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

Teil 2

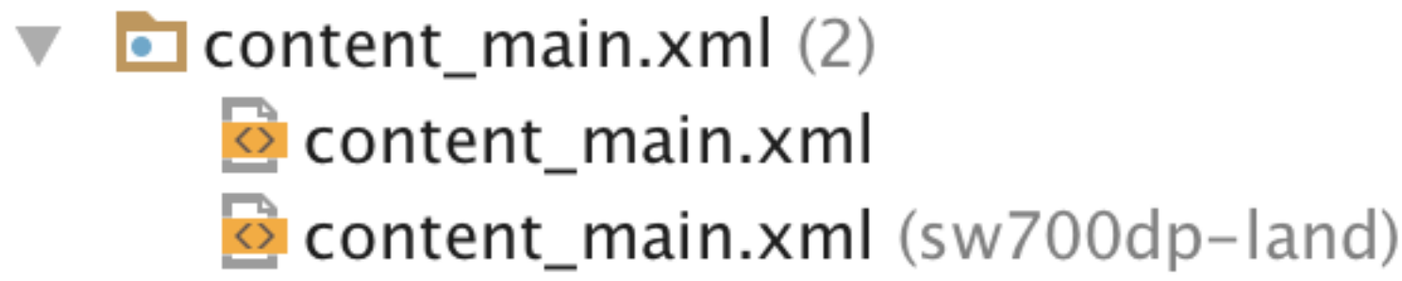
content_main.xml

- 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

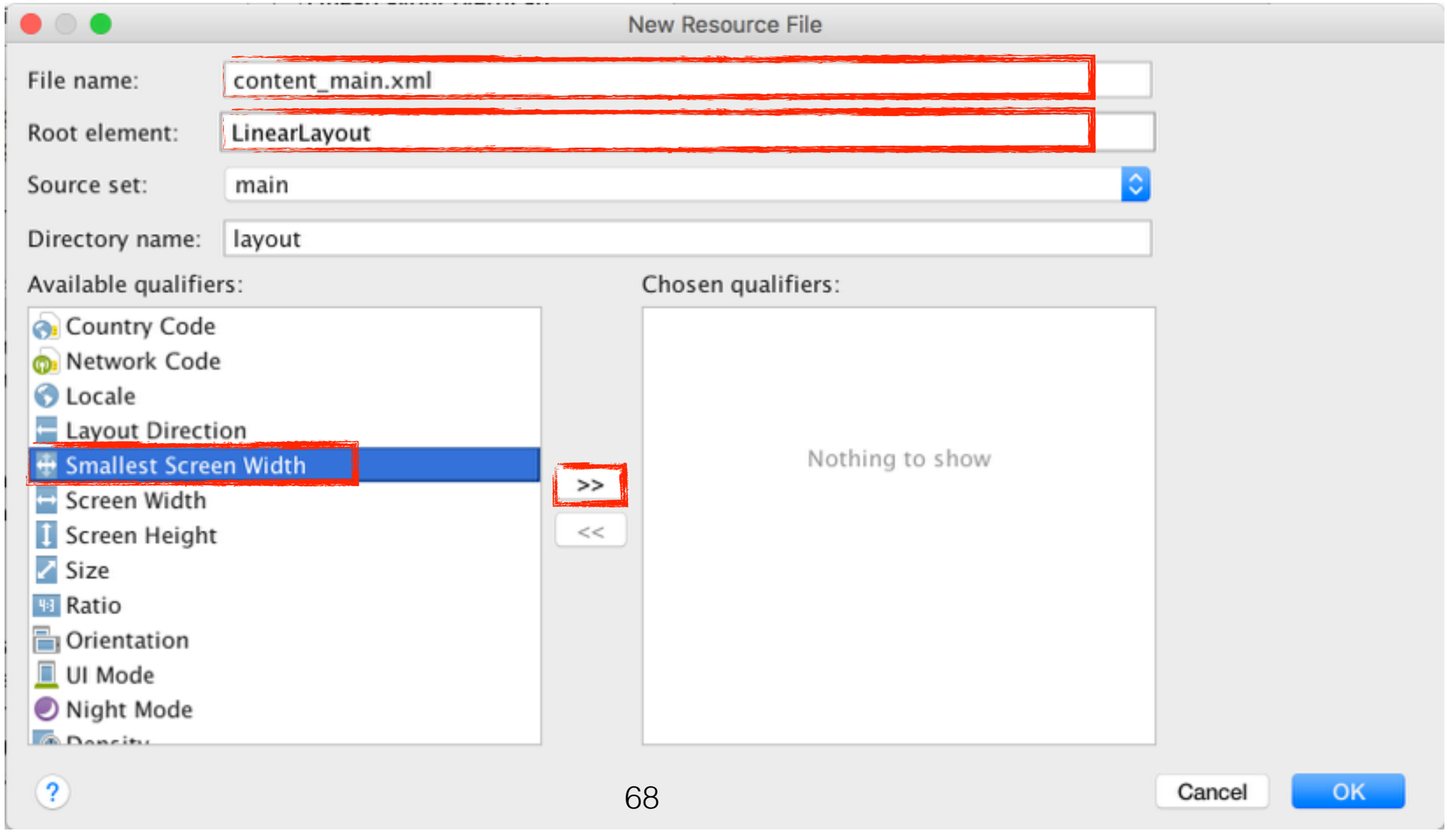
- Dieses Layout wird nur bei einer minimalen Screen Width von 700dp (sw) sowie im Landscape-Mode (land) verwendet —> layout-sw700dp-land

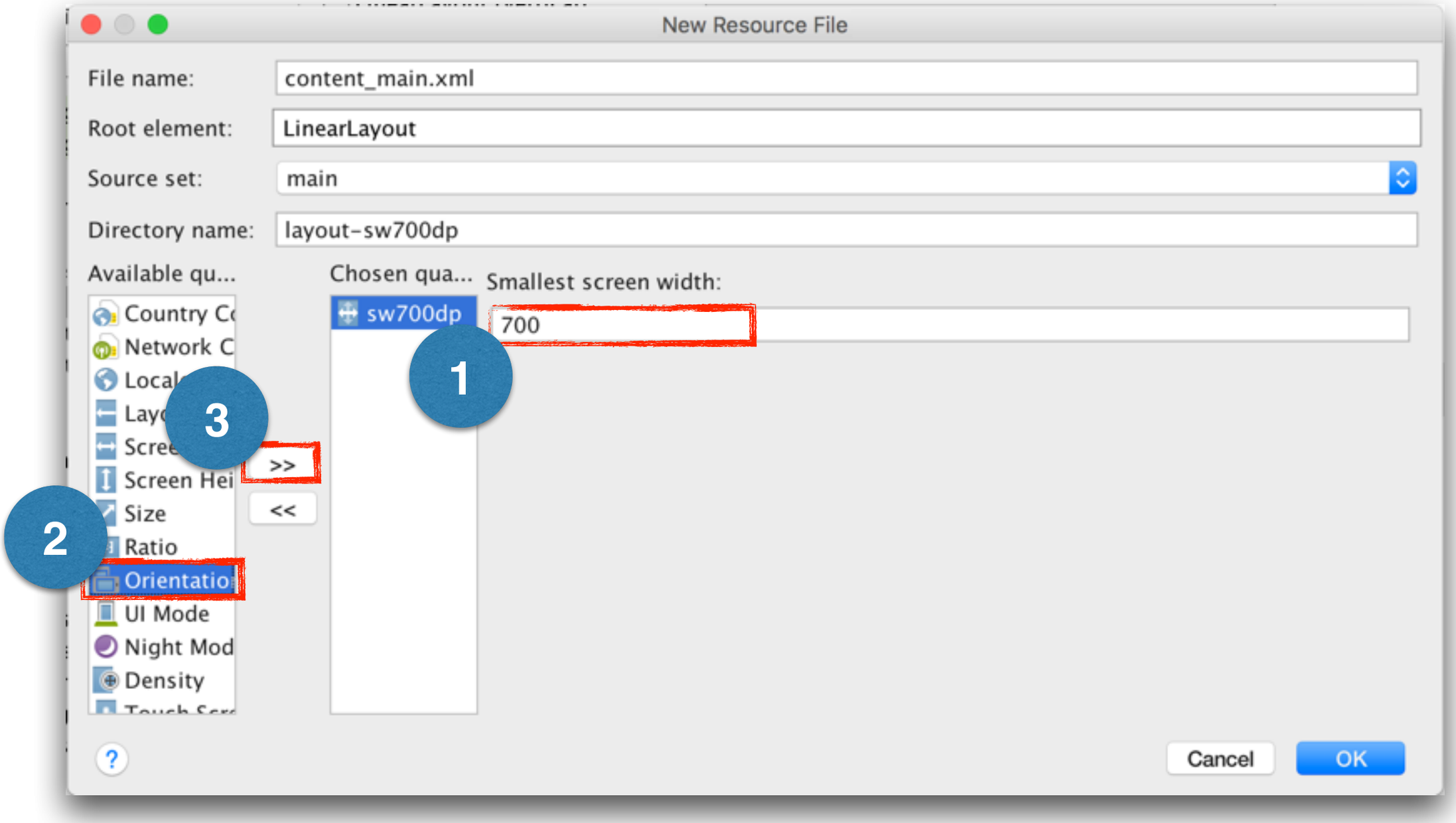
- Das Layout ohne Qualifier in den Klammern ist das Standard-Layout.

