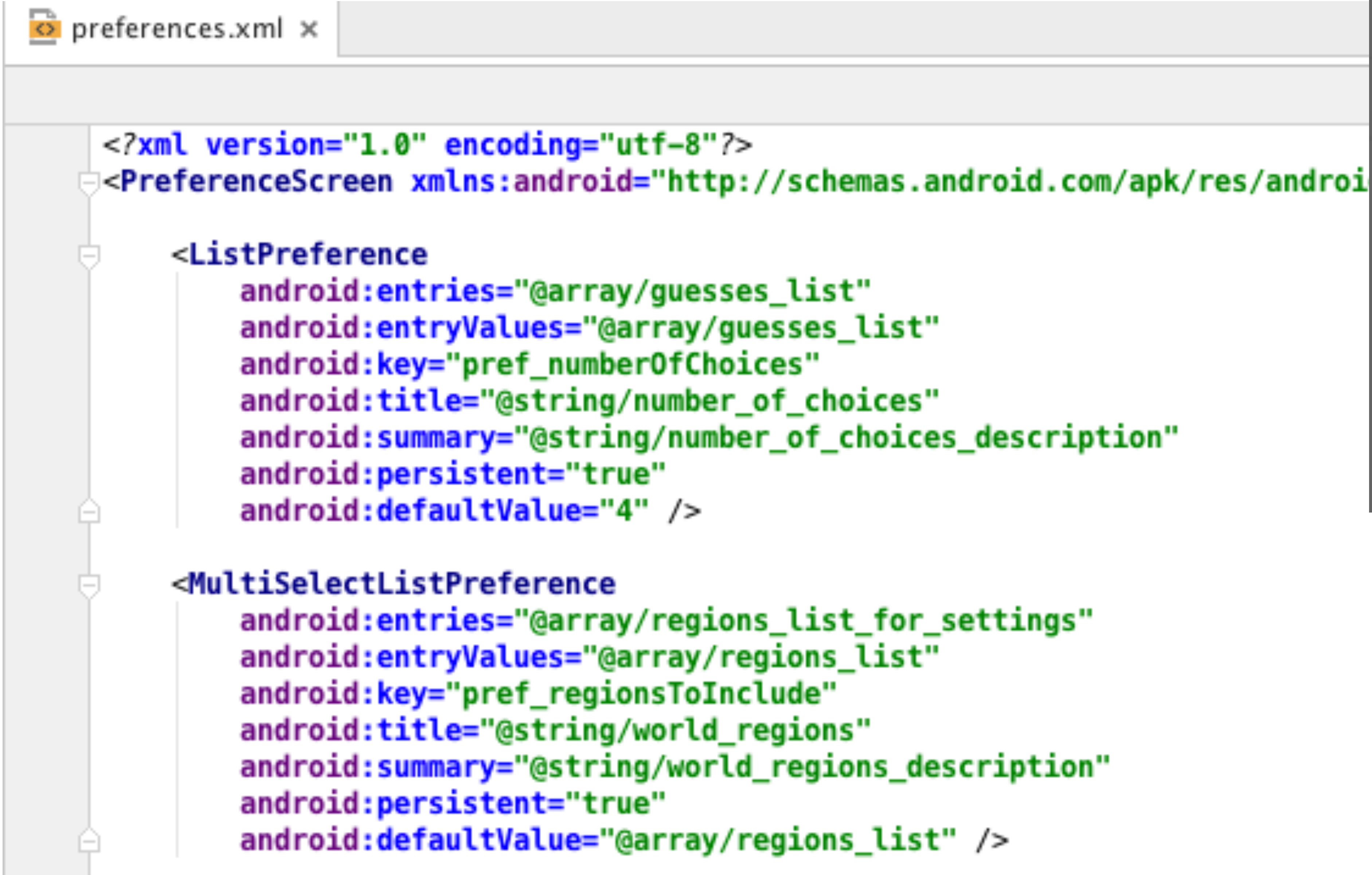
Resource type: XML

Root element: PreferenceScreen

Source set: main

Directory name: xml

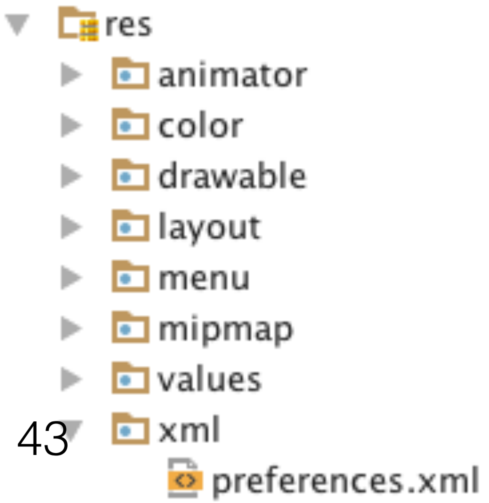
Available qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
>>
<<
Chosen qualifiers:
Nothing to show
?
Cancel
OK



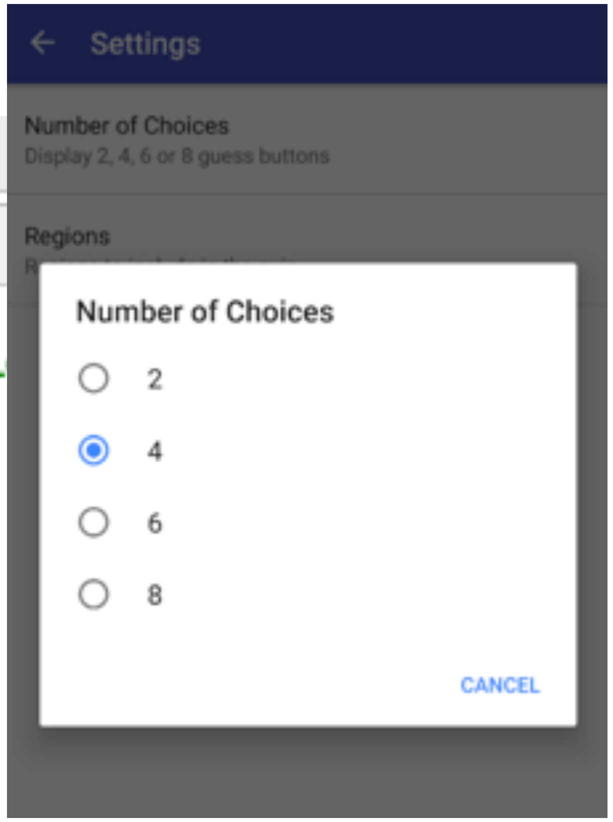
preferences.xml

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

    <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



- res
  - animator
  - color
  - drawable
  - layout
  - menu
  - mipmap
  - values
  - xml
    - preferences.xml



Settings

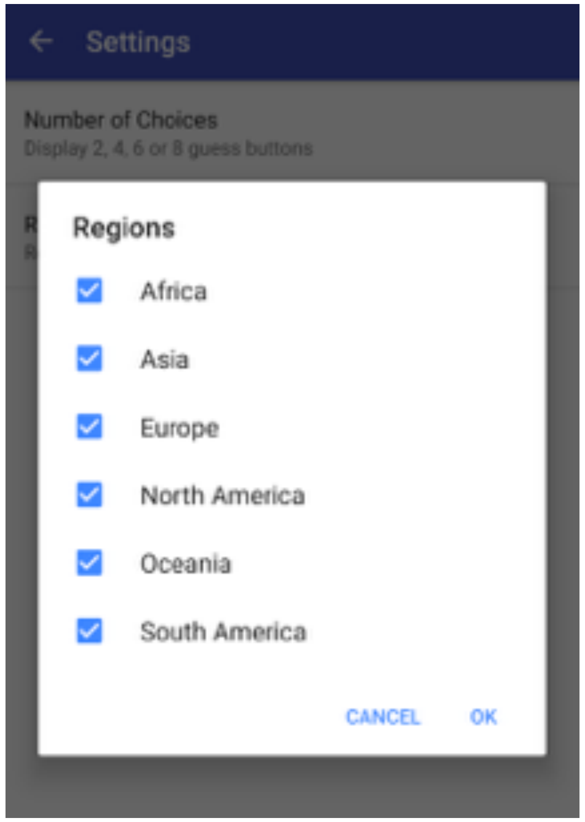
Number of Choices
Display 2, 4, 6 or 8 guess buttons

Regions

Number of Choices

○ 2
● 4
○ 6
○ 8

CANCEL



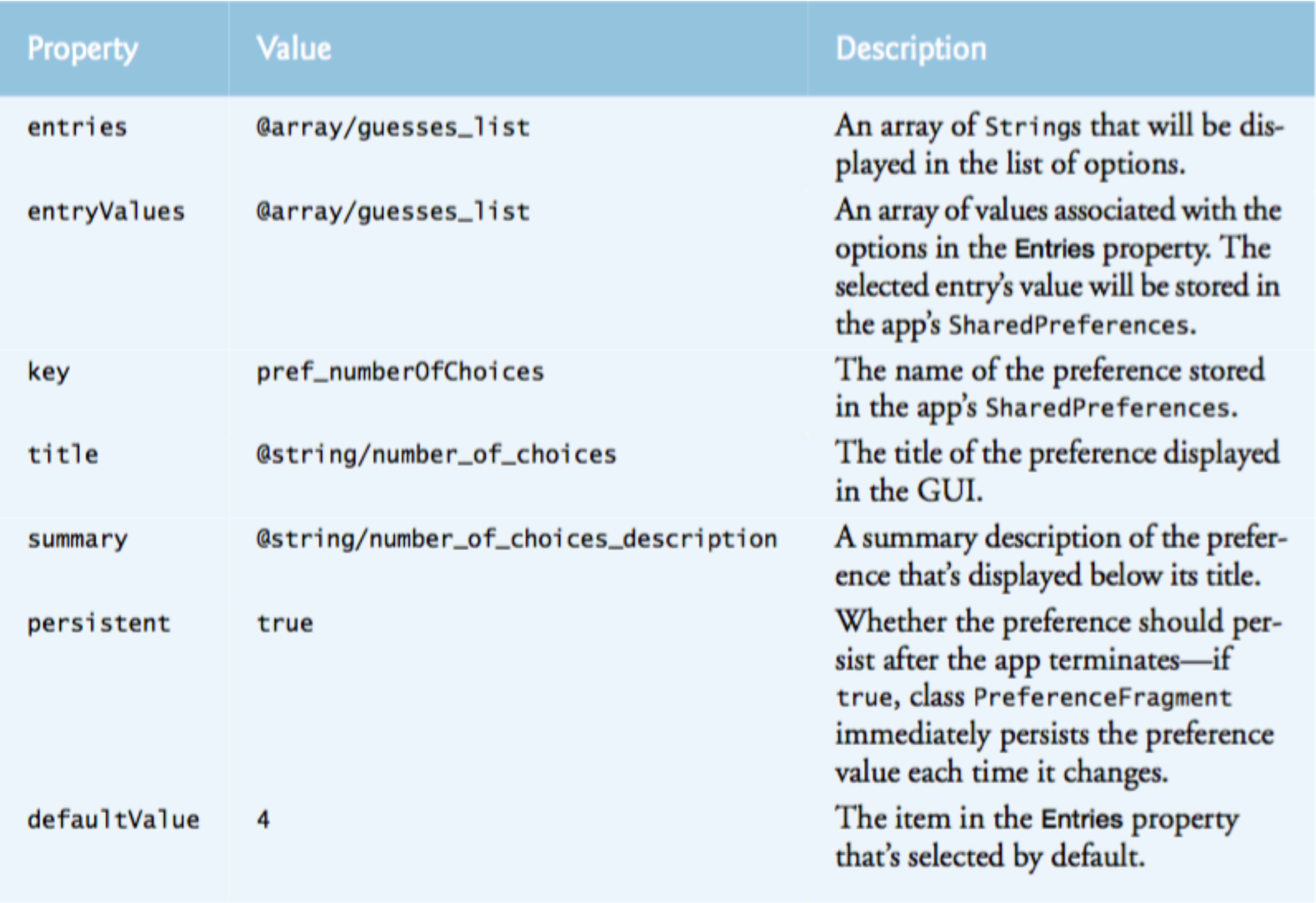
Settings

Number of Choices
Display 2, 4, 6 or 8 guess buttons

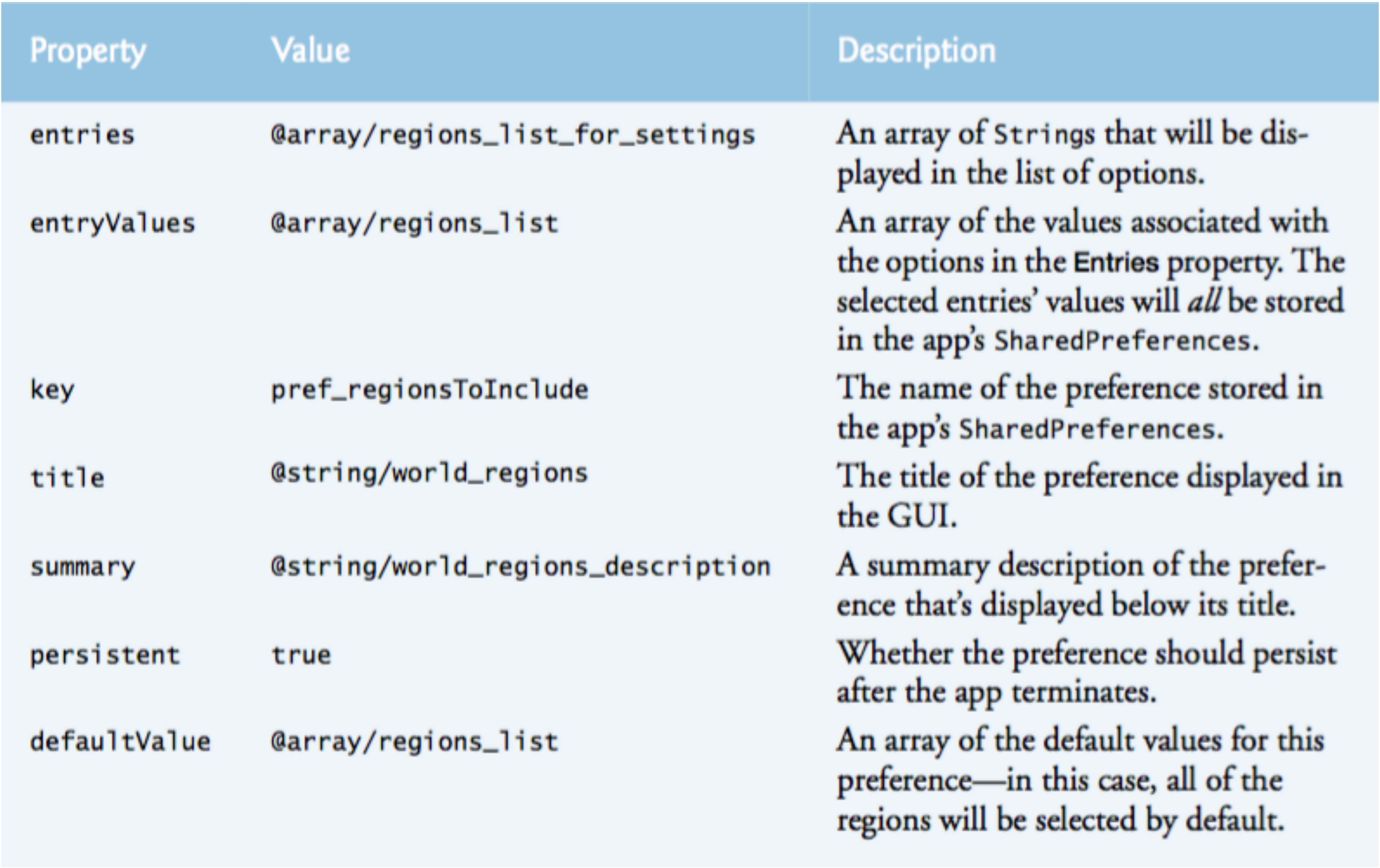
Regions

☑ Africa
☑ Asia
☑ Europe
☑ North America
☑ Oceania
☑ South America

CANCEL OK

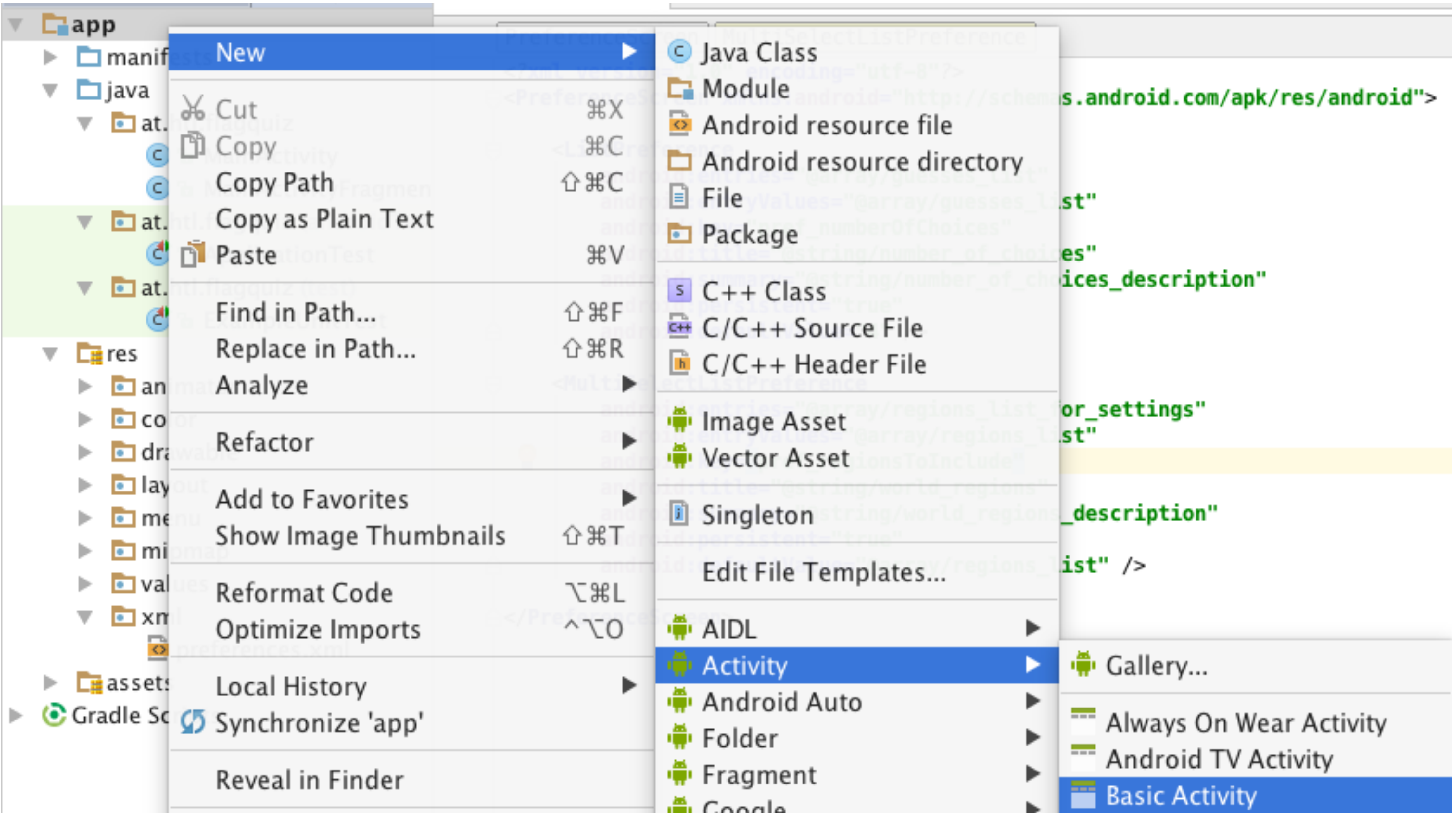


Property	Value	Description
`entries`	`@array/guesses_list`	An array of `Strings` that will be displayed in the list of options.
`entryValues`	`@array/guesses_list`	An array of values associated with the options in the **Entries** property. The selected entry's value will be stored in the app's `SharedPreferences`.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

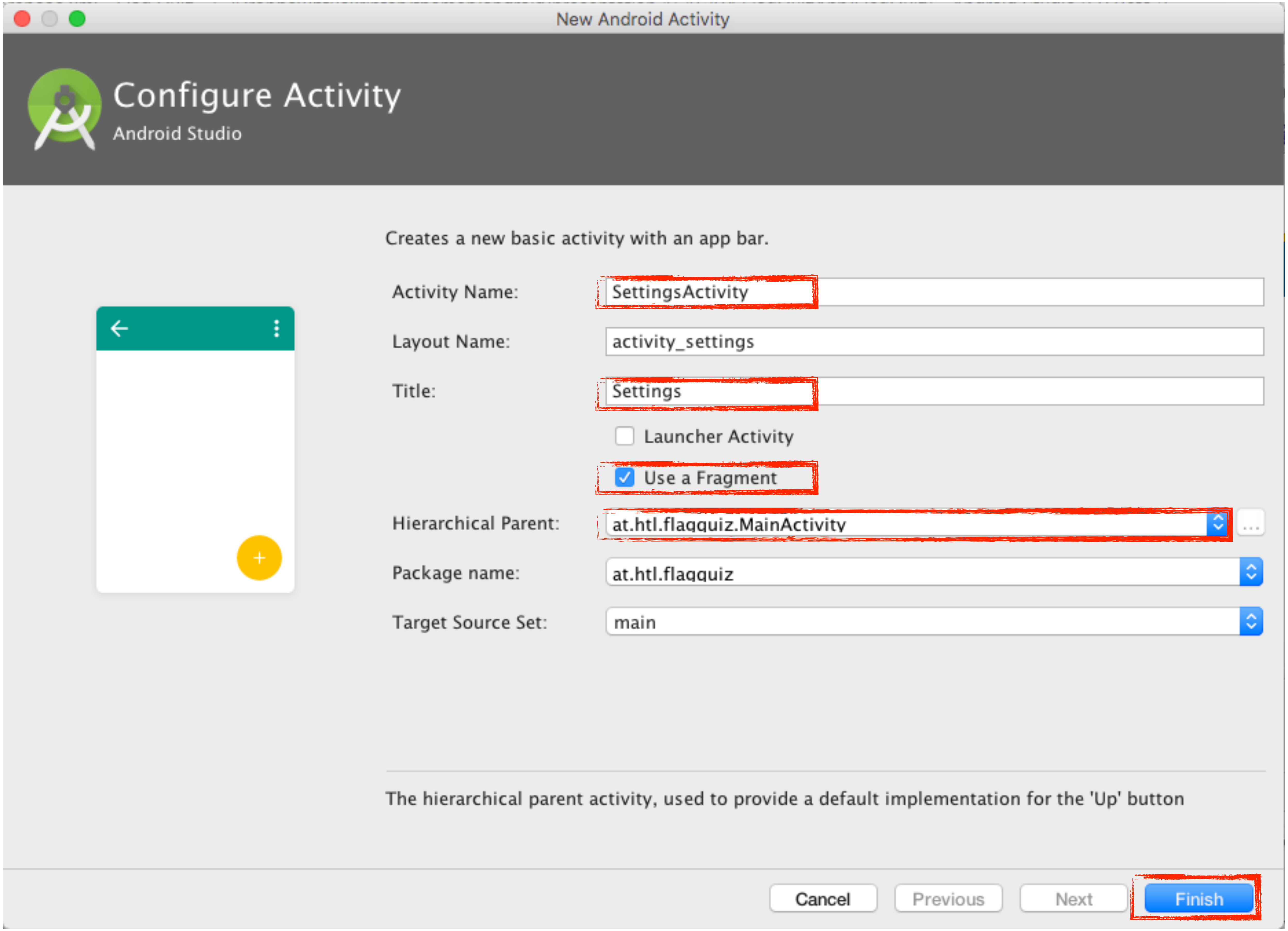


Property	Value	Description
`entries`	`@array/regions_list_for_settings`	An array of `Strings` that will be displayed in the list of options.
`entryValues`	`@array/regions_list`	An array of the values associated with the options in the **Entries** property. The selected entries' values will *all* be stored in the app's `SharedPreferences`.
`key`	`pref_regionsToInclude`	The name of the preference stored in the app's `SharedPreferences`.
`title`	`@string/world_regions`	The title of the preference displayed in the GUI.
`summary`	`@string/world_regions_description`	A summary description of the preference that's displayed below its title.
`persistent`	`true`	Whether the preference should persist after the app terminates.
`defaultValue`	`@array/regions_list`	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
manifests
java
res
assets
Gradle Scripts
New
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy as Plain Text
Paste ⌘V
Find in Path... ⇧⌘F
Replace in Path... ⇧⌘R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails ⇧⌘T
Reformat Code ⌥⌘L
Optimize Imports ^⌥O
Local History
Synchronize 'app'
Reveal in Finder
Java Class
Module
Android resource file
Android resource directory
File
Package
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Singleton
Edit File Templates...
AIDL
Activity
Android Auto
Folder
Fragment
Google
Gallery...
Always On Wear Activity
Android TV Activity
Basic Activity

app - New -> Activity -> Basic Activity



New Android Activity
Configure Activity
Android Studio

Creates a new basic activity with an app bar.

Activity Name: SettingsActivity

Layout Name: activity_settings

Title: Settings

☐ Launcher Activity

☑ Use a Fragment

Hierarchical Parent: at.htl.flagquiz.MainActivity

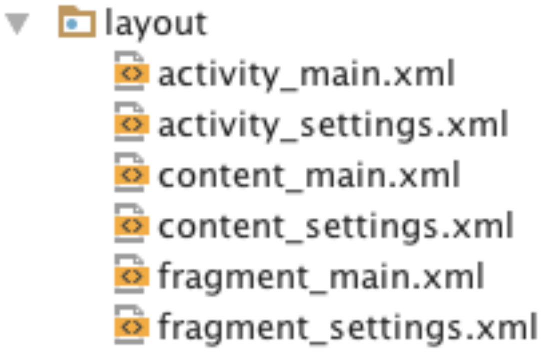
Package name: at.htl.flagquiz

Target Source Set: main

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

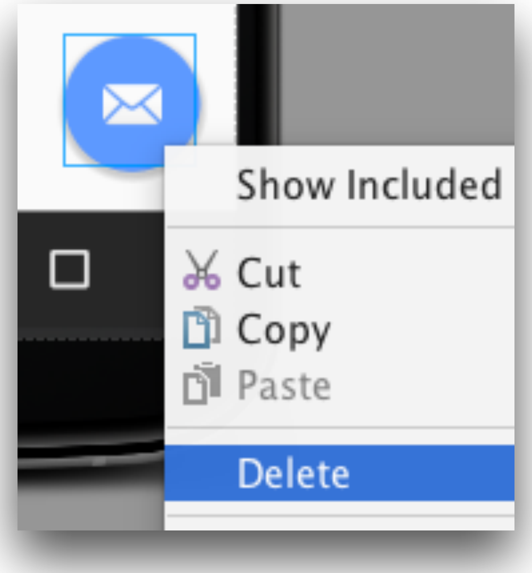
Cancel
Previous
Next
Finish

Settings



- layout
  - activity_main.xml
  - activity_settings.xml
  - content_main.xml
  - content_settings.xml
  - fragment_main.xml
  - fragment_settings.xml

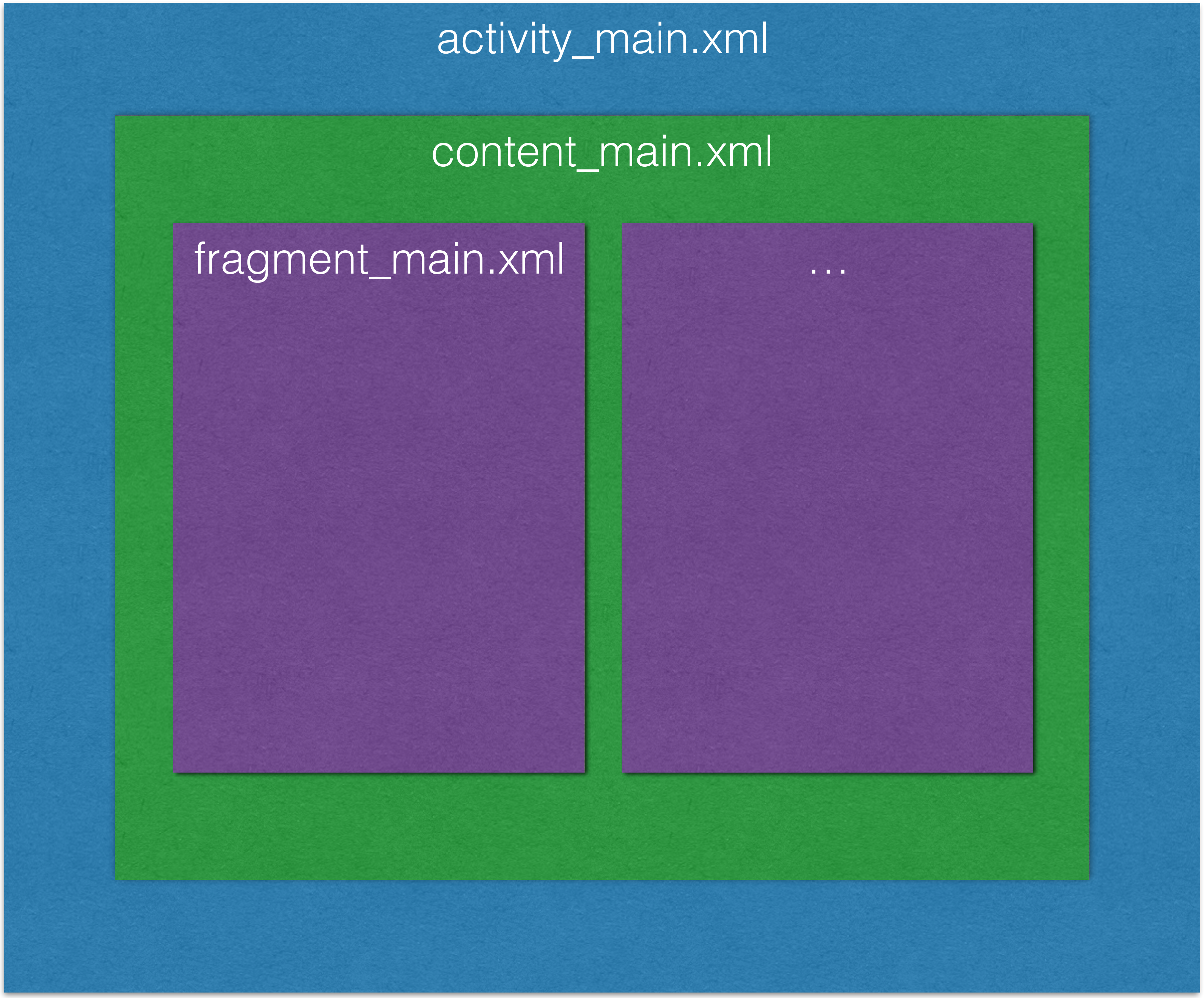
Die settings Files wurden in layout erstellt


Show Included
Cut
Copy
Paste
Delete

Delete the FloatingActionButton

Erstellen der GUI

Derzeitige GUI

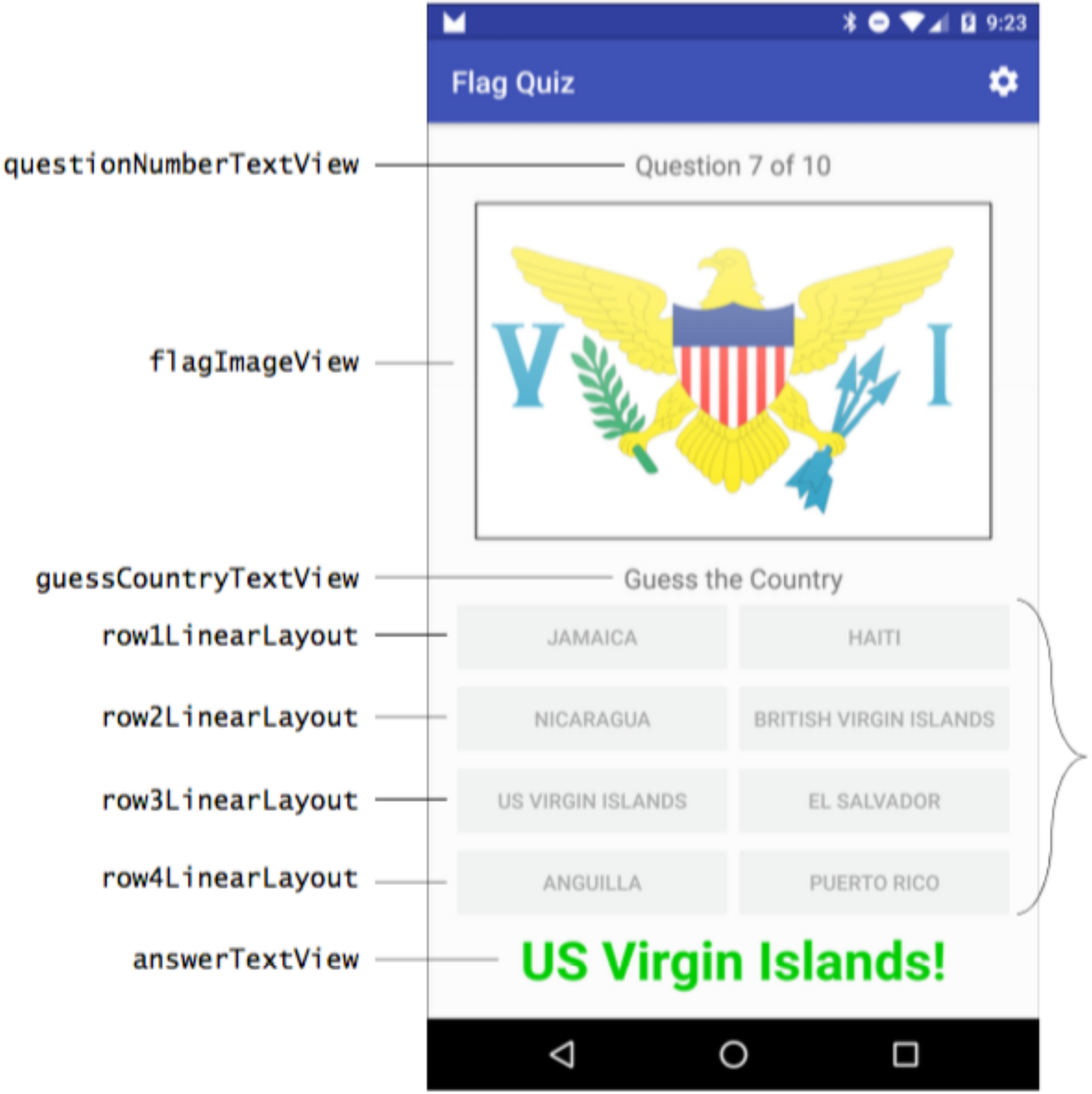


activity_main.xml

content_main.xml

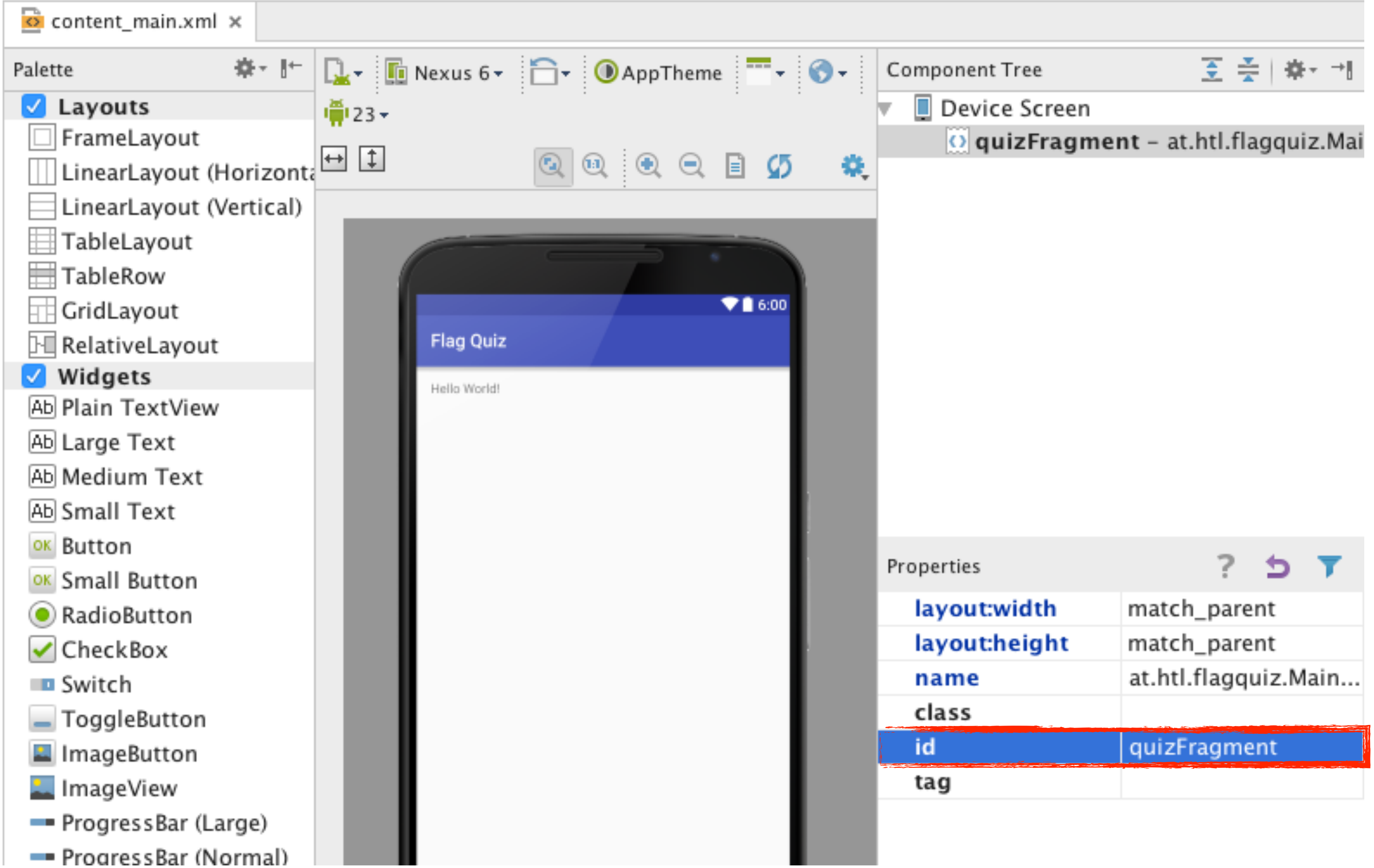
fragment_main.xml

…


9:23
Flag Quiz
questionNumberTextView
Question 7 of 10
flagImageView
V
I
guessCountryTextView
Guess the Country
row1LinearLayout
JAMAICA
HAITI
row2LinearLayout
NICARAGUA
BRITISH VIRGIN ISLANDS
row3LinearLayout
US VIRGIN ISLANDS
EL SALVADOR
row4LinearLayout
ANGUILLA
PUERTO RICO
answerTextView
US Virgin Islands!

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

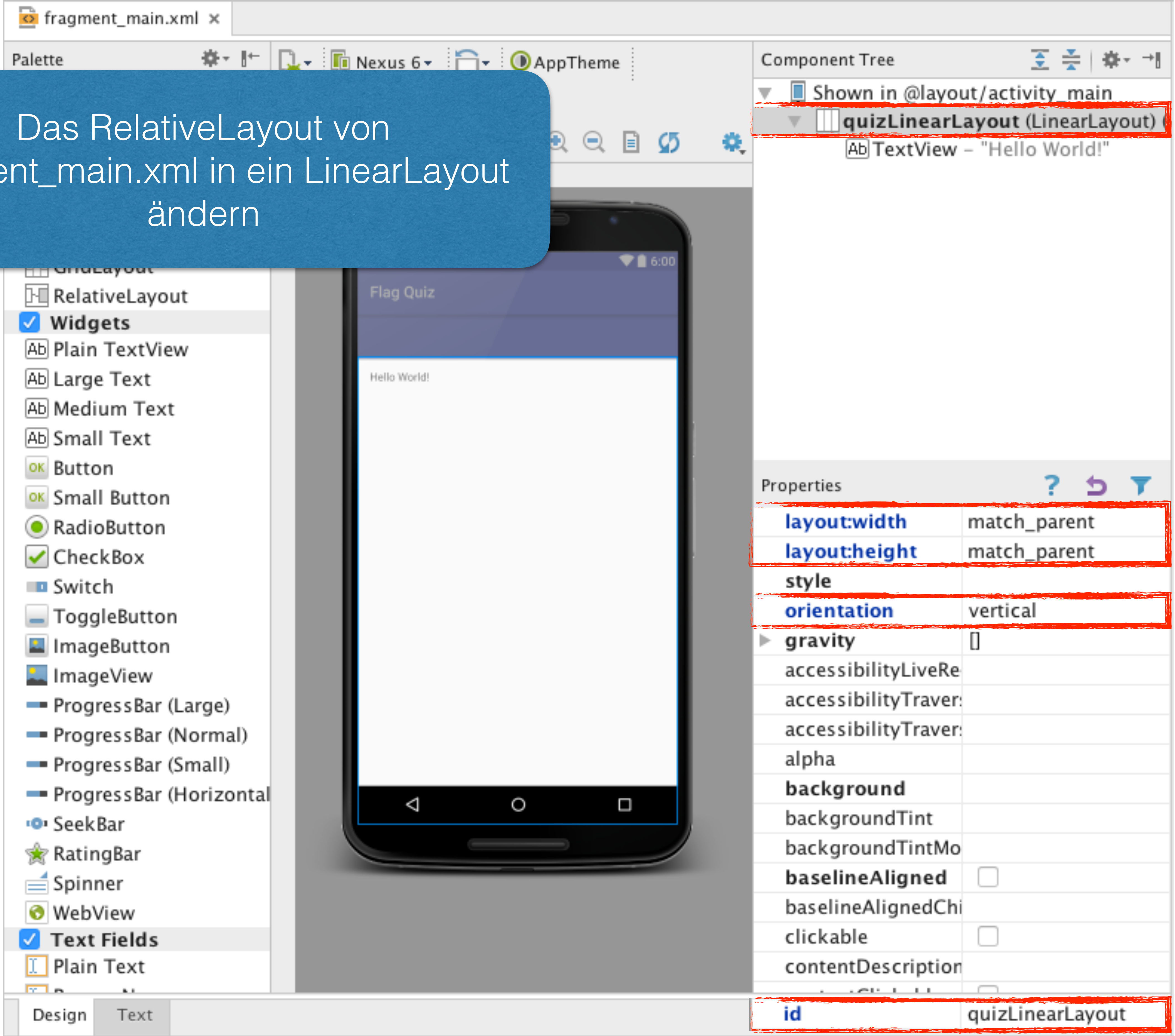
Fragment umbenennen


content_main.xml
Palette
Layouts
FrameLayout
LinearLayout (Horizonta
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)
Nexus 6
AppTheme
23
6:00
Flag Quiz
Hello World!
Component Tree
Device Screen
Properties

quizFragment – at.htl.flagquiz.Mai

layout:width	match_parent
layout:height	match_parent
name	at.htl.flagquiz.Main...
class	
id	quizFragment
tag	

Das RelativeLayout von fragment_main.xml in ein LinearLayout ändern

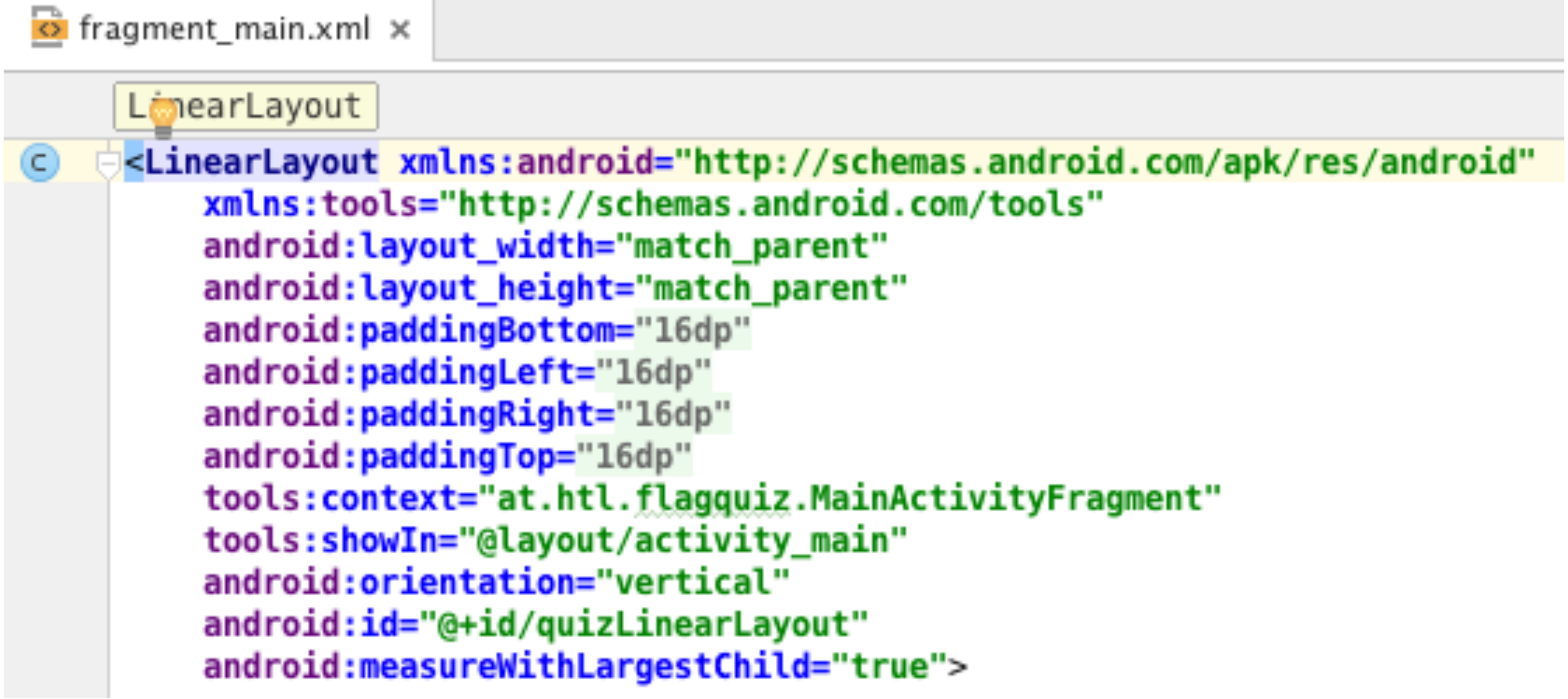

fragment_main.xml
Palette
Nexus 6
AppTheme
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizontal
SeekBar
RatingBar
Spinner
WebView
Text Fields
Plain Text
6:00
Flag Quiz
Hello World!
Design
Text
Component Tree
Properties