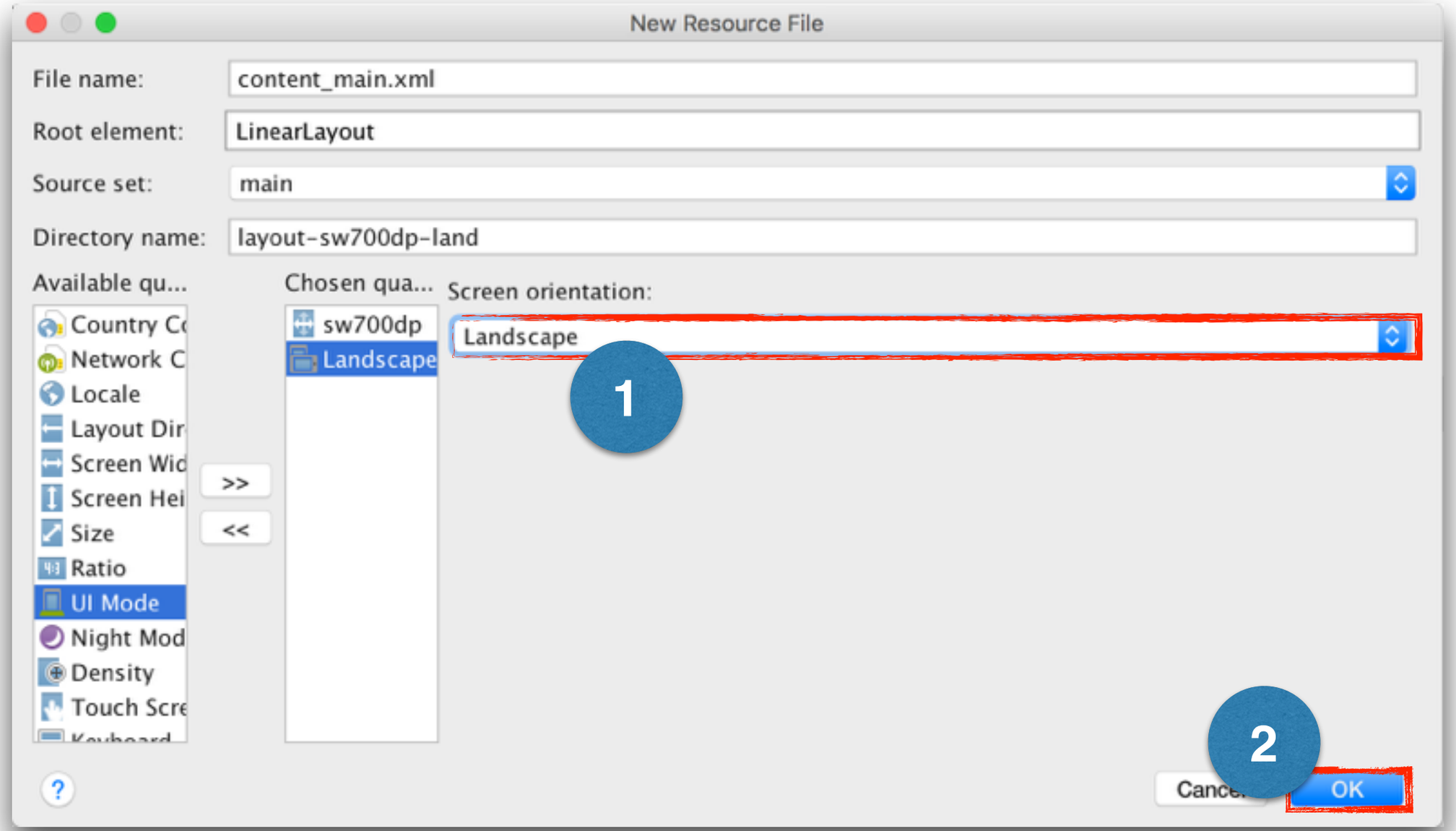


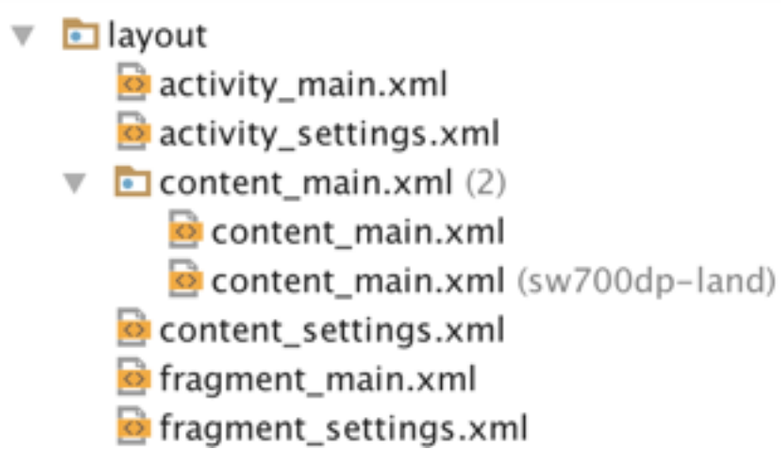
content_main.xml
Vorbereitung für Landscape

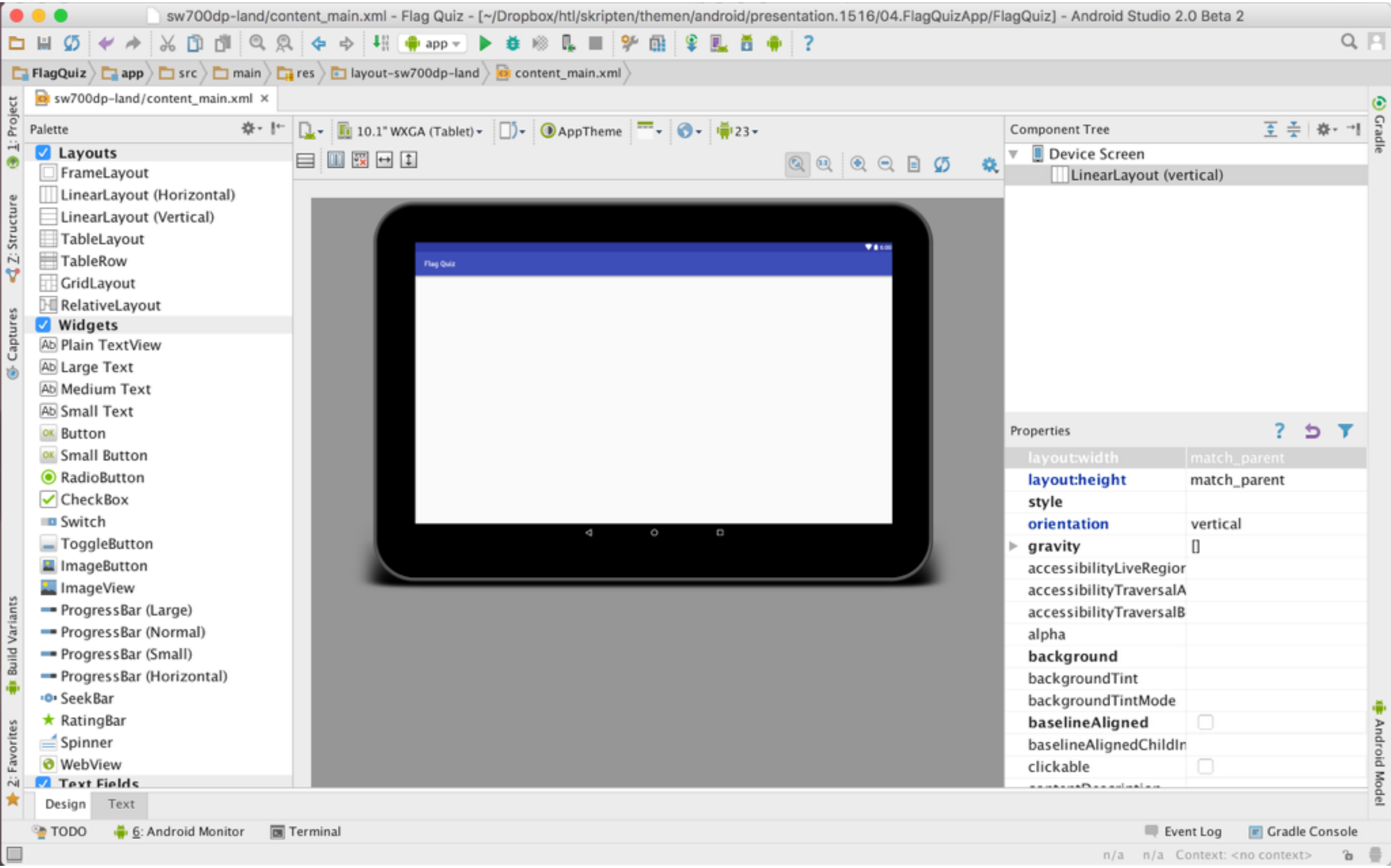
res/layout - New - Layout Resource File

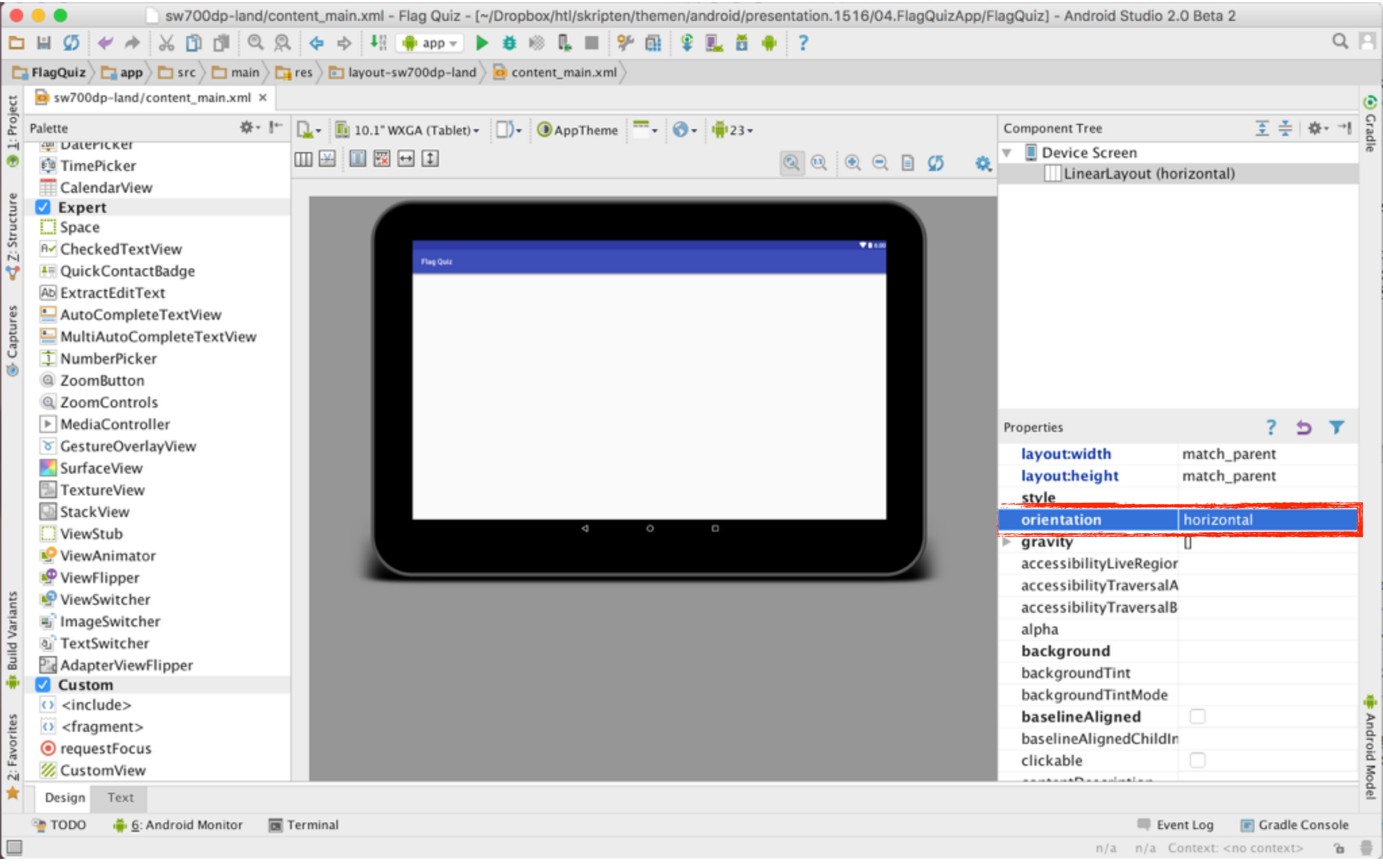



New Resource File
File name:
content_main.xml
Root element:
LinearLayout
Source set:
main
Directory name:
layout-sw700dp
Available qu...
Chosen qua...
Smallest screen width:
sw700dp
700
Country C
Network C
Local
Lay
Scree
Screen Hei
Size
Ratio
Orientatio
UI Mode
Night Mod
Density
>>
<<
1
2
3
Cancel
OK

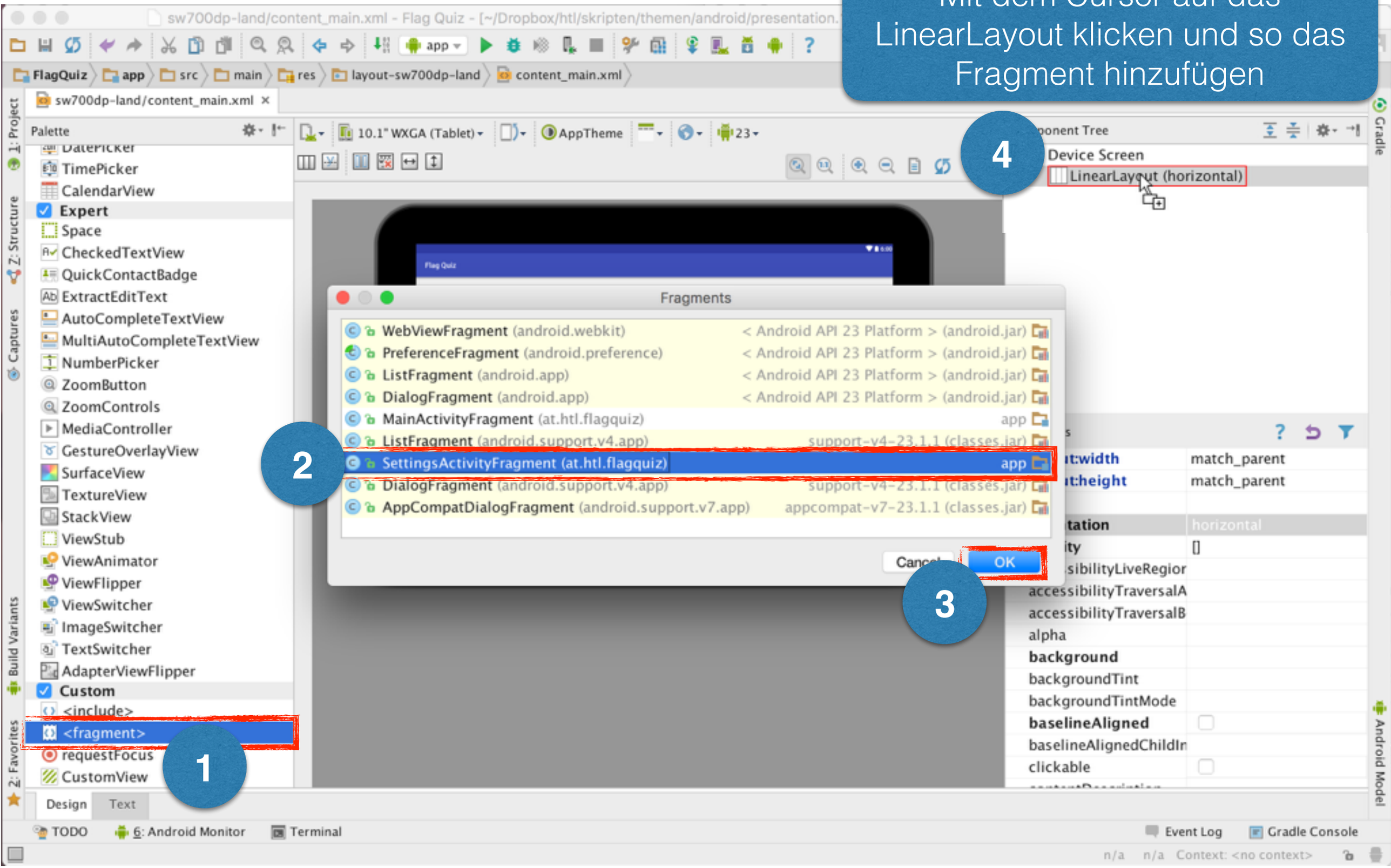

New Resource File
File name:
content_main.xml
Root element:
LinearLayout
Source set:
main
Directory name:
layout-sw700dp-land
Available qu...
Chosen qua...
Screen orientation:
Country Co
Network C
Locale
Layout Dir
Screen Wid
Screen Hei
Size
Ratio
UI Mode
Night Mod
Density
Touch Scre
sw700dp
Landscape
>>
<<
Landscape
1
2
Cancel
OK