Shown in @layout/activity_main
quizLinearLayout (LinearLayout)
TextView – "Hello World!"

layout:width	match_parent
layout:height	match_parent
style	
orientation	vertical
gravity	[]
accessibilityLiveRe	
accessibilityTraver	
accessibilityTraver	
alpha	
background	
backgroundTint	
backgroundTintMo	
baselineAligned	☐
baselineAlignedChi	
clickable	☐
contentDescription	
id	quizLinearLayout

Look-and-Feel Observation 4.2

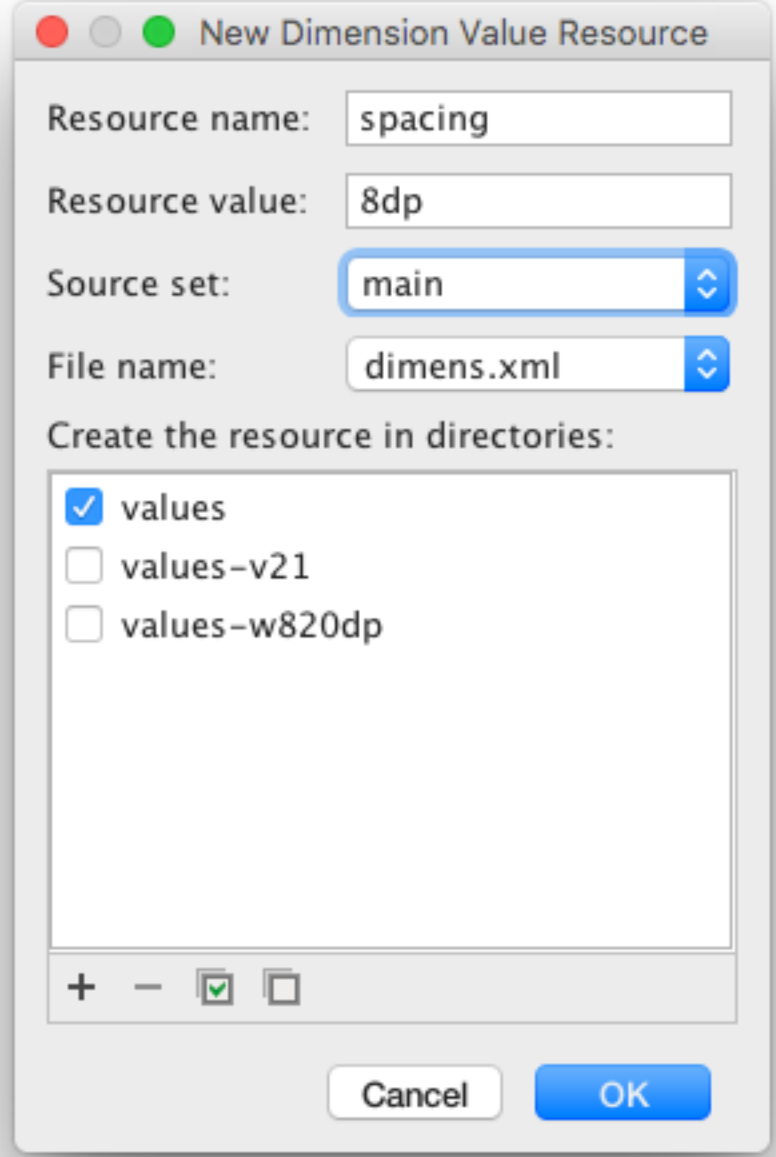
According to the Android design guidelines, 16dp *is the recommended space between the edges of a device's touchable screen area and the app's content; however, many apps (such as games) use the full screen.*


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



New Dimension Value Resource

Resource name: spacing

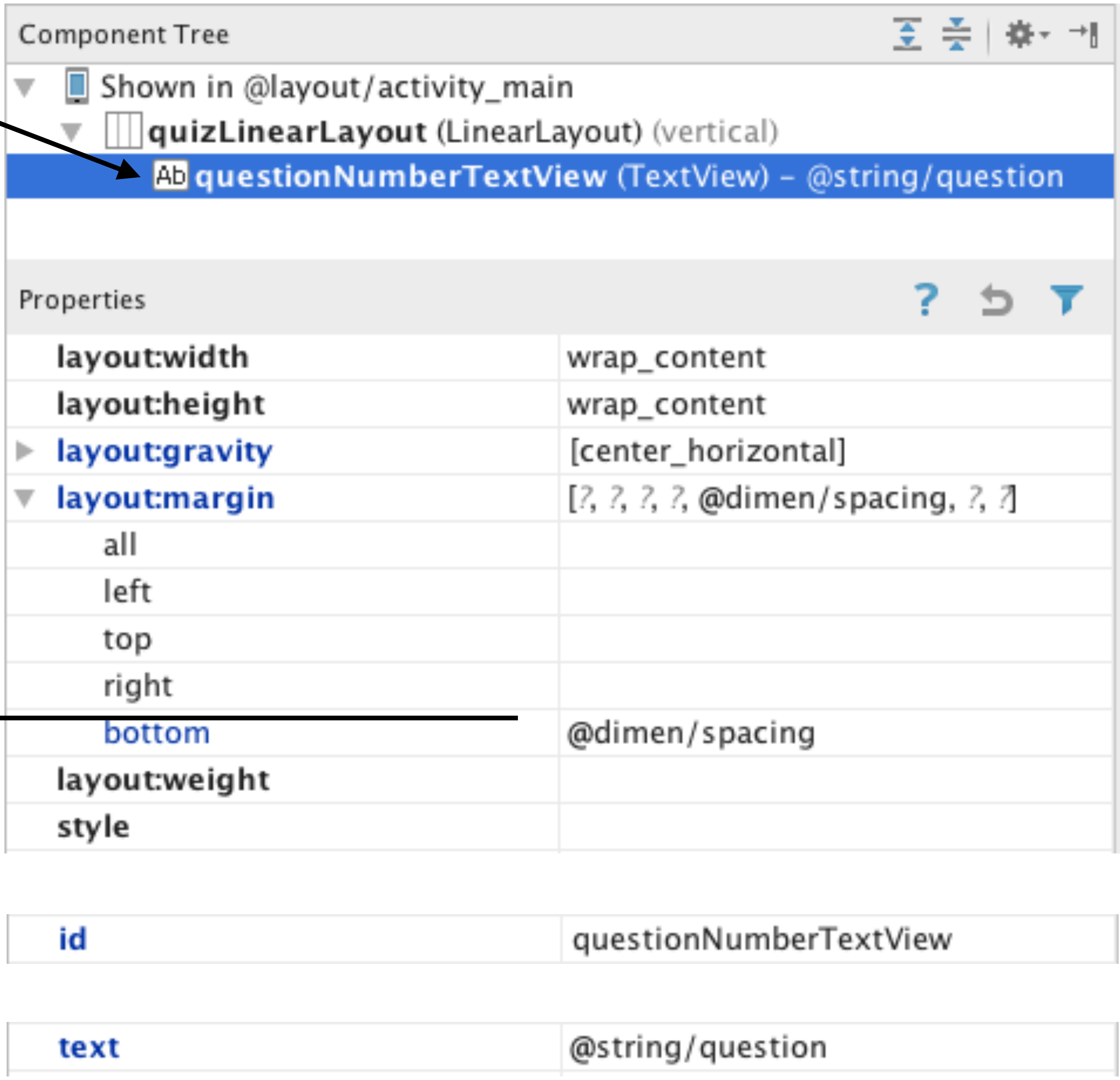
Resource value: 8dp

Source set: main

File name: dimens.xml

Create the resource in directories:

- ☑ values
- ☐ values-v21
- ☐ values-w820dp

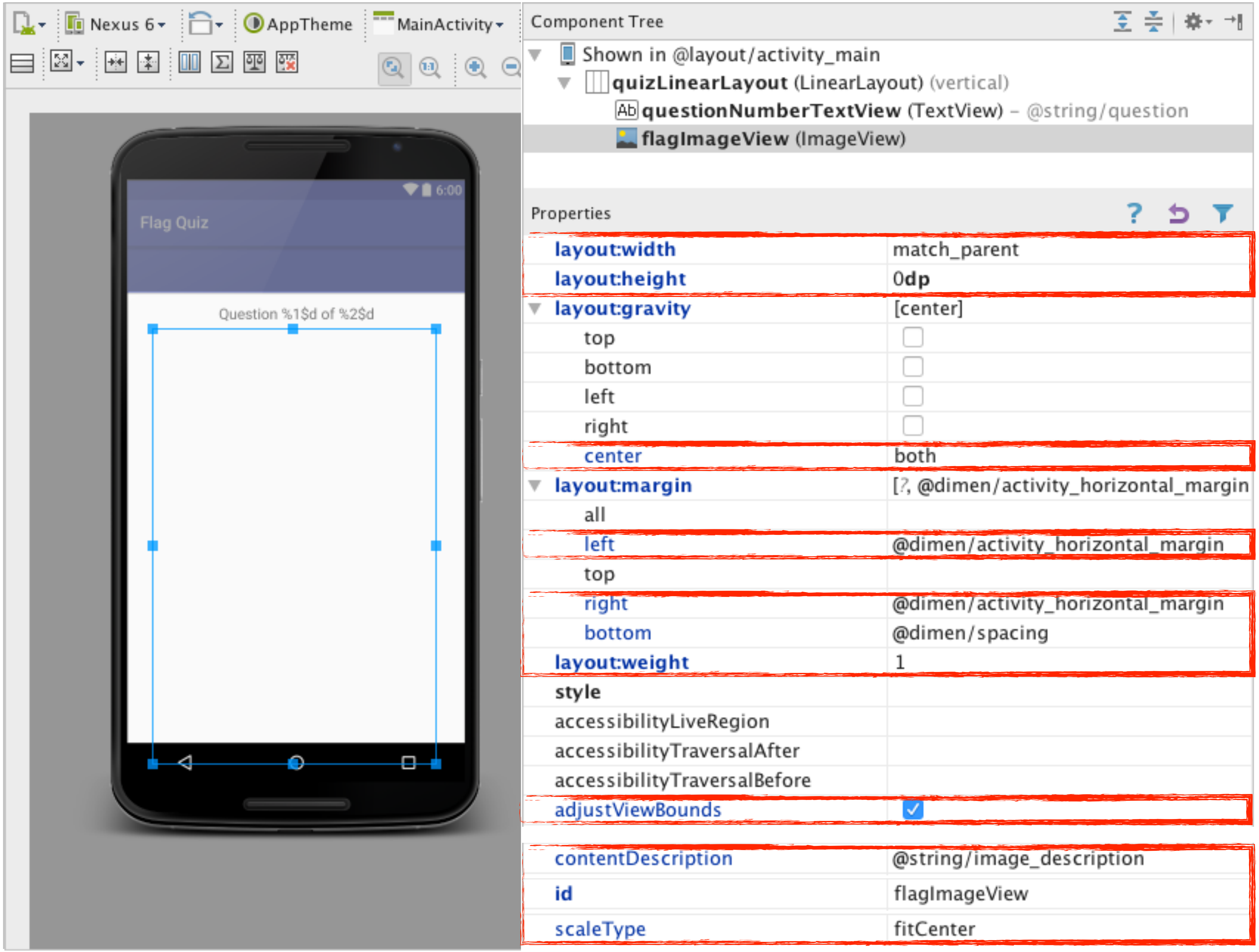


Component Tree

- Shown in @layout/activity_main
  - **quizLinearLayout** (LinearLayout) (vertical)
    - **questionNumberTextView** (TextView) - @string/question

Properties

layout:width	wrap_content
layout:height	wrap_content
layout:gravity	[center_horizontal]
layout:margin	[?, ?, ?, ?, @dimen/spacing, ?, ?]
all	
left	
top	
right	
bottom	@dimen/spacing
layout:weight	
style	
id	questionNumberTextView
text	@string/question



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	☐
bottom	☐
left	☐
right	☐
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	☑
contentDescription	@string/image_description
id	flagImageView
scaleType	fitCenter

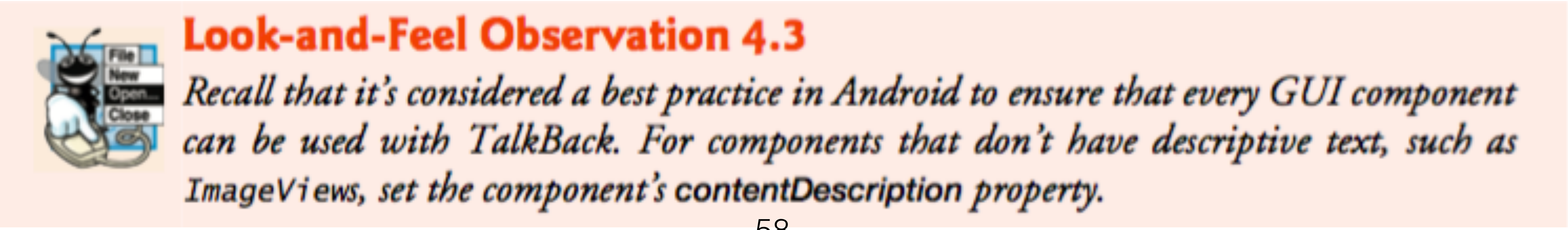
- layout:height: 0dp … dadurch wird die Höhe der ImageView durch das layout:weight - Property bestimmt

- 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

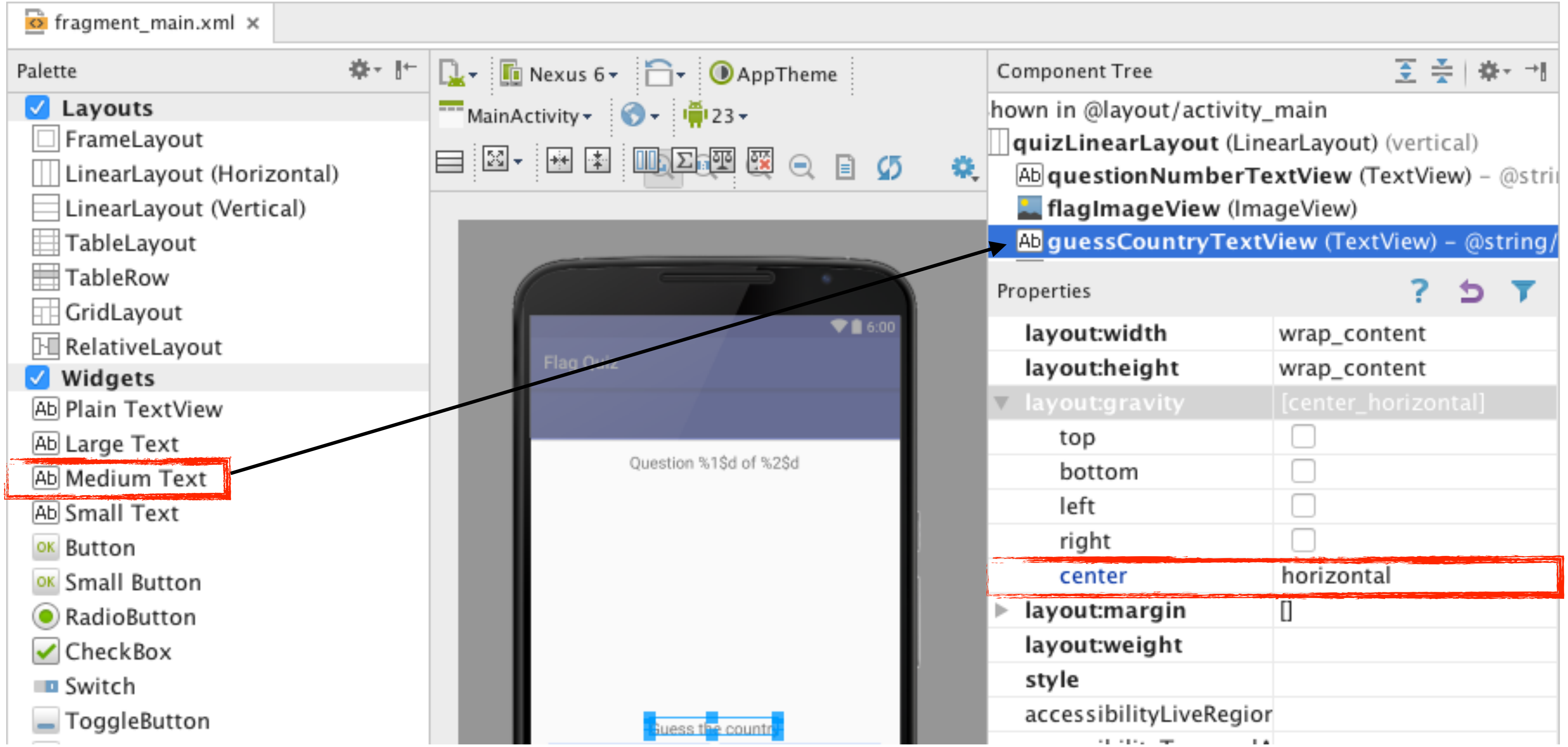
- scaleType: fitCenter … die ImageView sollte soll so skaliert werden, dass entweder die gesamte verfügbare Höhe oder Breite verwendet wird unter Einhaltung der aspect ratio. Die jeweils andere Dimension wird zentriert



Look-and-Feel Observation 4.3

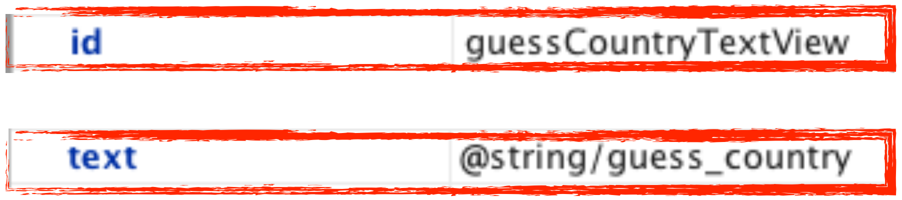
Recall that it's considered a best practice in Android to ensure that every GUI component can be used with TalkBack. For components that don't have descriptive text, such as `ImageViews`*, set the component's* `contentDescription` *property.*

guessCountryTextView

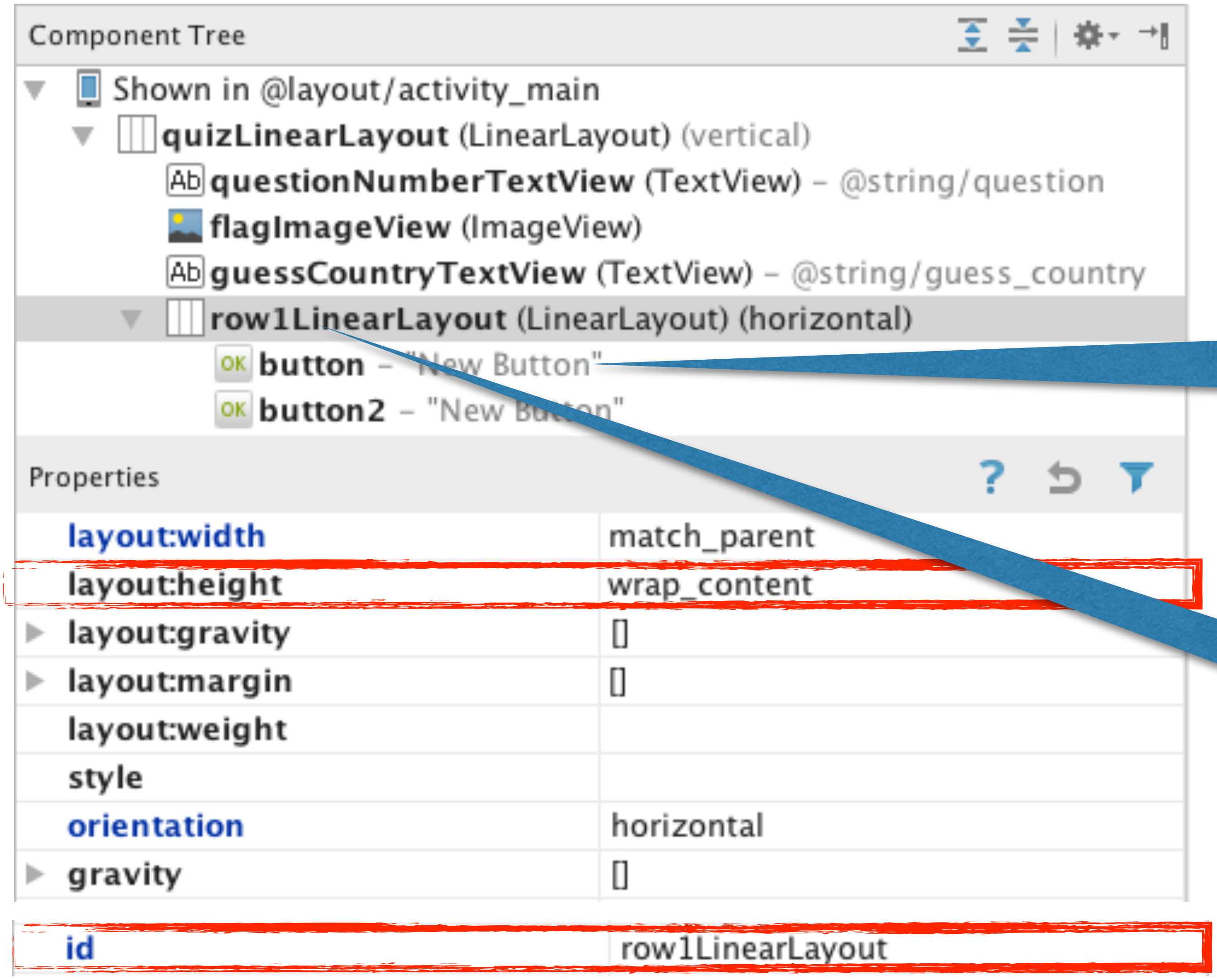

fragment_main.xml
Palette
Layouts
FrameLayout
LinearLayout (Horizontal)
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
Nexus 6
AppTheme
MainActivity
23
6:00
Flag Quiz
Question %1$d of %2$d
Guess the country
Component Tree
Properties

hown in @layout/activity_main
quizLinearLayout (LinearLayout) (vertical)
questionNumberTextView (TextView) – @stri
flagImageView (ImageView)
guessCountryTextView (TextView) – @string/

layout:width	wrap_content
layout:height	wrap_content
layout:gravity	[center_horizontal]
top	☐
bottom	☐
left	☐
right	☐
center	horizontal
layout:margin	[]
layout:weight	
style	
accessibilityLiveRegior	



id	guessCountryTextView
text	@string/guess_country



Component Tree

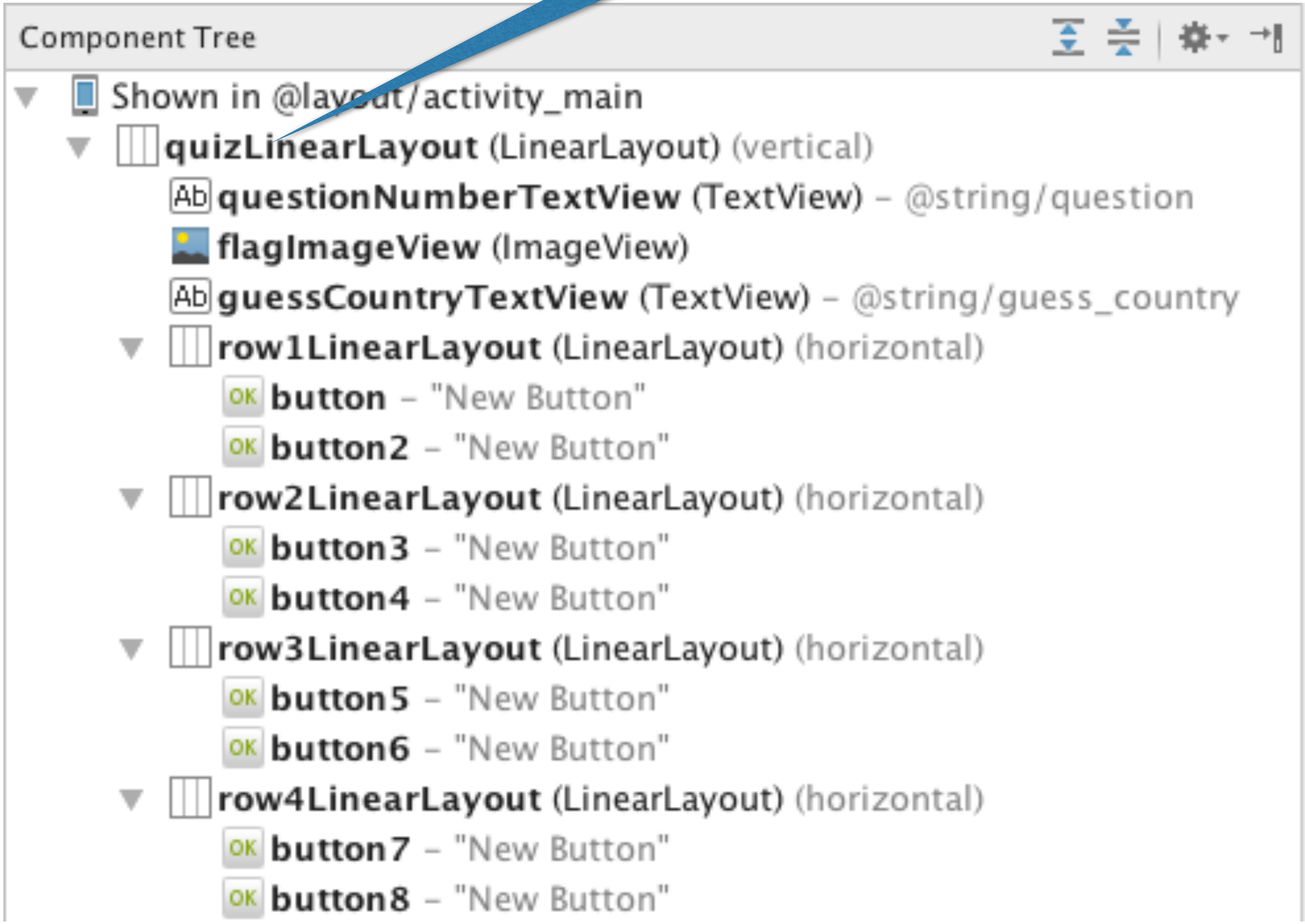
- Shown in @layout/activity_main
  - **quizLinearLayout** (LinearLayout) (vertical)
    - **questionNumberTextView** (TextView) – @string/question
    - **flagImageView** (ImageView)
    - **guessCountryTextView** (TextView) – @string/guess_country
    - **row1LinearLayout** (LinearLayout) (horizontal)
      - **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

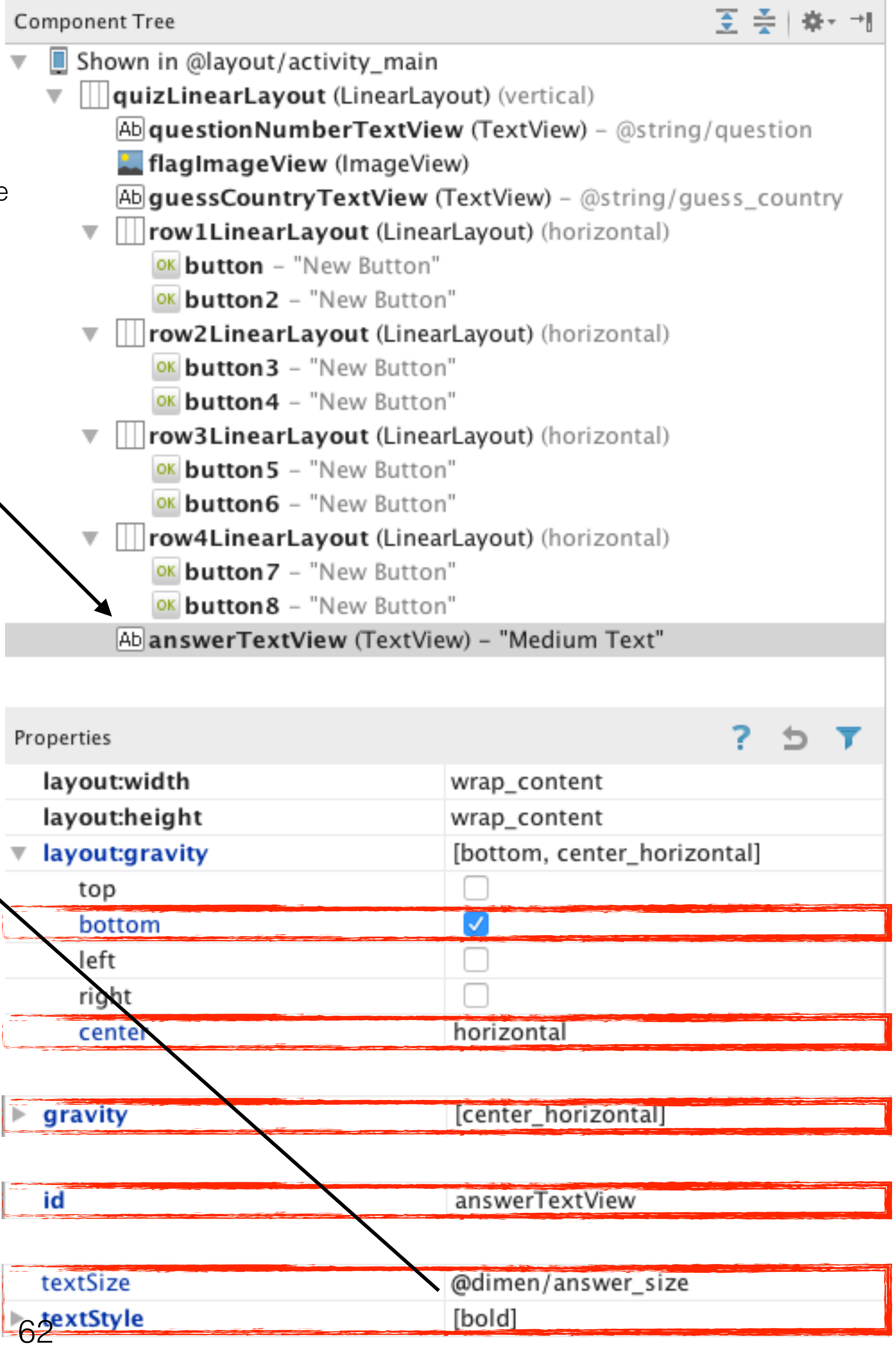
- Shown in @layout/activity_main
  - **quizLinearLayout** (LinearLayout) (vertical)
    - **questionNumberTextView** (TextView) – @string/question
    - **flagImageView** (ImageView)
    - **guessCountryTextView** (TextView) – @string/guess_country
    - **row1LinearLayout** (LinearLayout) (horizontal)
      - **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

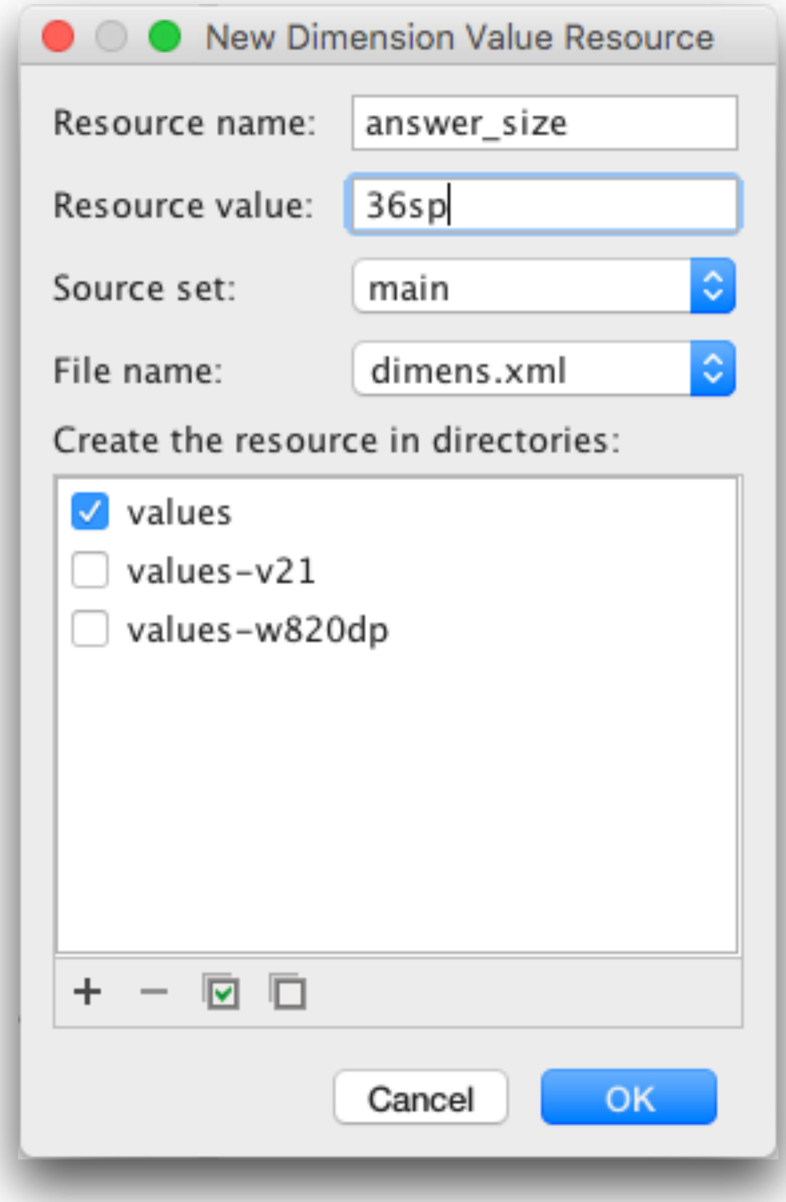
scale… zB Fontgröße

add a Medium Text

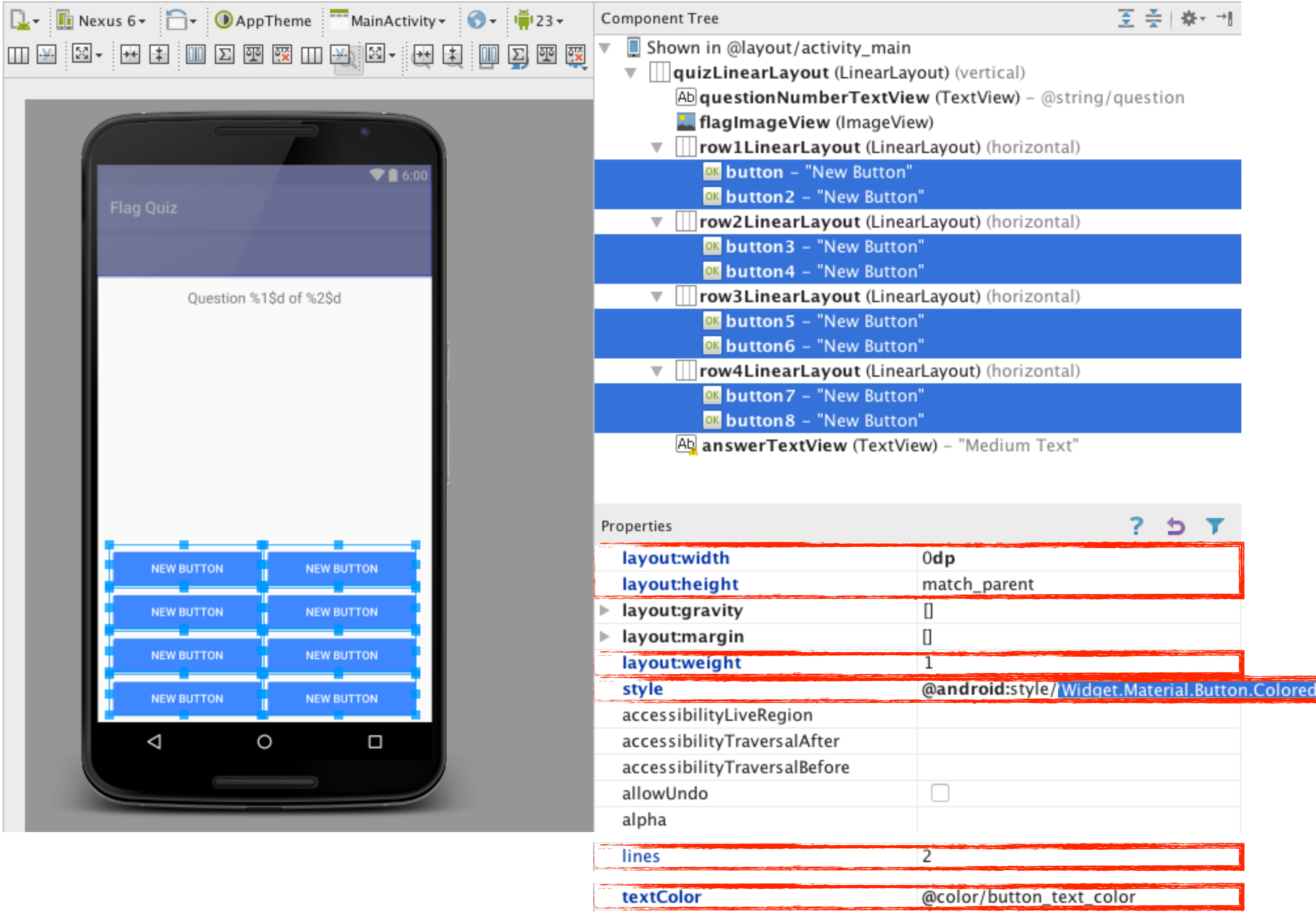


- Shown in @layout/activity_main
  - **quizLinearLayout** (LinearLayout) (vertical)
    - **questionNumberTextView** (TextView) – @string/question
    - **flagImageView** (ImageView)
    - **guessCountryTextView** (TextView) – @string/guess_country
    - **row1LinearLayout** (LinearLayout) (horizontal)
      - **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"

layout:width	wrap_content
layout:height	wrap_content
layout:gravity	[bottom, center_horizontal]
top	☐
bottom	☑
left	☐
right	☐
center	horizontal
gravity	[center_horizontal]
id	answerTextView
textSize	@dimen/answer_size
textStyle	[bold]



Resource name: answer_size
Resource value: 36sp
Source set: main
File name: dimens.xml
Create the resource in directories:
- ☑ values
- ☐ values-v21
- ☐ values-w820dp



Nexus 6
AppTheme
MainActivity
23
6:00
Flag Quiz
Question %1$d of %2$d
NEW BUTTON
NEW BUTTON
NEW BUTTON
NEW BUTTON
NEW BUTTON
NEW BUTTON
NEW BUTTON
NEW BUTTON

Component Tree

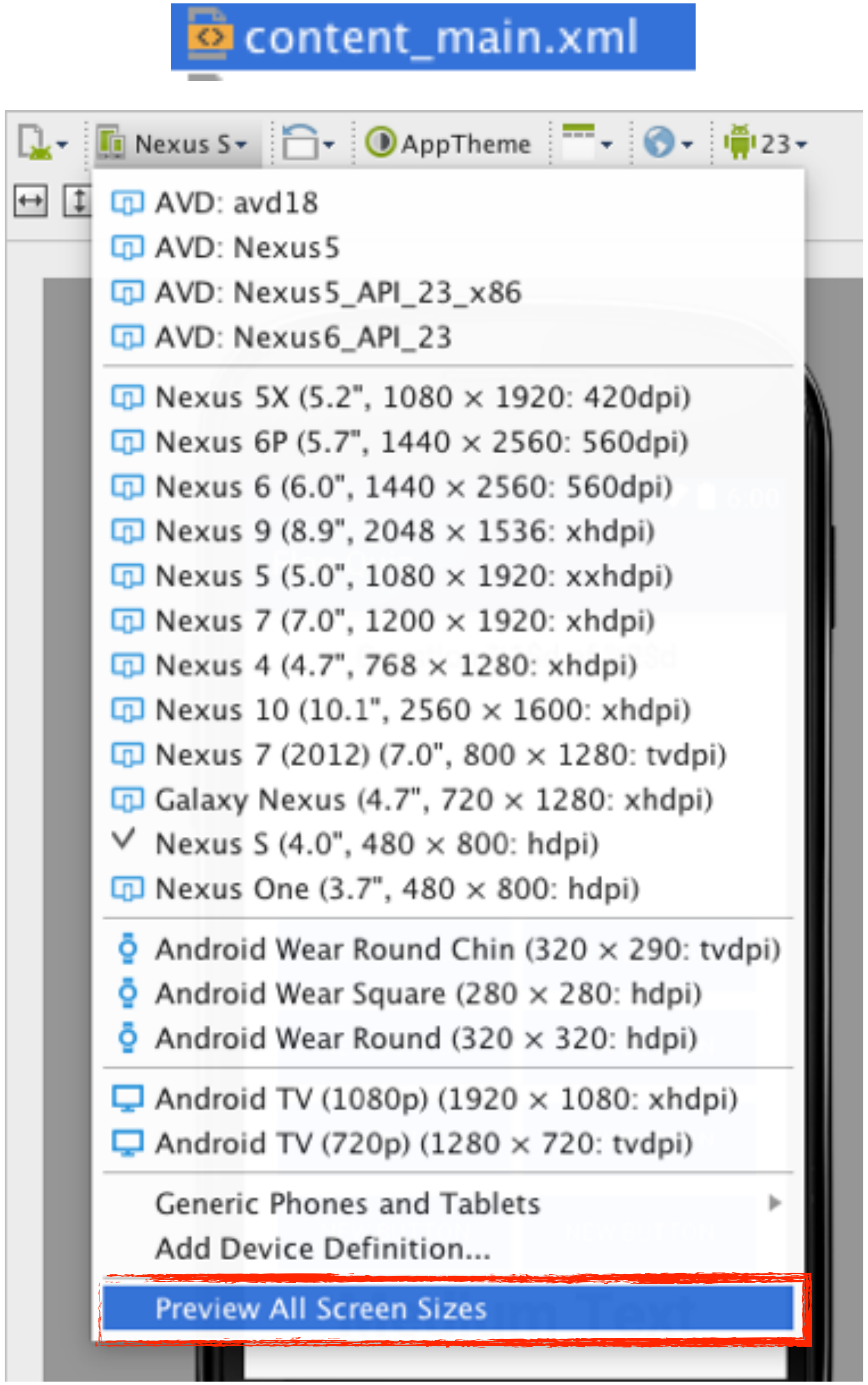
- Shown in @layout/activity_main
  - **quizLinearLayout** (LinearLayout) (vertical)
    - **questionNumberTextView** (TextView) – @string/question
    - **flagImageView** (ImageView)
    - **row1LinearLayout** (LinearLayout) (horizontal)
      - **button** – "New Button"
      - **button2** – "New Button"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

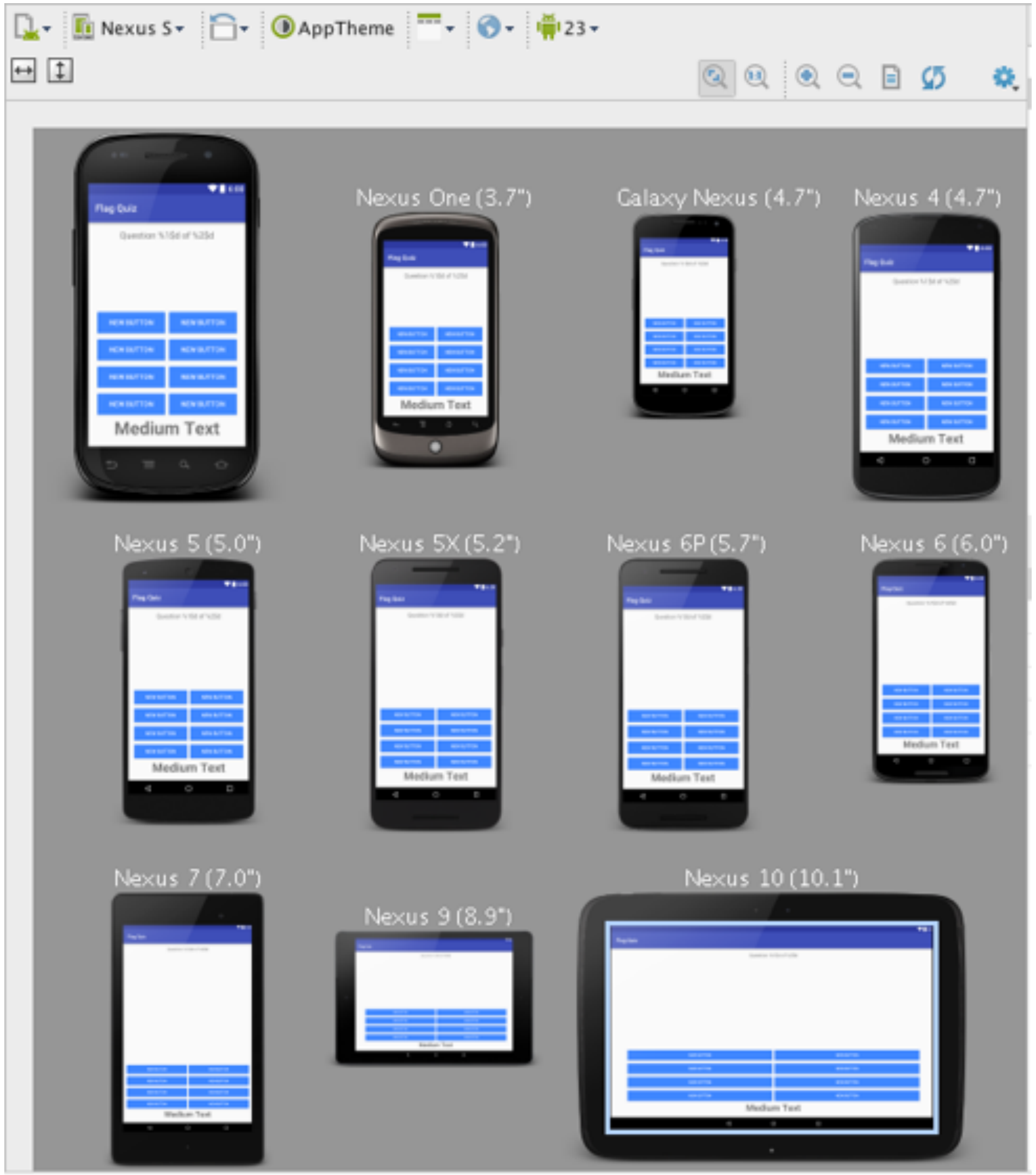
Properties

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	☐
alpha	
lines	2
textColor	@color/button_text_color