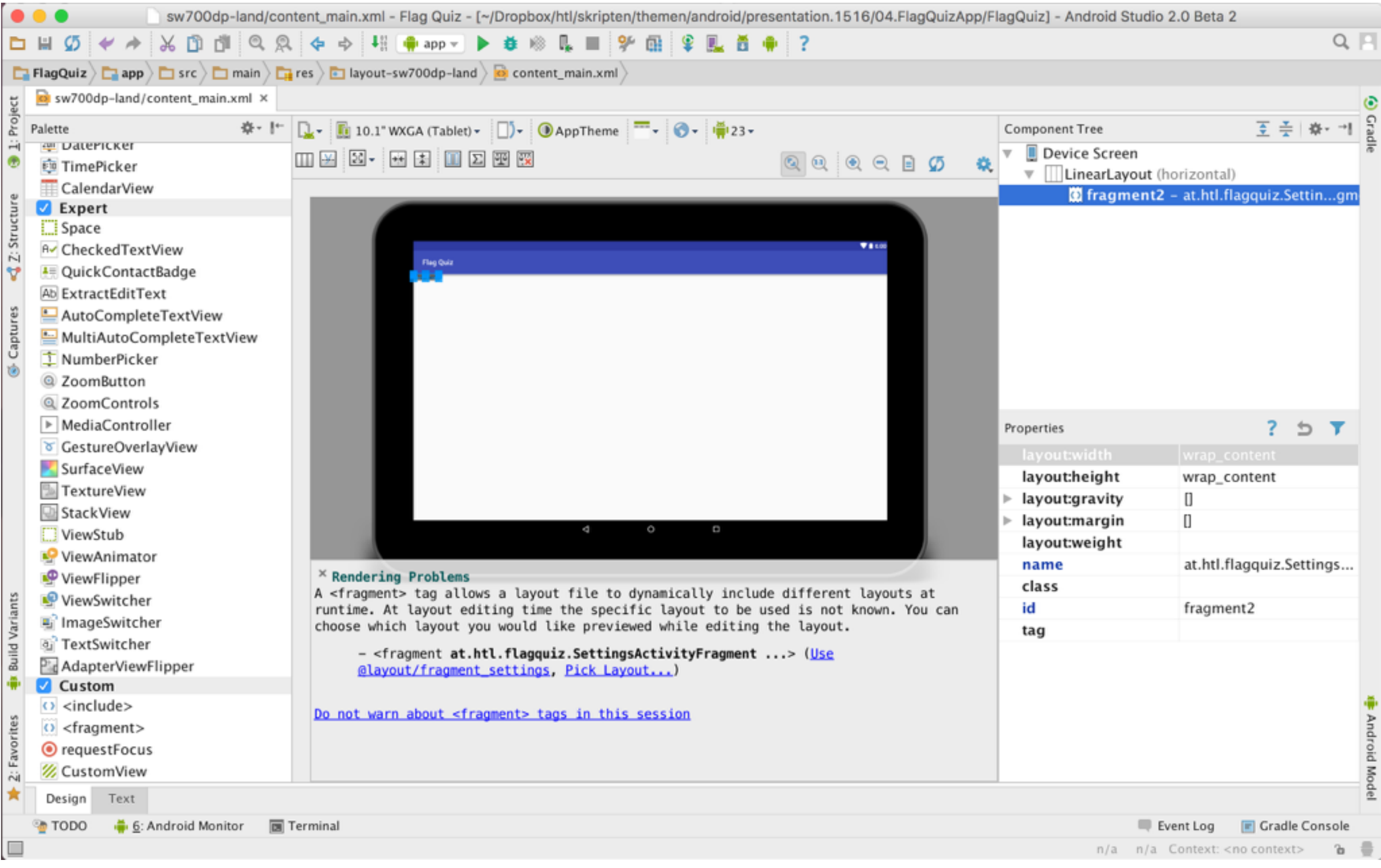

layout
activity_main.xml
activity_settings.xml
content_main.xml (2)
content_main.xml
content_main.xml (sw700dp-land)
content_settings.xml
fragment_main.xml
fragment_settings.xml

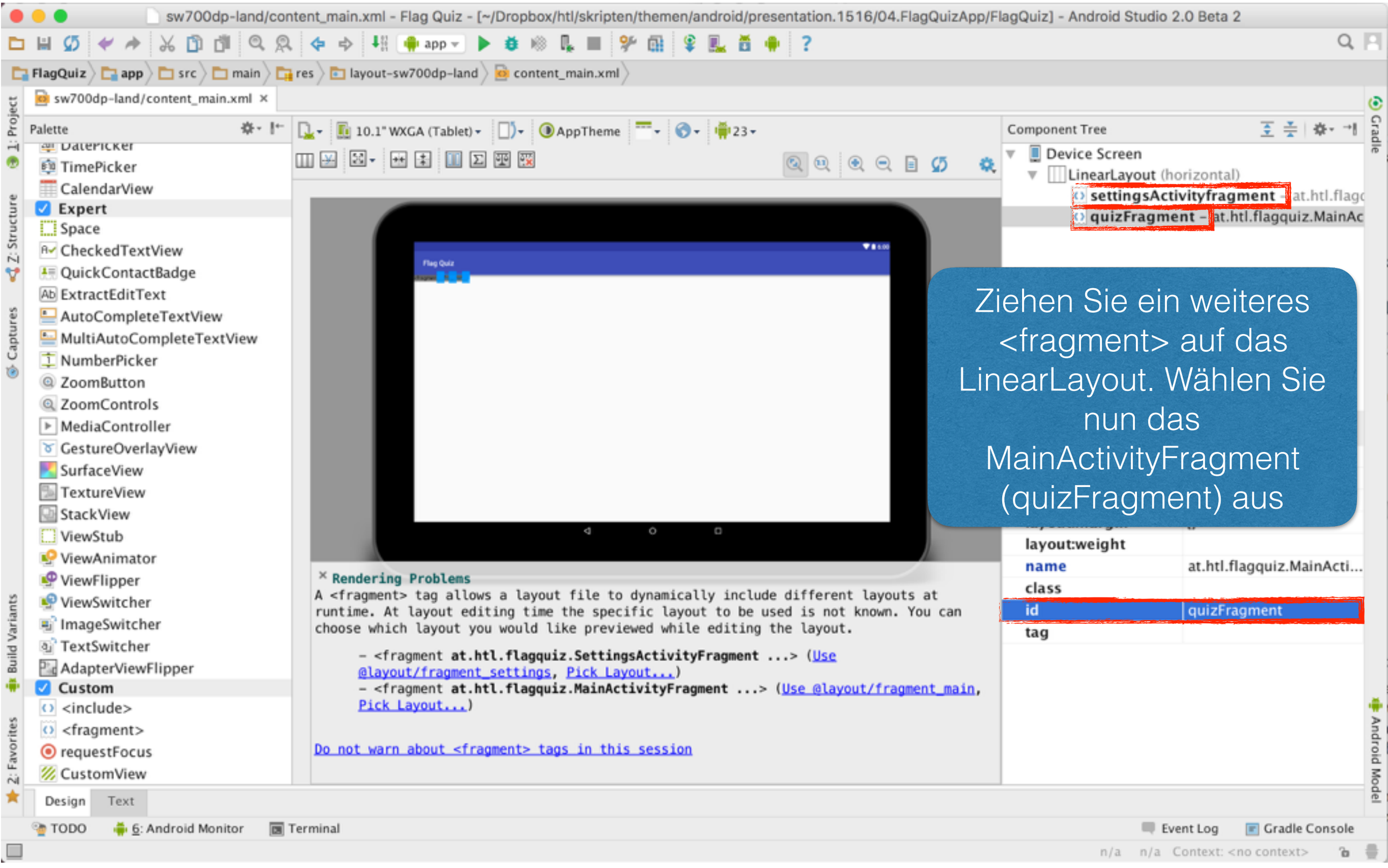

sw700dp-land/content_main.xml - Flag Quiz - [~/Dropbox/htl/skripten/themen/android/presentation.1516/04.FlagQuizApp/FlagQuiz] - Android Studio 2.0 Beta 2
app
FlagQuiz
app
src
main
res
layout-sw700dp-land
content_main.xml
sw700dp-land/content_main.xml
Palette
Layouts
FrameLayout
LinearLayout (Horizontal)
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
layout:width
match_parent
layout:height
match_parent
style
orientation
vertical
gravity
[]
accessibilityLiveRegion
accessibilityTraversalA
accessibilityTraversalB
alpha
background
backgroundTint
backgroundTintMode
baselineAligned
baselineAlignedChildIn
clickable
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


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
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)
Properties
layout:width
match_parent
layout:height
match_parent
style
orientation
horizontal
gravity
accessibilityLiveRegion
accessibilityTraversalA
accessibilityTraversalB
alpha
background
backgroundTint
backgroundTintMode
baselineAligned
baselineAlignedChildIn
clickable
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




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
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
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
Component Tree
Device Screen
LinearLayout (horizontal)
fragment2 - at.htl.flagquiz.Settin...gm
Properties
layout:width
wrap_content
layout:height
wrap_content
layout:gravity
[]
layout:margin
[]
layout:weight
name
at.htl.flagquiz.Settings...
class
id
fragment2
tag
Design
Text
TODO
6: Android Monitor
Terminal
Event Log
Gradle Console
n/a
n/a
Context: <no context>
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Gradle
Android Model


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
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
Component Tree
Device Screen
LinearLayout (horizontal)
settingsActivityfragment
quizFragment
layout:weight
name
at.htl.flagquiz.MainActi...
class
id
quizFragment
tag
Design
Text
TODO
6: Android Monitor
Terminal
Event Log
Gradle Console
Context: <no context>
Ziehen Sie ein weiteres <fragment> auf das LinearLayout. Wählen Sie nun das MainActivityFragment (quizFragment) aus

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

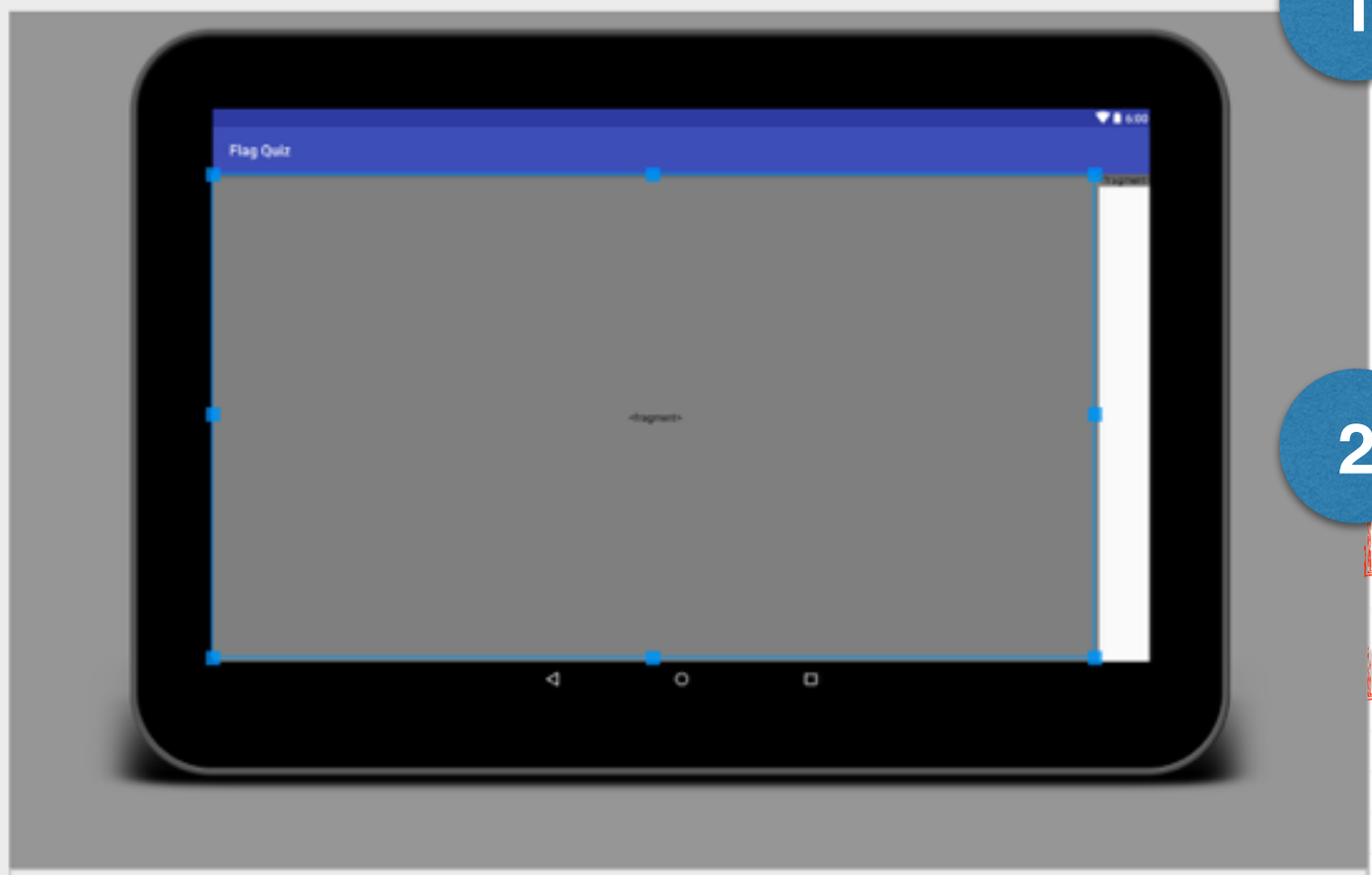
FlagQuiz › app › src › main › res › layout-sw700dp-land › content_main.xml

sw700dp-land/content_main.xml

Palette

- 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

10.1" WXGA (Tablet) · AppTheme · 23



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...)