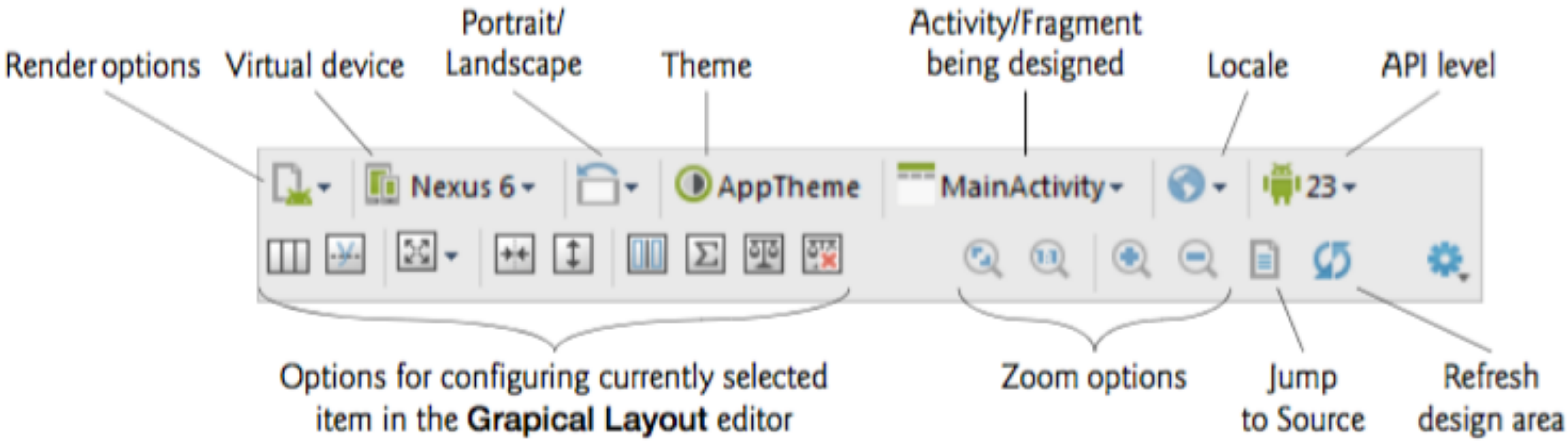
- Durch das Setzen von layout:width auf 0dp und layout:weight auf 1 wird der verfügbare horizontale Raum auf die beiden Buttons gleich aufgeteilt
- Durch layout:height auf match_parent nutzen die Buttons die verfügbare Höhe der horizontalen Layouts
- lines : 2 bewirkt, dass die Buttons die gleiche Höhe 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 färbige 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


content_main.xml
Nexus S
AppTheme
23
AVD: avd18
AVD: Nexus5
AVD: Nexus5_API_23_x86
AVD: Nexus6_API_23
Nexus 5X (5.2", 1080 × 1920: 420dpi)
Nexus 6P (5.7", 1440 × 2560: 560dpi)
Nexus 6 (6.0", 1440 × 2560: 560dpi)
Nexus 9 (8.9", 2048 × 1536: xhdpi)
Nexus 5 (5.0", 1080 × 1920: xxhdpi)
Nexus 7 (7.0", 1200 × 1920: xhdpi)
Nexus 4 (4.7", 768 × 1280: xhdpi)
Nexus 10 (10.1", 2560 × 1600: xhdpi)
Nexus 7 (2012) (7.0", 800 × 1280: tvdpi)
Galaxy Nexus (4.7", 720 × 1280: xhdpi)
✓ Nexus S (4.0", 480 × 800: hdpi)
Nexus One (3.7", 480 × 800: hdpi)
Android Wear Round Chin (320 × 290: tvdpi)
Android Wear Square (280 × 280: hdpi)
Android Wear Round (320 × 320: hdpi)
Android TV (1080p) (1920 × 1080: xhdpi)
Android TV (720p) (1280 × 720: tvdpi)
Generic Phones and Tablets
Add Device Definition...
Preview All Screen Sizes


Nexus S
AppTheme
23
Nexus One (3.7")
Galaxy Nexus (4.7")
Nexus 4 (4.7")
Nexus 5 (5.0")
Nexus 5X (5.2")
Nexus 6P (5.7")
Nexus 6 (6.0")
Nexus 7 (7.0")
Nexus 9 (8.9")
Nexus 10 (10.1")
Medium Text

Canvas Configuration Options


Render options
Virtual device
Portrait/ Landscape
Theme
Activity/Fragment being designed
Locale
API level
Nexus 6
AppTheme
MainActivity
23
Options for configuring currently selected item in the **Grapical Layout** editor
Zoom options
Jump to Source
Refresh design area

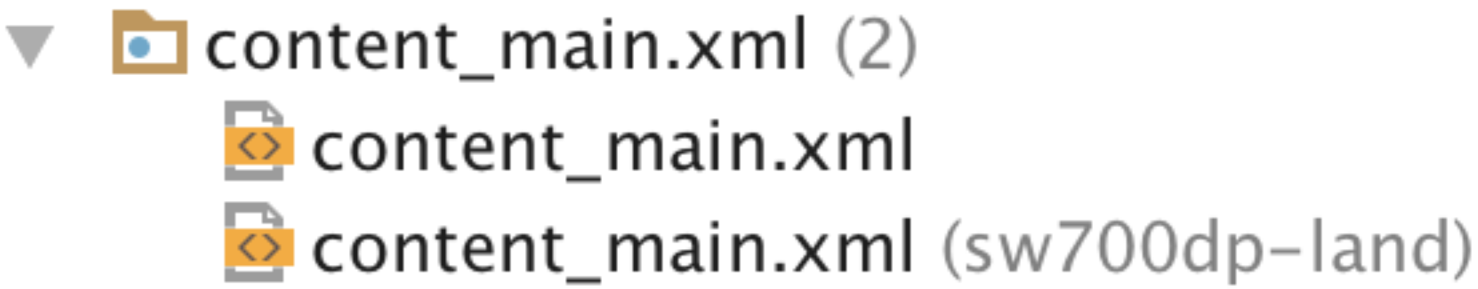
content_main.xml

- Auf Tablets sollen im Landscape - Mode das MainActivityFragment und das SettingsActivityFragment nebeneinander angezeigt werden

- Dazu wird ein weiteres content-main.xml für diese Einstellungen erstellt

- Dieses Layout wird nur bei einer minimalen Screen Width von 700dp (sw) sowie im Landscape-Mode (land) verwendet —> layout-sw700dp-land

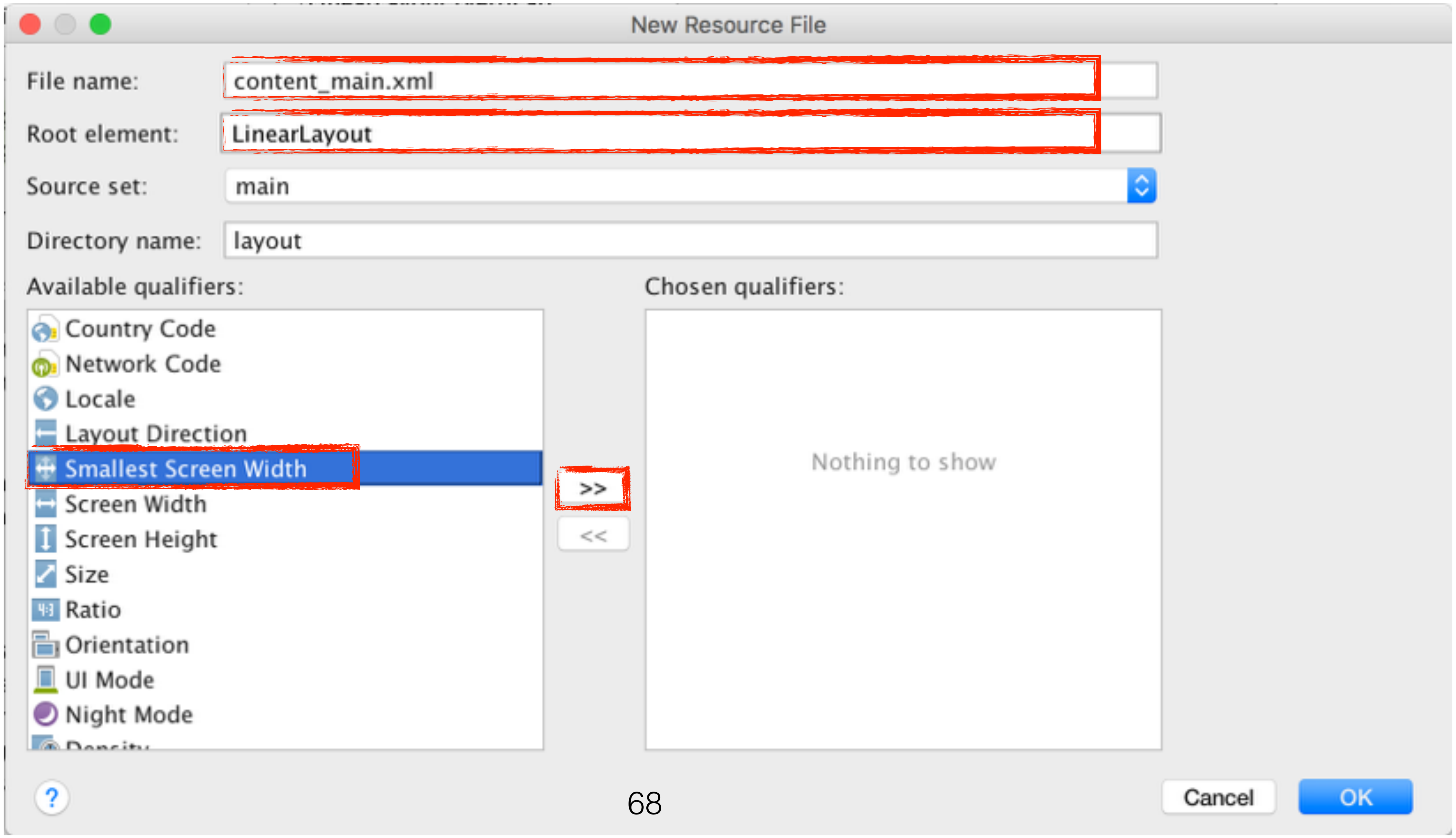
- Das Layout ohne Qualifier in den Klammern ist das Standard-Layout.



content_main.xml (2)
content_main.xml
content_main.xml (sw700dp-land)

content_main.xml
Vorbereitung für Landscape

res/layout - New - Layout Resource File

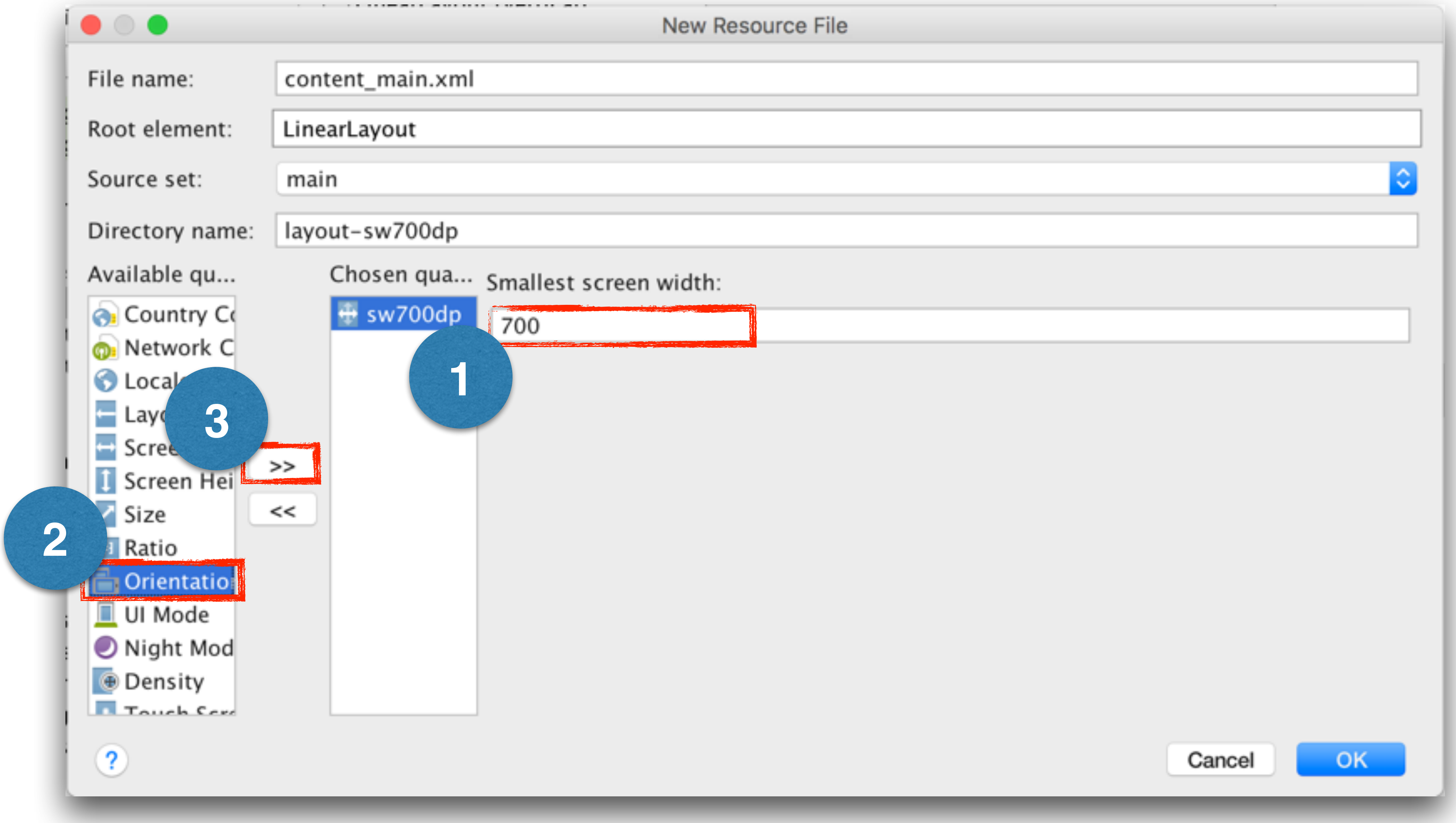

New Resource File
Available qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
Night Mode
Density
>>
<<
Chosen qualifiers:
Nothing to show
?
Cancel
OK

File name: content_main.xml

Root element: LinearLayout

Source set: main

Directory name: layout


New Resource File
Available qu...
Country Cc
Network C
Local
Lay
Scree
Screen Hei
Size
Ratio
Orientatio
UI Mode
Night Mod
Density
Chosen qua...
>>
<<
?
Cancel
OK
1
2
3

File name: content_main.xml

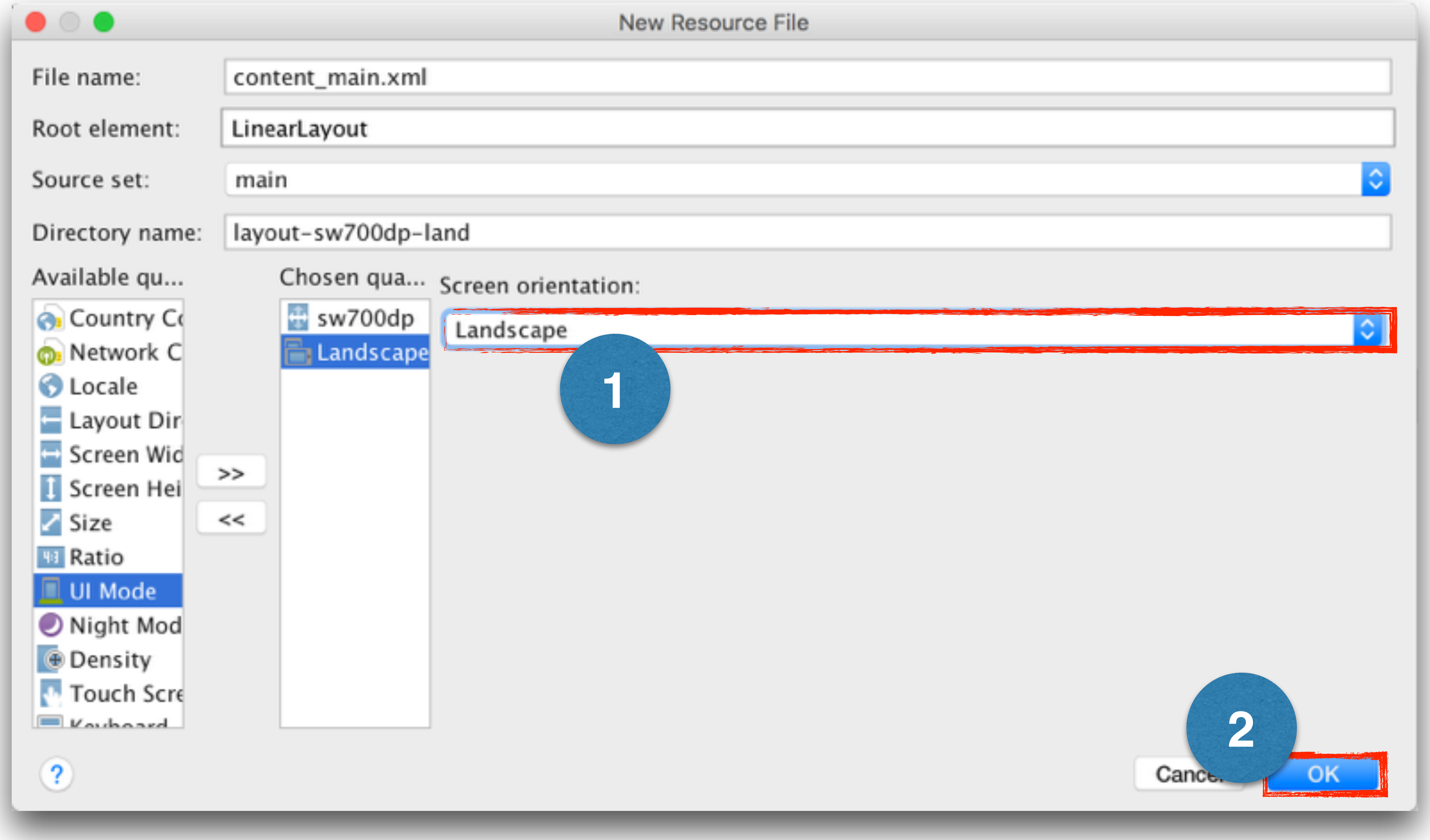
Root element: LinearLayout

Source set: main

Directory name: layout-sw700dp

Chosen qua...: sw700dp

Smallest screen width: 700


New Resource File
Available qu...
Country C
Network C
Locale
Layout Dir
Screen Wid
Screen Hei
Size
Ratio
UI Mode
Night Mod
Density
Touch Scre
Keyboard
>>
<<
Chosen qua...
1
2
?
Cancel
OK

File name: content_main.xml

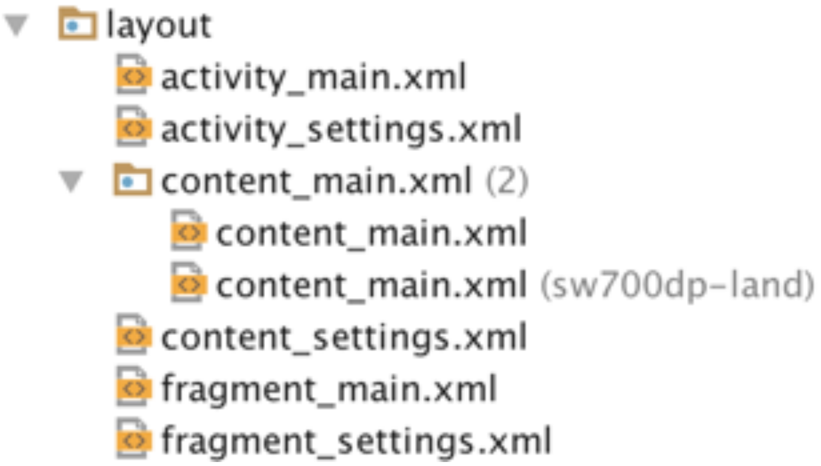
Root element: LinearLayout

Source set: main

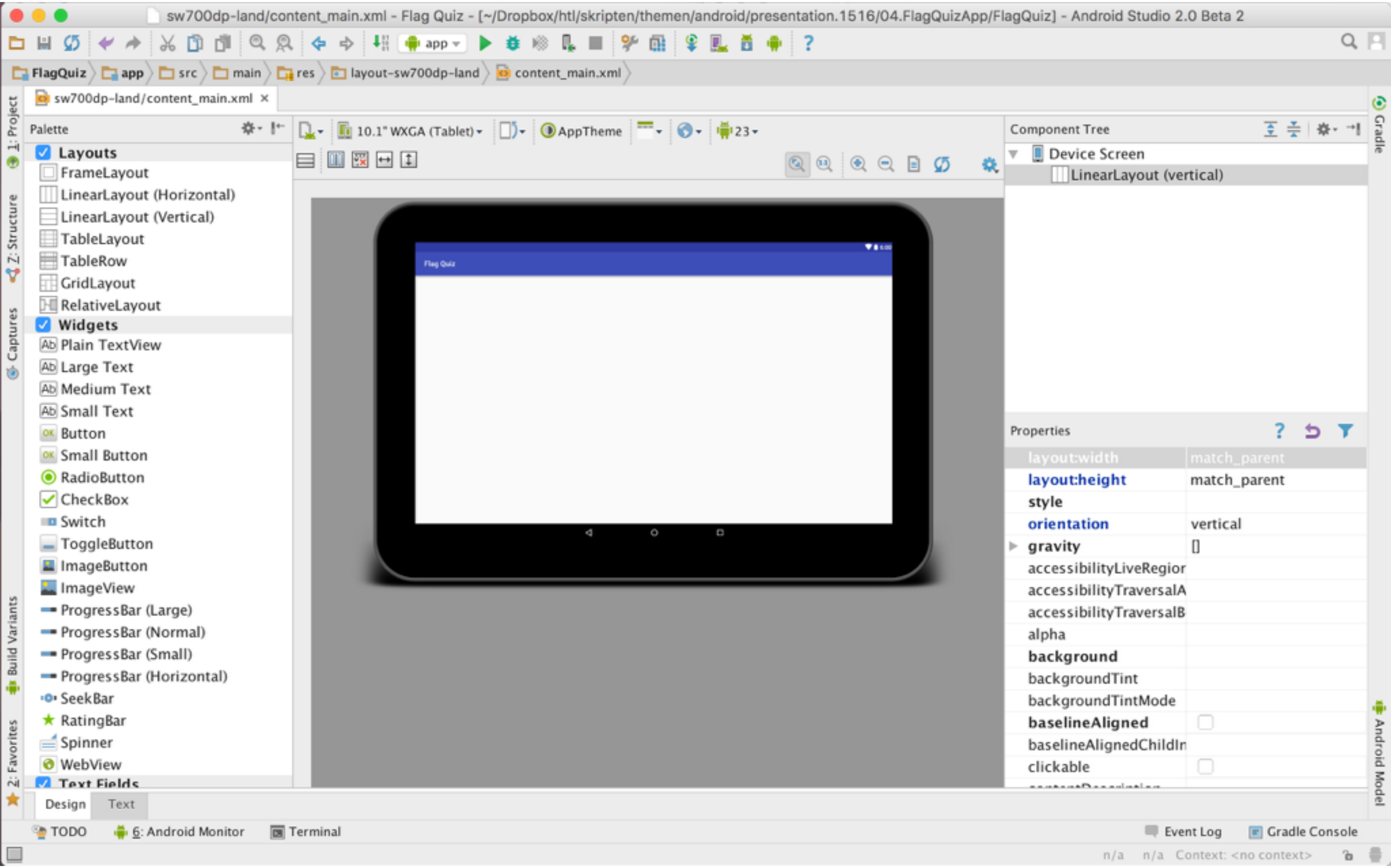
Directory name: layout-sw700dp-land

Chosen qua...: sw700dp, Landscape

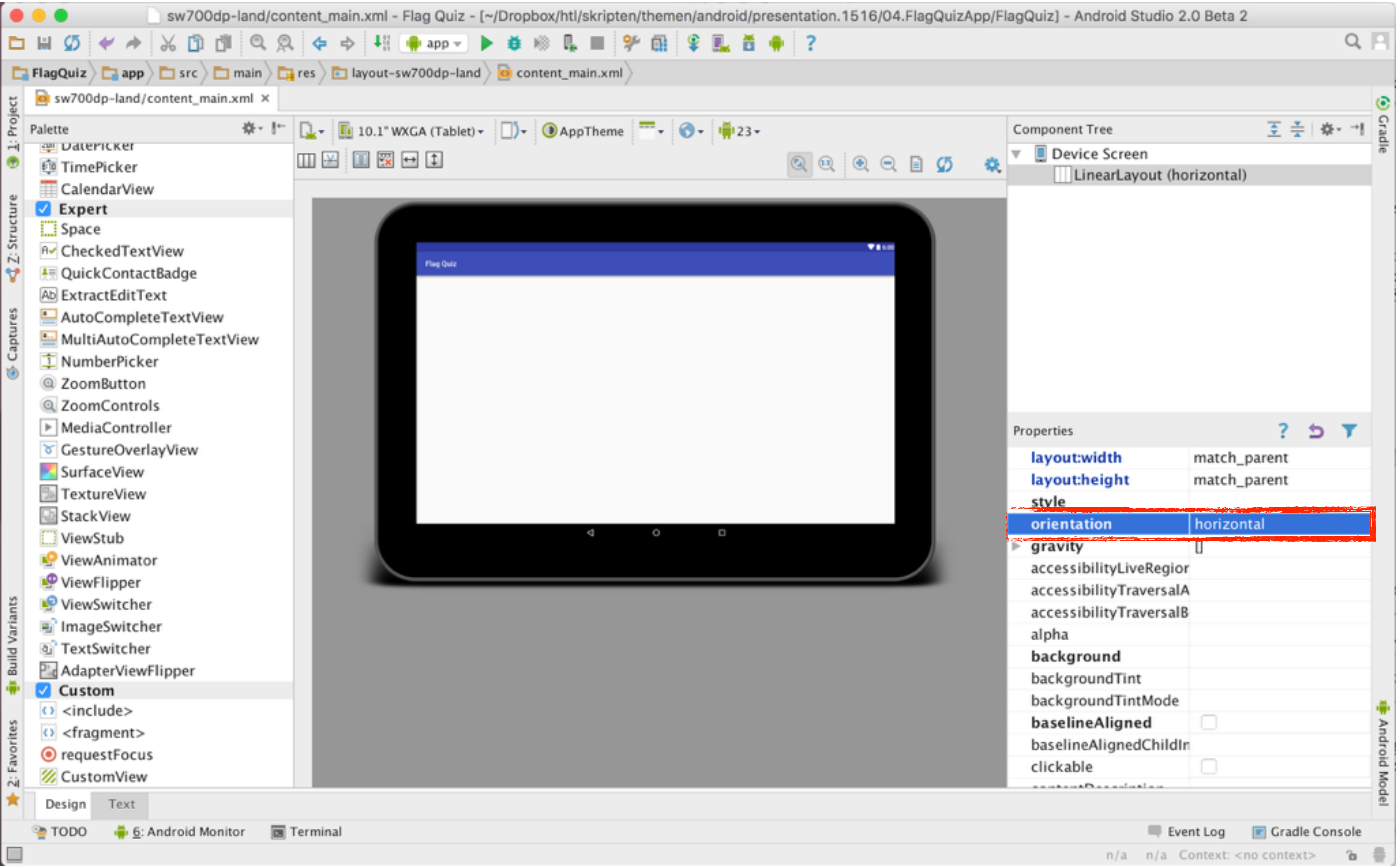
Screen orientation: Landscape



- layout
  - activity_main.xml
  - activity_settings.xml
  - content_main.xml (2)
    - content_main.xml
    - content_main.xml (sw700dp-land)
  - content_settings.xml
  - fragment_main.xml
  - fragment_settings.xml


sw700dp-land/content_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
app
FlagQuiz
app
src
main
res
layout-sw700dp-land
content_main.xml
sw700dp-land/content_main.xml
1: Project
Z: Structure
Captures
Build Variants
2: Favorites
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
10.1" WXGA (Tablet)
AppTheme
23
Flag Quiz
Component Tree
Device Screen
LinearLayout (vertical)
Properties
Gradle
Android Model
Design
Text
TODO
6: Android Monitor
Terminal
Event Log
Gradle Console
n/a
n/a
Context: <no context>

layout:width match_parent
layout:height match_parent
style
orientation vertical
gravity []
accessibilityLiveRegior
accessibilityTraversalA
accessibilityTraversalB
alpha
background
backgroundTint
backgroundTintMode
baselineAligned ☐
baselineAlignedChildIn
clickable ☐



sw700dp-land/content_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2

FlagQuiz > app > src > main > res > layout-sw700dp-land > content_main.xml

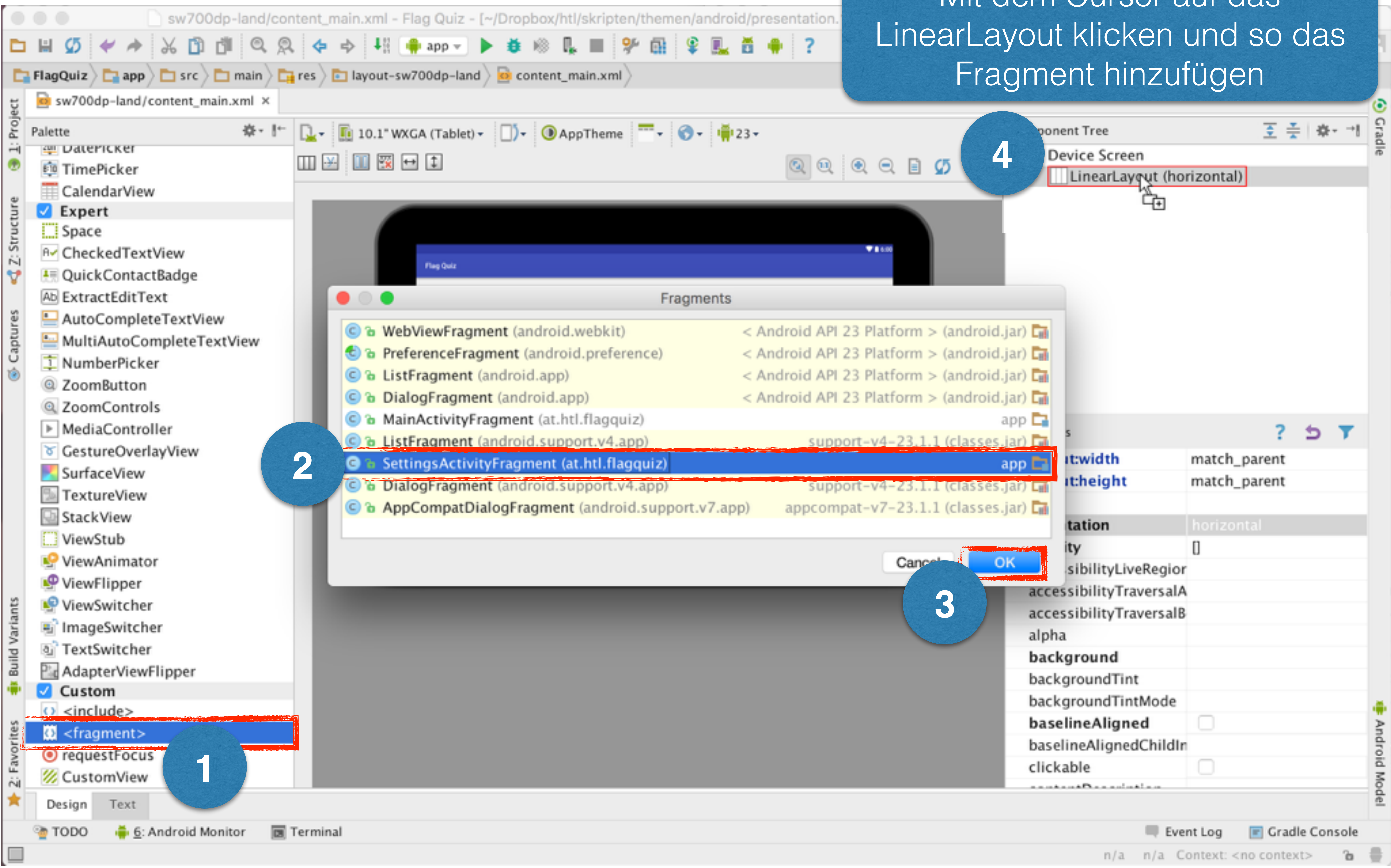
Component Tree
- Device Screen
  - LinearLayout (horizontal)

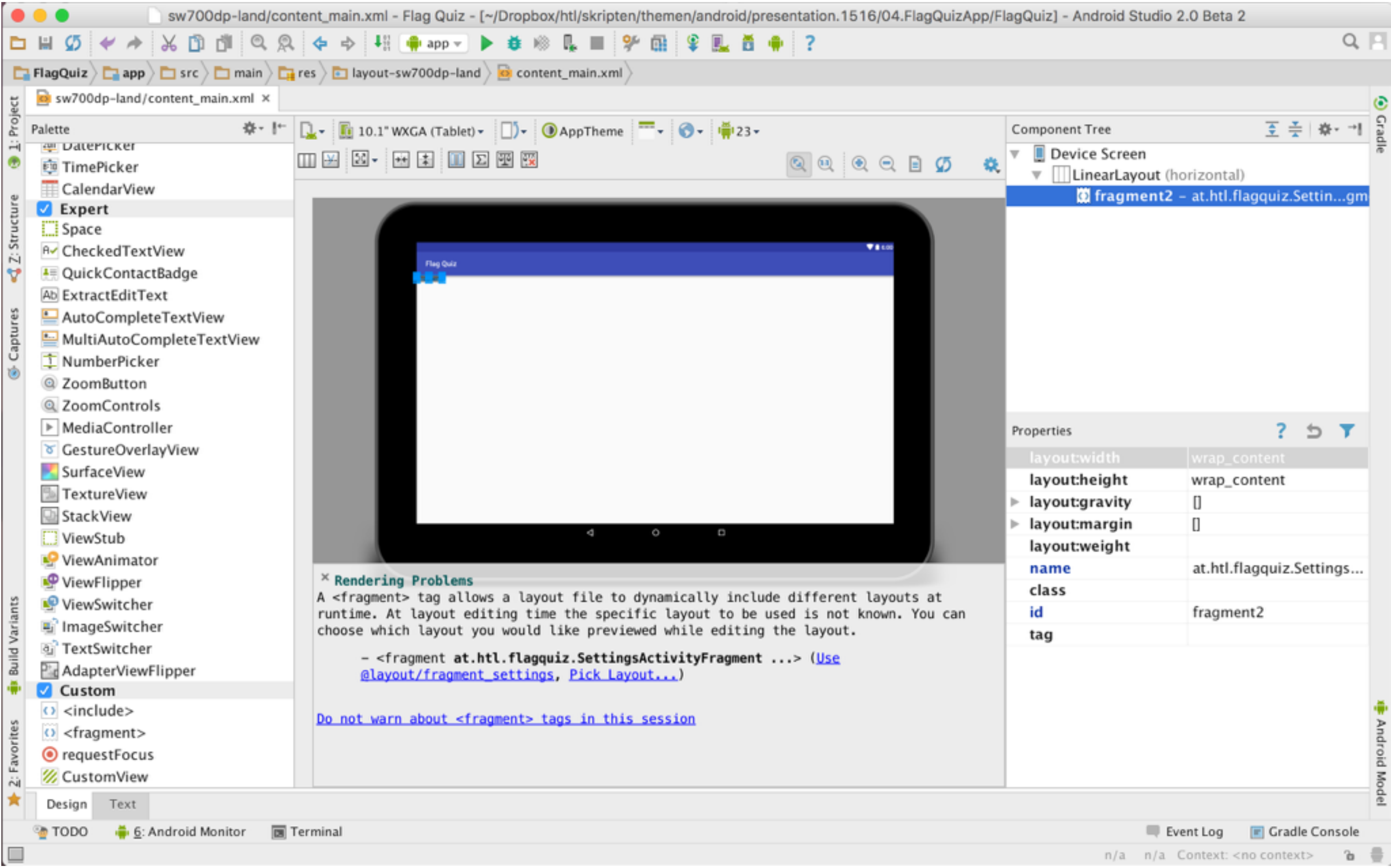
Properties

layout:width	match_parent
layout:height	match_parent
style	
orientation	horizontal
gravity	[]
accessibilityLiveRegior	
accessibilityTraversalA	
accessibilityTraversalB	
alpha	
background	
backgroundTint	
backgroundTintMode	
baselineAligned	☐
baselineAlignedChildIn	
clickable	☐

sw700dp-land/content_main.xml
Palette
DatePicker
TimePicker
CalendarView
Expert
Space
CheckedTextView
QuickContactBadge
ExtractEditText
AutoCompleteTextView
MultiAutoCompleteTextView
NumberPicker
ZoomButton
ZoomControls
MediaController
GestureOverlayView
SurfaceView
TextureView
StackView
ViewStub
ViewAnimator
ViewFlipper
ViewSwitcher
ImageSwitcher
TextSwitcher
AdapterViewFlipper
Custom
<include>
<fragment>
requestFocus
CustomView
app
10.1" WXGA (Tablet)
AppTheme
23
Flag Quiz
Design
Text
TODO
6: Android Monitor
Terminal
Event Log
Gradle Console
n/a
n/a
Context: <no context>
1: Project
Z: Structure
Captures
Build Variants
2: Favorites
Gradle
Android Model

Mit dem Cursor auf das LinearLayout klicken und so das Fragment hinzufügen


1
2
3
4
<fragment>
Fragments
WebViewFragment (android.webkit)
PreferenceFragment (android.preference)
ListFragment (android.app)
DialogFragment (android.app)
MainActivityFragment (at.htl.flagquiz)
ListFragment (android.support.v4.app)
SettingsActivityFragment (at.htl.flagquiz)
DialogFragment (android.support.v4.app)
AppCompatDialogFragment (android.support.v7.app)
Cancel
OK
Device Screen
LinearLayout (horizontal)



sw700dp-land/content_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
app
FlagQuiz
app
src
main
res
layout-sw700dp-land
content_main.xml
sw700dp-land/content_main.xml
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Palette
DatePicker
TimePicker
CalendarView
Expert
Space
CheckedTextView
QuickContactBadge
ExtractEditText
AutoCompleteTextView
MultiAutoCompleteTextView
NumberPicker
ZoomButton
ZoomControls
MediaController
GestureOverlayView
SurfaceView
TextureView
StackView
ViewStub
ViewAnimator
ViewFlipper
ViewSwitcher
ImageSwitcher
TextSwitcher
AdapterViewFlipper
Custom
<include>
<fragment>
requestFocus
CustomView
10.1" WXGA (Tablet)
AppTheme
23
Flag Quiz
Component Tree
Device Screen
LinearLayout (horizontal)
fragment2 – at.htl.flagquiz.Settin...gm
Properties
Design
Text
TODO
6: Android Monitor
Terminal
Event Log
Gradle Console
n/a
n/a
Context: <no context>
Gradle
Android Model

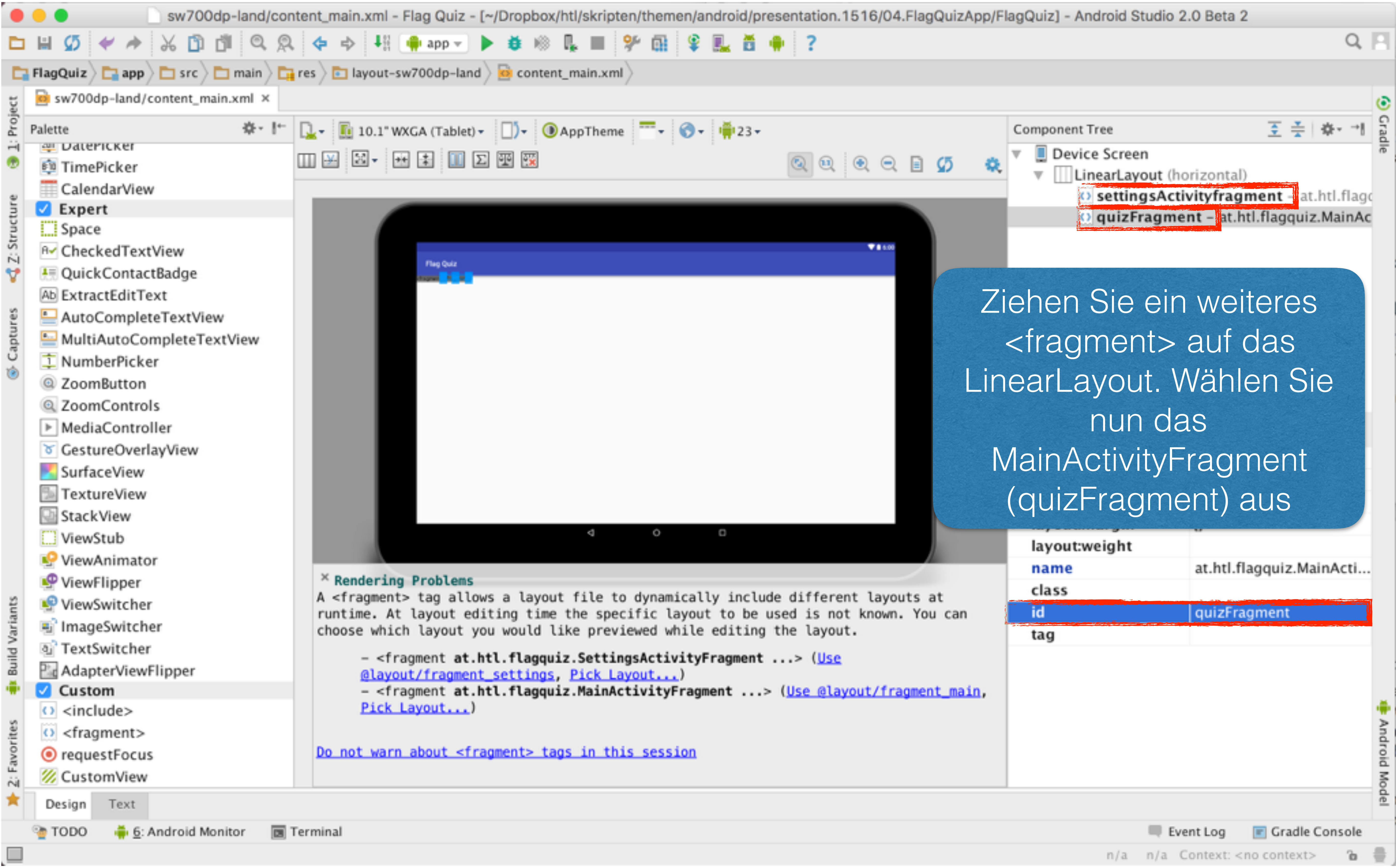
Rendering Problems

A <fragment> tag allows a layout file to dynamically include different layouts at runtime. At layout editing time the specific layout to be used is not known. You can choose which layout you would like previewed while editing the layout.

- <fragment **at.htl.flagquiz.SettingsActivityFragment** ...> (Use @layout/fragment_settings, Pick Layout...)