Component Tree

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

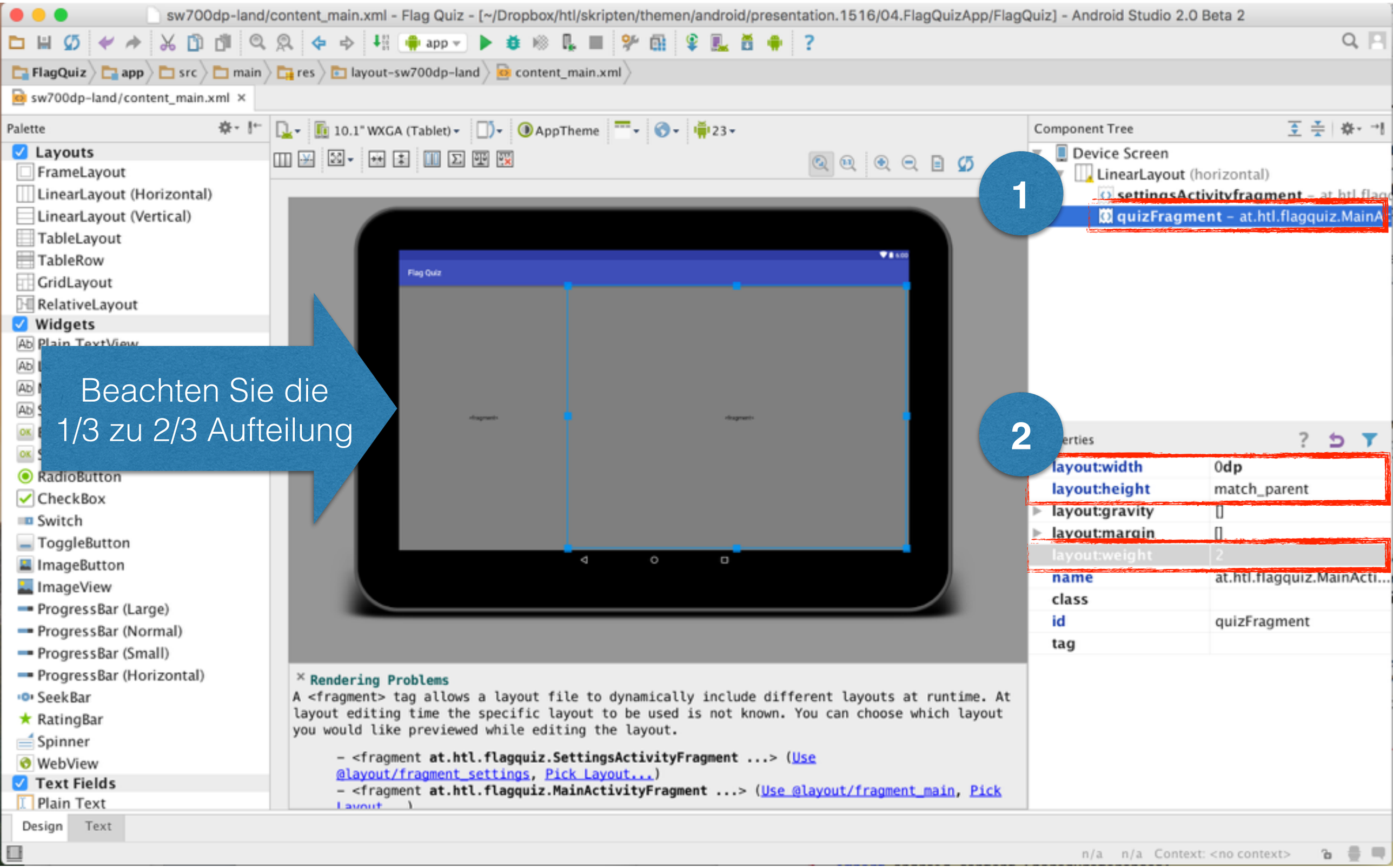
1

2

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

Design | Text

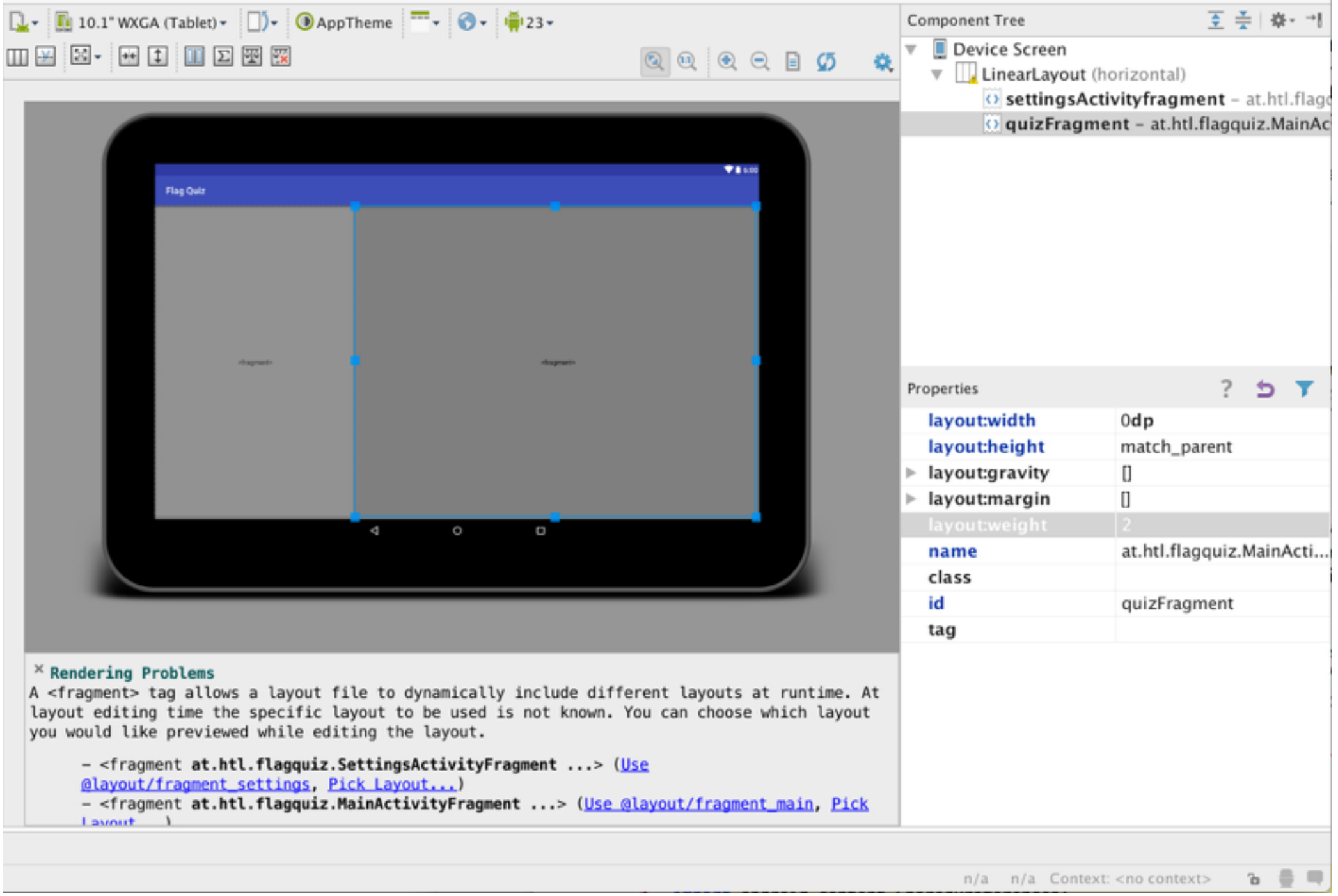
n/a n/a Context: <no context>


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
Palette
Layouts
FrameLayout
LinearLayout (Horizontal)
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
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)
quizFragment – at.htl.flagquiz.MainA
1
2
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
Design
Text
n/a
n/a
Context: <no context>

sw700dp-land/content_main.xml

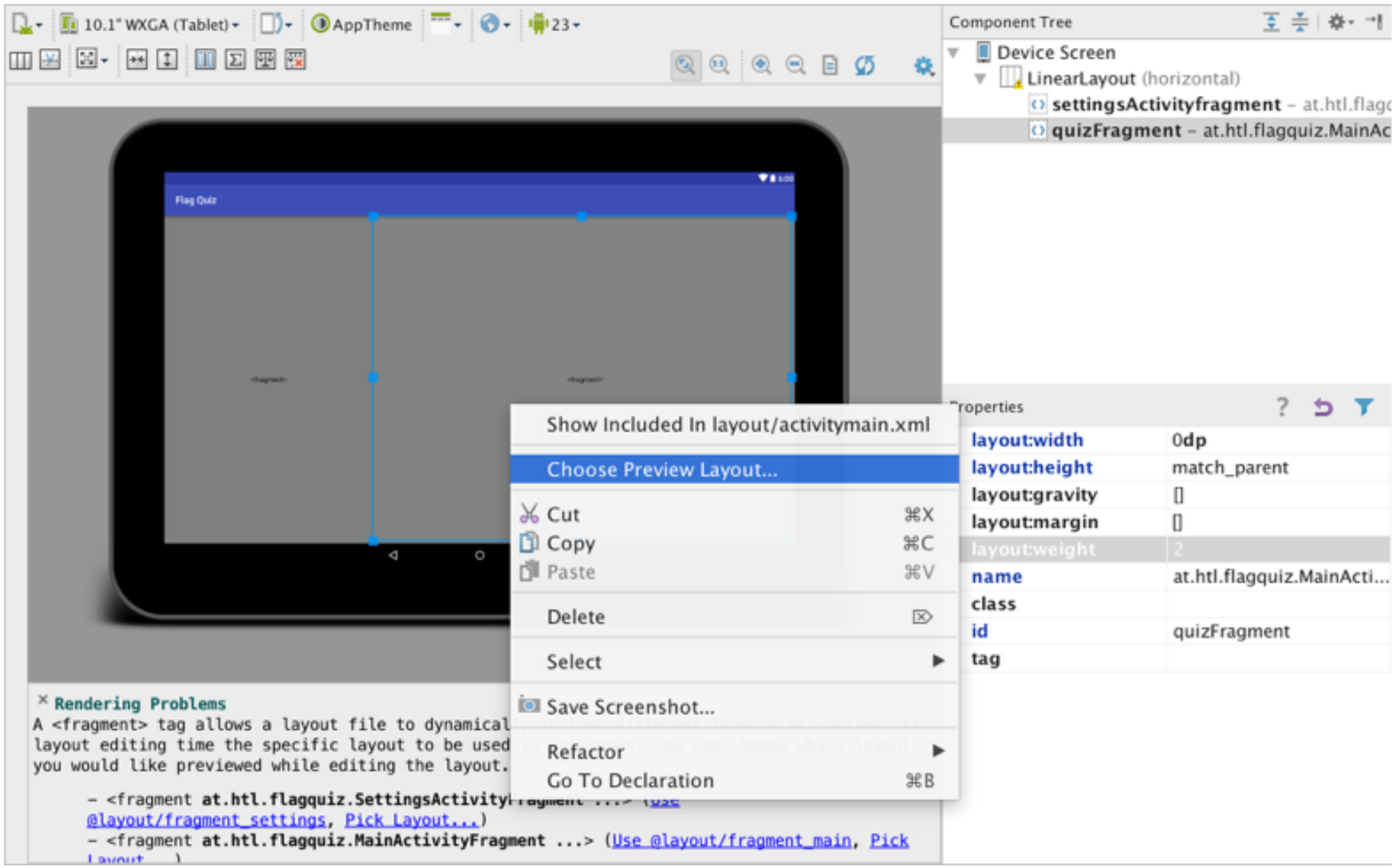
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    android:orientation="horizontal" android:layout_width="match_parent"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

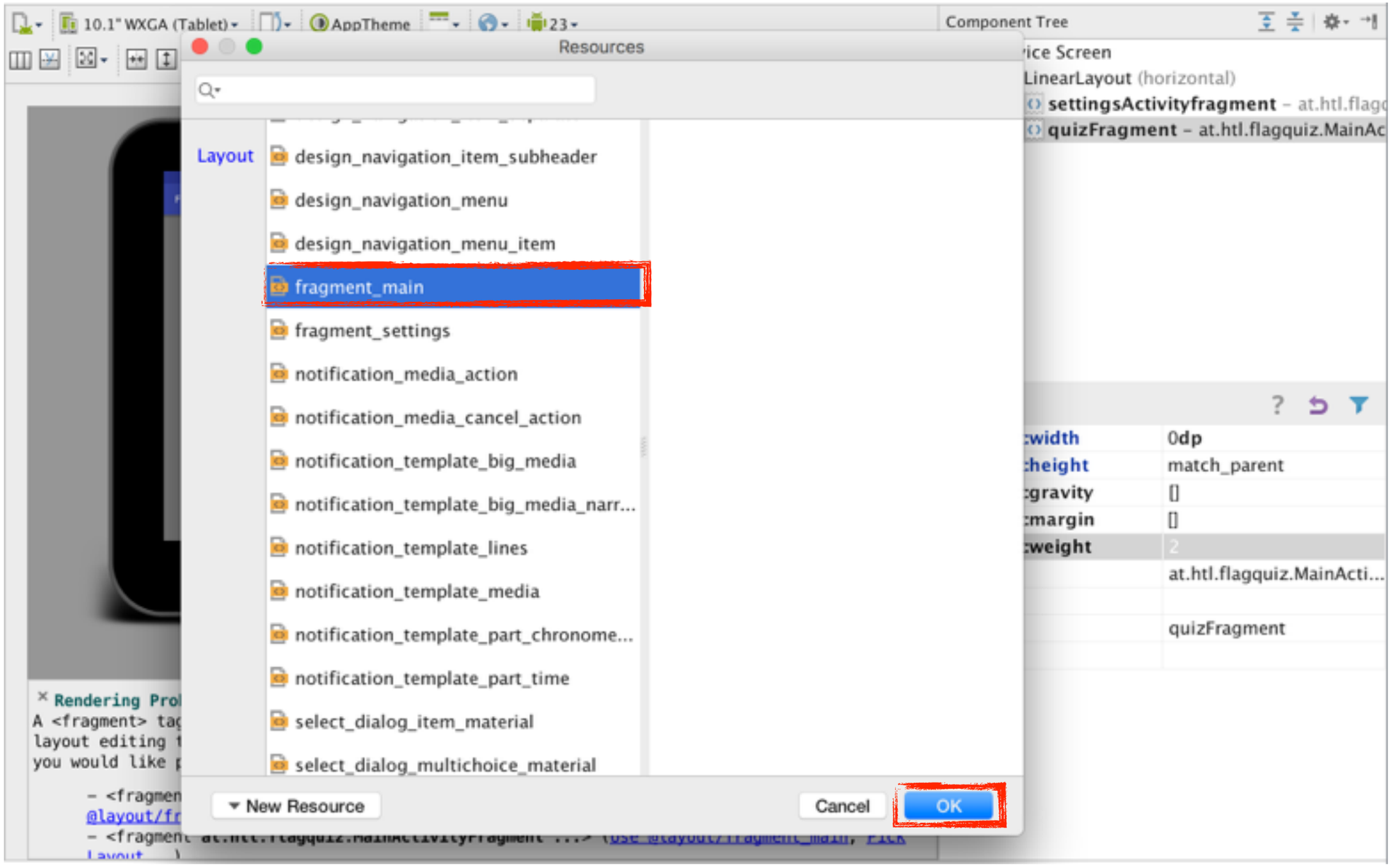
Dadurch wird bewirkt, dass der obere Teil des Layout unter der AppBar sichtbar bleibt und nicht hinter der AppBar verschwindet