Do not warn about <fragment> tags in this session

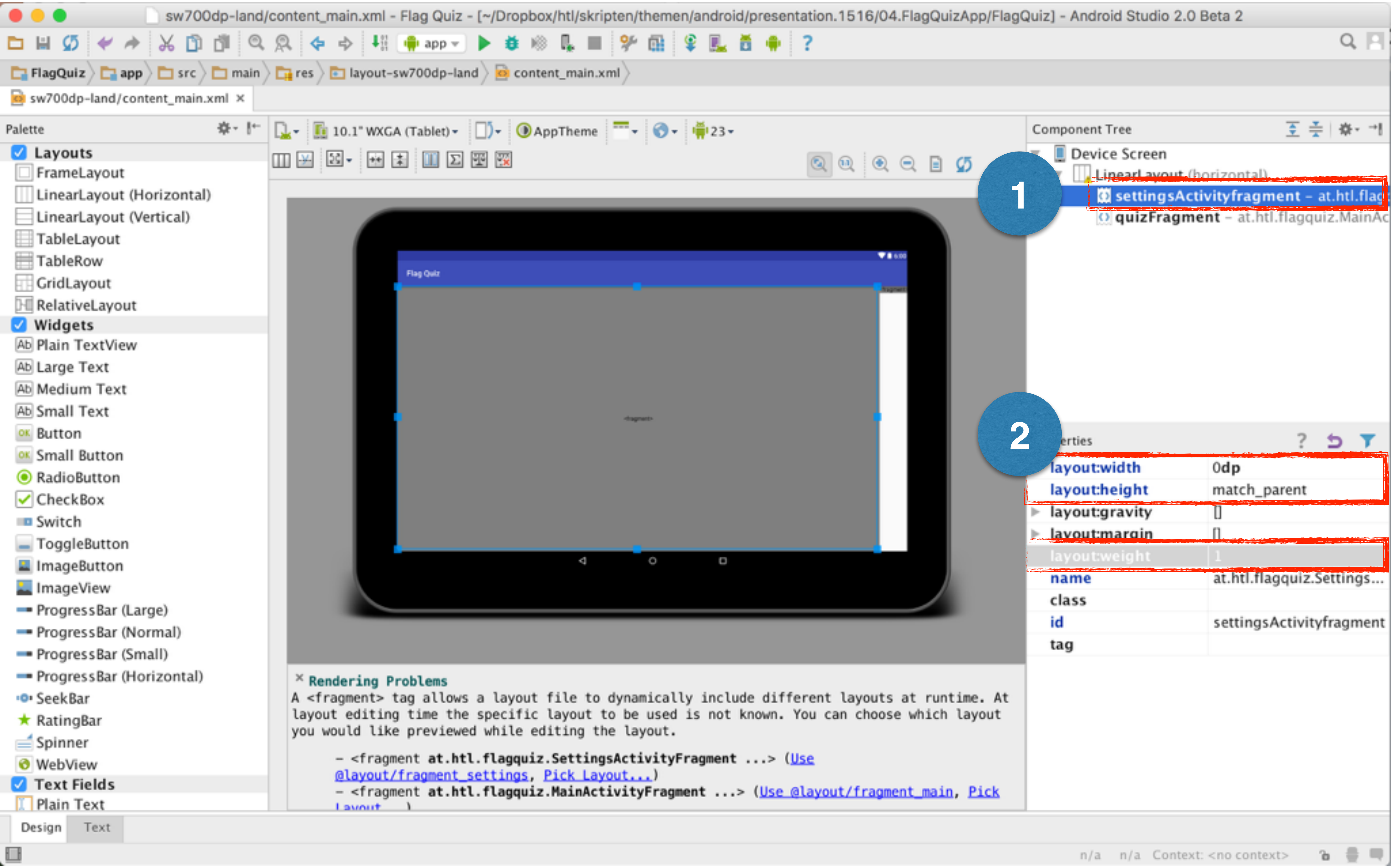
Properties	
layout:width	wrap_content
layout:height	wrap_content
layout:gravity	[]
layout:margin	[]
layout:weight	
name	at.htl.flagquiz.Settings...
class	
id	fragment2
tag	


sw700dp-land/content_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
FlagQuiz
app
src
main
res
layout-sw700dp-land
content_main.xml
sw700dp-land/content_main.xml
Palette
TimePicker
CalendarView
Expert
Space
CheckedTextView
QuickContactBadge
ExtractEditText
AutoCompleteTextView
MultiAutoCompleteTextView
NumberPicker
ZoomButton
ZoomControls
MediaController
GestureOverlayView
SurfaceView
TextureView
StackView
ViewStub
ViewAnimator
ViewFlipper
ViewSwitcher
ImageSwitcher
TextSwitcher
AdapterViewFlipper
Custom
<include>
<fragment>
requestFocus
CustomView
10.1" WXGA (Tablet)
AppTheme
23
Flag Quiz
Component Tree
Device Screen
LinearLayout (horizontal)
settingsActivityfragment - at.htl.flagq
quizFragment - at.htl.flagquiz.MainAc
Design
Text
TODO
6: Android Monitor
Terminal
Event Log
Gradle Console
Context: <no context>
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Gradle
Android Model

Ziehen Sie ein weiteres <fragment> auf das LinearLayout. Wählen Sie nun das MainActivityFragment (quizFragment) aus

```
× Rendering Problems
A <fragment> tag allows a layout file to dynamically include different layouts at
runtime. At layout editing time the specific layout to be used is not known. You can
choose which layout you would like previewed while editing the layout.

    - <fragment at.htl.flagquiz.SettingsActivityFragment ...> (Use
    @layout/fragment_settings, Pick Layout...)
    - <fragment at.htl.flagquiz.MainActivityFragment ...> (Use @layout/fragment_main,
    Pick Layout...)

Do not warn about <fragment> tags in this session
```

layout:weight	
name	at.htl.flagquiz.MainActi...
class	
id	quizFragment
tag	


1
2
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
Design
Text
app
10.1" WXGA (Tablet)
AppTheme
23
Flag Quiz
<fragment>
n/a
n/a
Context: <no context>

sw700dp-land/content_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2

FlagQuiz > app > src > main > res > layout-sw700dp-land > content_main.xml

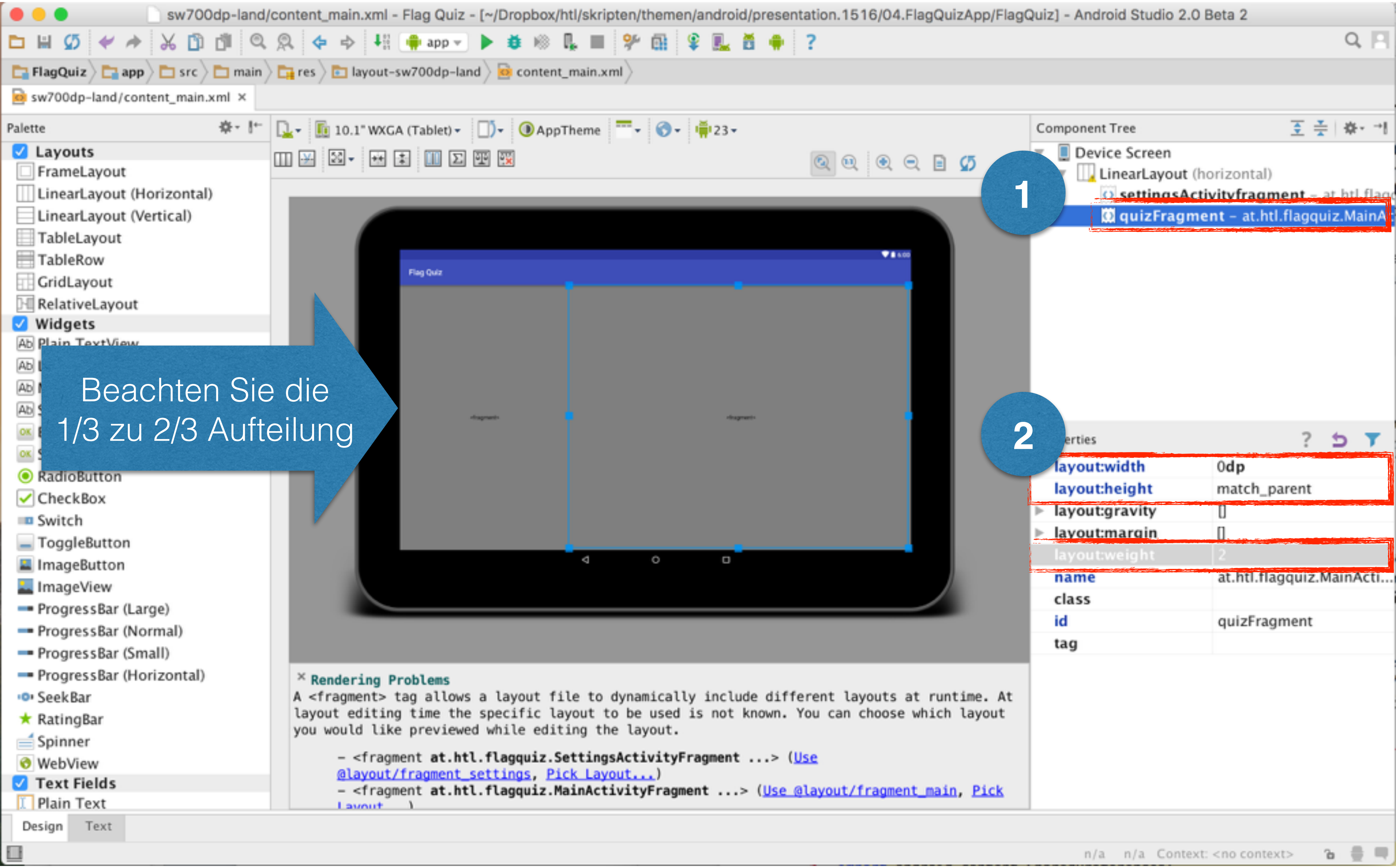
sw700dp-land/content_main.xml

Component Tree
- Device Screen
  - LinearLayout (horizontal)
    - settingsActivityfragment – at.htl.flag
    - quizFragment – at.htl.flagquiz.MainAc

erties

layout:width	0dp
layout:height	match_parent
layout:gravity	[]
layout:margin	[]
layout:weight	1
name	at.htl.flagquiz.Settings...
class	
id	settingsActivityfragment
tag	

```
Rendering Problems
A <fragment> tag allows a layout file to dynamically include different layouts at runtime. At
layout editing time the specific layout to be used is not known. You can choose which layout
you would like previewed while editing the layout.

    - <fragment at.htl.flagquiz.SettingsActivityFragment ...> (Use
    @layout/fragment_settings, Pick Layout...)
    - <fragment at.htl.flagquiz.MainActivityFragment ...> (Use @layout/fragment_main, Pick
    Layout...)
```


sw700dp-land/content_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
app
FlagQuiz
app
src
main
res
layout-sw700dp-land
content_main.xml
sw700dp-land/content_main.xml
Palette
Layouts
FrameLayout
LinearLayout (Horizontal)
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizontal)
SeekBar
RatingBar
Spinner
WebView
Text Fields
Plain Text
10.1" WXGA (Tablet)
AppTheme
23
Flag Quiz
Component Tree
Device Screen
LinearLayout (horizontal)
settingsActivityfragment – at.htl.flag
quizFragment – at.htl.flagquiz.MainA
1
2
Properties
Design
Text
n/a
n/a
Context: <no context>

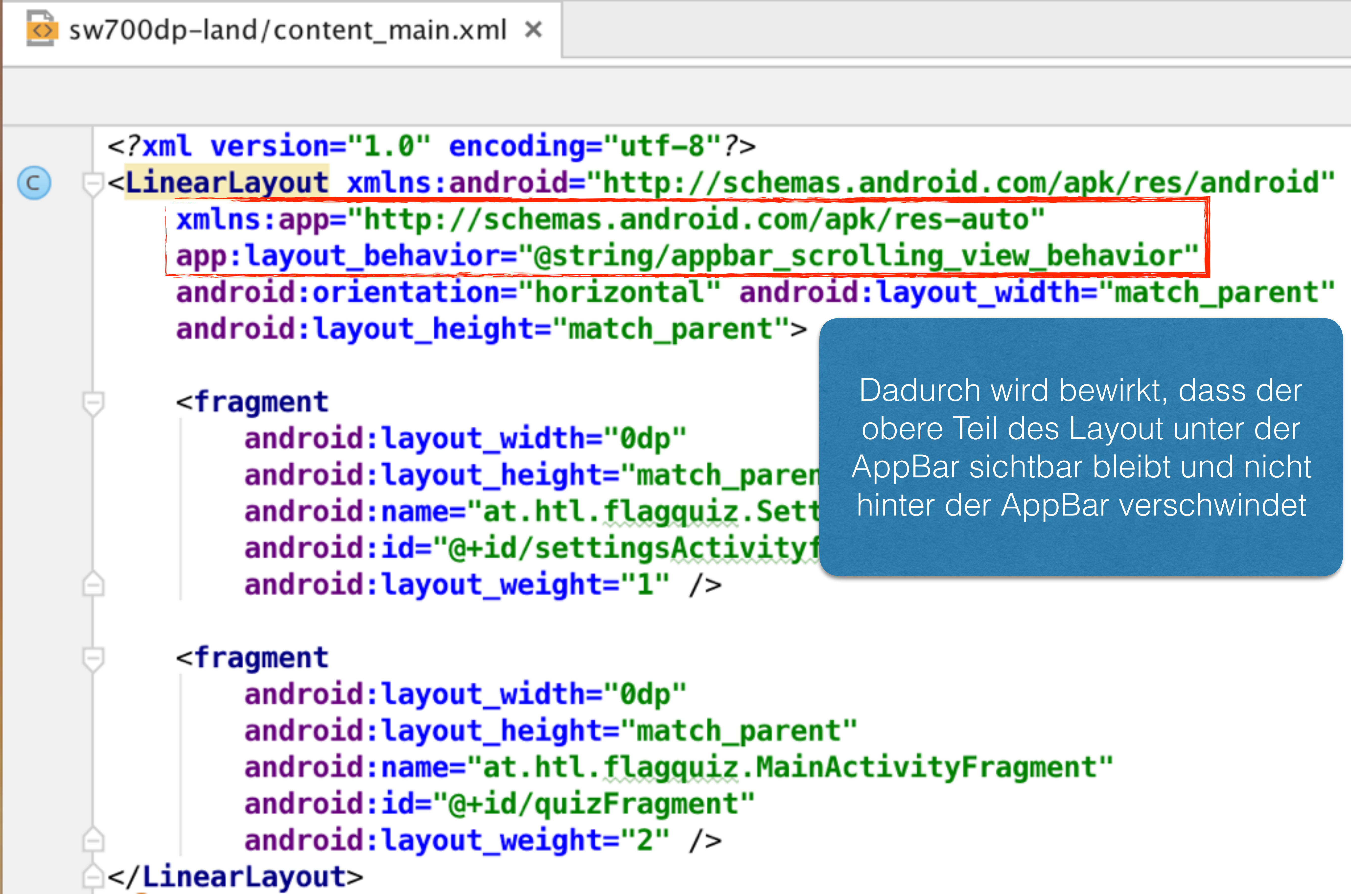
Beachten Sie die 1/3 zu 2/3 Aufteilung

layout:width	0dp
layout:height	match_parent
layout:gravity	[]
layout:margin	[]
layout:weight	2
name	at.htl.flagquiz.MainActi...
class	
id	quizFragment
tag	

Rendering Problems

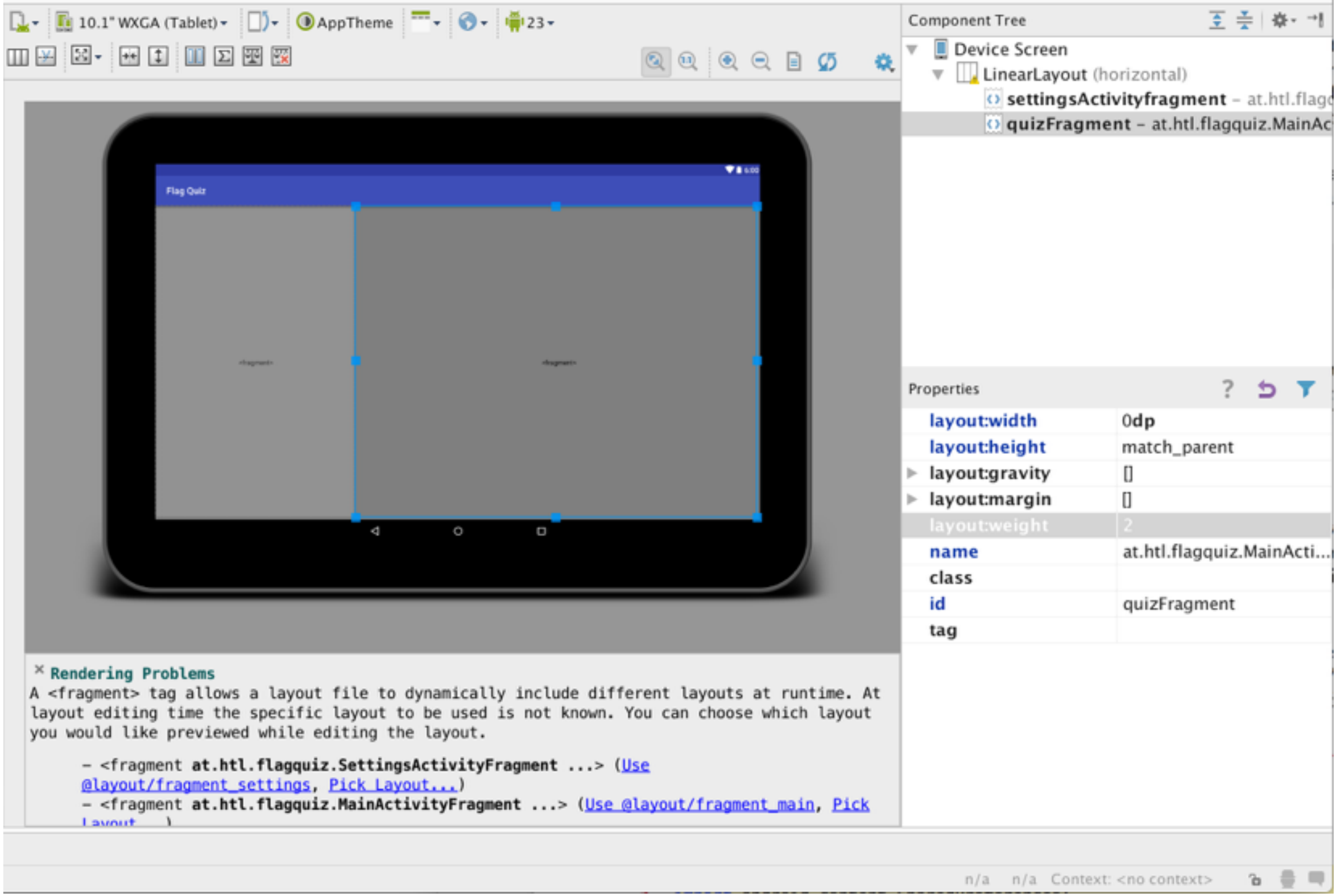
A <fragment> tag allows a layout file to dynamically include different layouts at runtime. At layout editing time the specific layout to be used is not known. You can choose which layout you would like previewed while editing the layout.

- <fragment **at.htl.flagquiz.SettingsActivityFragment** ...> (Use @layout/fragment_settings, Pick Layout...)
- <fragment **at.htl.flagquiz.MainActivityFragment** ...> (Use @layout/fragment_main, Pick Layout...)


sw700dp-land/content_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <fragment
        android:layout_width="0dp"
        android:layout_height="match_paren
        android:name="at.htl.flagquiz.Sett
        android:id="@+id/settingsActivityf
        android:layout_weight="1" />

    <fragment
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:name="at.htl.flagquiz.MainActivityFragment"
        android:id="@+id/quizFragment"
        android:layout_weight="2" />
</LinearLayout>
```

Dadurch wird bewirkt, dass der obere Teil des Layout unter der AppBar sichtbar bleibt und nicht hinter der AppBar verschwindet



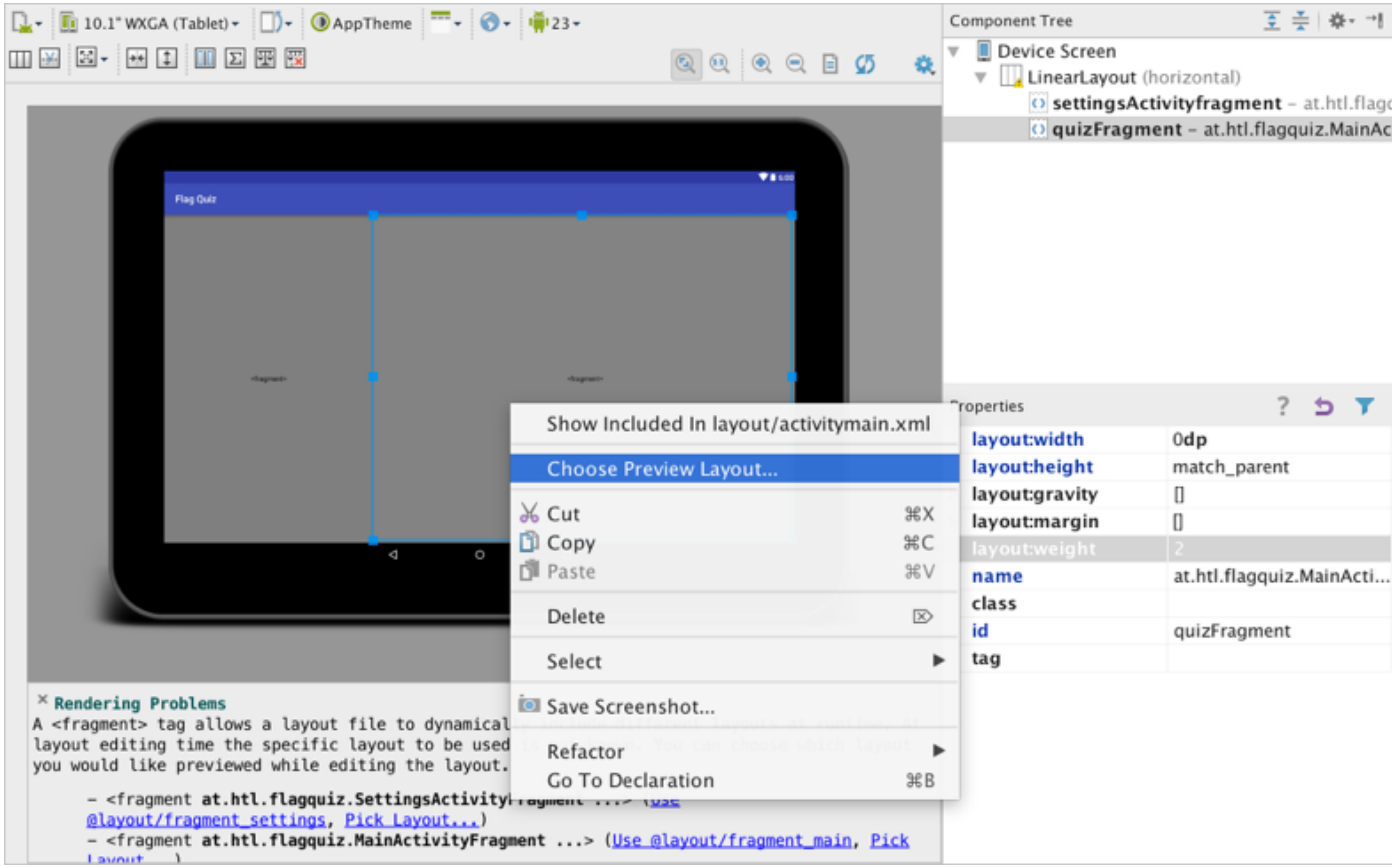
Device Screen
- LinearLayout (horizontal)
  - settingsActivityfragment – at.htl.flag
  - quizFragment – at.htl.flagquiz.MainAc

layout:width	0dp
layout:height	match_parent
layout:gravity	[]
layout:margin	[]
layout:weight	2
name	at.htl.flagquiz.MainActi...
class	
id	quizFragment
tag	

```
Rendering Problems
A <fragment> tag allows a layout file to dynamically include different layouts at runtime. At
layout editing time the specific layout to be used is not known. You can choose which layout
you would like previewed while editing the layout.
    - <fragment at.htl.flagquiz.SettingsActivityFragment ...> (Use
    @layout/fragment_settings, Pick Layout...)
    - <fragment at.htl.flagquiz.MainActivityFragment ...> (Use @layout/fragment_main, Pick
    Layout...)
```

- Die Design View des Layout Editors kann jegliches Fragment als PreView anzeigen. Wird hierfür kein Fragment festgelegt, erscheinen Rendering Problems

- Rechtsklicken Sie daher auf das gewünschte Fragment und wählen Sie „Choose Preview Layout …“ (entweder in der Design View oder im Component Tree)

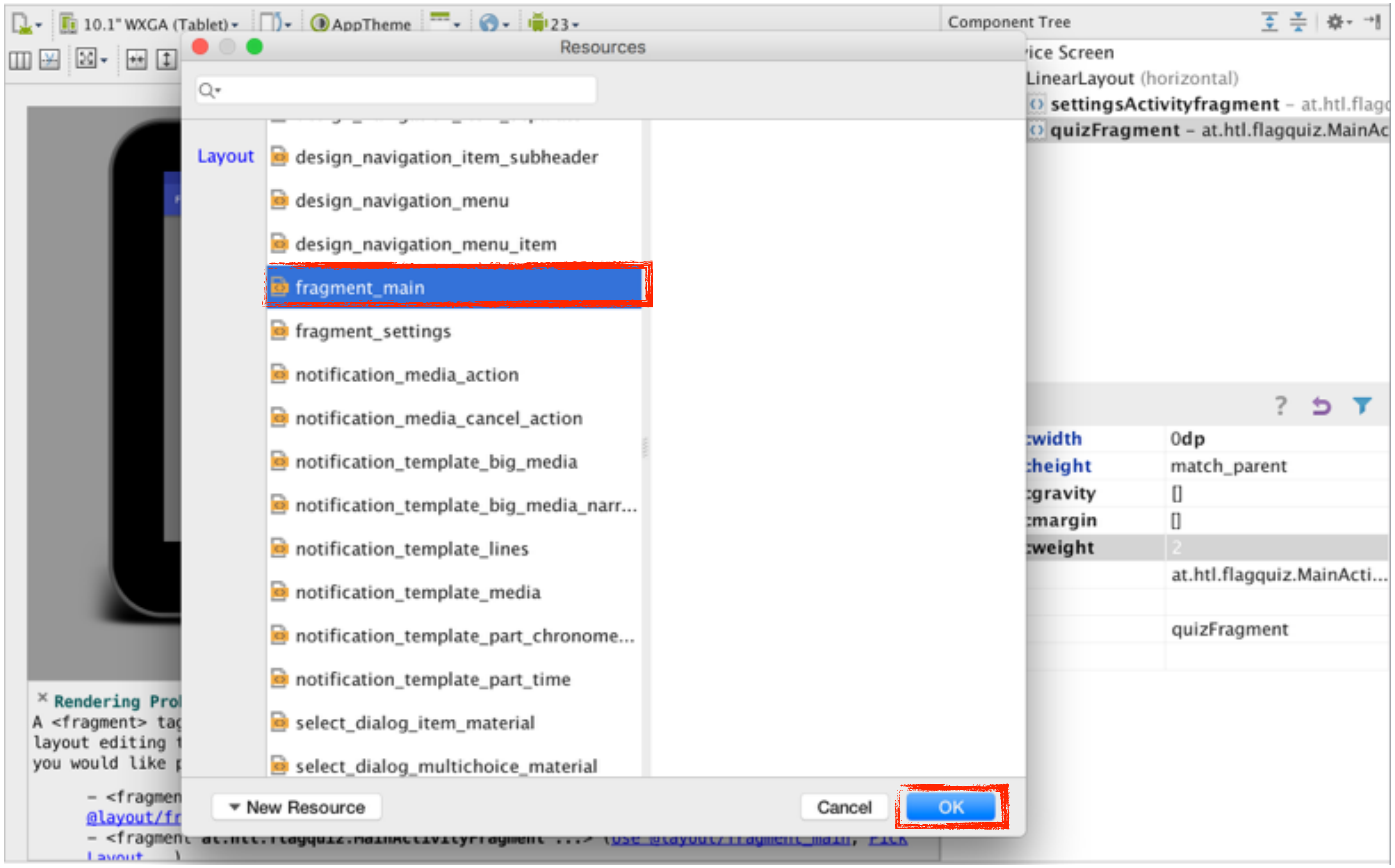


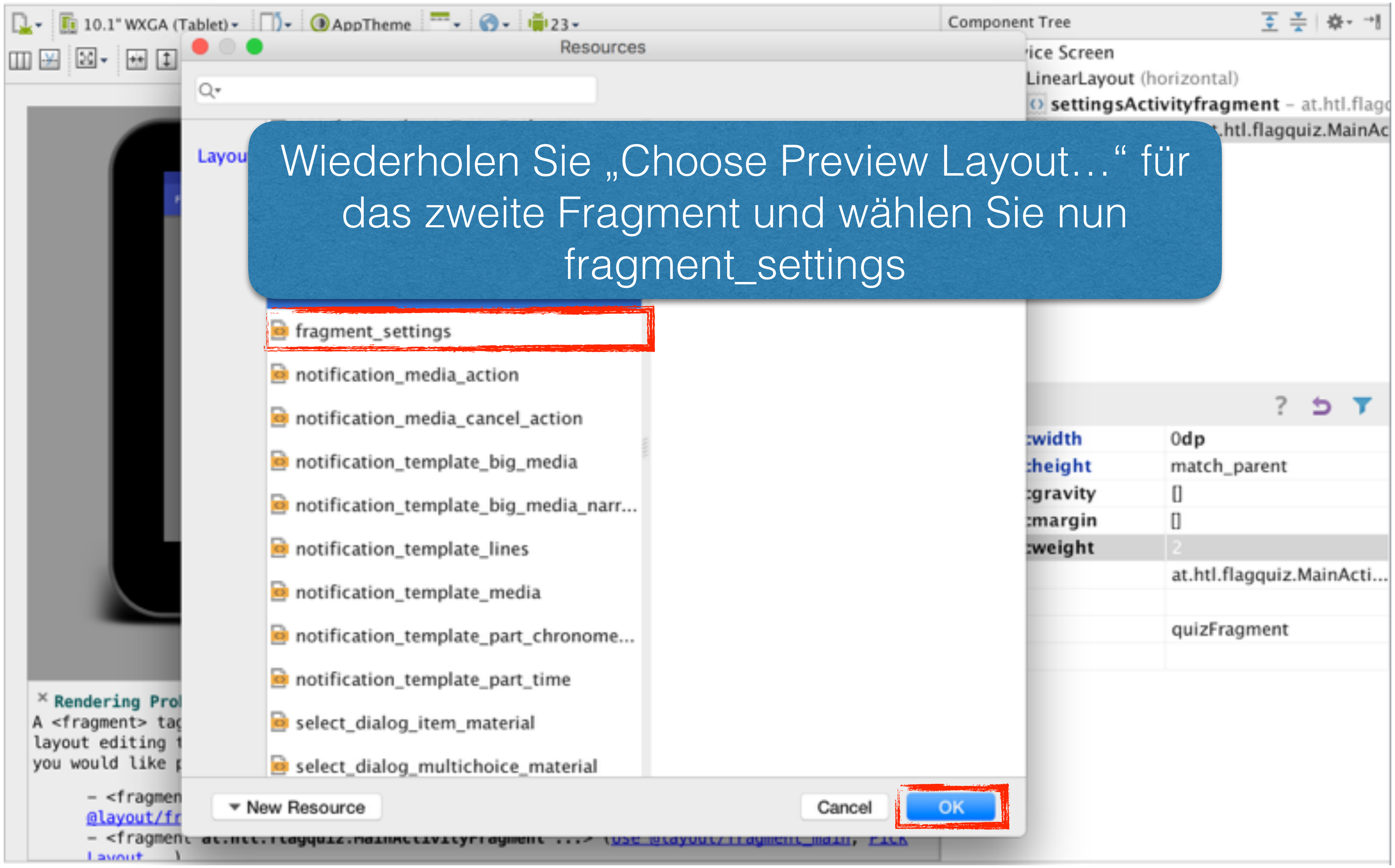
10.1" WXGA (Tablet)
AppTheme
23
Flag Quiz
<fragment>
<fragment>
Show Included In layout/activitymain.xml
Choose Preview Layout...
Cut ⌘X
Copy ⌘C
Paste ⌘V
Delete
Select
Save Screenshot...
Refactor
Go To Declaration ⌘B
Component Tree
Properties

Device Screen
LinearLayout (horizontal)
settingsActivityfragment – at.htl.flagq
quizFragment – at.htl.flagquiz.MainAc

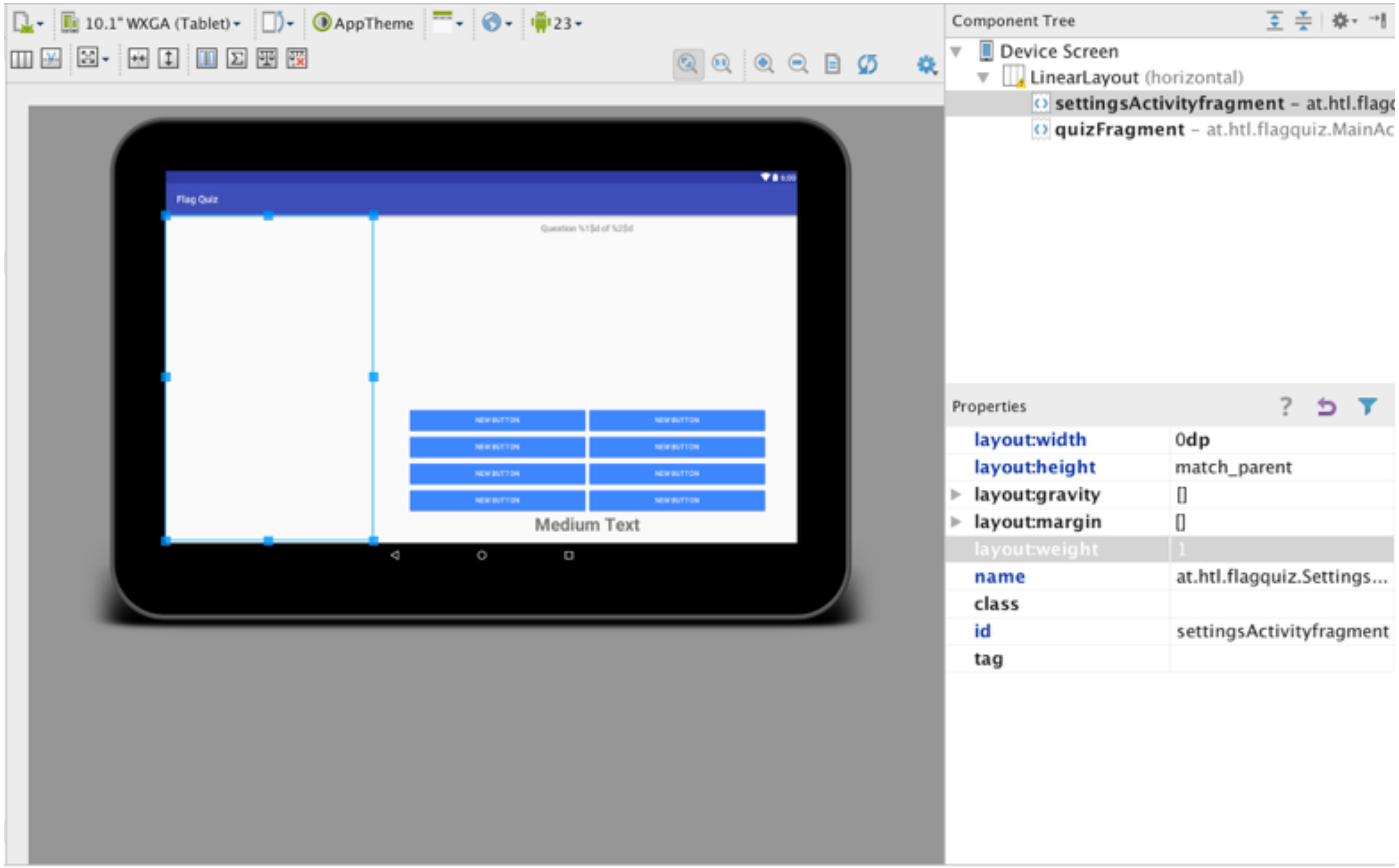
layout:width	0dp
layout:height	match_parent
layout:gravity	[]
layout:margin	[]
layout:weight	2
name	at.htl.flagquiz.MainActi...
class	
id	quizFragment
tag	

```
Rendering Problems
A <fragment> tag allows a layout file to dynamical
layout editing time the specific layout to be used
you would like previewed while editing the layout.

     - <fragment at.htl.flagquiz.SettingsActivityFragment ...> (Use
     @layout/fragment_settings, Pick Layout...)
     - <fragment at.htl.flagquiz.MainActivityFragment ...> (Use @layout/fragment_main, Pick
     Layout...)
```


10.1" WXGA (Tablet)
AppTheme
23
Component Tree
Resources
Layout
design_navigation_item_subheader
design_navigation_menu
design_navigation_menu_item
fragment_main
fragment_settings
notification_media_action
notification_media_cancel_action
notification_template_big_media
notification_template_big_media_narr...
notification_template_lines
notification_template_media
notification_template_part_chronome...
notification_template_part_time
select_dialog_item_material
select_dialog_multichoice_material
New Resource
Cancel
OK
LinearLayout (horizontal)
settingsActivityfragment – at.htl.flagq
quizFragment – at.htl.flagquiz.MainAc
width
height
gravity
margin
weight
0dp
match_parent
[]
[]
at.htl.flagquiz.MainActi...
quizFragment
Rendering Pro
A <fragment> tag
layout editing t
you would like p
- <fragmen
@layout/fr
- <fragment at.htl.flagquiz.MainActivityFragment ...> (Use @layout/fragment_main, Pick
Layout...)


10.1" WXGA (Tablet)
AppTheme
23
Resources
Layou
Component Tree
ice Screen
LinearLayout (horizontal)
settingsActivityfragment – at.htl.flagq
.htl.flagquiz.MainAc
fragment_settings
notification_media_action
notification_media_cancel_action
notification_template_big_media
notification_template_big_media_narr...
notification_template_lines
notification_template_media
notification_template_part_chronome...
notification_template_part_time
select_dialog_item_material
select_dialog_multichoice_material
New Resource
Cancel
OK
:width
0dp
:height
match_parent
:gravity
[]
:margin
[]
:weight
2
at.htl.flagquiz.MainActi...
quizFragment
Rendering Pro
A <fragment> tag
layout editing t
you would like p
– <fragmen
@layout/fr
– <fragment at.htl.flagquiz.MainActivityFragment ...> (Use @layout/fragment_main, Pick
Layout...)

Wiederholen Sie „Choose Preview Layout…“ für das zweite Fragment und wählen Sie nun fragment_settings


10.1" WXGA (Tablet)
AppTheme
23
Flag Quiz
Question %1$d of %2$d
NEW BUTTON
Medium Text

Component Tree

- Device Screen
  - LinearLayout (horizontal)
    - settingsActivityfragment – at.htl.flagq
    - quizFragment – at.htl.flagquiz.MainAc

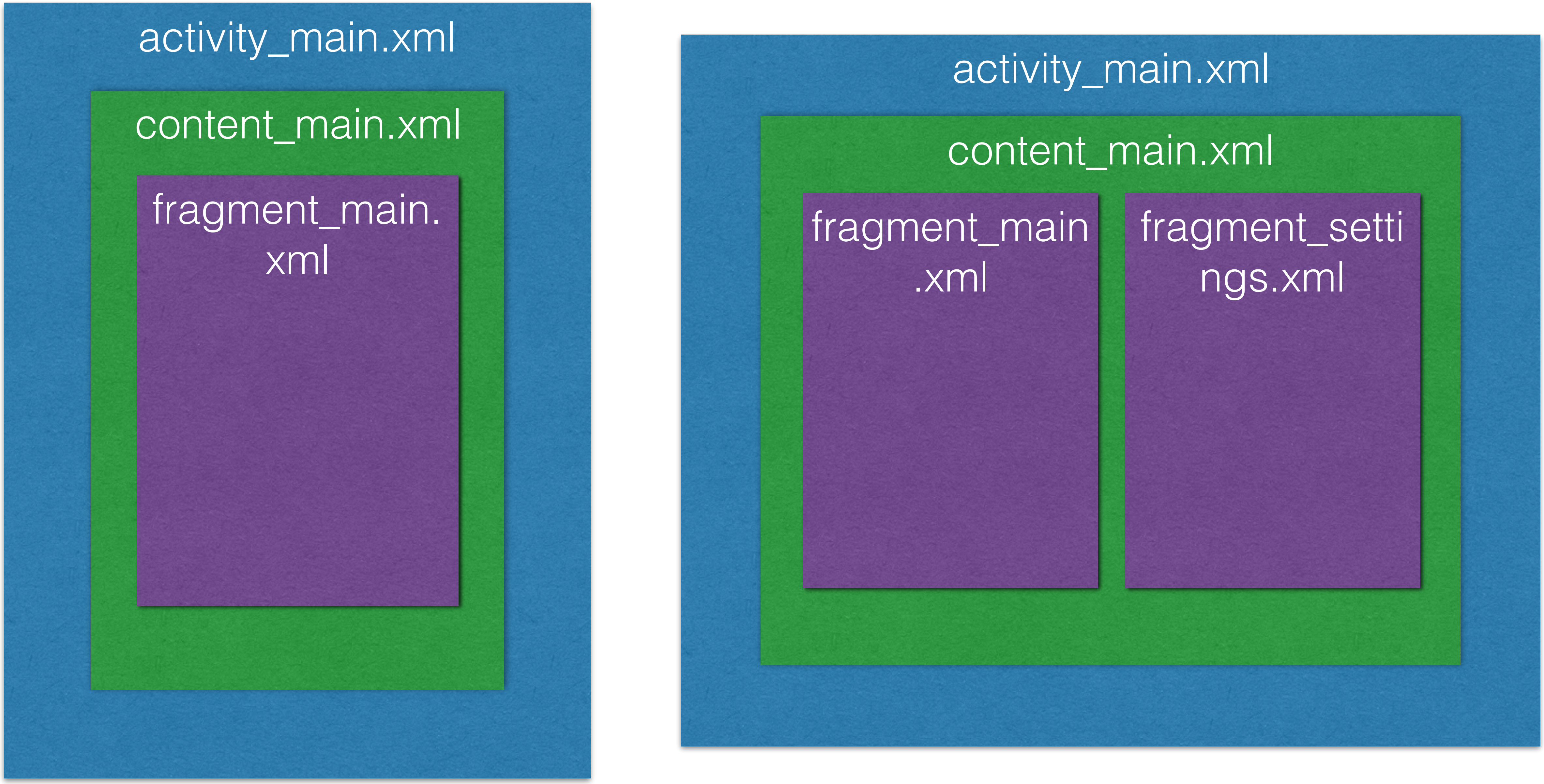
Properties

layout:width	0dp
layout:height	match_parent
layout:gravity	[]
layout:margin	[]
layout:weight	1
name	at.htl.flagquiz.Settings...
class	
id	settingsActivityfragment
tag	

Endlich coden!!!

MainActivity.java

MainActivity.java



activity_main.xml

content_main.xml

fragment_main.xml

activity_main.xml

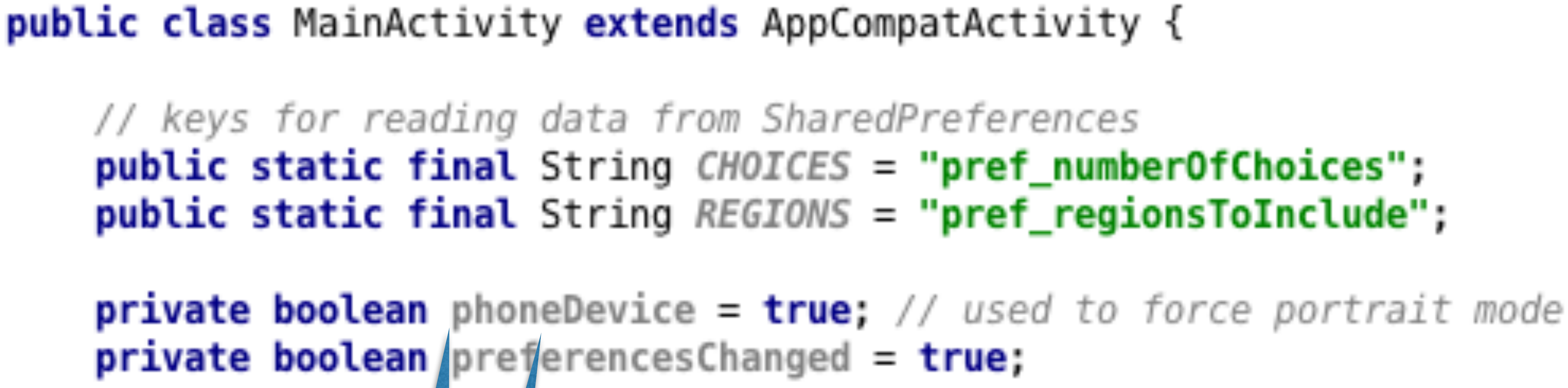
content_main.xml

fragment_main.xml

fragment_settings.xml

Im Portrait-Mode beinhaltet die MainActivity das MainActivityFragment. Im Landscape-Mode zusätzlich das SettingsActivityFragment

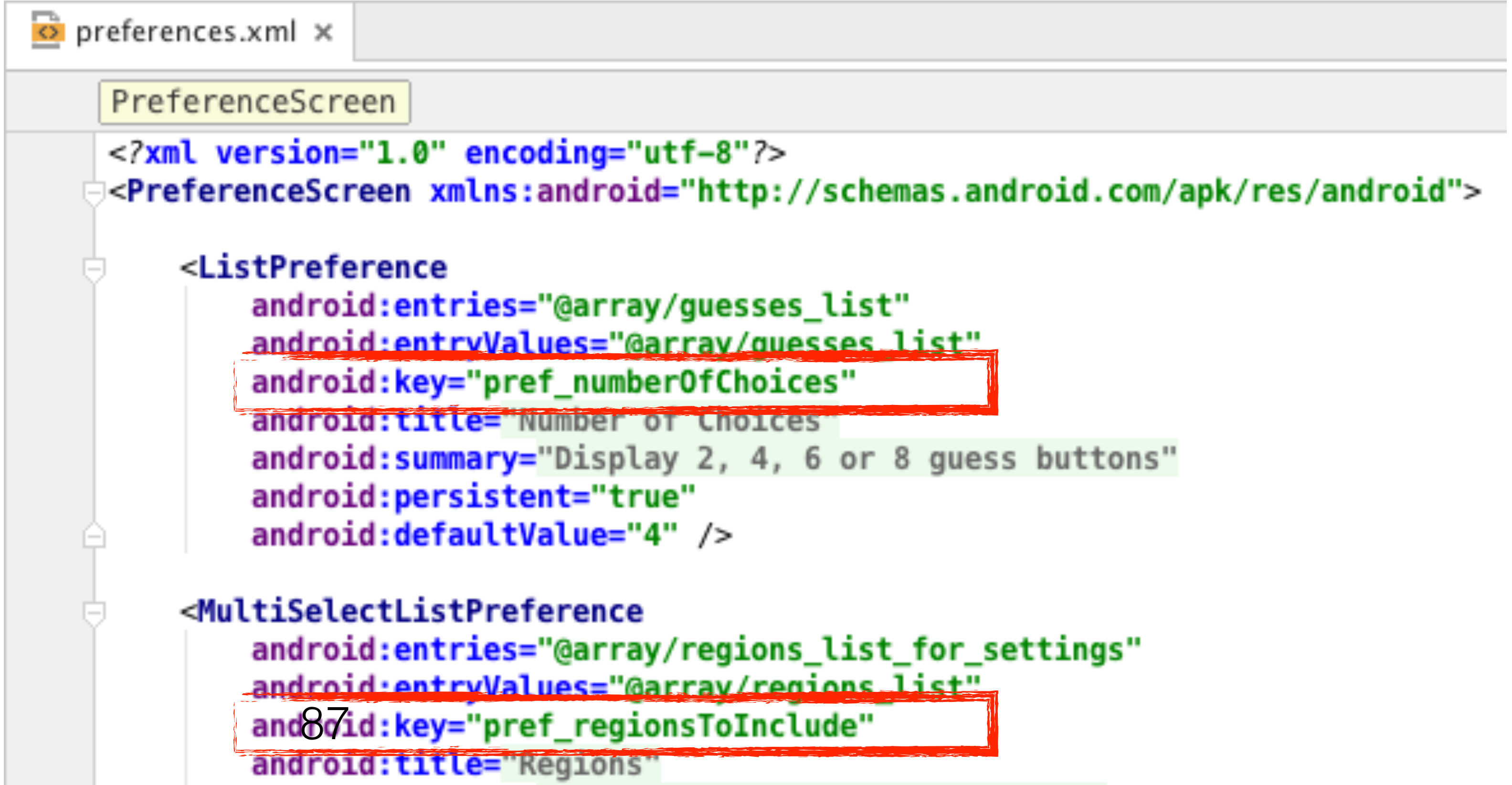
MainActivity.java



```java
public class MainActivity extends AppCompatActivity {

    // keys for reading data from SharedPreferences
    public static final String CHOICES = "pref_numberOfChoices";
    public static final String REGIONS = "pref_regionsToInclude";

    private boolean phoneDevice = true; // used to force portrait mode
    private boolean preferencesChanged = true;
```

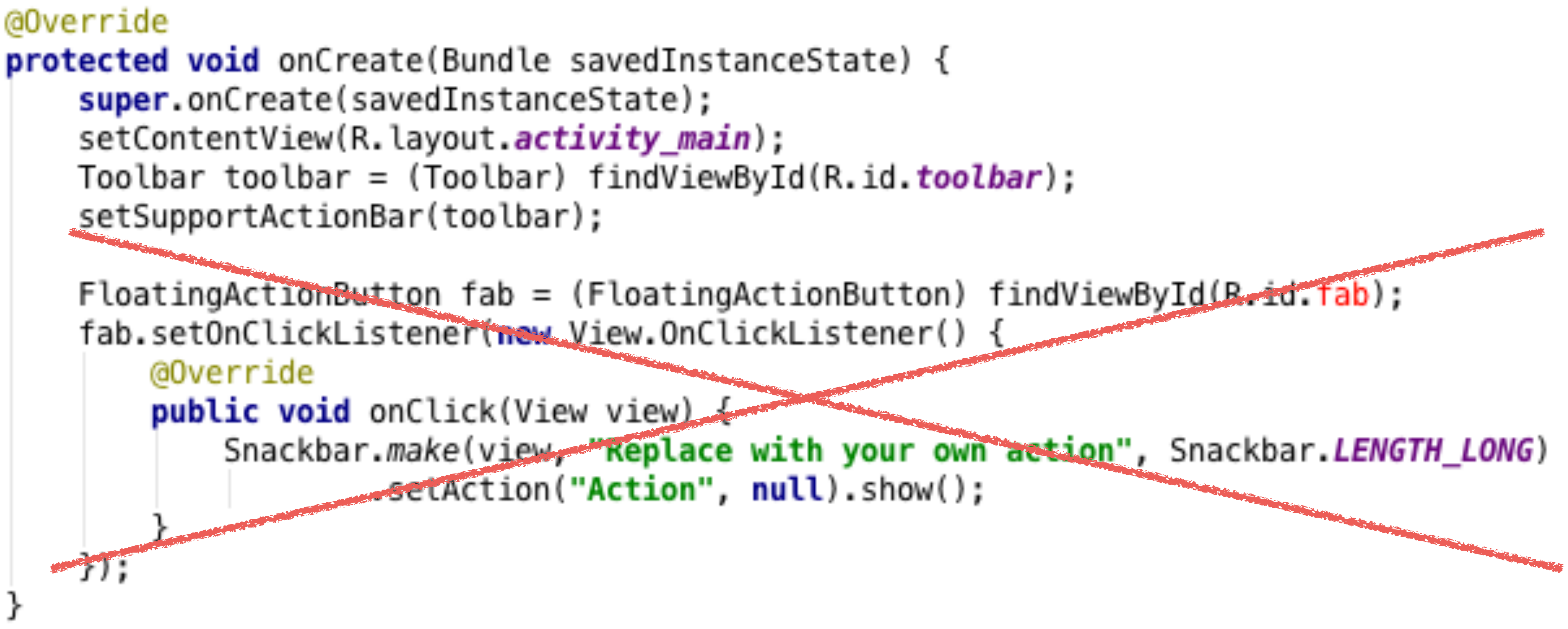
Auf Phones darf die App nur im Portrait Modus laufen!

Werden die Preferences geändert ruft onStart() die updateGuessRows()- und updateRegions()-Methoden des Fragments auf



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

    <ListPreference
        android:entries="@array/guesses_list"
        android:entryValues="@array/guesses_list"
        android:key="pref_numberOfChoices"
        android:title="Number of Choices"
        android:summary="Display 2, 4, 6 or 8 guess buttons"
        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="Regions"
```

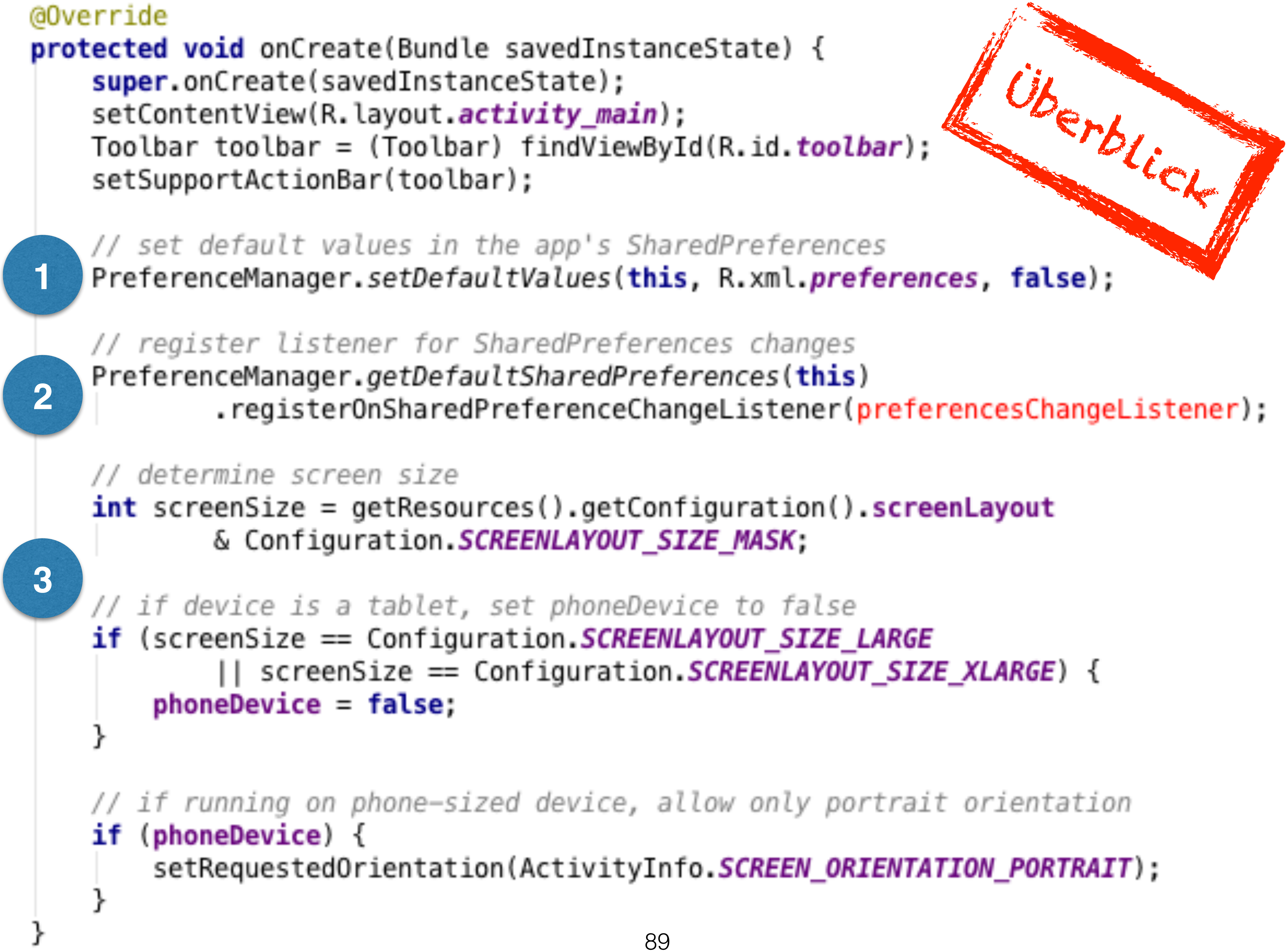
onCreate()



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
        }
    });
}
```

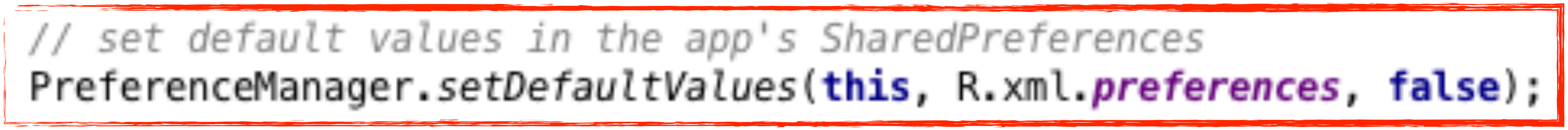
Der EventHandler für den (bereits gelöschten) FloatingActionButton wird nun entfernt

Überblick



```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    // set default values in the app's SharedPreferences
    PreferenceManager.setDefaultValues(this, R.xml.preferences, false);

    // register listener for SharedPreferences changes
    PreferenceManager.getDefaultSharedPreferences(this)
            .registerOnSharedPreferenceChangeListener(preferencesChangeListener);

    // determine screen size
    int screenSize = getResources().getConfiguration().screenLayout
            & Configuration.SCREENLAYOUT_SIZE_MASK;

    // if device is a tablet, set phoneDevice to false
    if (screenSize == Configuration.SCREENLAYOUT_SIZE_LARGE
            || screenSize == Configuration.SCREENLAYOUT_SIZE_XLARGE) {
        phoneDevice = false;
    }

    // if running on phone-sized device, allow only portrait orientation
    if (phoneDevice) {
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    }
}
```

1 defaultPreferences



```java
// set default values in the app's SharedPreferences
PreferenceManager.setDefaultValues(this, R.xml.preferences, false);
```

- Shared Preferences sind im Grunde eine Datei mit Key- und Value-Werten, die als einfache Möglichkeit zum dauerhaften speichern (persistieren) von Einstellungen zur Verfügung steht. Zum Zugriff dient das Framework PreferenceManager.

- Nach dem Installieren und dem allerersten Start der App, wird durch den Aufruf von PreferenceManager.setDefaultValues(…) die SharedPreferences-Datei der Applikation erstellt und mit den Werten aus der von uns erstellten preferences.xml initialisiert.

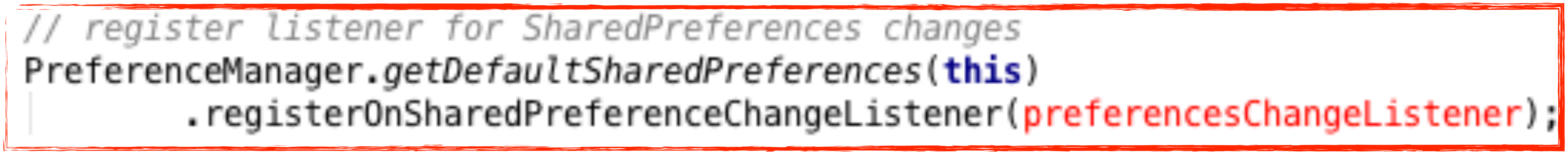
- Parameter der Methode:

    - this … android.content.Context, der Zugriff auf die Informationen der Umgebung gewährt, in der die App läuft. Dadurch erhält man auch Zugriff auf verschiedene Dienste (Services). Hier ist der Context die Activity (this), für die die SharedPreferences gesetzt werden

    - R.xml.preferences … Die Ressourcen-Id des preferences.xml-Files

    - false … Ein bool'scher Wert, der angibt, ob die Standardwerte (default values) bei jedem Aufruf von setDefaultValues(…) zurückgesetzt (reset) werden sollen. „false" bedeutet, dass die Standardwerte NUR beim ERSTEN AUFRUF gesetzt werden (!)

② preferencesChangeListener



```
// register listener for SharedPreferences changes
PreferenceManager.getDefaultSharedPreferences(this)
        .registerOnSharedPreferenceChangeListener(preferencesChangeListener);
```

- Jedes Mal, wenn der Benutzer die preferences ändert, muss die MainActivity die Methoden von MainActivityFragment - updateGuessRows und updateRegions - aufrufen, um das Quiz neu zu konfigurieren

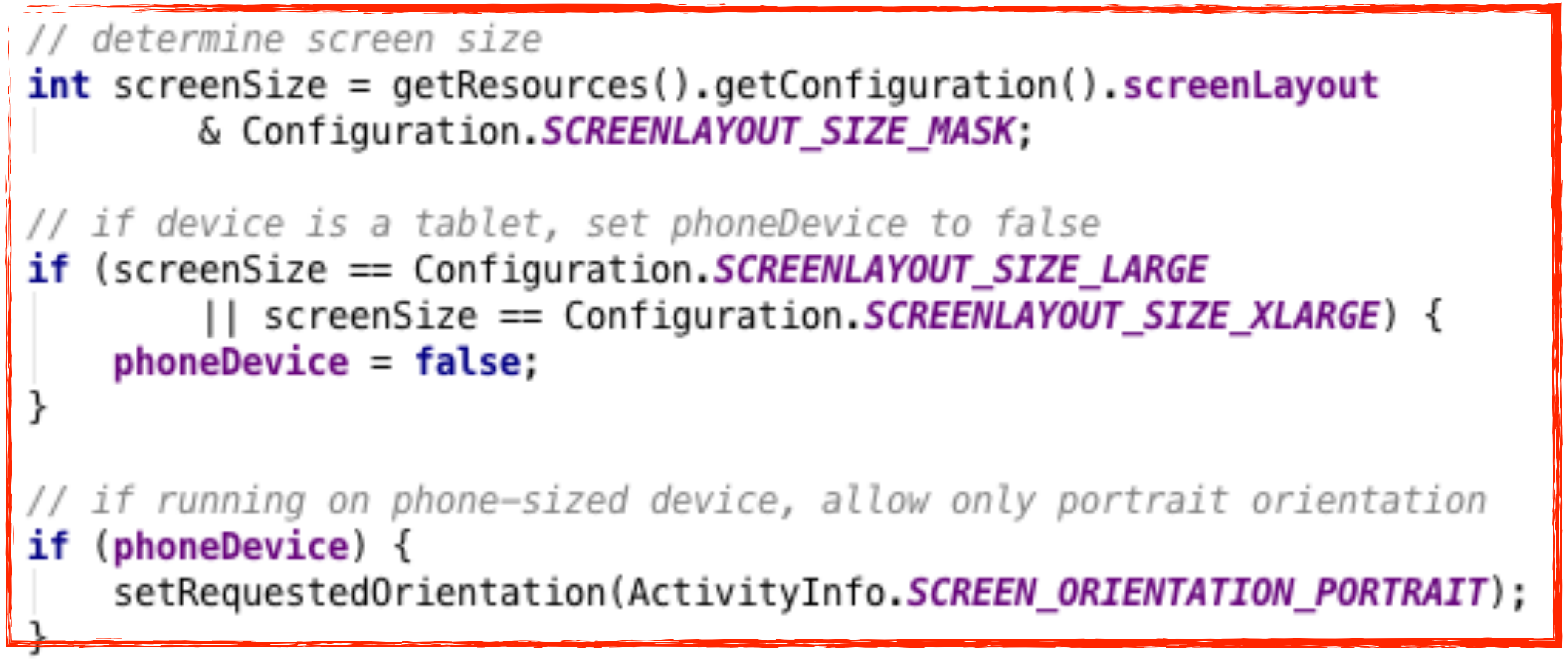
- Dies wird durch Definition eines OnSharedPreferenceChangedListener’s ermöglicht.

- Die Methode getDefaultSharedPreferences() gibt eine Referenz auf das **SharedPreference-Objekt** zurück

- Der Listener wird später erstellt (daher im Editor rot dargestellt)

Bildschirmbreite bestimmen

3



```
// determine screen size
int screenSize = getResources().getConfiguration().screenLayout
        & Configuration.SCREENLAYOUT_SIZE_MASK;

// if device is a tablet, set phoneDevice to false
if (screenSize == Configuration.SCREENLAYOUT_SIZE_LARGE
        || screenSize == Configuration.SCREENLAYOUT_SIZE_XLARGE) {
    phoneDevice = false;
}

// if running on phone-sized device, allow only portrait orientation
if (phoneDevice) {
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
}
```

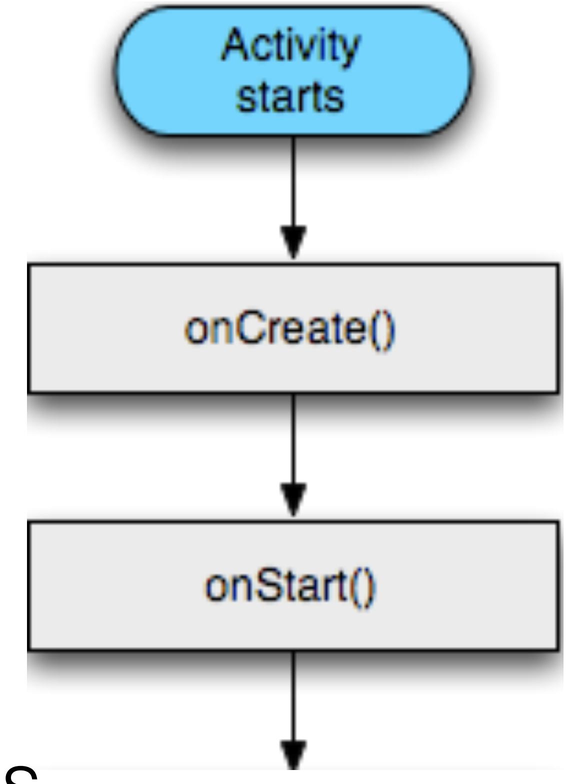
- getResources() gibt die Referenz auf das **Ressourcen-Objekt** (android.content.res.Resource) zurück.

- Damit kann man die Ressourcen der App zugreifen und auch Informationen über die Umgebung abrufen.

- getConfiguration() gibt eine Referenz auf das **Configuration-Objekt** (android.content.res.Configuration) zurück. Hier ist eine öffentliche Variable „screenLayout“ enthalten, welche die screen-size Category bestimmt

- Zuerst wird mit einem Bitoperator & (AND) die Bildschirmgröße ermittelt. Anschließend wird ermittelt, ob die Bildschirmgröße LARGE oder XLARGE ist. Falls ja, wird die Portrait-Orientierung (Hochformat) erzwungen

onStart()



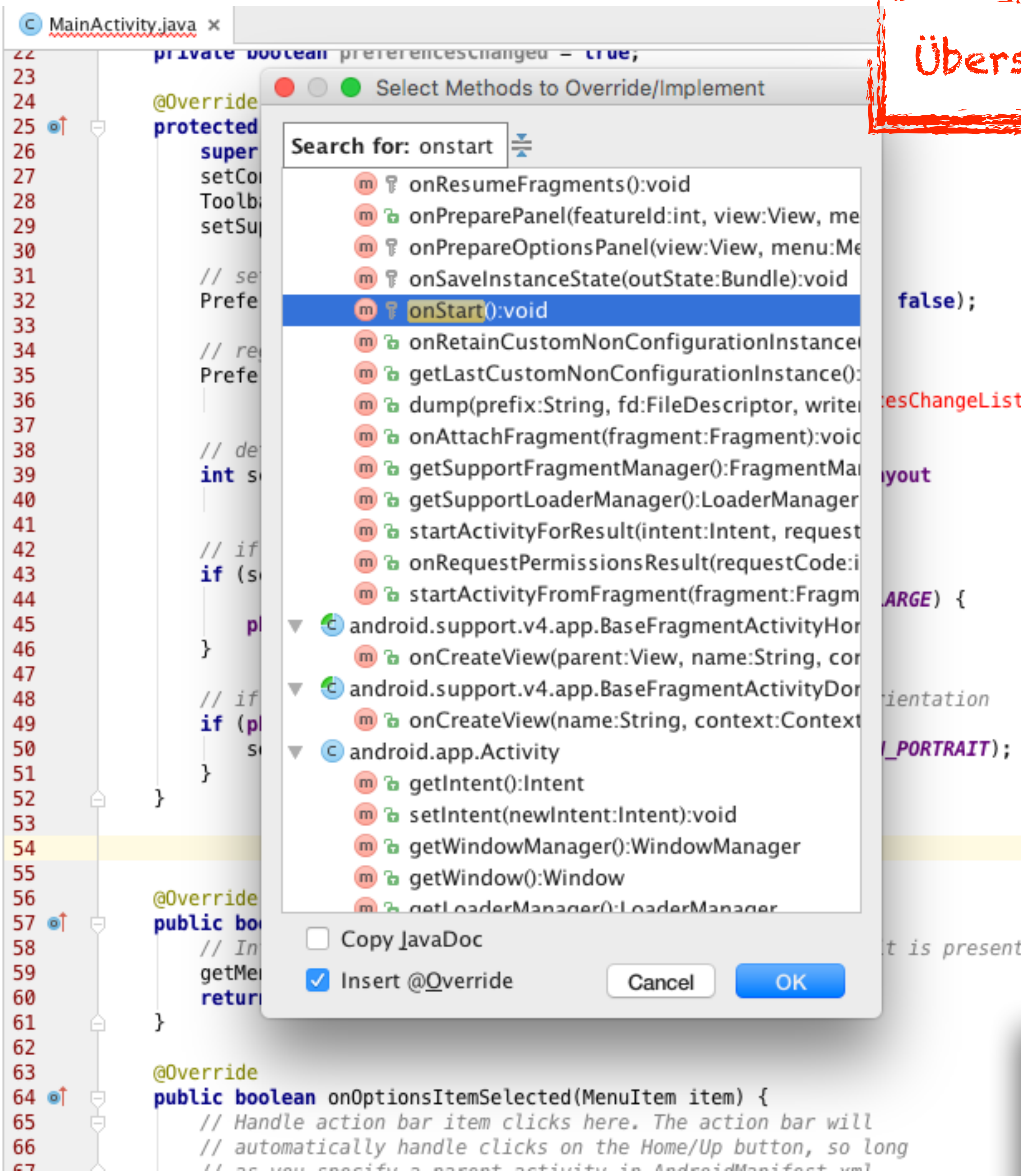
- onStart() wird nach onCreate() ausgeführt

- In unserer App wird onStart() verwendet, um sicherzustellen, dass sie den SharedPreferences entsprechend, korrekt konfiguriert ist

- Wenn die App im Portrait-Modus läuft und die SettingsActivity wird durch den Benutzer aufgerufen, stoppt die MainActivity während die SettingsActivity angezeigt wird

- Nach der Rückkehr zur MainActivity wird onStart() wiederum aufgerufen. Fall eine Neukonfiguration des Quizzes notwendig ist, erfolgt diese hier

Überschreiben von Methoden


MainActivity.java
Select Methods to Override/Implement
onResumeFragments():void
onPreparePanel(featureId:int, view:View, me
onPrepareOptionsPanel(view:View, menu:Me
onSaveInstanceState(outState:Bundle):void
onStart():void
onRetainCustomNonConfigurationInstance
getLastCustomNonConfigurationInstance():
dump(prefix:String, fd:FileDescriptor, write
onAttachFragment(fragment:Fragment):voic
getSupportFragmentManager():FragmentMa
getSupportLoaderManager():LoaderManager
startActivityForResult(intent:Intent, request
onRequestPermissionsResult(requestCode:i
startActivityFromFragment(fragment:Fragm
android.support.v4.app.BaseFragmentActivityHor
onCreateView(parent:View, name:String, cor
android.support.v4.app.BaseFragmentActivityDor
onCreateView(name:String, context:Context
android.app.Activity
getIntent():Intent
setIntent(newIntent:Intent):void
getWindowManager():WindowManager
getWindow():Window
getLoaderManager():LoaderManager
Cancel
OK

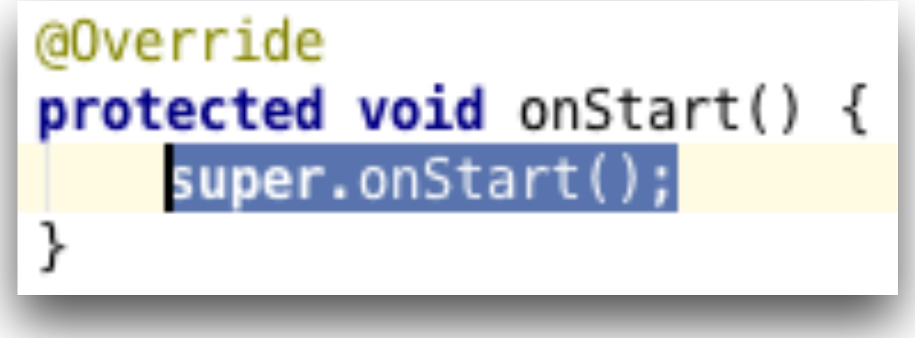
Search for: onstart

☐ Copy JavaDoc

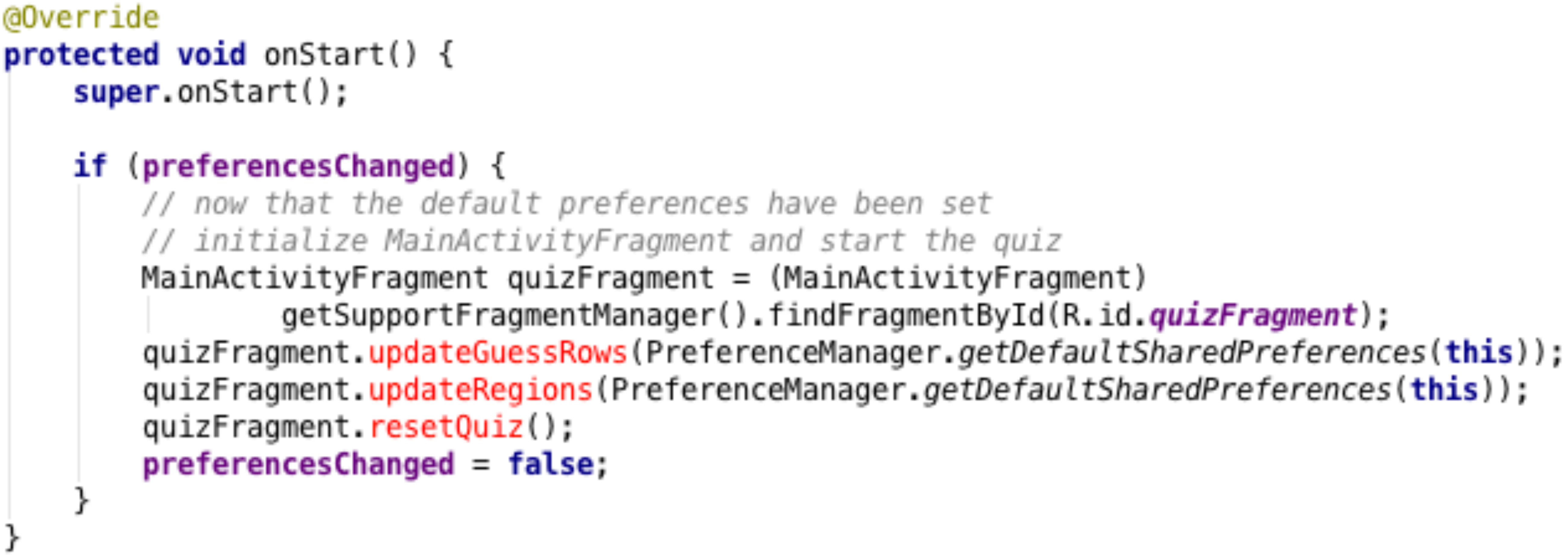
☑ Insert @Override

```
    @Override
    protected
        super
        setCo
        Toolb
        setSu

        // se
        Prefe                                    false);

        // re
        Prefe                                    esChangeList

        // de
        int s                                    yout

        // if
        if (s                                    ARGE) {

            p
        }

        // if                                    ientation
        if (p
            s                                    _PORTRAIT);
        }
    }

    @Override
    public boo
        // In                                    t is present
        getMe
        retur
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
```

- setzen Sie den Cursor unter onCreate()
- drücken Sie Ctrl-O
- geben Sie „onstart“ ein
- wählen Sie die entsprechende Methode aus
- klicken Sie Ok



```
@Override
protected void onStart() {
    super.onStart();
}
```

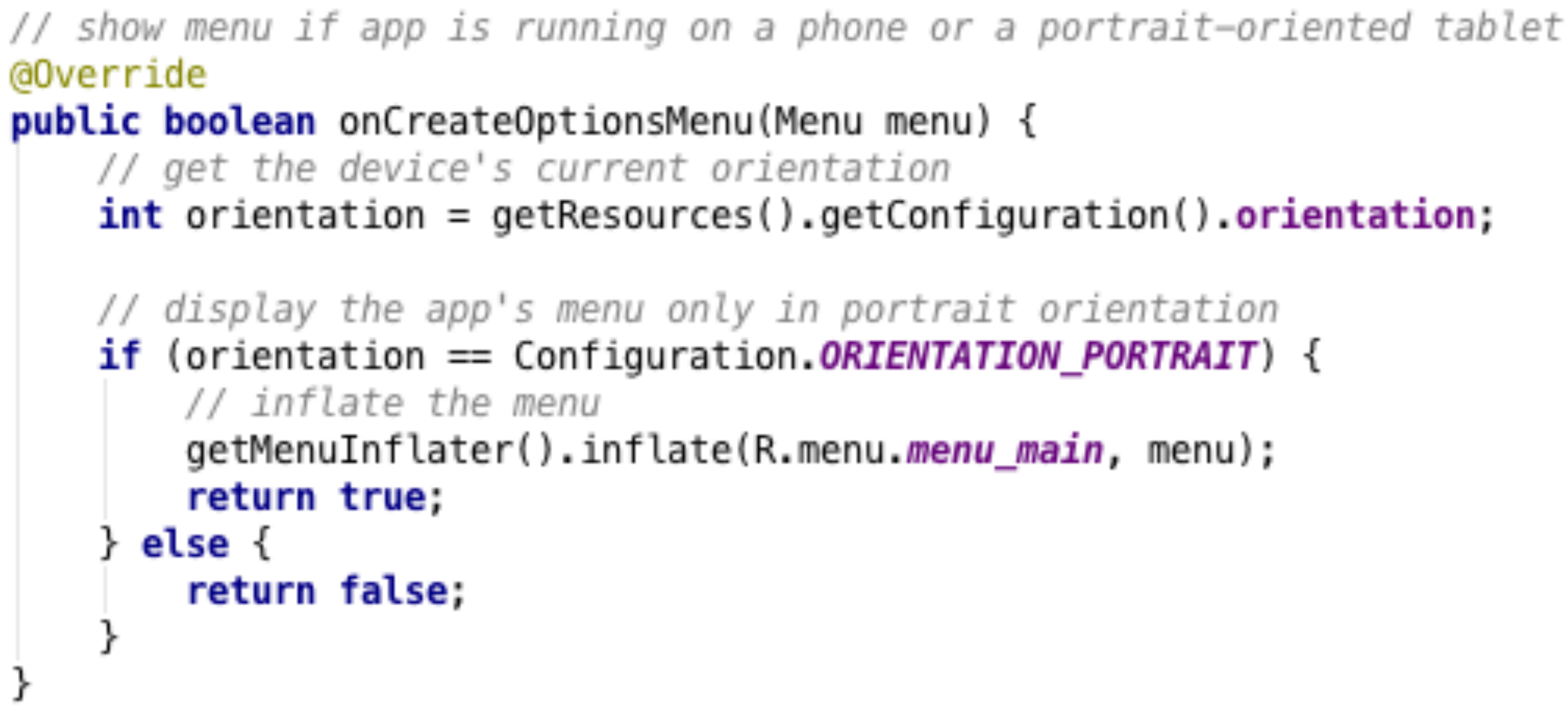


```
@Override
protected void onStart() {
    super.onStart();

    if (preferencesChanged) {
        // now that the default preferences have been set
        // initialize MainActivityFragment and start the quiz
        MainActivityFragment quizFragment = (MainActivityFragment)
                getSupportFragmentManager().findFragmentById(R.id.quizFragment);
        quizFragment.updateGuessRows(PreferenceManager.getDefaultSharedPreferences(this));
        quizFragment.updateRegions(PreferenceManager.getDefaultSharedPreferences(this));
        quizFragment.resetQuiz();
        preferencesChanged = false;
    }
}
```

- Wir „befinden“ uns in der Activity und benötigen daher eine Referenz auf das Fragment —> findFragmentById(…)

- Bei updateGuessRow() und updateRegions() wird das SharedPreferences-Objekt als Parameter übergeben

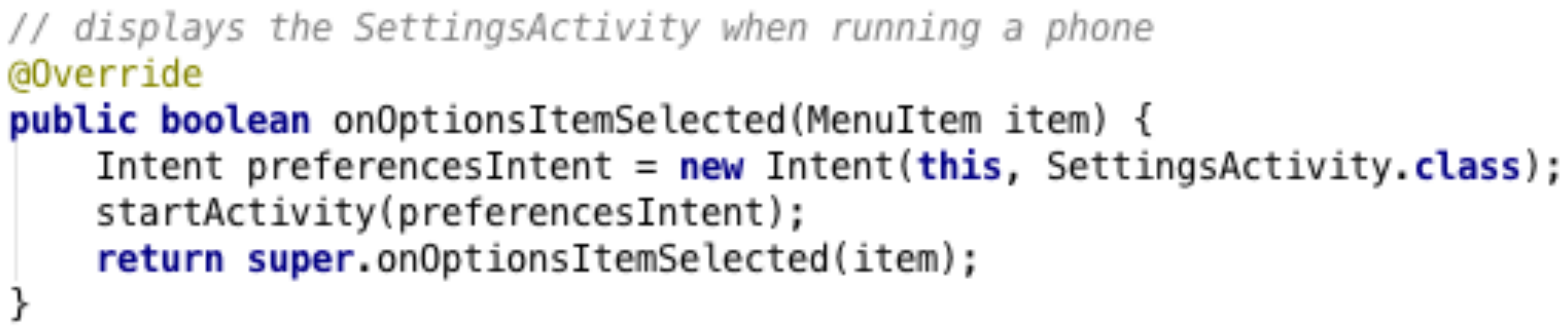
onCreateOptionsMenu()



```
// show menu if app is running on a phone or a portrait-oriented tablet
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // get the device's current orientation
    int orientation = getResources().getConfiguration().orientation;

    // display the app's menu only in portrait orientation
    if (orientation == Configuration.ORIENTATION_PORTRAIT) {
        // inflate the menu
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    } else {
        return false;
    }
}
```

- Die Methode onCreateOptionsMenu() initialisiert das Menu
- Der Methode wird das **Menu-Objekt** vom System übergeben
- Das Menu wird mit der Methode inflate() erstellt
- Der Rückgabewert von onCreateOptionsMenu() gibt an, ob das Menu angezeigt wird (true) oder nicht (false)

onOptionsItemSelected()

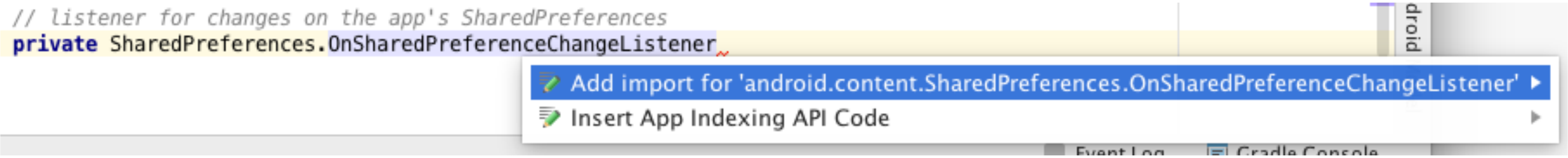


```
// displays the SettingsActivity when running a phone
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    Intent preferencesIntent = new Intent(this, SettingsActivity.class);
    startActivity(preferencesIntent);
    return super.onOptionsItemSelected(item);
}
```

- In der Methode werden i.N. die Aktionen der verschiedenen Menüeinträge behandelt. In unserer App gibt es nur den Aufruf der SettingActivity

- Beim Anlegen des expliziten Intents gibt der Parameter this, den Context an, von welchen die SettingActivity gestartet wird. sowie die zu erzeugende Activity

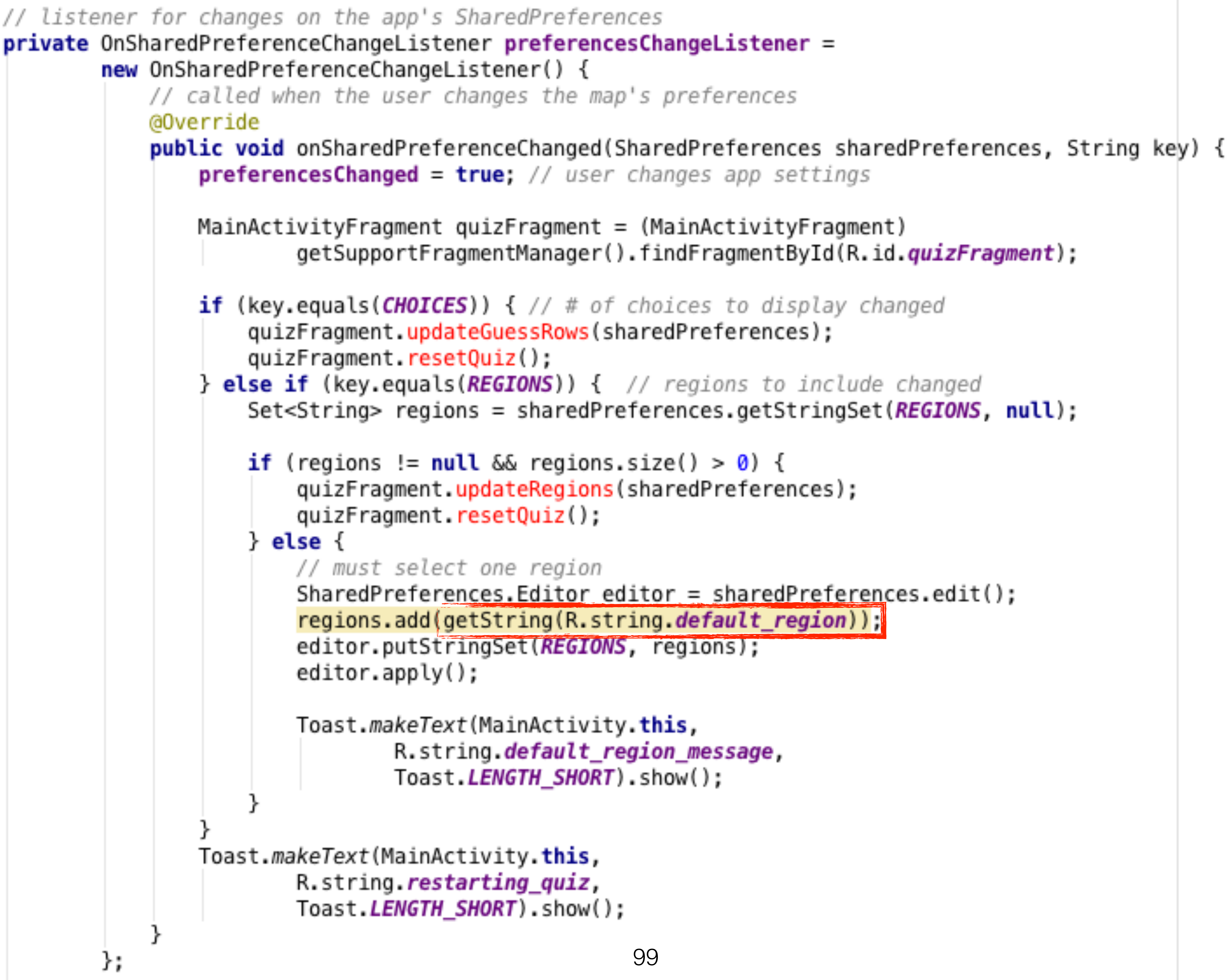
preferencesChangeListener



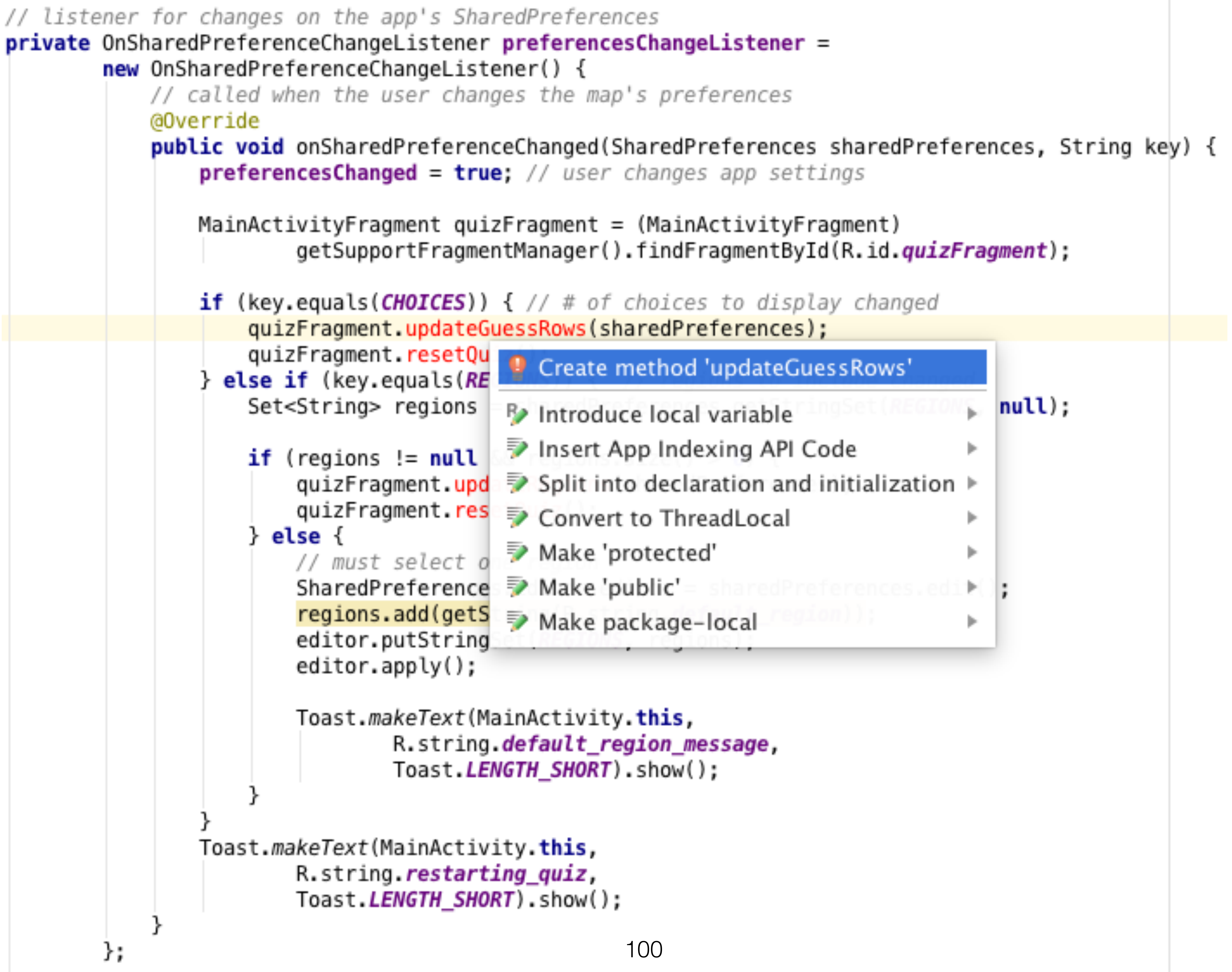
```
// listener for changes on the app's SharedPreferences
private SharedPreferences.OnSharedPreferenceChangeListener
```

Add import for 'android.content.SharedPreferences.OnSharedPreferenceChangeListener'
Insert App Indexing API Code
Event Log
Gradle Console
droid

<alt> + <Eingabe> zum Erstellen des Imports



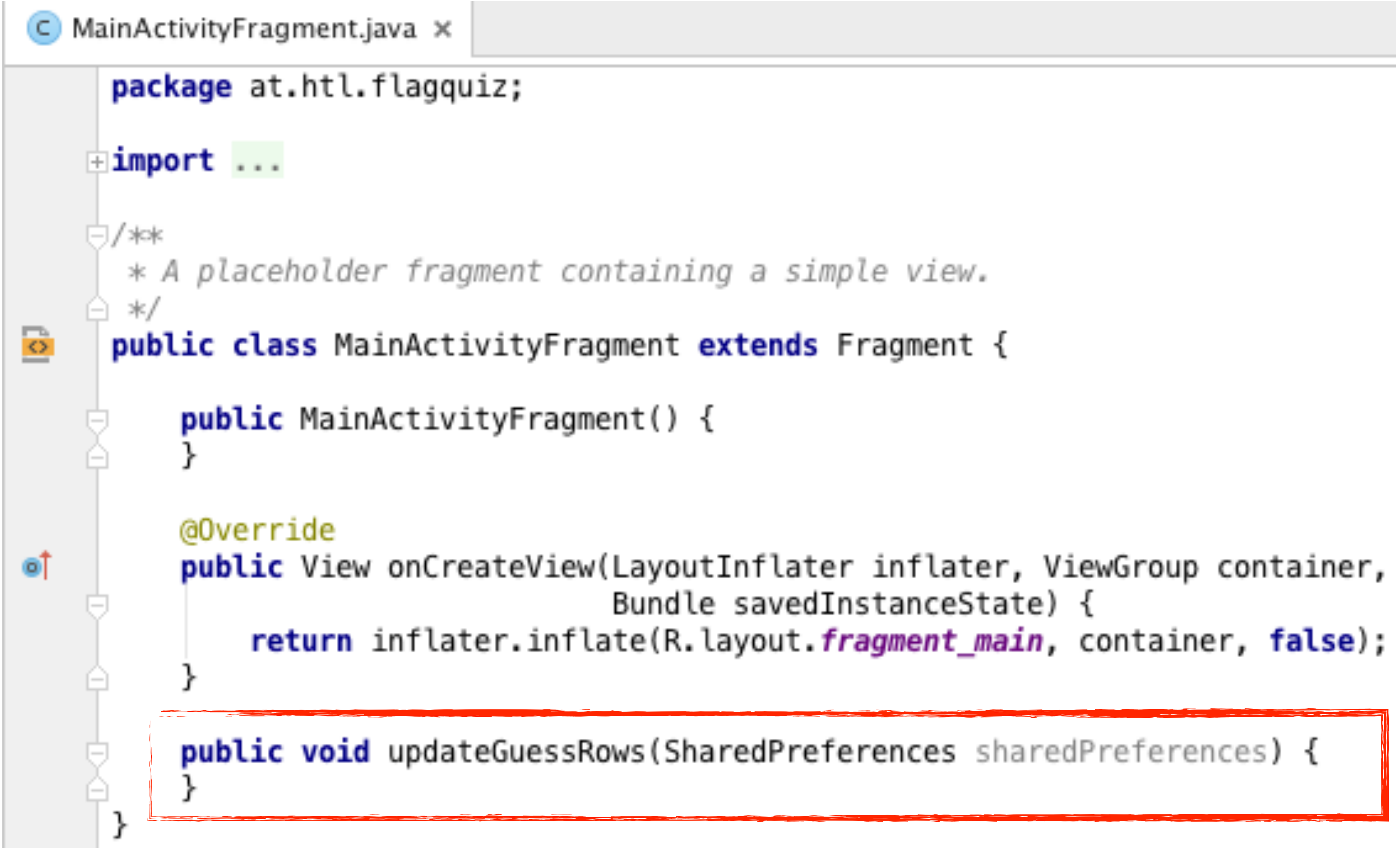
```java
// listener for changes on the app's SharedPreferences
private OnSharedPreferenceChangeListener preferencesChangeListener =
    new OnSharedPreferenceChangeListener() {
      // called when the user changes the map's preferences
      @Override
      public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
        preferencesChanged = true; // user changes app settings

        MainActivityFragment quizFragment = (MainActivityFragment)
            getSupportFragmentManager().findFragmentById(R.id.quizFragment);

        if (key.equals(CHOICES)) { // # of choices to display changed
          quizFragment.updateGuessRows(sharedPreferences);
          quizFragment.resetQuiz();
        } else if (key.equals(REGIONS)) {  // regions to include changed
          Set<String> regions = sharedPreferences.getStringSet(REGIONS, null);

          if (regions != null && regions.size() > 0) {
            quizFragment.updateRegions(sharedPreferences);
            quizFragment.resetQuiz();
          } else {
            // must select one region
            SharedPreferences.Editor editor = sharedPreferences.edit();
            regions.add(getString(R.string.default_region));
            editor.putStringSet(REGIONS, regions);
            editor.apply();

            Toast.makeText(MainActivity.this,
                R.string.default_region_message,
                Toast.LENGTH_SHORT).show();
          }
        }
        Toast.makeText(MainActivity.this,
            R.string.restarting_quiz,
            Toast.LENGTH_SHORT).show();
      }
    };
```



```
// listener for changes on the app's SharedPreferences
private OnSharedPreferenceChangeListener preferencesChangeListener =
        new OnSharedPreferenceChangeListener() {
            // called when the user changes the map's preferences
            @Override
            public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
                preferencesChanged = true; // user changes app settings

                MainActivityFragment quizFragment = (MainActivityFragment)
                        getSupportFragmentManager().findFragmentById(R.id.quizFragment);

                if (key.equals(CHOICES)) { // # of choices to display changed
                    quizFragment.updateGuessRows(sharedPreferences);
                    quizFragment.resetQu
                } else if (key.equals(RE
                    Set<String> regions                                              null);

                    if (regions != null
                        quizFragment.upd
                        quizFragment.res
                    } else {
                        // must select o
                        SharedPreference                                              ;
                        regions.add(getS
                        editor.putString
                        editor.apply();

                        Toast.makeText(MainActivity.this,
                                R.string.default_region_message,
                                Toast.LENGTH_SHORT).show();
                    }
                }
                Toast.makeText(MainActivity.this,
                        R.string.restarting_quiz,
                        Toast.LENGTH_SHORT).show();
            }
        };
```

Create method 'updateGuessRows'
Introduce local variable
Insert App Indexing API Code
Split into declaration and initialization
Convert to ThreadLocal
Make 'protected'
Make 'public'
Make package-local

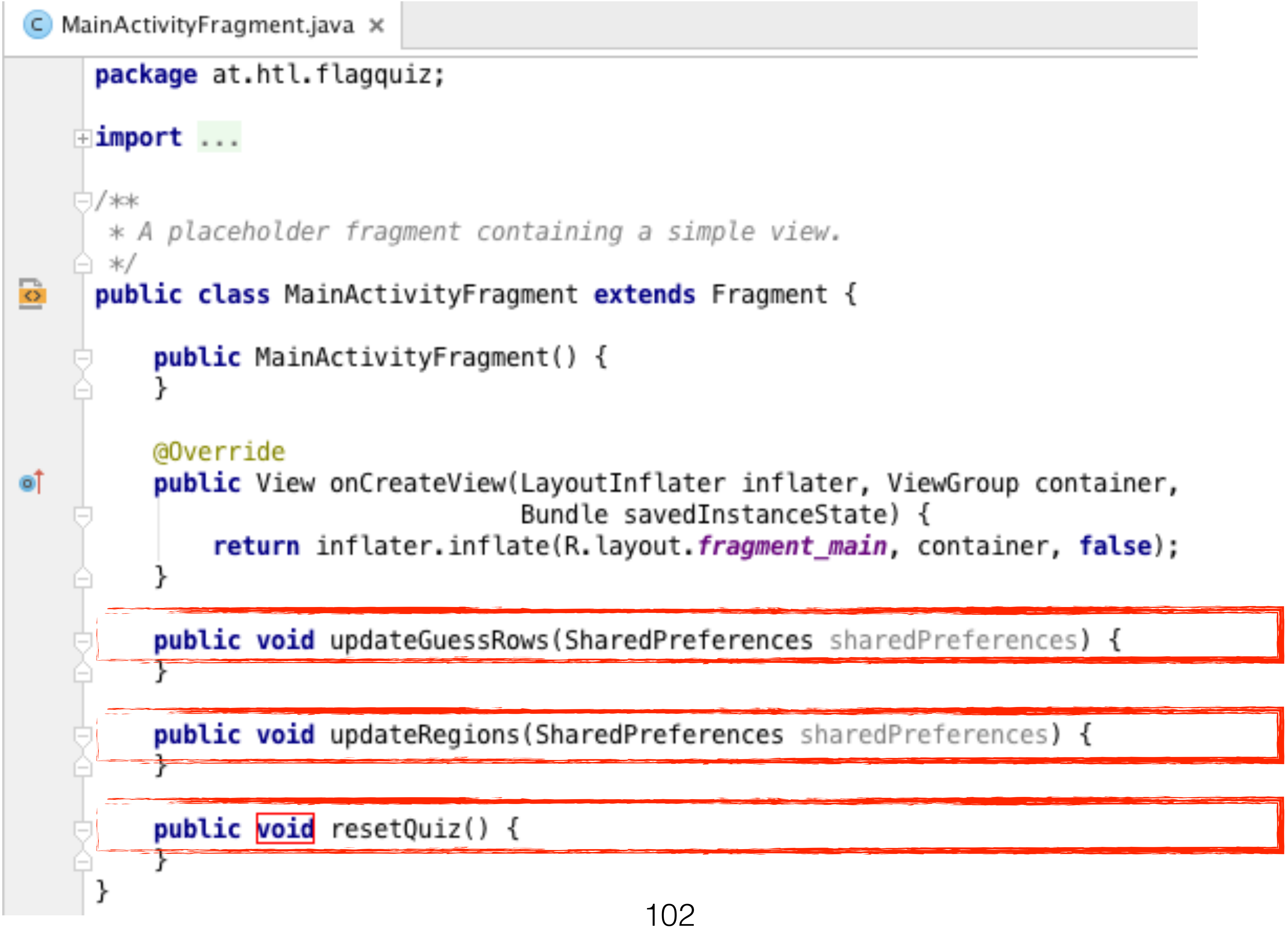
Erstellte Methode



MainActivityFragment.java

```
package at.htl.flagquiz;

import ...

/**
 * A placeholder fragment containing a simple view.
 */
public class MainActivityFragment extends Fragment {

    public MainActivityFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_main, container, false);
    }

    public void updateGuessRows(SharedPreferences sharedPreferences) {
    }
}
```

Alle generierten Methoden

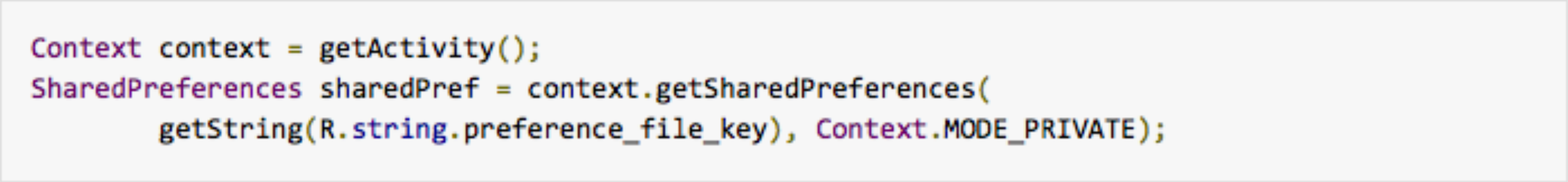


```java
package at.htl.flagquiz;

import ...

/**
 * A placeholder fragment containing a simple view.
 */
public class MainActivityFragment extends Fragment {

    public MainActivityFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_main, container, false);
    }

    public void updateGuessRows(SharedPreferences sharedPreferences) {
    }

    public void updateRegions(SharedPreferences sharedPreferences) {
    }

    public void resetQuiz() {
    }
}
```

Zusammenfassung

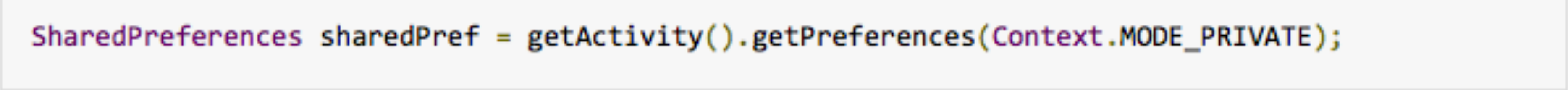
SharedPreferences

- Erstellen:
  - getSharedPreferences(…) … wenn mehrere Files



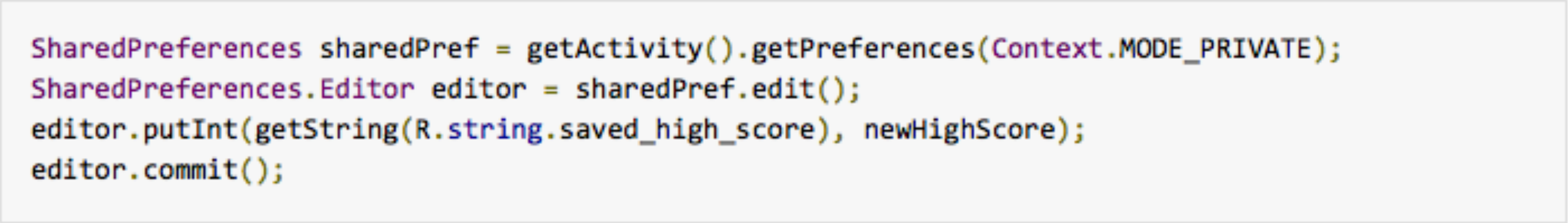
```
Context context = getActivity();
SharedPreferences sharedPref = context.getSharedPreferences(
        getString(R.string.preference_file_key), Context.MODE_PRIVATE);
```

  - getPreferences … wenn nur ein preferences-File ev. anstelle von getDefaultSharedPreferences()



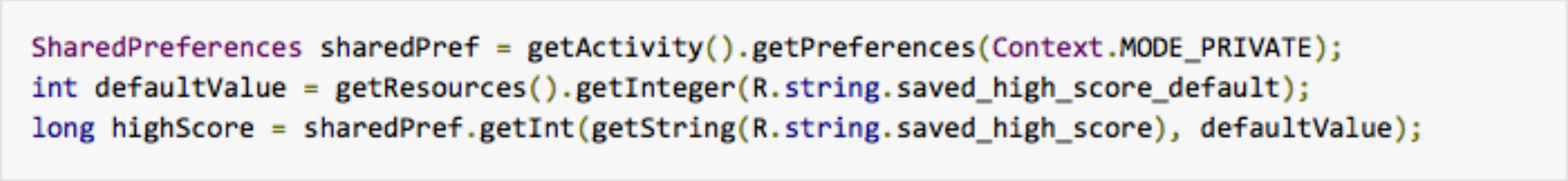
```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
```

- Schreiben:



```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPref.edit();
editor.putInt(getString(R.string.saved_high_score), newHighScore);
editor.commit();
```

- Lesen:

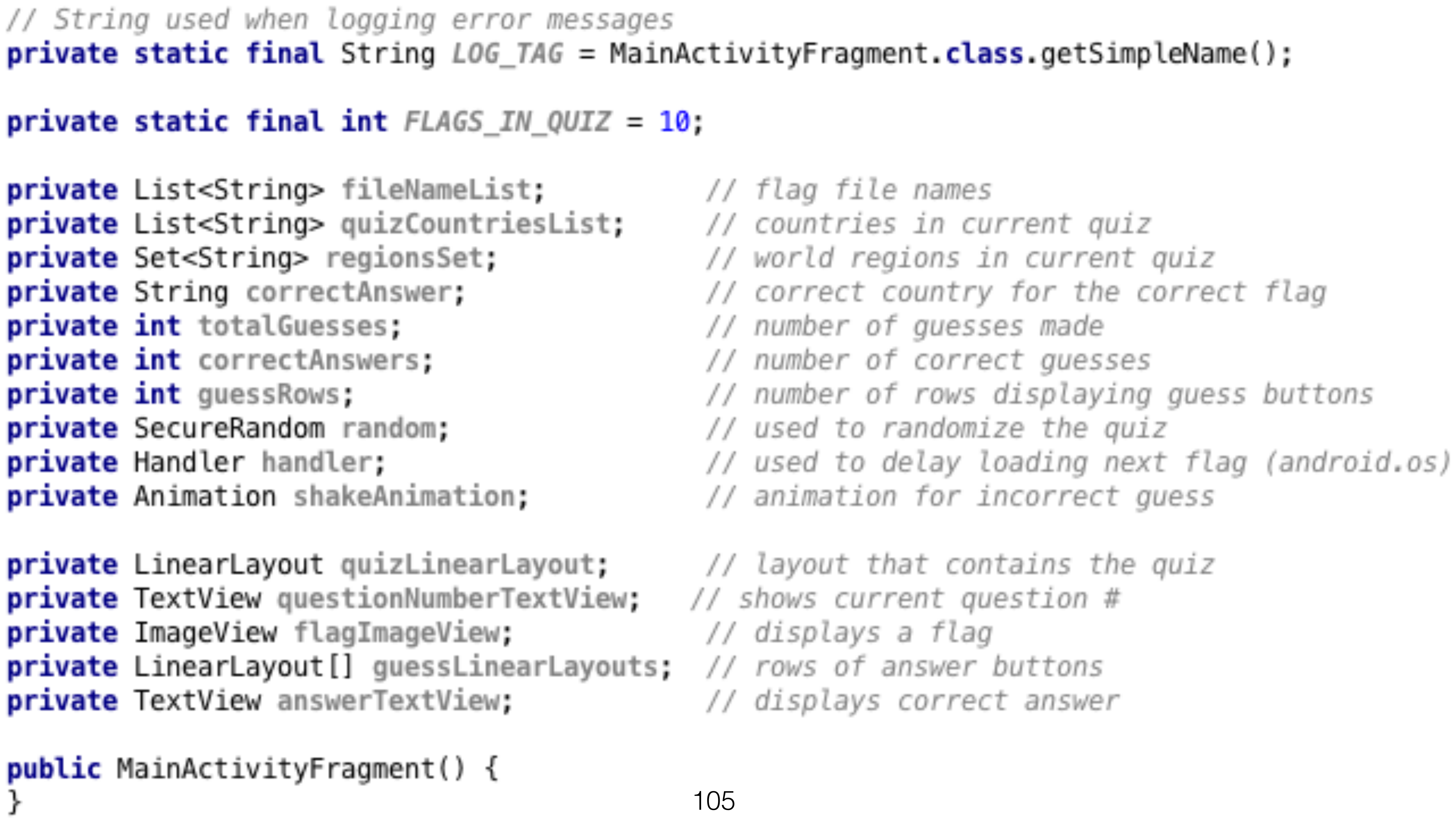


```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
int defaultValue = getResources().getInteger(R.string.saved_high_score_default);
long highScore = sharedPref.getInt(getString(R.string.saved_high_score), defaultValue);
```

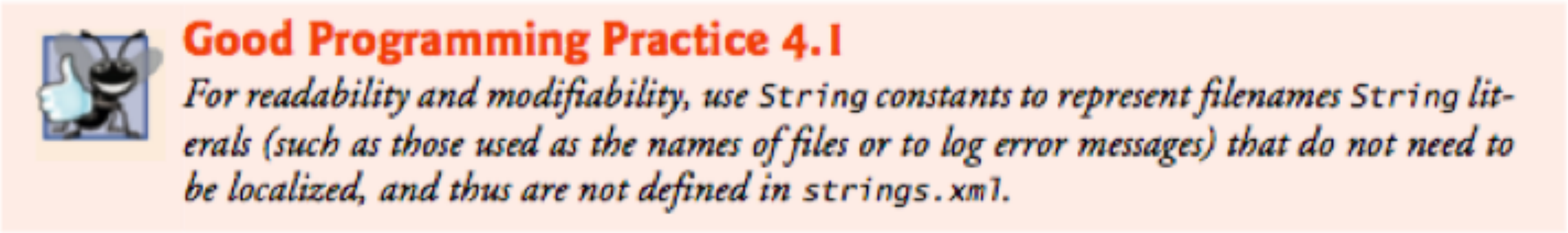
http://developer.android.com/training/basics/data-storage/shared-preferences.html

MainActivityFragment.java

Fields



```
// String used when logging error messages
private static final String LOG_TAG = MainActivityFragment.class.getSimpleName();

private static final int FLAGS_IN_QUIZ = 10;

private List<String> fileNameList;          // flag file names
private List<String> quizCountriesList;     // countries in current quiz
private Set<String> regionsSet;             // world regions in current quiz
private String correctAnswer;               // correct country for the correct flag
private int totalGuesses;                   // number of guesses made
private int correctAnswers;                 // number of correct guesses
private int guessRows;                      // number of rows displaying guess buttons
private SecureRandom random;                // used to randomize the quiz
private Handler handler;                    // used to delay loading next flag (android.os)
private Animation shakeAnimation;           // animation for incorrect guess

private LinearLayout quizLinearLayout;      // layout that contains the quiz
private TextView questionNumberTextView;   // shows current question #
private ImageView flagImageView;            // displays a flag
private LinearLayout[] guessLinearLayouts;  // rows of answer buttons
private TextView answerTextView;            // displays correct answer

public MainActivityFragment() {
}
```

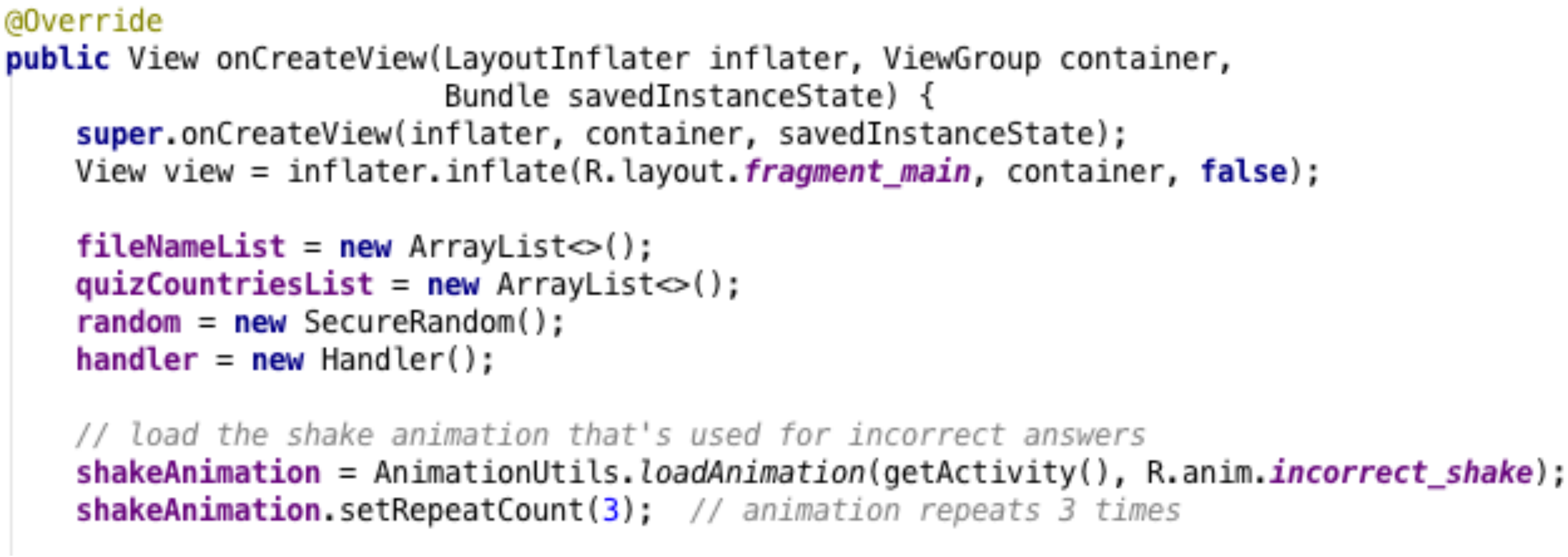


Good Programming Practice 4.1

For readability and modifiability, use `String` constants to represent filenames `String` literals (such as those used as the names of files or to log error messages) that do not need to be localized, and thus are not defined in `strings.xml`.

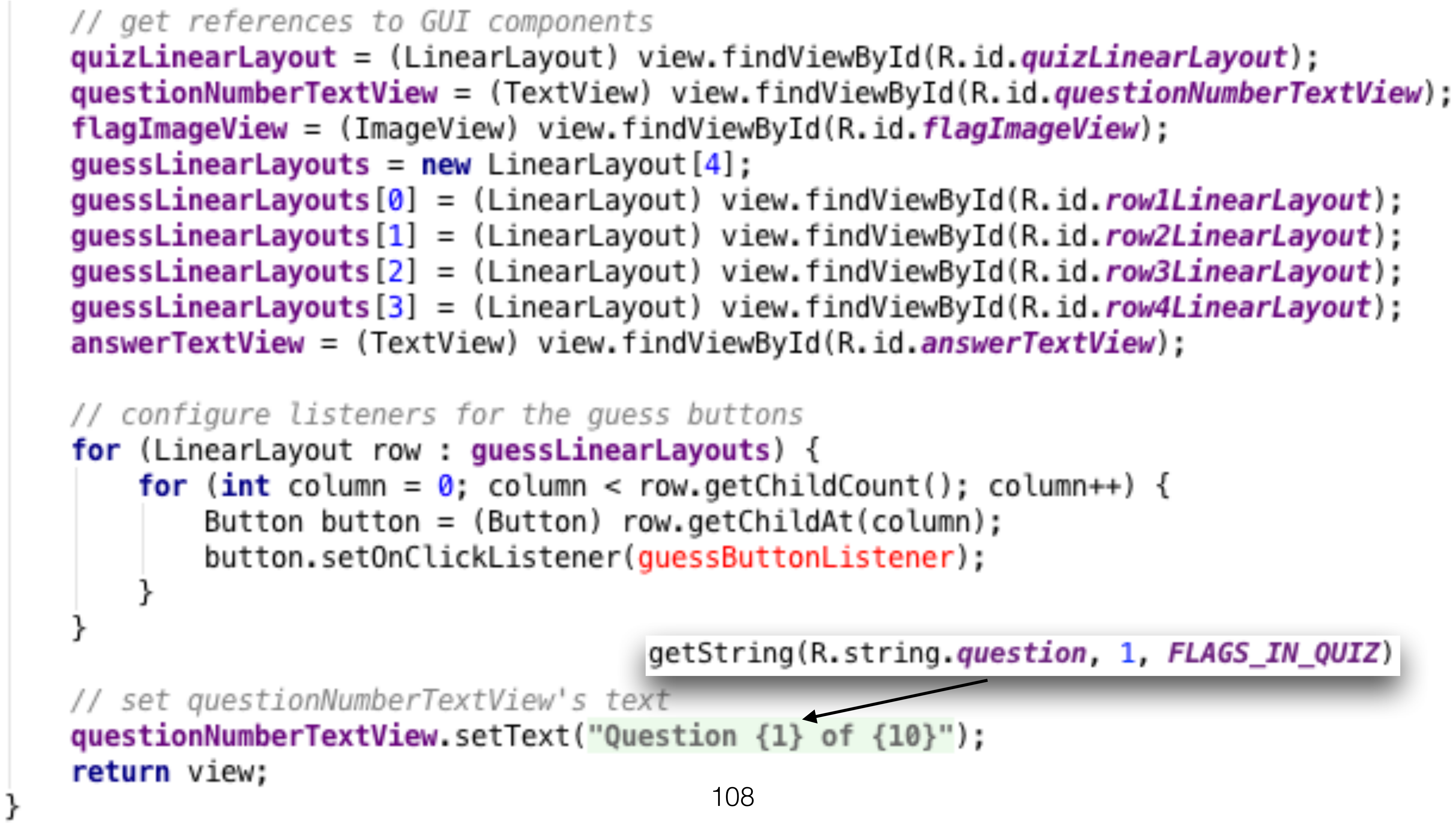
- Unterschied Random vs. SecureRandom
  - java.util.**Random** … ist ein Linear Congruential Generator (LCG) und ziemlich vorherbestimmbar (vgl. diesen Artikel)
  - java.security.**SecureRandom** verwendet einen schwerer bestimmbaren Seed und ist daher schwerer vorhersehbar

onCreateView() 1



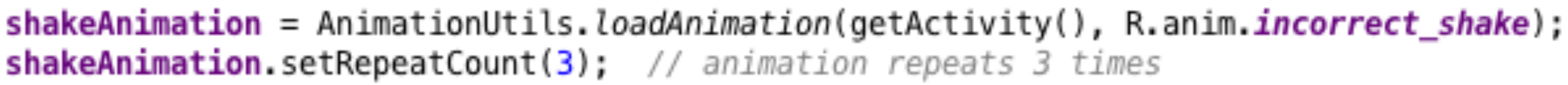
```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    super.onCreateView(inflater, container, savedInstanceState);
    View view = inflater.inflate(R.layout.fragment_main, container, false);

    fileNameList = new ArrayList<>();
    quizCountriesList = new ArrayList<>();
    random = new SecureRandom();
    handler = new Handler();

    // load the shake animation that's used for incorrect answers
    shakeAnimation = AnimationUtils.loadAnimation(getActivity(), R.anim.incorrect_shake);
    shakeAnimation.setRepeatCount(3);  // animation repeats 3 times
```

onCreateView() 2



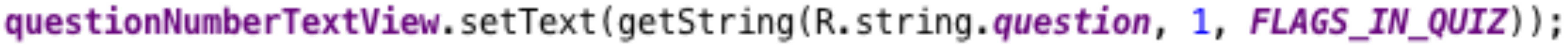
```
    // get references to GUI components
    quizLinearLayout = (LinearLayout) view.findViewById(R.id.quizLinearLayout);
    questionNumberTextView = (TextView) view.findViewById(R.id.questionNumberTextView);
    flagImageView = (ImageView) view.findViewById(R.id.flagImageView);
    guessLinearLayouts = new LinearLayout[4];
    guessLinearLayouts[0] = (LinearLayout) view.findViewById(R.id.row1LinearLayout);
    guessLinearLayouts[1] = (LinearLayout) view.findViewById(R.id.row2LinearLayout);
    guessLinearLayouts[2] = (LinearLayout) view.findViewById(R.id.row3LinearLayout);
    guessLinearLayouts[3] = (LinearLayout) view.findViewById(R.id.row4LinearLayout);
    answerTextView = (TextView) view.findViewById(R.id.answerTextView);

    // configure listeners for the guess buttons
    for (LinearLayout row : guessLinearLayouts) {
        for (int column = 0; column < row.getChildCount(); column++) {
            Button button = (Button) row.getChildAt(column);
            button.setOnClickListener(guessButtonListener);
        }
    }

    // set questionNumberTextView's text
    questionNumberTextView.setText("Question {1} of {10}");
    return view;
}
```

`getString(R.string.question, 1, FLAGS_IN_QUIZ)`



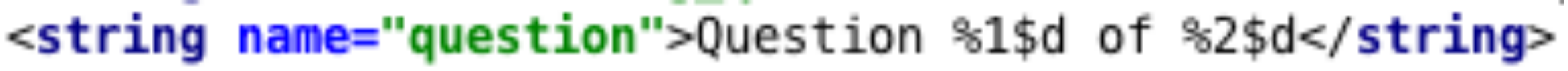
```
shakeAnimation = AnimationUtils.loadAnimation(getActivity(), R.anim.incorrect_shake);
shakeAnimation.setRepeatCount(3);  // animation repeats 3 times
```

- getActivity() gibt die Activity zurück, welche dieses Fragment hostet. Activity ist eine direkte Subklasse von Context.



```
questionNumberTextView.setText(getString(R.string.question, 1, FLAGS_IN_QUIZ));
```

- R.string.question sieht wie folgt aus

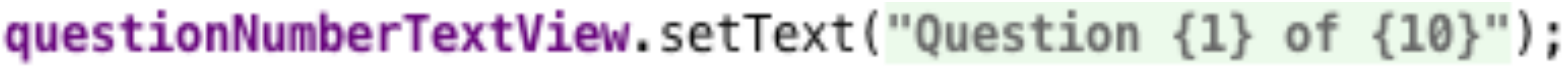


```
<string name="question">Question %1$d of %2$d</string>
```

und benötigt daher zwei Parameter

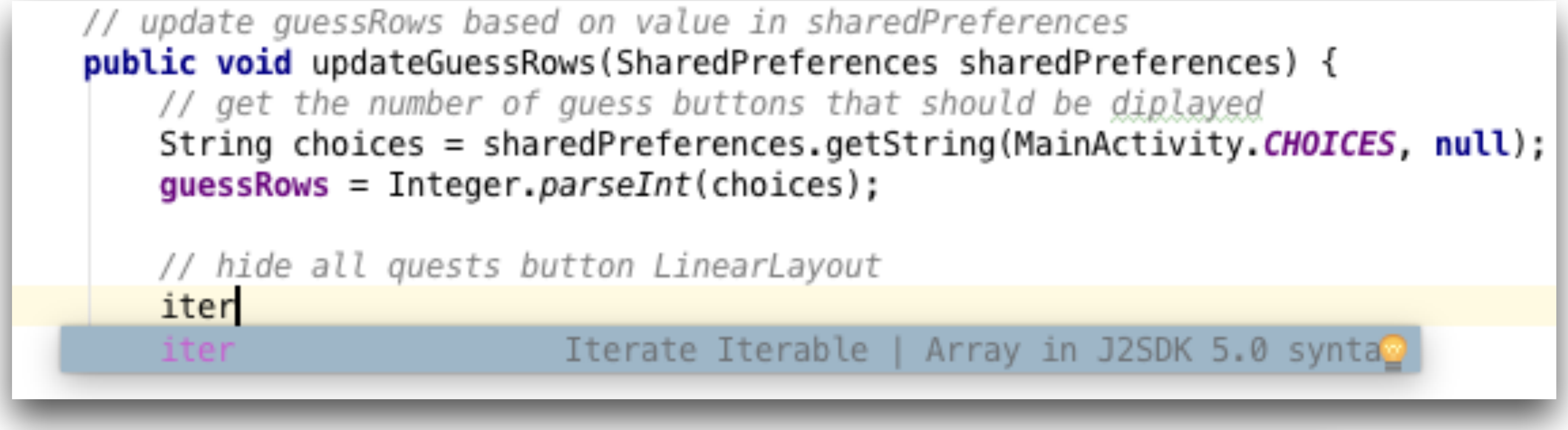
  - 1 … als Startwert für die erste Frage

  - Anzahl der gesamten Flaggen im Quiz



```
questionNumberTextView.setText("Question {1} of {10}");
```

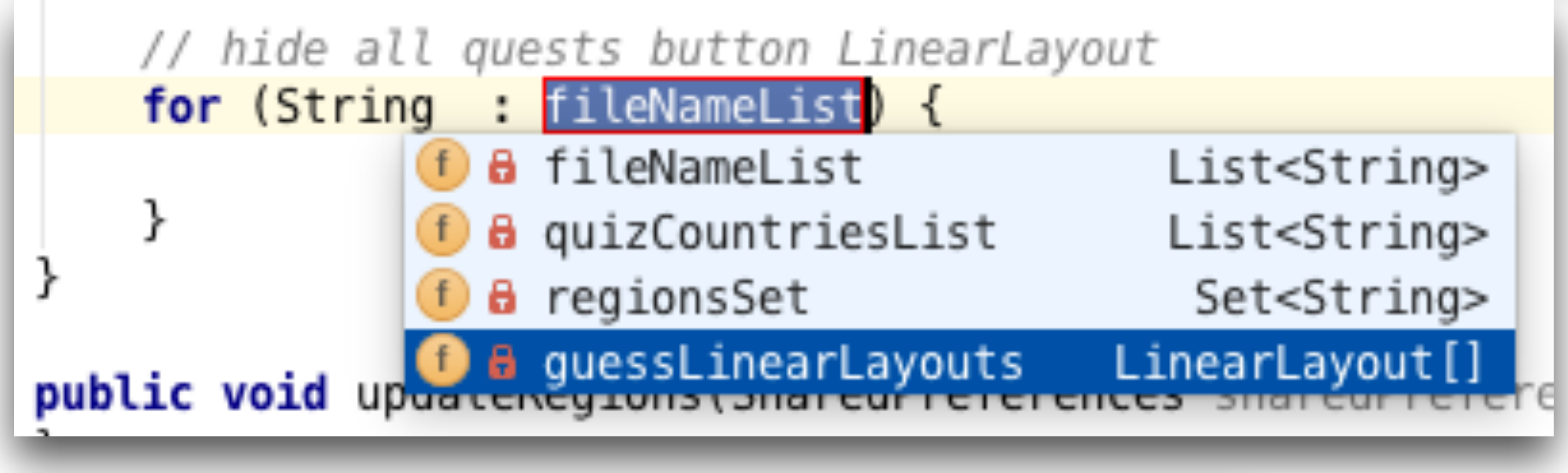
Live Template iter



```
// update guessRows based on value in sharedPreferences
public void updateGuessRows(SharedPreferences sharedPreferences) {
    // get the number of guess buttons that should be diplayed
    String choices = sharedPreferences.getString(MainActivity.CHOICES, null);
    guessRows = Integer.parseInt(choices);

    // hide all quests button LinearLayout
    iter
```

iter   Iterate Iterable | Array in J2SDK 5.0 synta

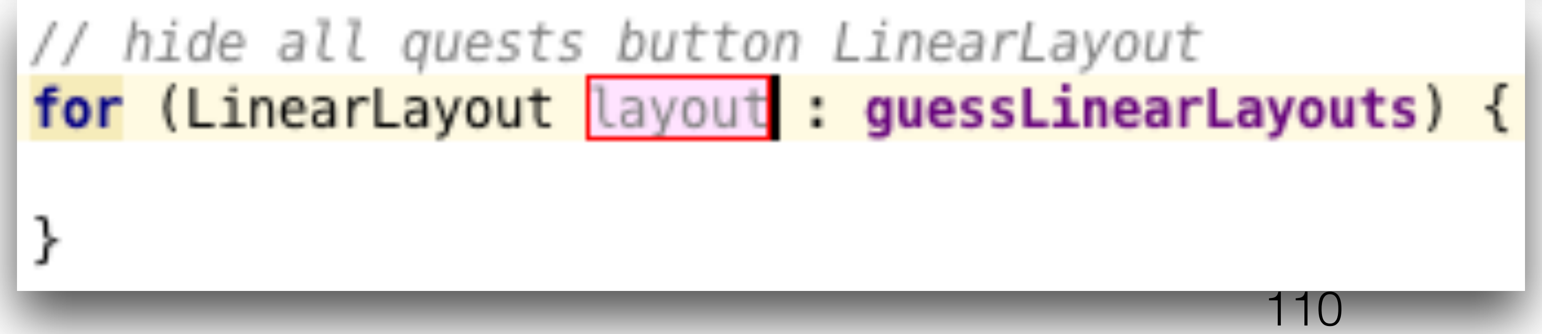
iter<tab>



```
    // hide all quests button LinearLayout
    for (String  : fileNameList) {

    }
}

public void updateRegions(SharedPreferences sharedPrefere
```

fileNameList List<String>
quizCountriesList List<String>
regionsSet Set<String>
guessLinearLayouts LinearLayout[]

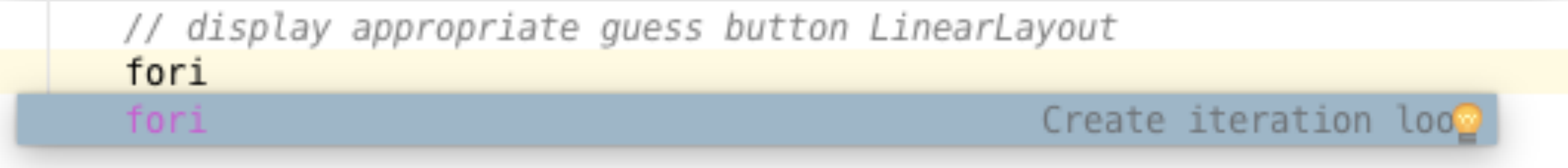
⬇⬇⬇<Enter>



```
// hide all quests button LinearLayout
for (LinearLayout layout : guessLinearLayouts) {

}
```

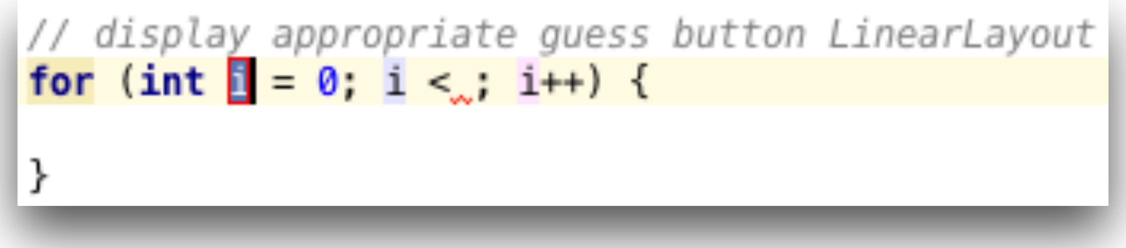
layout<Enter>

Live Template fori



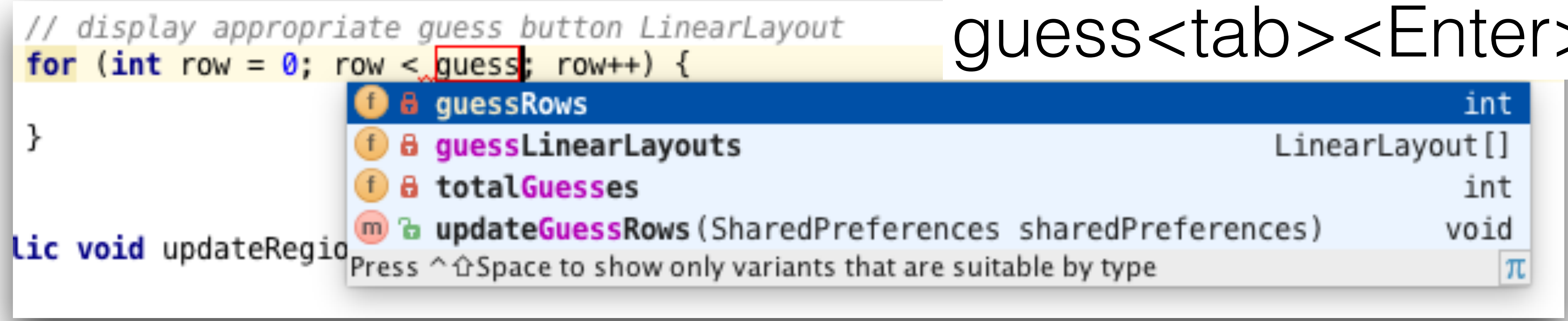
```
// display appropriate guess button LinearLayout
fori
fori                                        Create iteration loo
```

fori<tab>



```
// display appropriate guess button LinearLayout
for (int i = 0; i < ; i++) {

}
```

row<Enter>

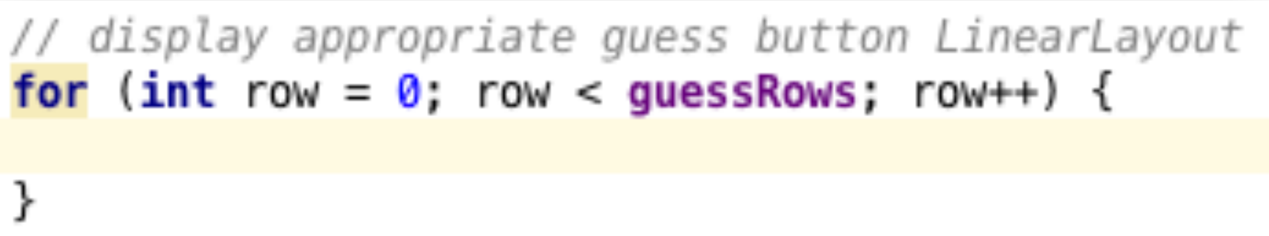


```
// display appropriate guess button LinearLayout
for (int row = 0; row < guess; row++) {

}

lic void updateRegio
```

guessRows	int
guessLinearLayouts	LinearLayout[]
totalGuesses	int
updateGuessRows(SharedPreferences sharedPreferences)	void

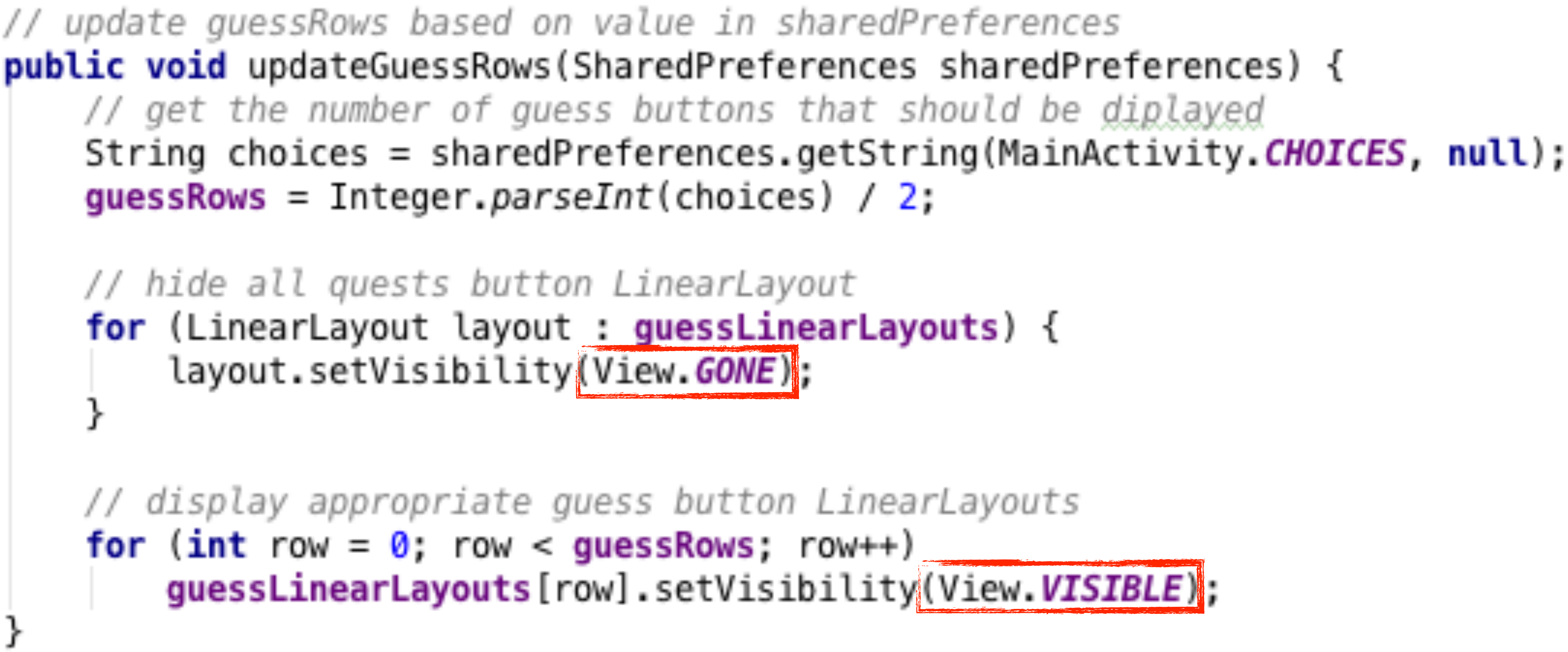
Press ^⇧Space to show only variants that are suitable by type π

guess<tab><Enter>



```
// display appropriate guess button LinearLayout
for (int row = 0; row < guessRows; row++) {

}
```

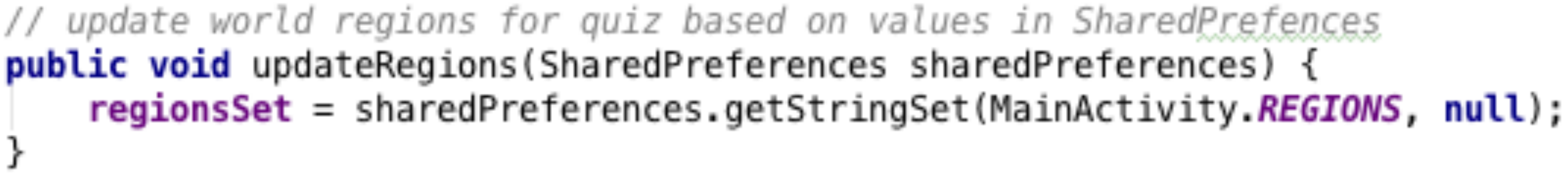
updateGuessRows()



```
// update guessRows based on value in sharedPreferences
public void updateGuessRows(SharedPreferences sharedPreferences) {
    // get the number of guess buttons that should be diplayed
    String choices = sharedPreferences.getString(MainActivity.CHOICES, null);
    guessRows = Integer.parseInt(choices) / 2;

    // hide all quests button LinearLayout
    for (LinearLayout layout : guessLinearLayouts) {
        layout.setVisibility(View.GONE);
    }

    // display appropriate guess button LinearLayouts
    for (int row = 0; row < guessRows; row++)
        guessLinearLayouts[row].setVisibility(View.VISIBLE);
}
```

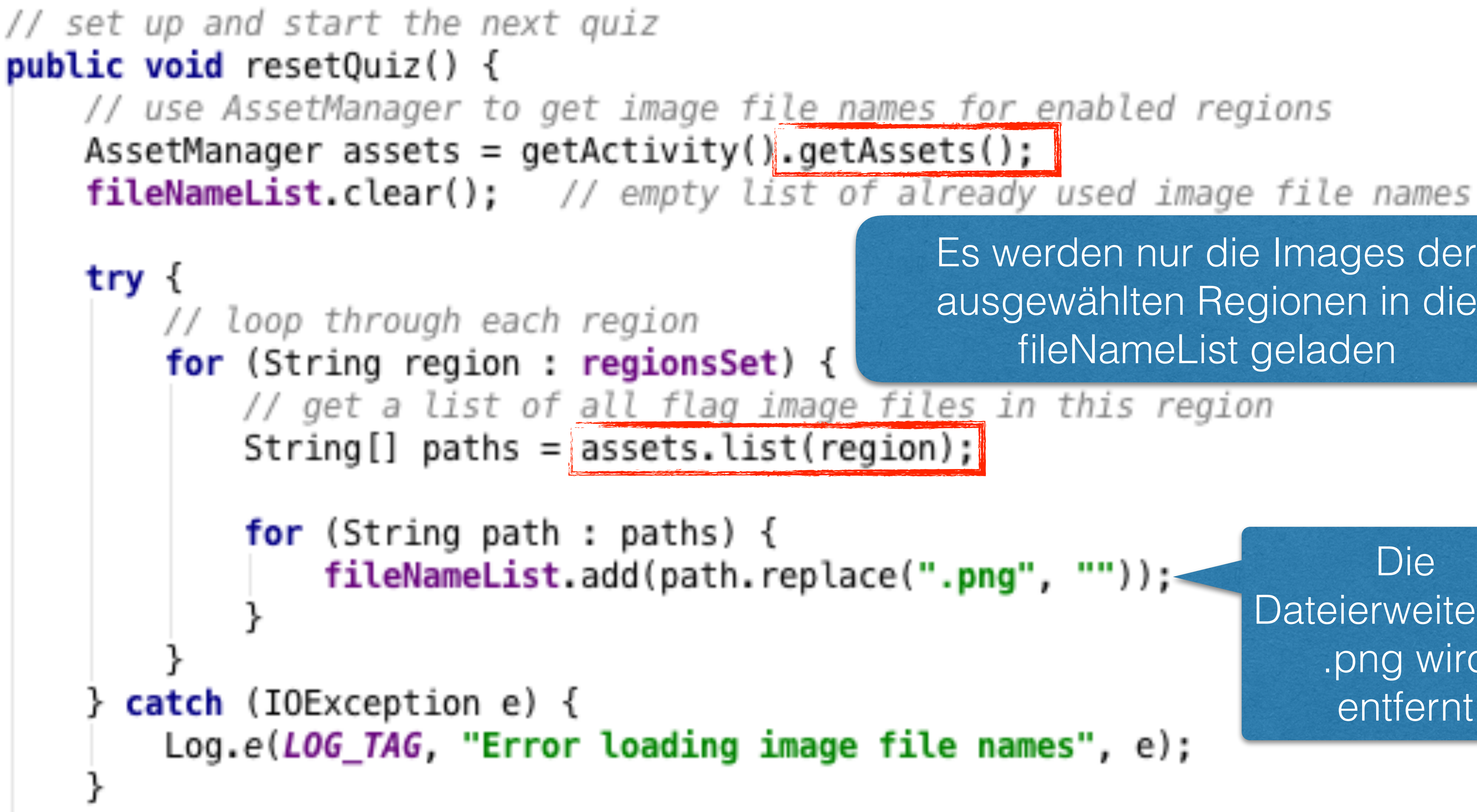
- View.VISIBLE … View wird angezeigt
- View.INVISIBLE … View wird nicht angezeigt, Platz bleibt - wegen Layout - reserviert
- View.GONE … View wird nicht angezeigt; Platz ist für andere Komponenten verfügbar

updateRegions()



```java
// update world regions for quiz based on values in SharedPrefences
public void updateRegions(SharedPreferences sharedPreferences) {
    regionsSet = sharedPreferences.getStringSet(MainActivity.REGIONS, null);
}
```

resetQuiz() 1



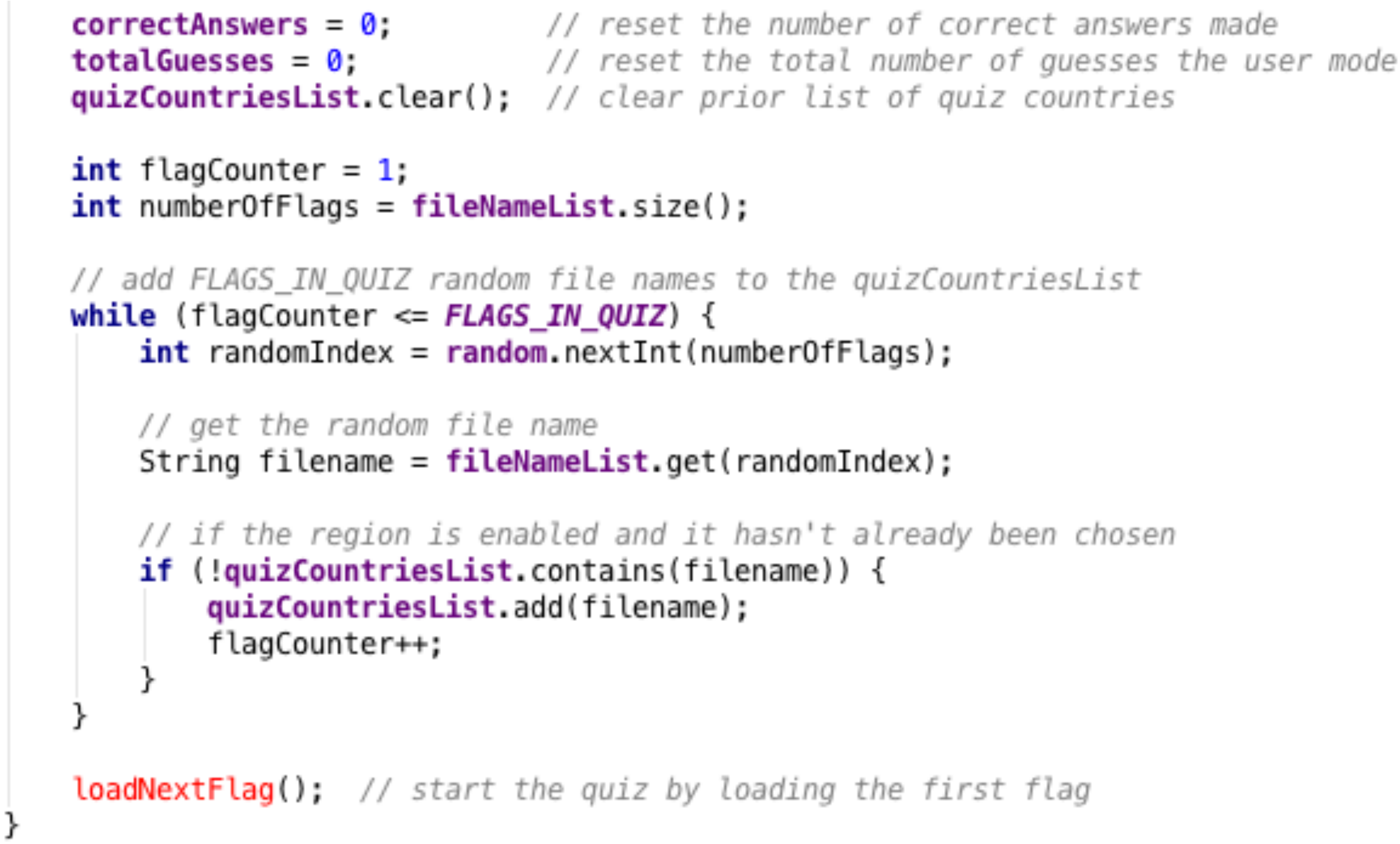
```java
// set up and start the next quiz
public void resetQuiz() {
    // use AssetManager to get image file names for enabled regions
    AssetManager assets = getActivity().getAssets();
    fileNameList.clear();   // empty list of already used image file names

    try {
        // loop through each region
        for (String region : regionsSet) {
            // get a list of all flag image files in this region
            String[] paths = assets.list(region);

            for (String path : paths) {
                fileNameList.add(path.replace(".png", ""));
            }
        }
    } catch (IOException e) {
        Log.e(LOG_TAG, "Error loading image file names", e);
    }
```

Es werden nur die Images der ausgewählten Regionen in die fileNameList geladen

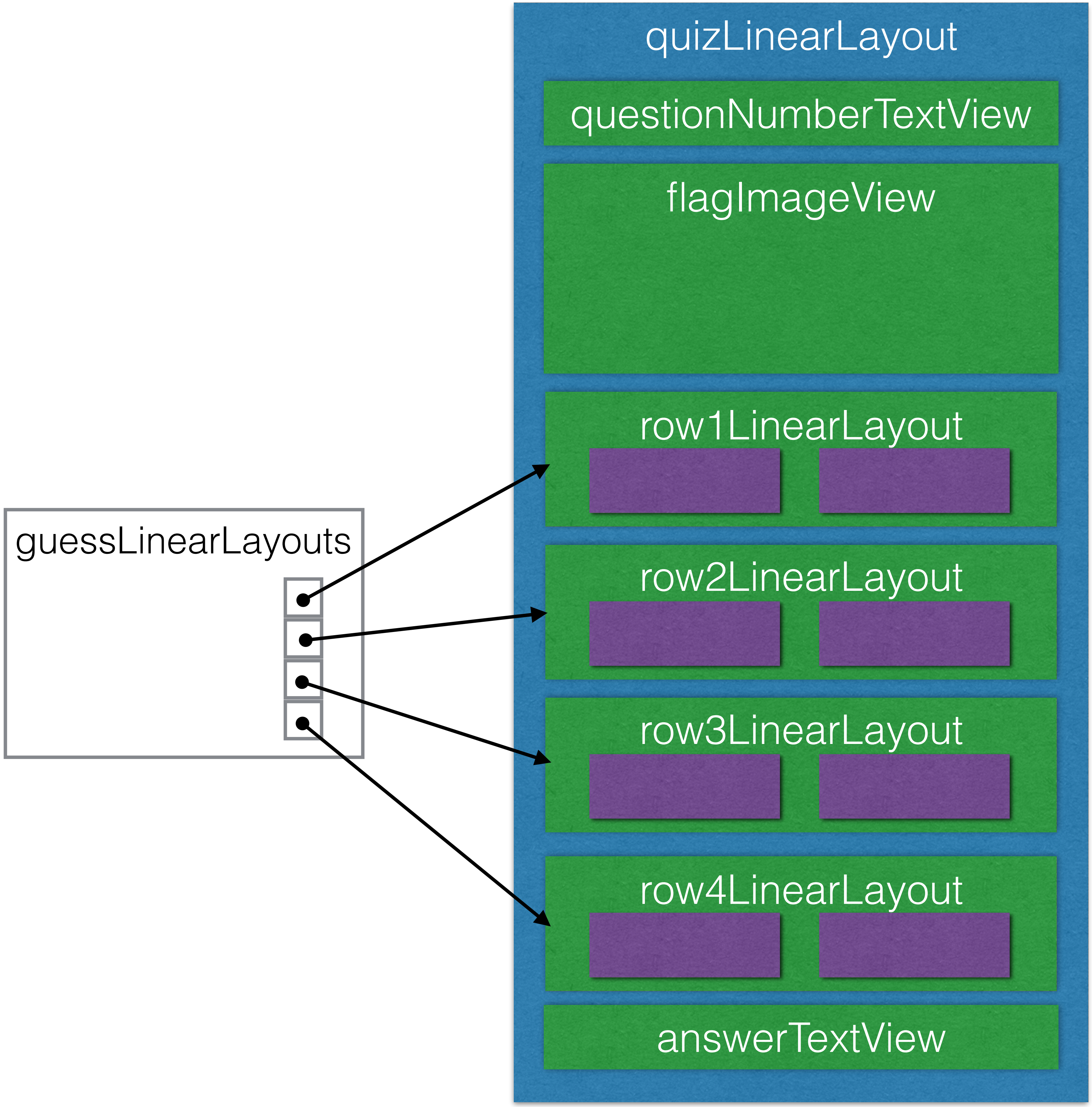
Die Dateierweiterung .png wird entfernt

http://developer.android.com/reference/android/util/Log.html

resetQuiz() 2



```
    correctAnswers = 0;         // reset the number of correct answers made
    totalGuesses = 0;           // reset the total number of guesses the user mode
    quizCountriesList.clear();  // clear prior list of quiz countries

    int flagCounter = 1;
    int numberOfFlags = fileNameList.size();

    // add FLAGS_IN_QUIZ random file names to the quizCountriesList
    while (flagCounter <= FLAGS_IN_QUIZ) {
        int randomIndex = random.nextInt(numberOfFlags);

        // get the random file name
        String filename = fileNameList.get(randomIndex);

        // if the region is enabled and it hasn't already been chosen
        if (!quizCountriesList.contains(filename)) {
            quizCountriesList.add(filename);
            flagCounter++;
        }
    }

    loadNextFlag();  // start the quiz by loading the first flag
}
```



quizLinearLayout

questionNumberTextView

flagImageView

row1LinearLayout

row2LinearLayout

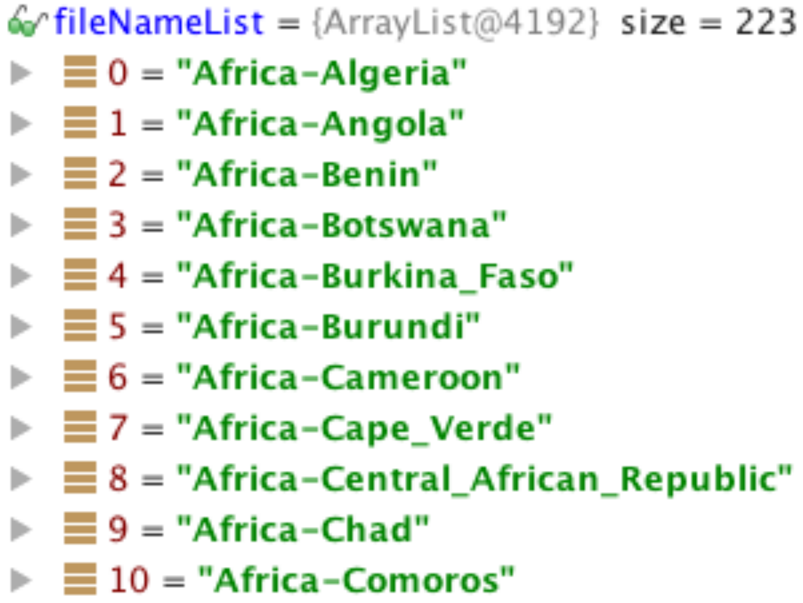
row3LinearLayout

row4LinearLayout

answerTextView

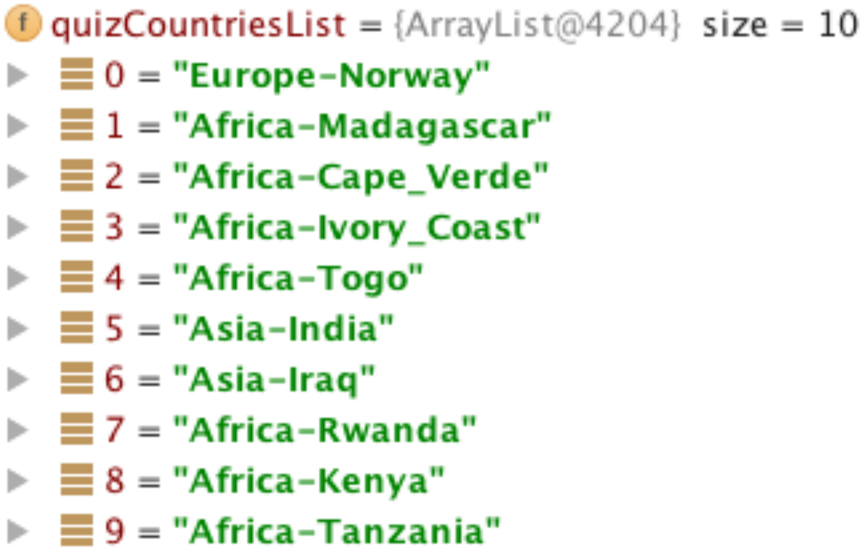
guessLinearLayouts

alle Länder, die für das Quiz verwendet werden **können**



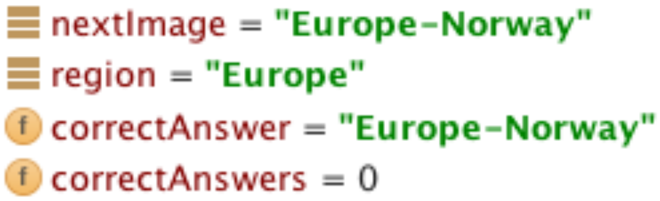
```
fileNameList = {ArrayList@4192}  size = 223
▶ 0 = "Africa-Algeria"
▶ 1 = "Africa-Angola"
▶ 2 = "Africa-Benin"
▶ 3 = "Africa-Botswana"
▶ 4 = "Africa-Burkina_Faso"
▶ 5 = "Africa-Burundi"
▶ 6 = "Africa-Cameroon"
▶ 7 = "Africa-Cape_Verde"
▶ 8 = "Africa-Central_African_Republic"
▶ 9 = "Africa-Chad"
▶ 10 = "Africa-Comoros"
```

alle Länder, die für das Quiz verwendet **werden** (korrekte Antw.)



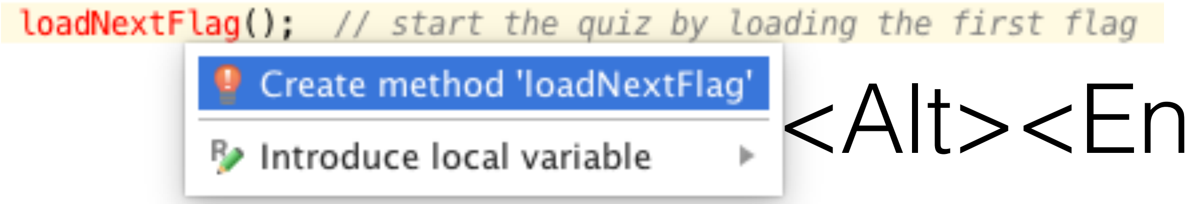
```
quizCountriesList = {ArrayList@4204}  size = 10
▶ 0 = "Europe-Norway"
▶ 1 = "Africa-Madagascar"
▶ 2 = "Africa-Cape_Verde"
▶ 3 = "Africa-Ivory_Coast"
▶ 4 = "Africa-Togo"
▶ 5 = "Asia-India"
▶ 6 = "Asia-Iraq"
▶ 7 = "Africa-Rwanda"
▶ 8 = "Africa-Kenya"
▶ 9 = "Africa-Tanzania"
```

Anzahl der Elemente = FLAGS_IN_QUIZ



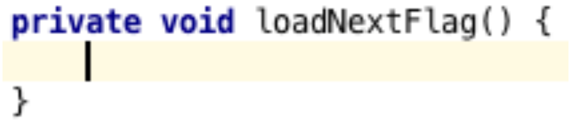
```
nextImage = "Europe-Norway"
region = "Europe"
correctAnswer = "Europe-Norway"
correctAnswers = 0
```

loadNextFlag() 1

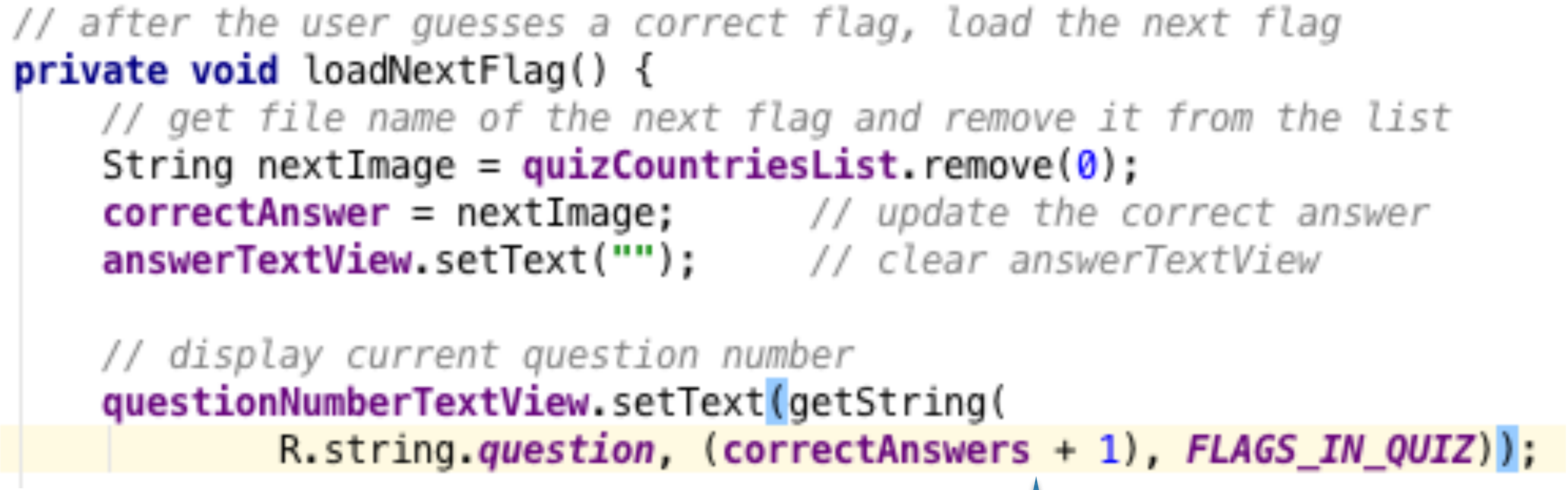


```
loadNextFlag(); // start the quiz by loading the first flag
```

<Alt><Enter>



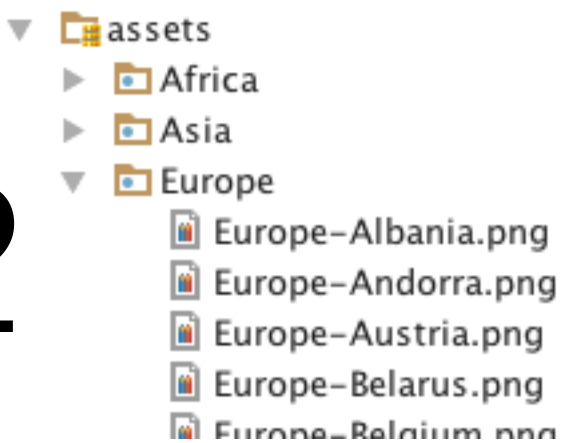
```
private void loadNextFlag() {
    
}
```



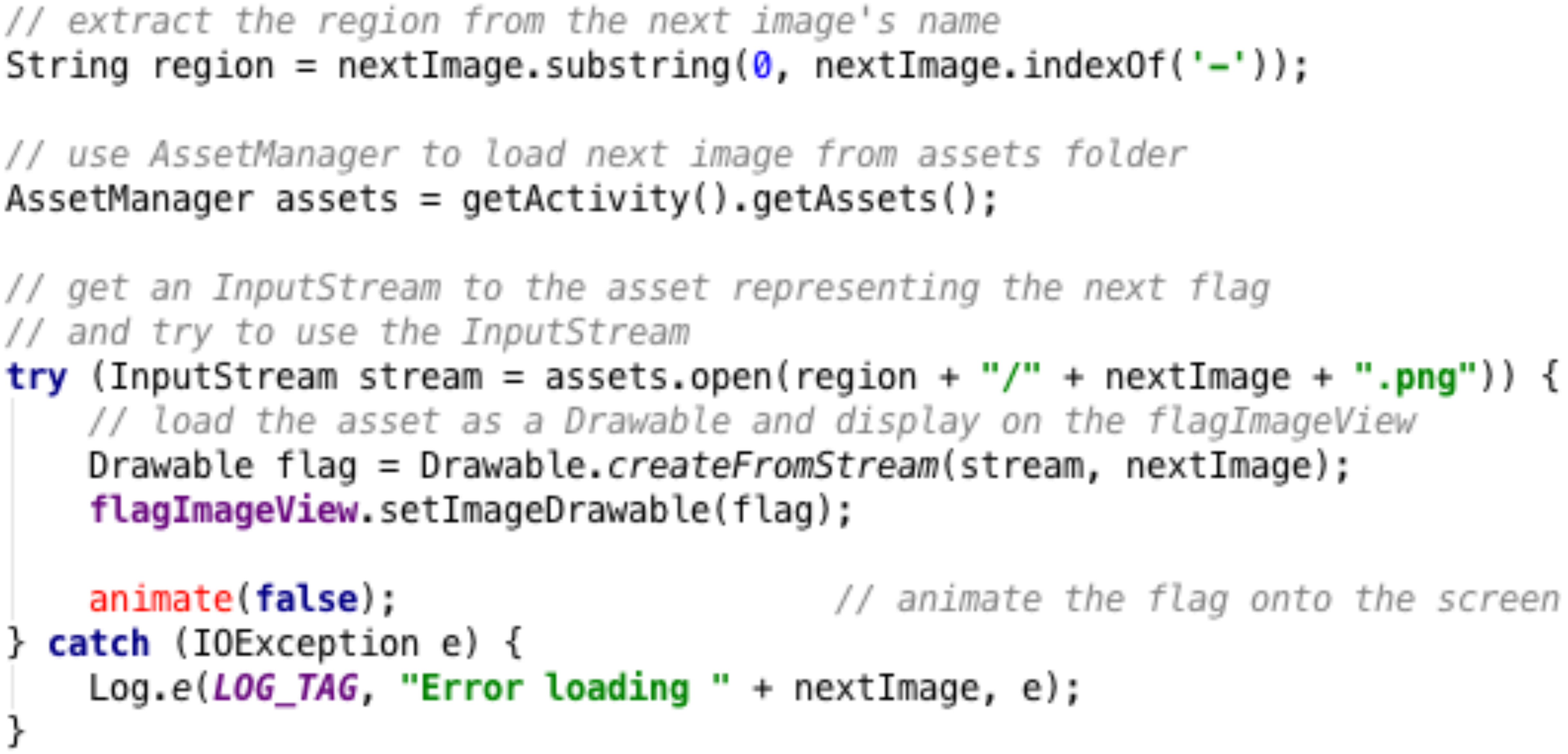
```
// after the user guesses a correct flag, load the next flag
private void loadNextFlag() {
    // get file name of the next flag and remove it from the list
    String nextImage = quizCountriesList.remove(0);
    correctAnswer = nextImage;      // update the correct answer
    answerTextView.setText("");     // clear answerTextView

    // display current question number
    questionNumberTextView.setText(getString(
            R.string.question, (correctAnswers + 1), FLAGS_IN_QUIZ));
```

Parameter für R.string.question

loadNextFlag() 2

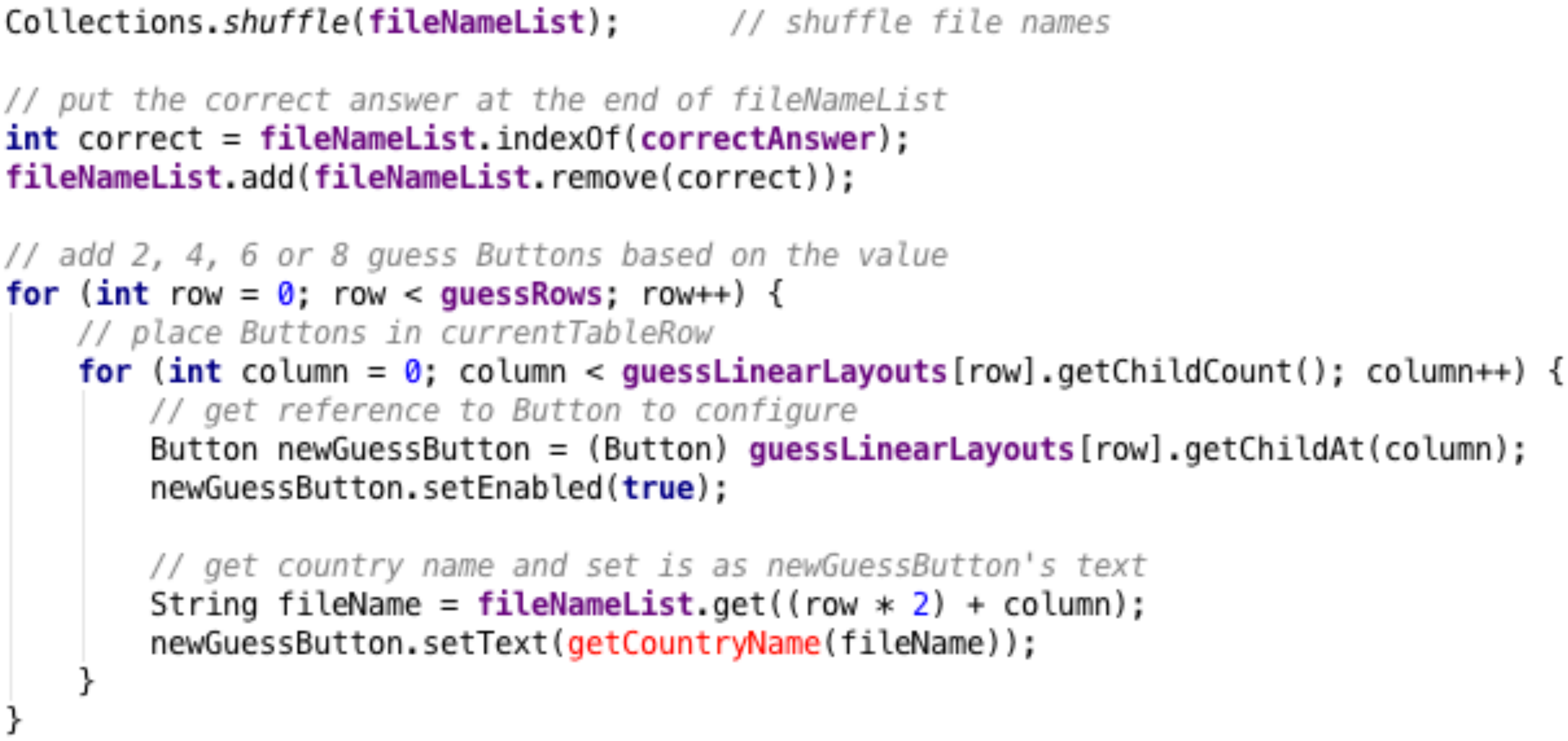


```
▼ assets
    ▶ Africa
    ▶ Asia
    ▼ Europe
        Europe-Albania.png
        Europe-Andorra.png
        Europe-Austria.png
        Europe-Belarus.png
        Europe-Belgium.png
```



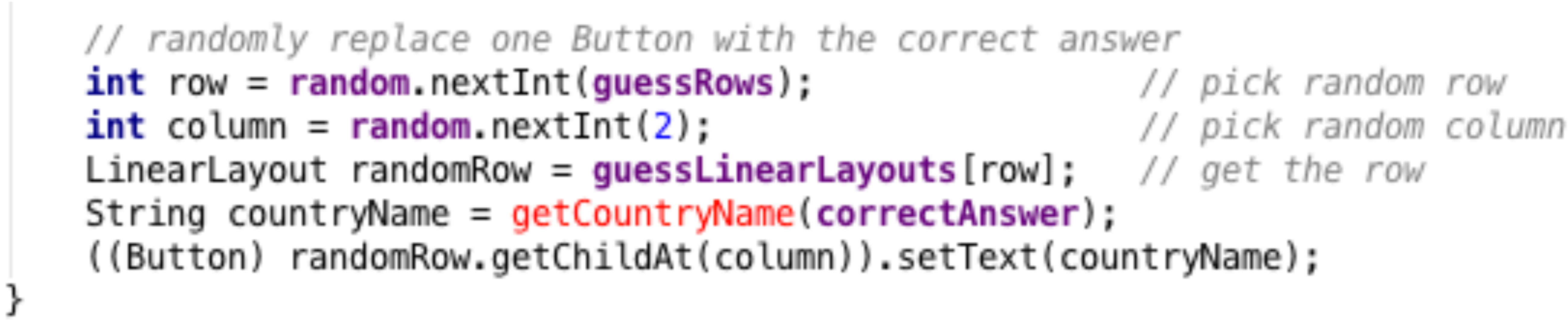
```java
// extract the region from the next image's name
String region = nextImage.substring(0, nextImage.indexOf('-'));

// use AssetManager to load next image from assets folder
AssetManager assets = getActivity().getAssets();

// get an InputStream to the asset representing the next flag
// and try to use the InputStream
try (InputStream stream = assets.open(region + "/" + nextImage + ".png")) {
    // load the asset as a Drawable and display on the flagImageView
    Drawable flag = Drawable.createFromStream(stream, nextImage);
    flagImageView.setImageDrawable(flag);

    animate(false);                    // animate the flag onto the screen
} catch (IOException e) {
    Log.e(LOG_TAG, "Error loading " + nextImage, e);
}
```

loadNextFlag() 3



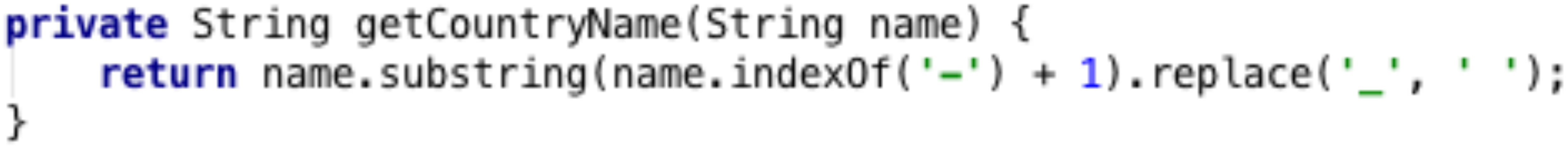
```
Collections.shuffle(fileNameList);      // shuffle file names

// put the correct answer at the end of fileNameList
int correct = fileNameList.indexOf(correctAnswer);
fileNameList.add(fileNameList.remove(correct));

// add 2, 4, 6 or 8 guess Buttons based on the value
for (int row = 0; row < guessRows; row++) {
    // place Buttons in currentTableRow
    for (int column = 0; column < guessLinearLayouts[row].getChildCount(); column++) {
        // get reference to Button to configure
        Button newGuessButton = (Button) guessLinearLayouts[row].getChildAt(column);
        newGuessButton.setEnabled(true);

        // get country name and set is as newGuessButton's text
        String fileName = fileNameList.get((row * 2) + column);
        newGuessButton.setText(getCountryName(fileName));
    }
}
```

loadNextFlag() 4



```
    // randomly replace one Button with the correct answer
    int row = random.nextInt(guessRows);                 // pick random row
    int column = random.nextInt(2);                      // pick random column
    LinearLayout randomRow = guessLinearLayouts[row];    // get the row
    String countryName = getCountryName(correctAnswer);
    ((Button) randomRow.getChildAt(column)).setText(countryName);
}
```

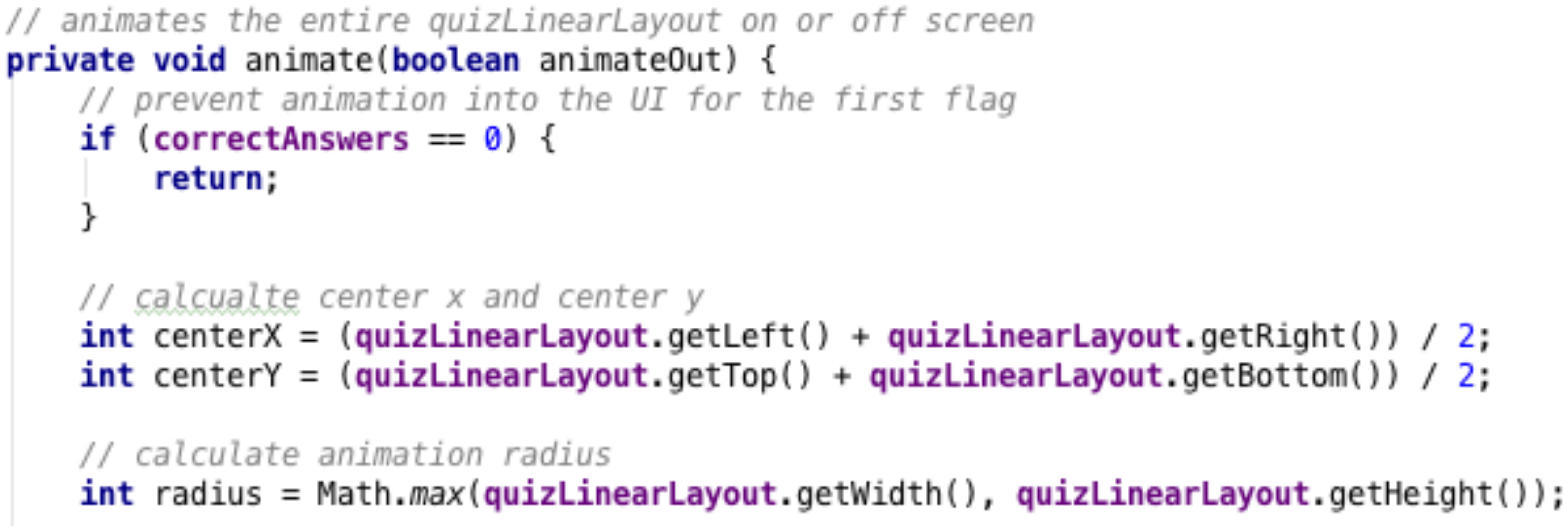
getCountryName()



```
private String getCountryName(String name) {
    return name.substring(name.indexOf('-') + 1).replace('_', ' ');
}
```

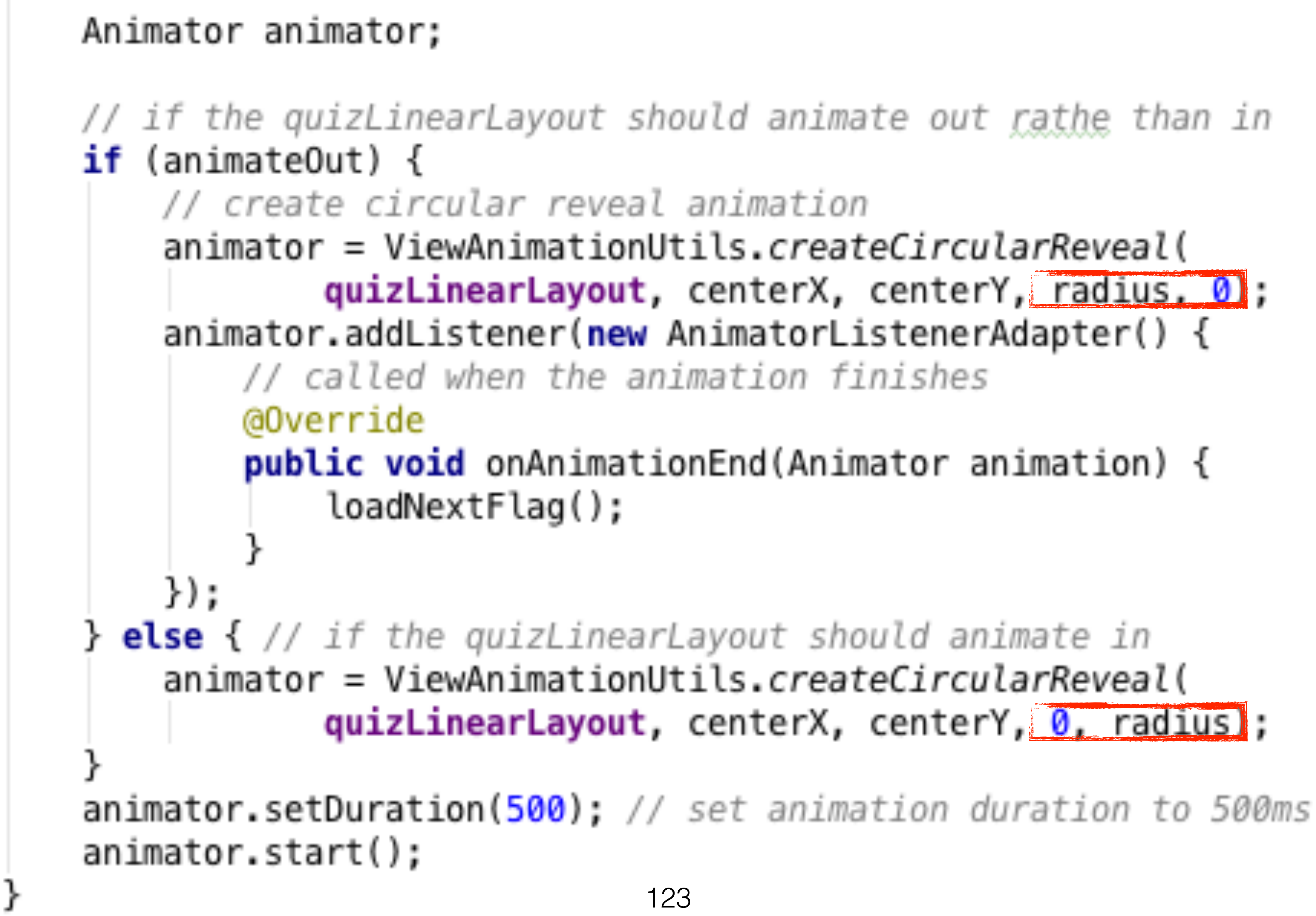
Africa-Cape_Verde —> Cape Verde

animate() 1



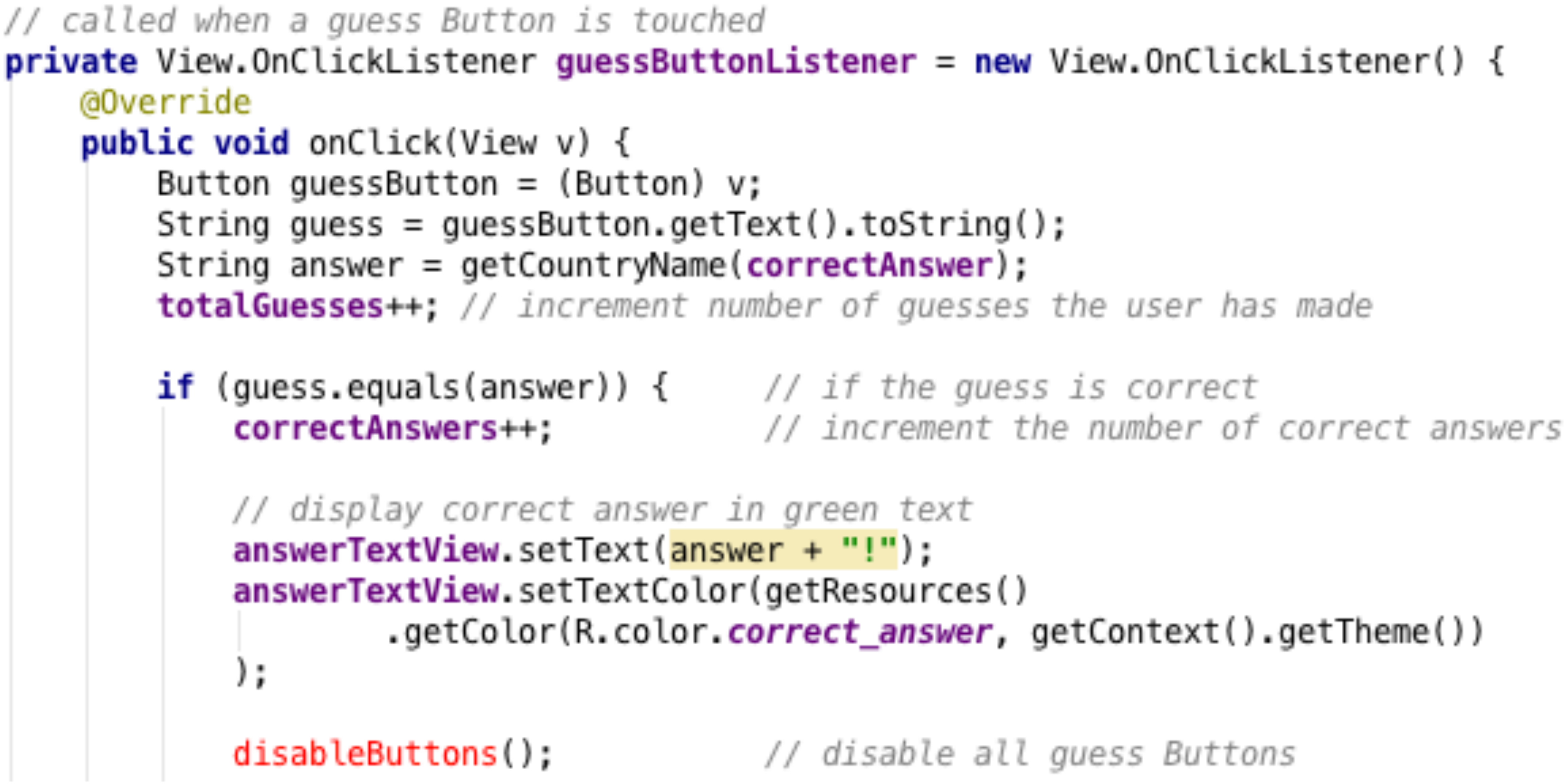
```
// animates the entire quizLinearLayout on or off screen
private void animate(boolean animateOut) {
    // prevent animation into the UI for the first flag
    if (correctAnswers == 0) {
        return;
    }

    // calcualte center x and center y
    int centerX = (quizLinearLayout.getLeft() + quizLinearLayout.getRight()) / 2;
    int centerY = (quizLinearLayout.getTop() + quizLinearLayout.getBottom()) / 2;

    // calculate animation radius
    int radius = Math.max(quizLinearLayout.getWidth(), quizLinearLayout.getHeight());
```

animate() 2



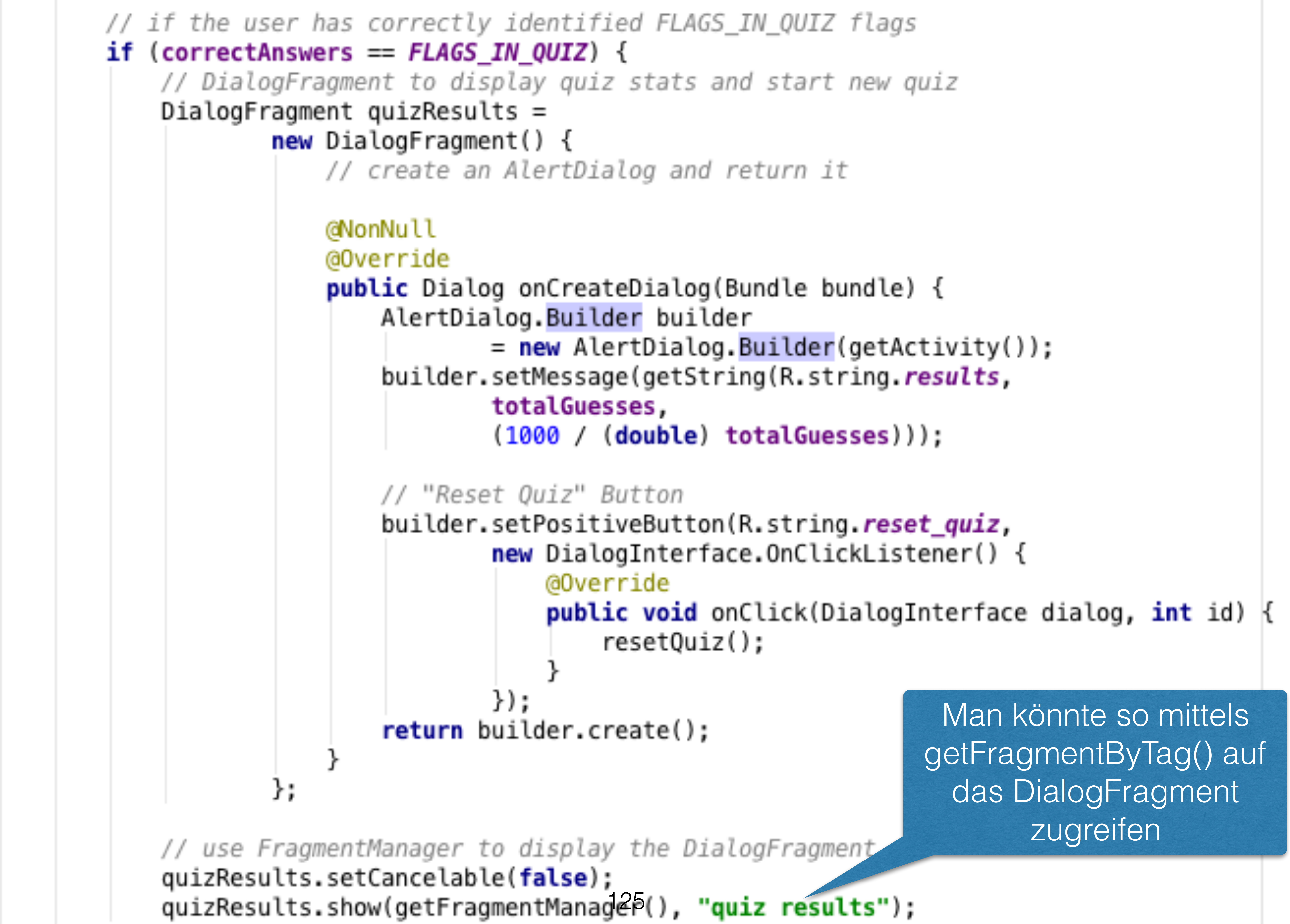
```
    Animator animator;

    // if the quizLinearLayout should animate out rathe than in
    if (animateOut) {
        // create circular reveal animation
        animator = ViewAnimationUtils.createCircularReveal(
                quizLinearLayout, centerX, centerY, radius, 0);
        animator.addListener(new AnimatorListenerAdapter() {
            // called when the animation finishes
            @Override
            public void onAnimationEnd(Animator animation) {
                loadNextFlag();
            }
        });
    } else { // if the quizLinearLayout should animate in
        animator = ViewAnimationUtils.createCircularReveal(
                quizLinearLayout, centerX, centerY, 0, radius);
    }
    animator.setDuration(500); // set animation duration to 500ms
    animator.start();
}
```

guessButtonListener 1



```
// called when a guess Button is touched
private View.OnClickListener guessButtonListener = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Button guessButton = (Button) v;
        String guess = guessButton.getText().toString();
        String answer = getCountryName(correctAnswer);
        totalGuesses++; // increment number of guesses the user has made

        if (guess.equals(answer)) {     // if the guess is correct
            correctAnswers++;           // increment the number of correct answers

            // display correct answer in green text
            answerTextView.setText(answer + "!");
            answerTextView.setTextColor(getResources()
                    .getColor(R.color.correct_answer, getContext().getTheme())
            );

            disableButtons();           // disable all guess Buttons
```

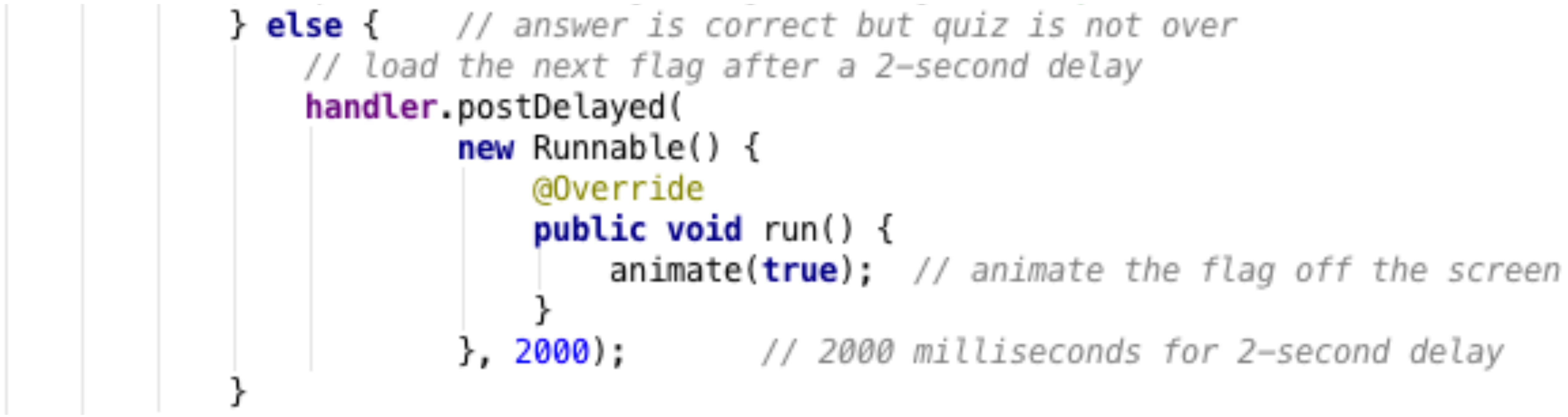
guessButtonListener 2



```
// if the user has correctly identified FLAGS_IN_QUIZ flags
if (correctAnswers == FLAGS_IN_QUIZ) {
    // DialogFragment to display quiz stats and start new quiz
    DialogFragment quizResults =
            new DialogFragment() {
                // create an AlertDialog and return it

                @NonNull
                @Override
                public Dialog onCreateDialog(Bundle bundle) {
                    AlertDialog.Builder builder
                            = new AlertDialog.Builder(getActivity());
                    builder.setMessage(getString(R.string.results,
                            totalGuesses,
                            (1000 / (double) totalGuesses)));

                    // "Reset Quiz" Button
                    builder.setPositiveButton(R.string.reset_quiz,
                            new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int id) {
                                    resetQuiz();
                                }
                            });
                    return builder.create();
                }
            };

    // use FragmentManager to display the DialogFragment
    quizResults.setCancelable(false);
    quizResults.show(getFragmentManager(), "quiz results");
```

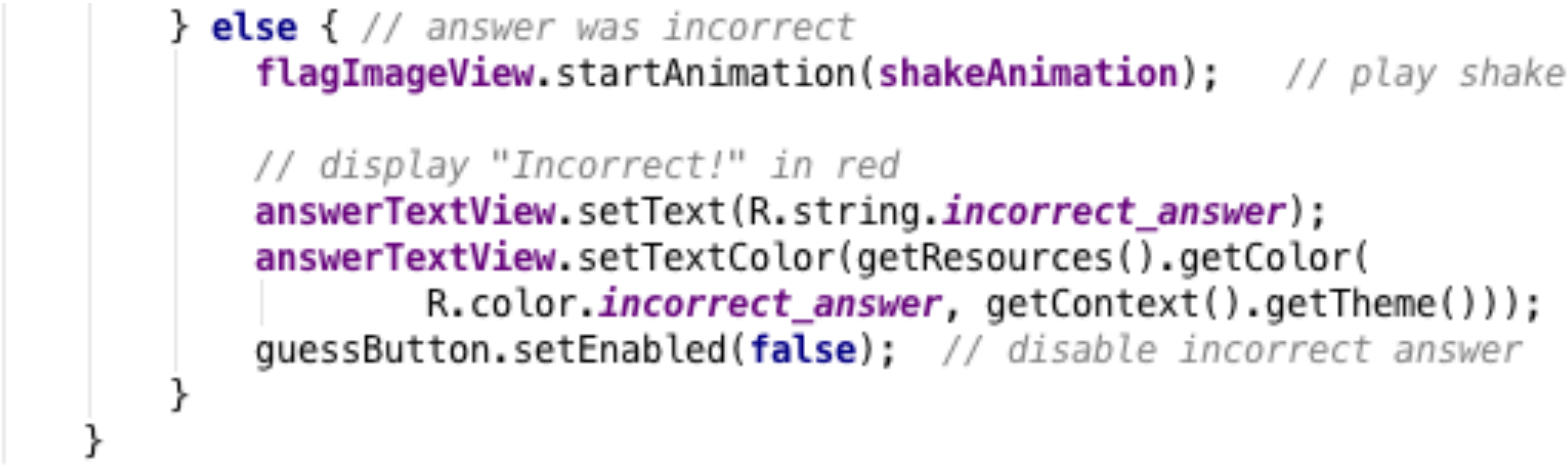
Man könnte so mittels getFragmentByTag() auf das DialogFragment zugreifen

guessButtonListener 3



```
} else {    // answer is correct but quiz is not over
    // load the next flag after a 2-second delay
    handler.postDelayed(
            new Runnable() {
                @Override
                public void run() {
                    animate(true);  // animate the flag off the screen
                }
            }, 2000);       // 2000 milliseconds for 2-second delay
}
```

guessButtonListener 4

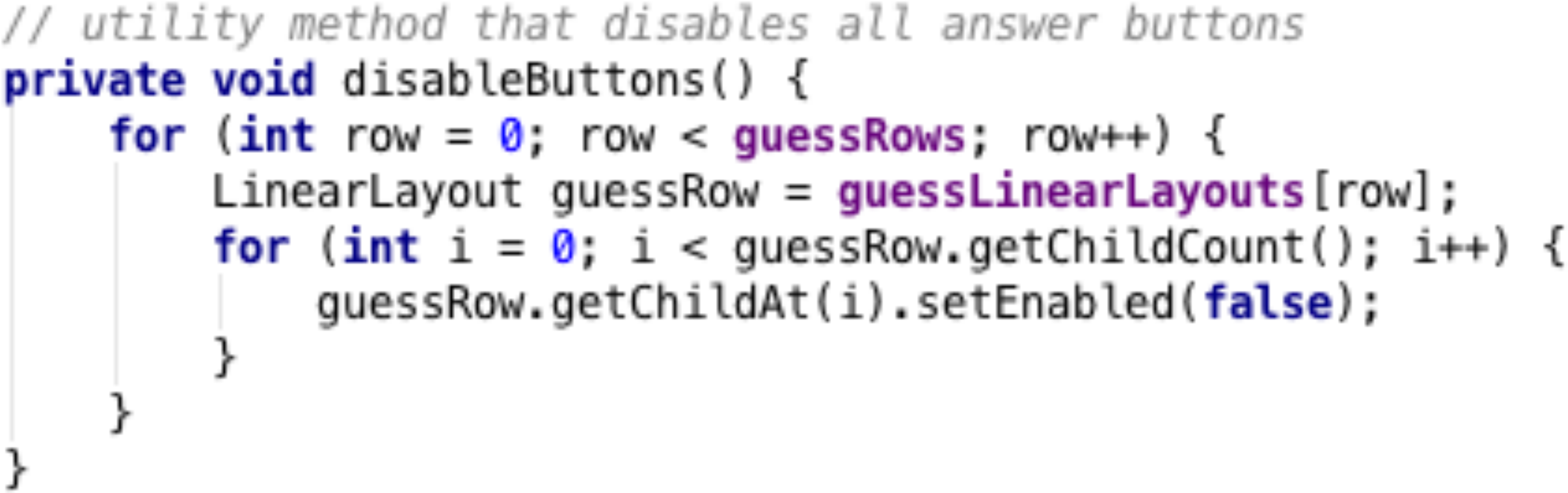


```
    } else { // answer was incorrect
        flagImageView.startAnimation(shakeAnimation);   // play shake

        // display "Incorrect!" in red
        answerTextView.setText(R.string.incorrect_answer);
        answerTextView.setTextColor(getResources().getColor(
                R.color.incorrect_answer, getContext().getTheme()));
        guessButton.setEnabled(false);  // disable incorrect answer
    }
}
```

Look-and-Feel Observation 4.4

You can set an `AlertDialog`*'s title (which appears above the dialog's message) with* `AlertDialog.Builder` *method* `setTitle`*. According to the Android design guidelines for dialogs (*`http://developer.android.com/design/building-blocks/dialogs.html`*), most dialogs do not need titles. A dialog should display a title for "a high-risk operation involving potential loss of data, connectivity, extra charges, and so on." Also, dialogs that display lists of options use the title to specify the dialog's purpose.*

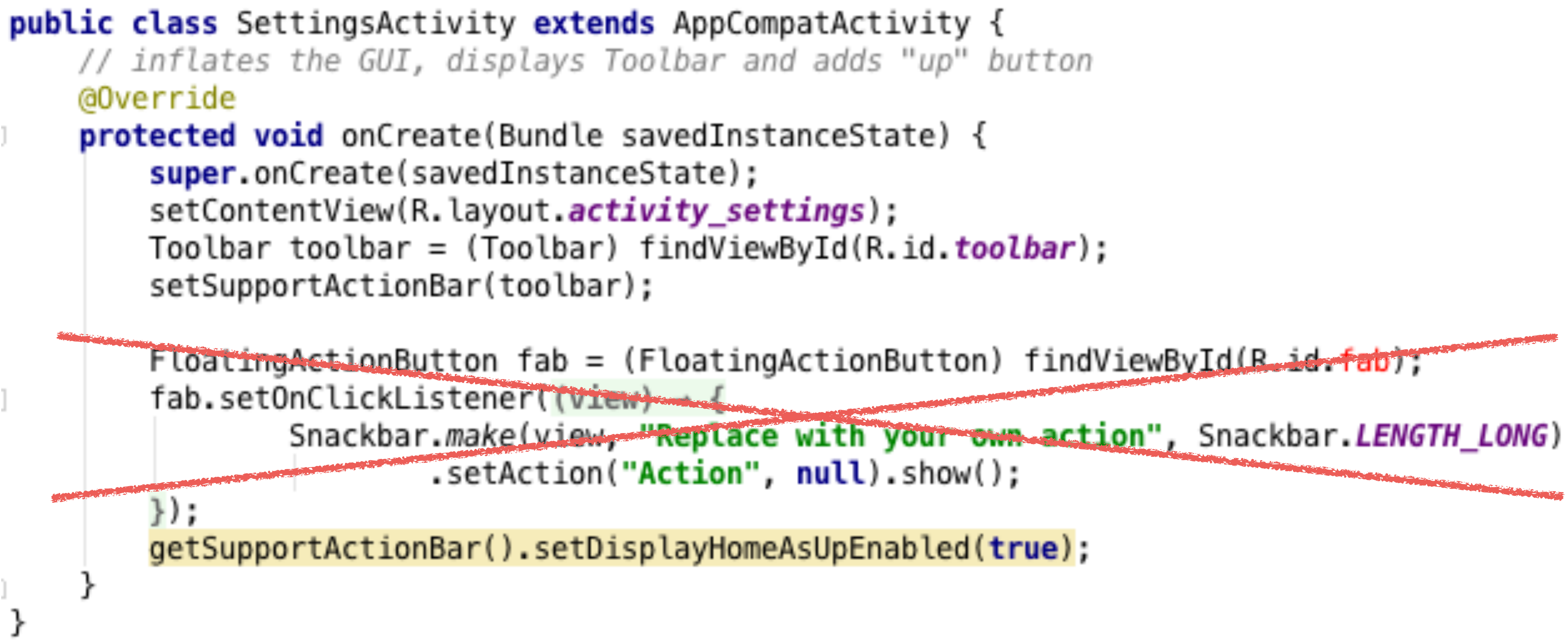
disableButtons()



```java
// utility method that disables all answer buttons
private void disableButtons() {
    for (int row = 0; row < guessRows; row++) {
        LinearLayout guessRow = guessLinearLayouts[row];
        for (int i = 0; i < guessRow.getChildCount(); i++) {
            guessRow.getChildAt(i).setEnabled(false);
        }
    }
}
```

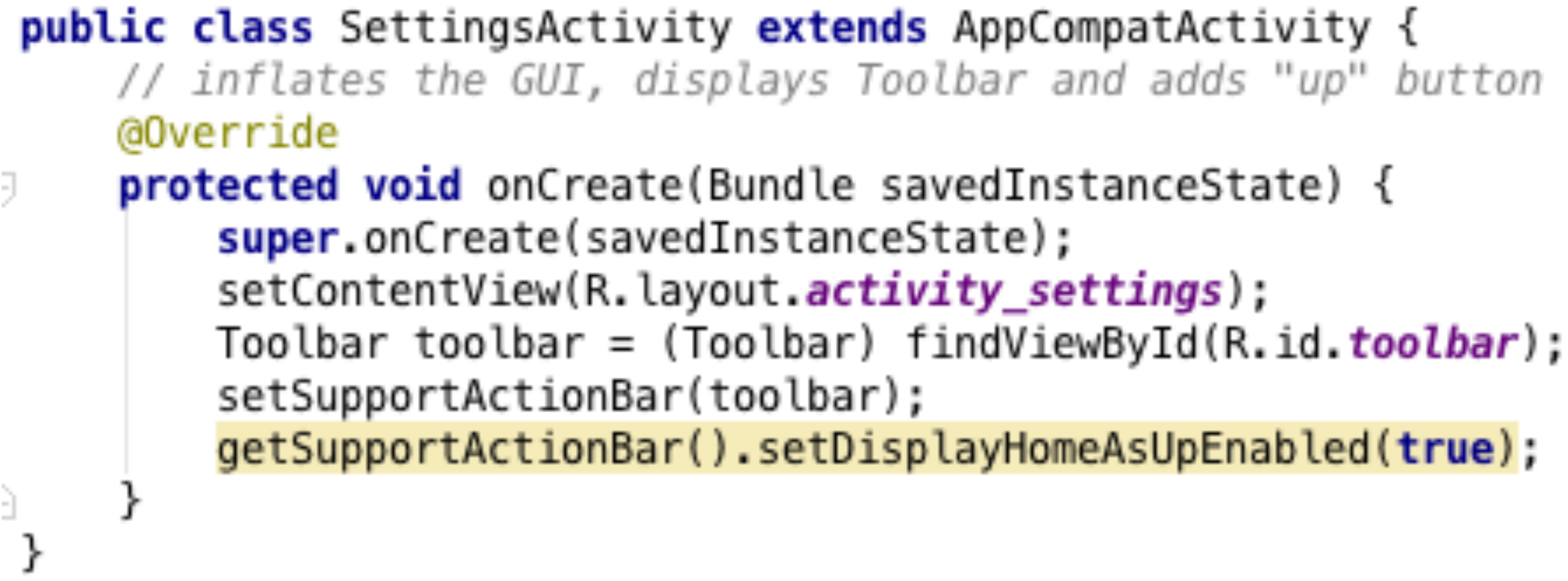
SettingsActivity.java

SettingsActivity.java



```java
public class SettingsActivity extends AppCompatActivity {
    // inflates the GUI, displays Toolbar and adds "up" button
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) -> {
            Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
        });
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
}
```

SettingsActivity.java

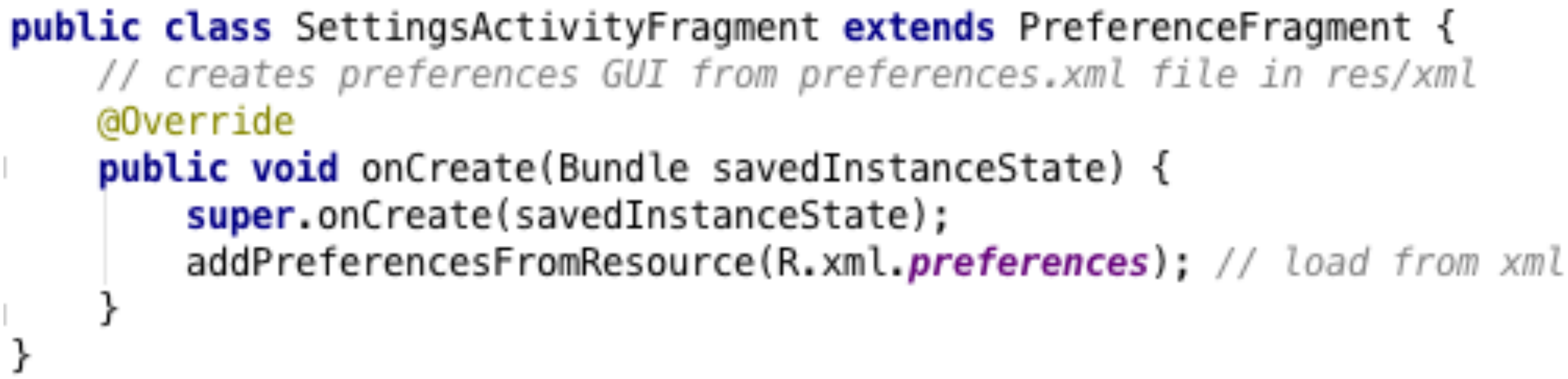


```java
public class SettingsActivity extends AppCompatActivity {
    // inflates the GUI, displays Toolbar and adds "up" button
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
}
```

menu_settings.xml kann gelöscht werden

SettingsActivityFragment .java

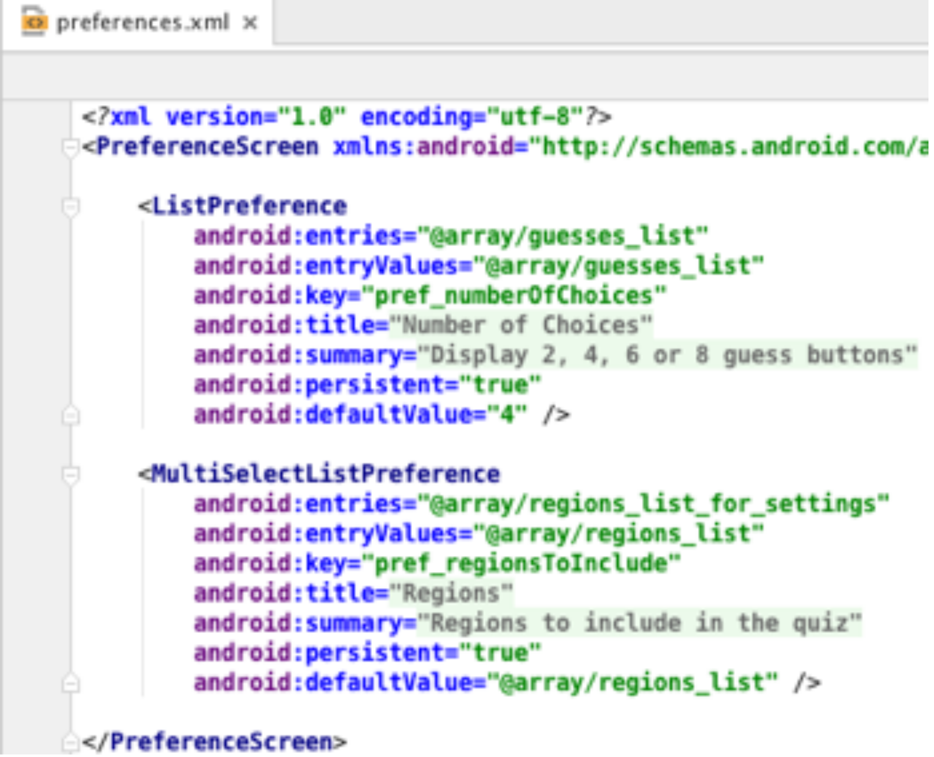
SettingsActivityFragment



```
public class SettingsActivityFragment extends PreferenceFragment {
    // creates preferences GUI from preferences.xml file in res/xml
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.preferences); // load from xml
    }
}
```

PreferencesFragment

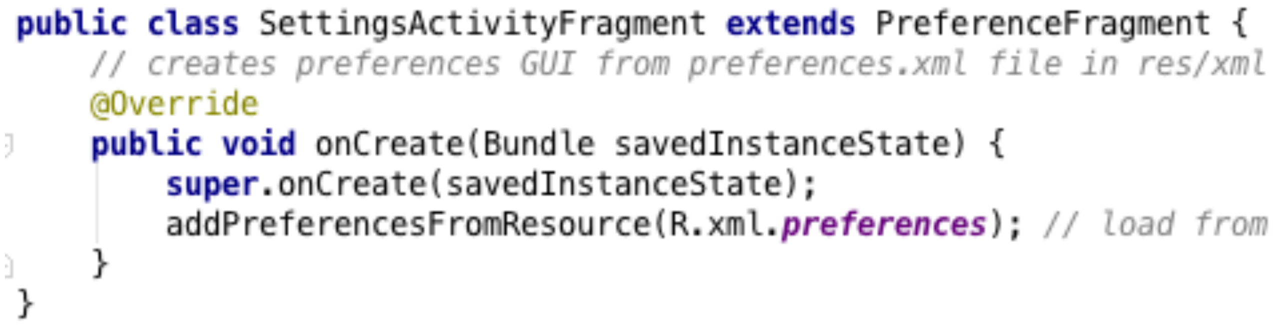
- Man kann die Settings in xml designen und mittels PreferencesFragment’s ausführen



preferences.xml ×

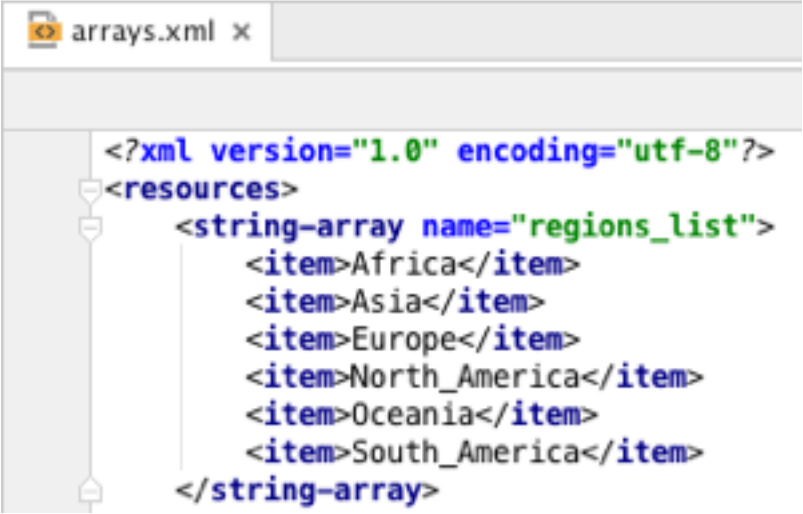
```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/a

    <ListPreference
        android:entries="@array/guesses_list"
        android:entryValues="@array/guesses_list"
        android:key="pref_numberOfChoices"
        android:title="Number of Choices"
        android:summary="Display 2, 4, 6 or 8 guess buttons"
        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="Regions"
        android:summary="Regions to include in the quiz"
        android:persistent="true"
        android:defaultValue="@array/regions_list" />

</PreferenceScreen>
```

+



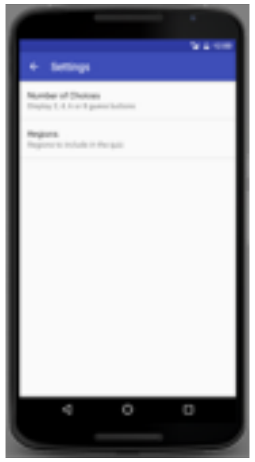
```
public class SettingsActivityFragment extends PreferenceFragment {
    // creates preferences GUI from preferences.xml file in res/xml
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.preferences); // load from
    }
}
```

+

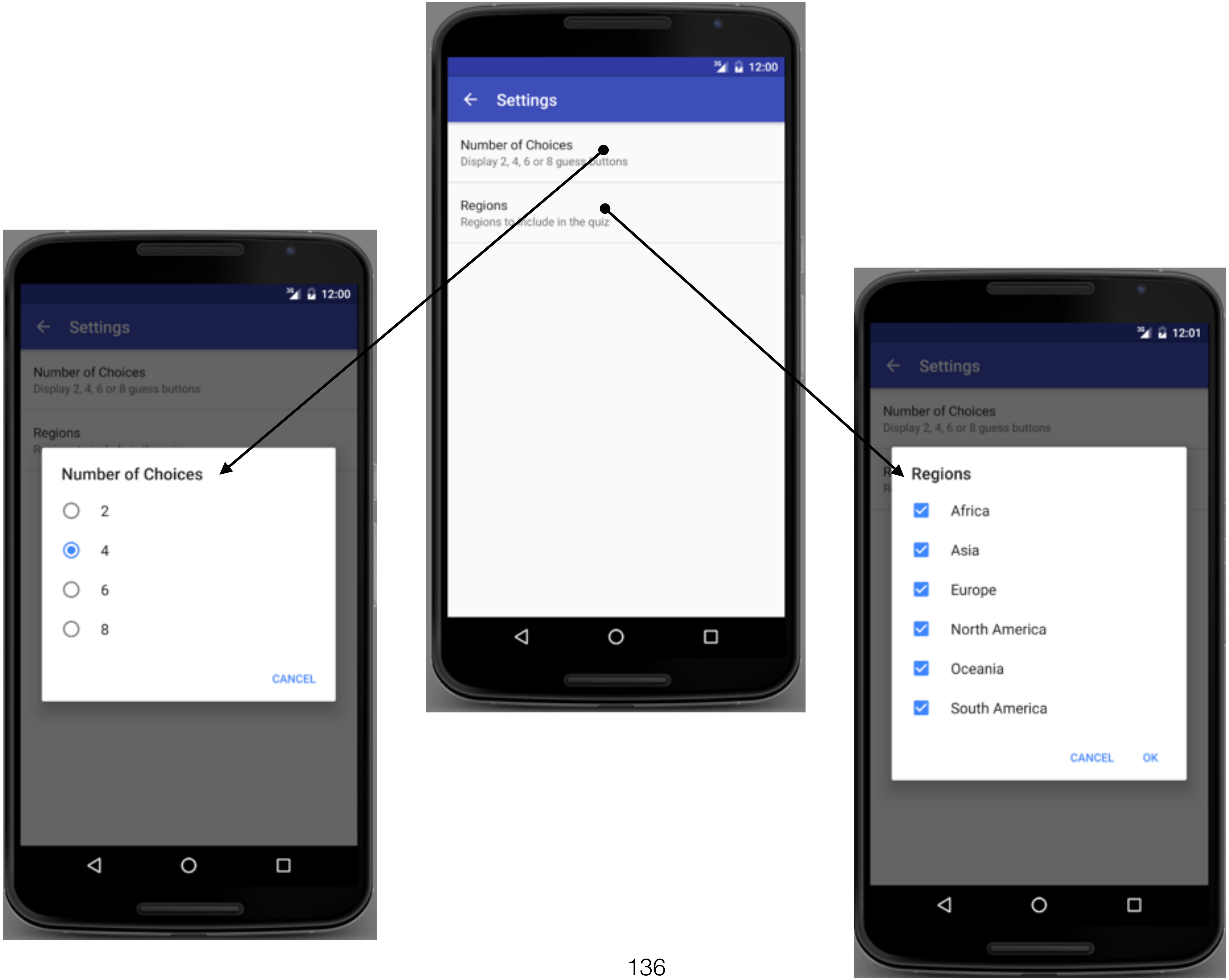


```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="regions_list">
        <item>Africa</item>
        <item>Asia</item>
        <item>Europe</item>
        <item>North_America</item>
        <item>Oceania</item>
        <item>South_America</item>
    </string-array>
```

=



Settings
Number of Choices
Regions



12:00
Settings

Number of Choices
Display 2, 4, 6 or 8 guess buttons

Regions
Regions to include in the quiz

12:00
Settings

Number of Choices

○ 2
● 4
○ 6
○ 8

CANCEL

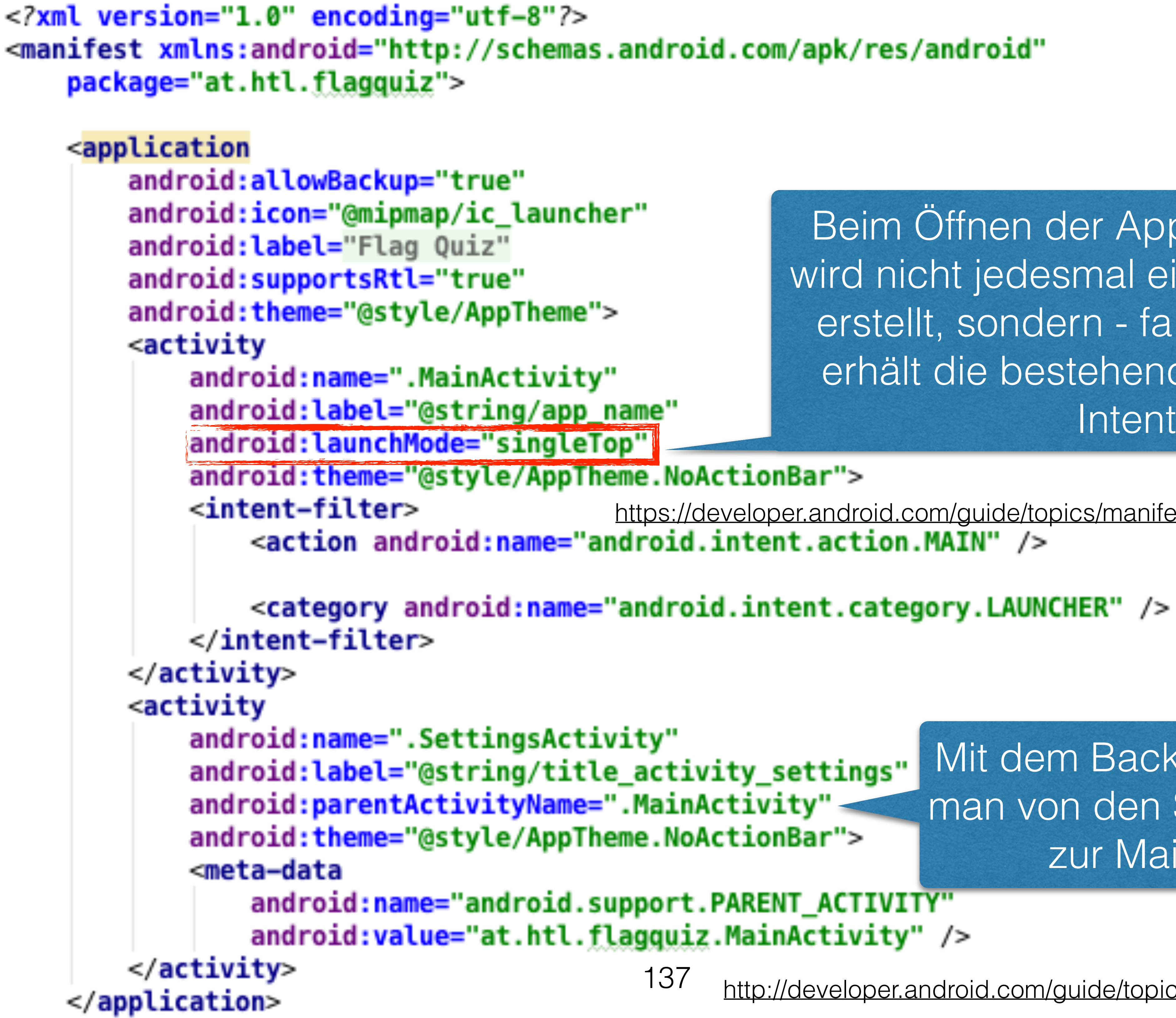
12:01
Settings

Regions

☑ Africa
☑ Asia
☑ Europe
☑ North America
☑ Oceania
☑ South America

CANCEL
OK

AndroidManifest.xml



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="at.htl.flagquiz">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Flag Quiz"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:launchMode="singleTop"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".SettingsActivity"
            android:label="@string/title_activity_settings"
            android:parentActivityName=".MainActivity"
            android:theme="@style/AppTheme.NoActionBar">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="at.htl.flagquiz.MainActivity" />
        </activity>
    </application>
```

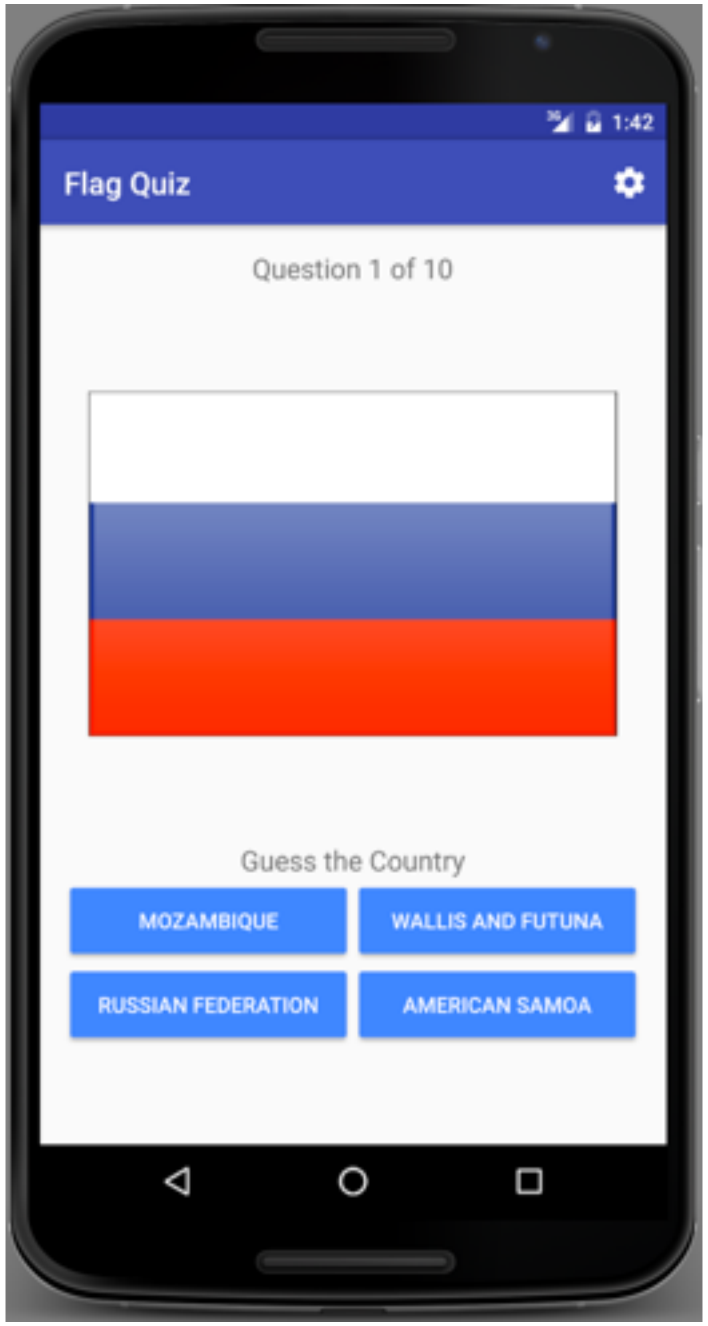
Beim Öffnen der App (neuer Intent) wird nicht jedesmal eine neue Instanz erstellt, sondern - falls vorhanden - erhält die bestehende Instanz den Intent

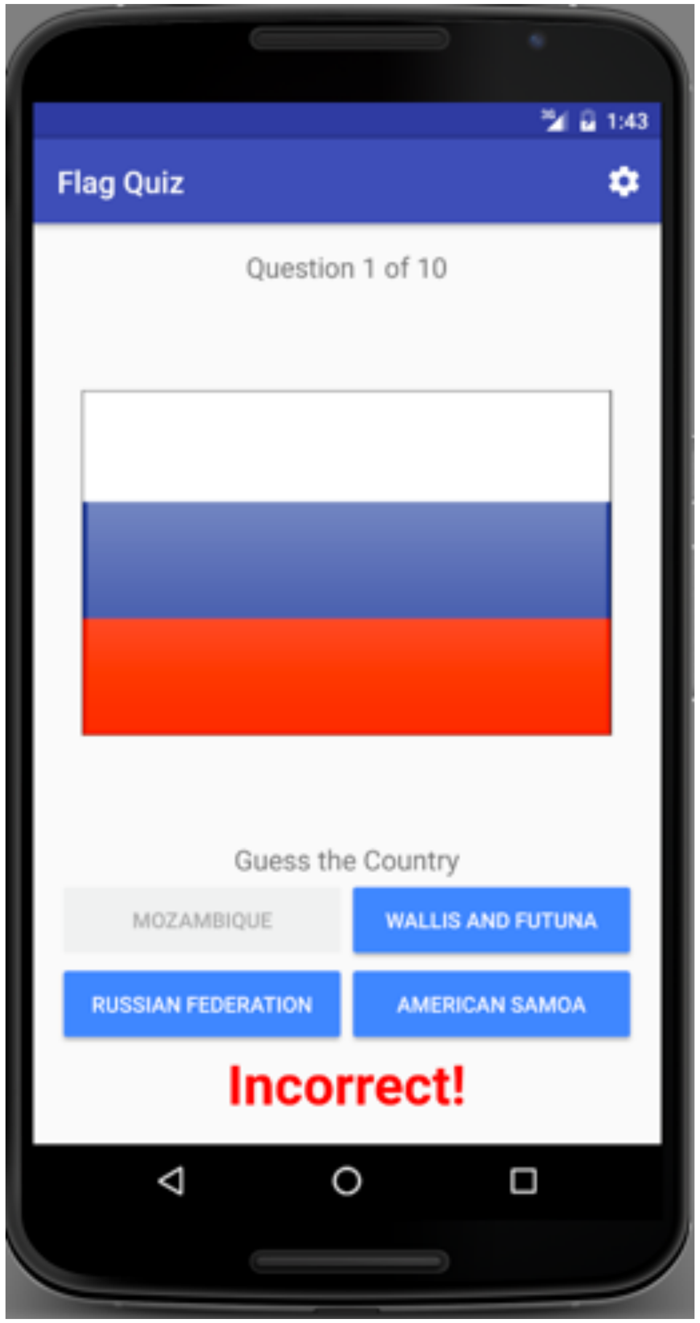
https://developer.android.com/guide/topics/manifest/activity-element.html#lmode

Mit dem Back-Button kommt man von den Settings zurück zur MainActivity

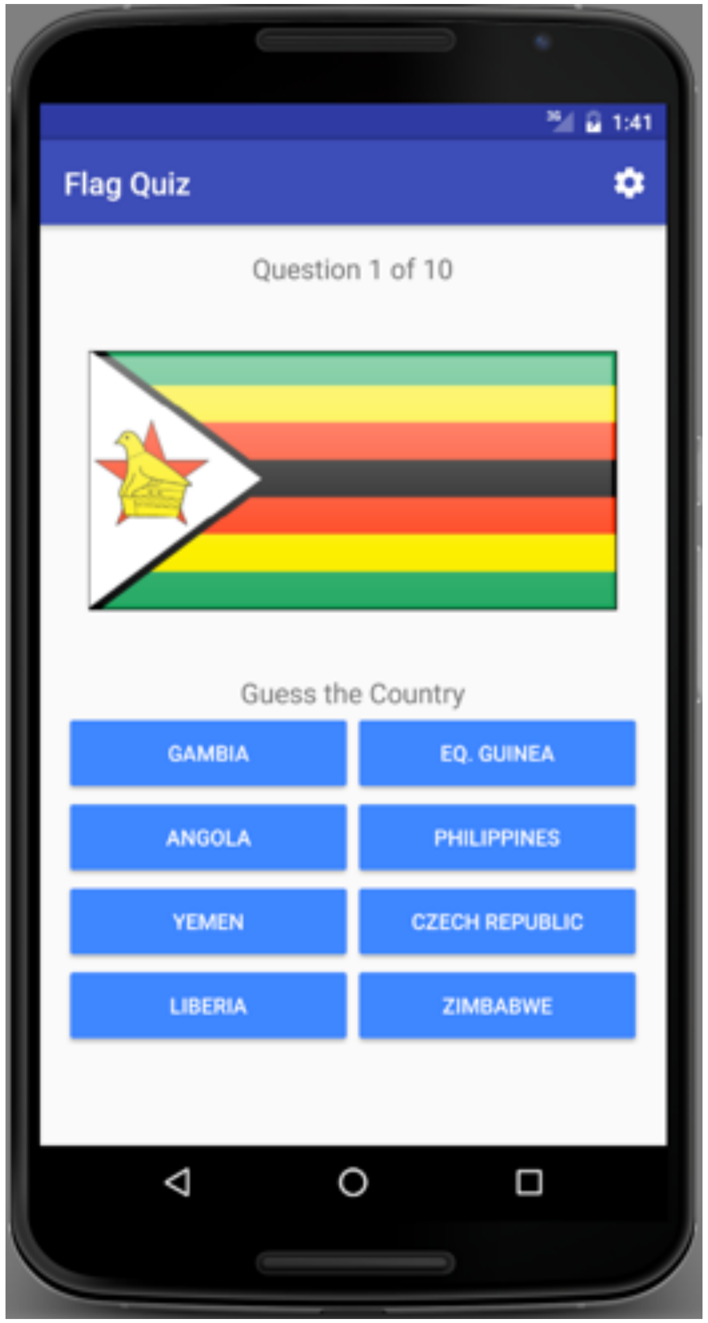
http://developer.android.com/guide/topics/manifest/manifest-intro.html

Let’s start …


1:42
Flag Quiz
Question 1 of 10
Guess the Country
MOZAMBIQUE
WALLIS AND FUTUNA
RUSSIAN FEDERATION
AMERICAN SAMOA

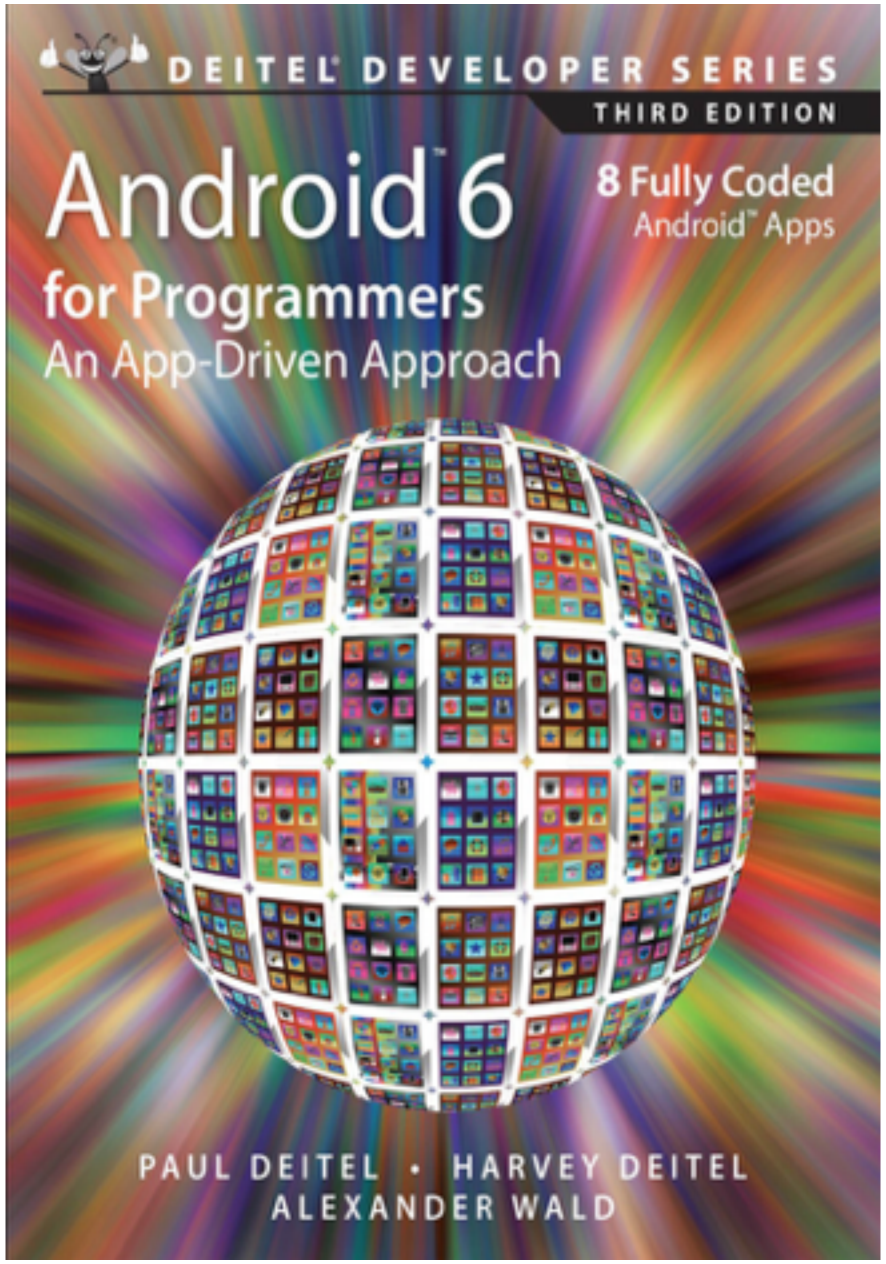

1:43
Flag Quiz
Question 1 of 10
Guess the Country
MOZAMBIQUE
WALLIS AND FUTUNA
RUSSIAN FEDERATION
AMERICAN SAMOA

Incorrect!


1:41
Flag Quiz
Question 1 of 10
Guess the Country
GAMBIA
EQ. GUINEA
ANGOLA
PHILIPPINES
YEMEN
CZECH REPUBLIC
LIBERIA
ZIMBABWE

Noch Fragen?

Quelle


DEITEL DEVELOPER SERIES
THIRD EDITION
Android 6
for Programmers
An App-Driven Approach
8 Fully Coded
Android Apps
PAUL DEITEL • HARVEY DEITEL
ALEXANDER WALD

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