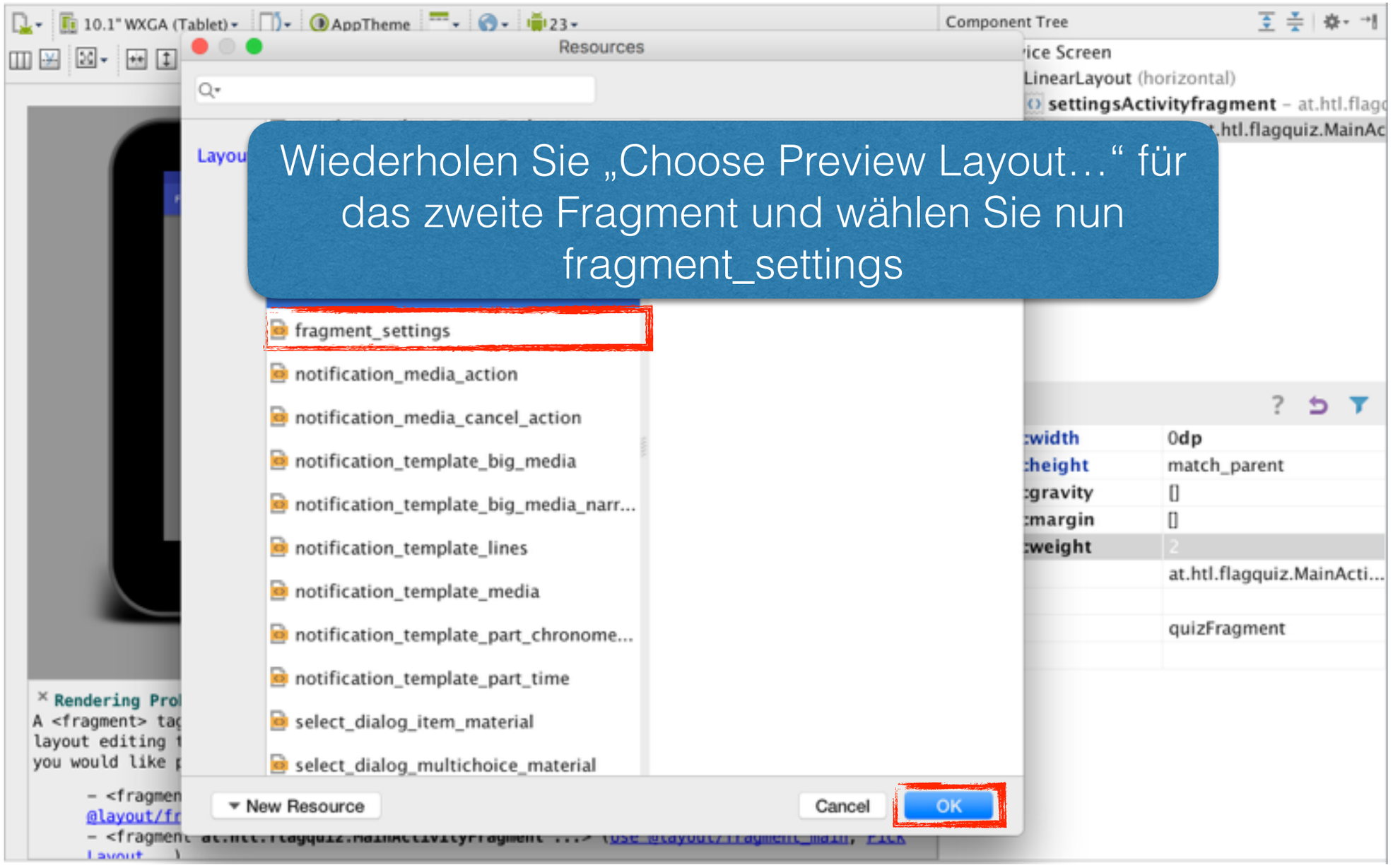


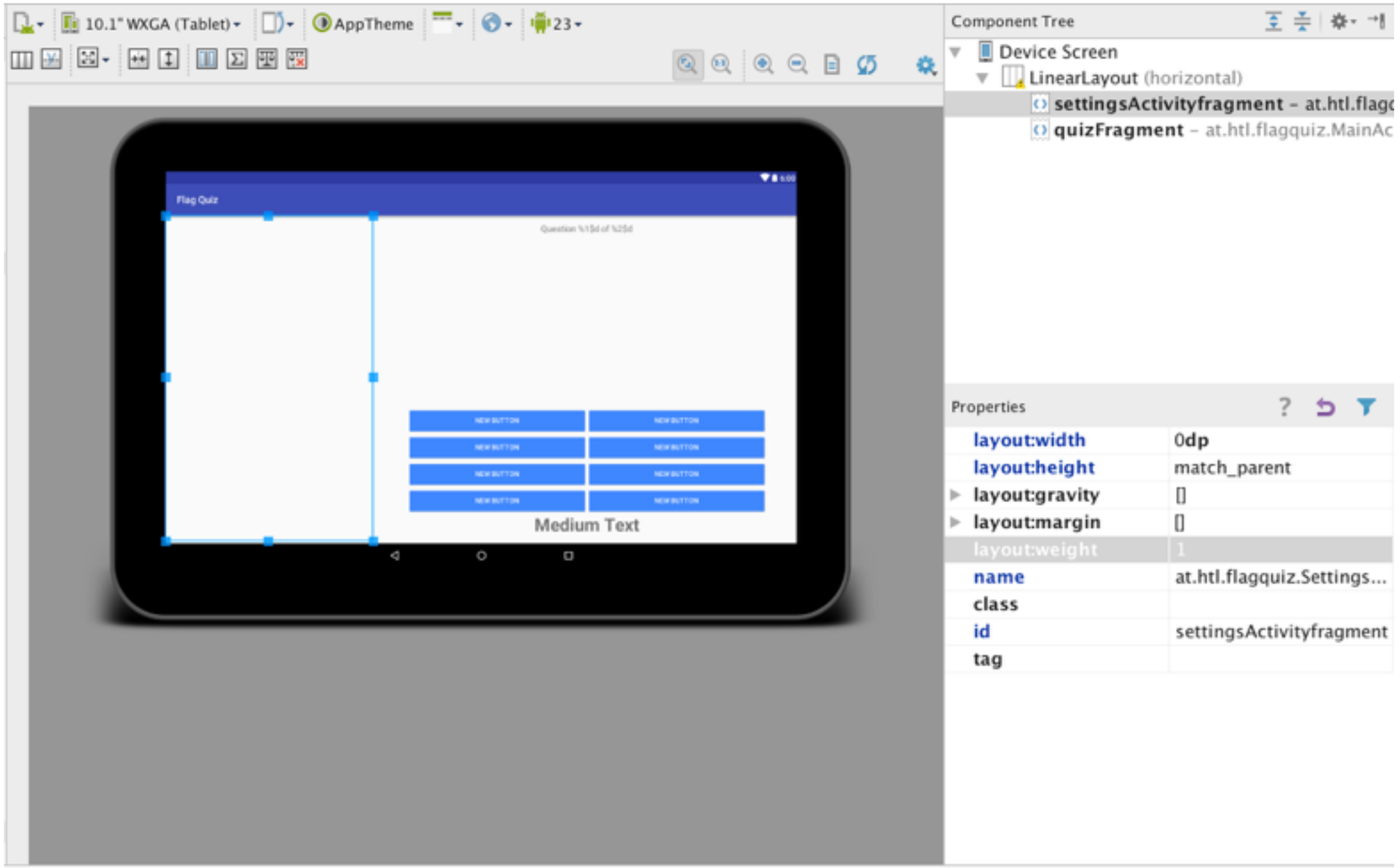
- Die Design View des Layout Editors kann jegliches Fragment als PreView anzeigen. Wird hierfür kein Fragment festgelegt, erscheinen Rendering Problems

- 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
Component Tree
Device Screen
LinearLayout (horizontal)
settingsActivityfragment – at.htl.flagc
quizFragment – at.htl.flagquiz.MainAc
Flag Quiz
Show Included In layout/activitymain.xml
Choose Preview Layout...
Cut ⌘X
Copy ⌘C
Paste ⌘V
Delete
Select
Save Screenshot...
Refactor
Go To Declaration ⌘B
Properties
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


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

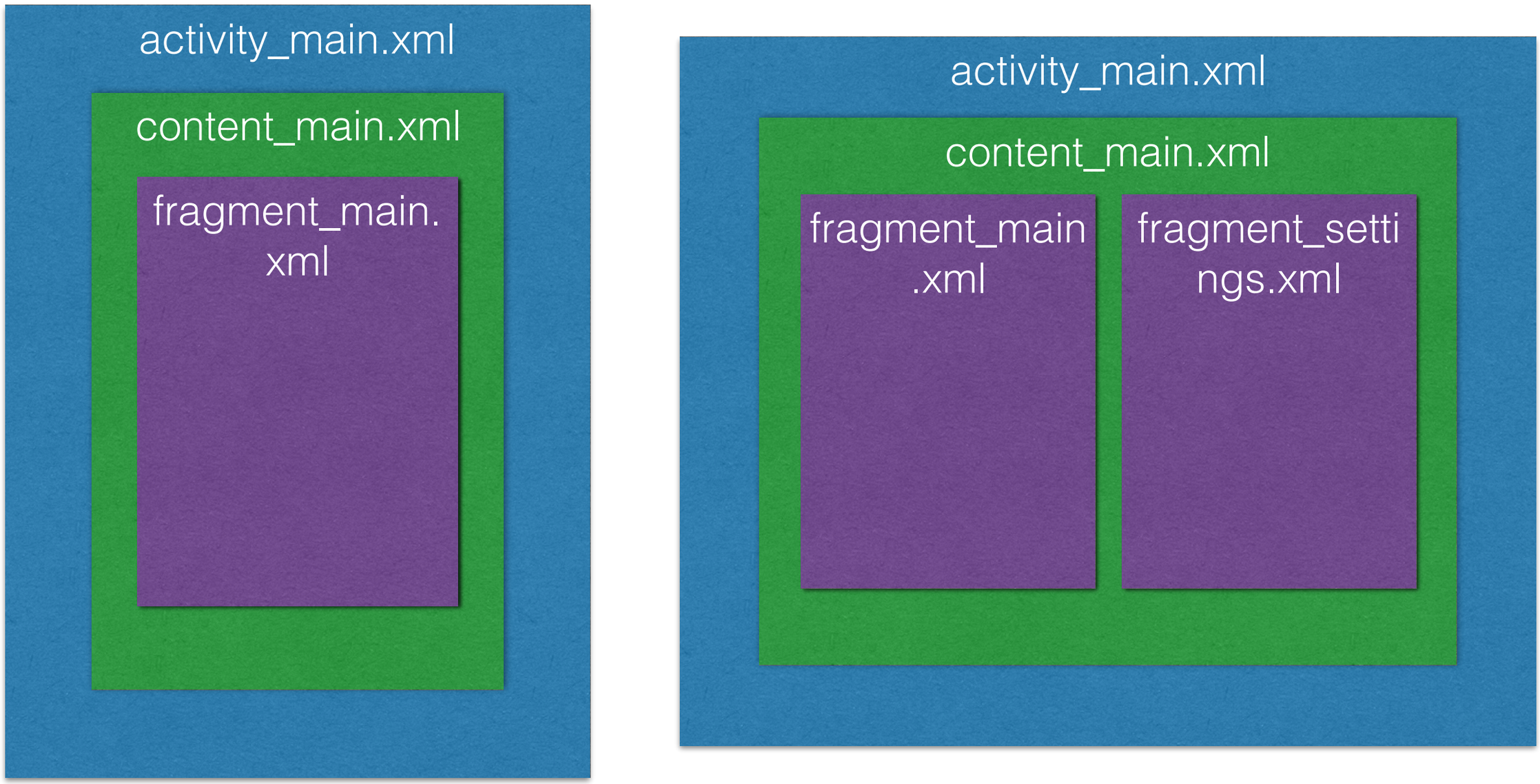

Wiederholen Sie „Choose Preview Layout…“ für das zweite Fragment und wählen Sie nun fragment_settings
Resources
fragment_settings
notification_media_action
notification_media_cancel_action
notification_template_big_media
notification_template_big_media_narr...
notification_template_lines
notification_template_media
notification_template_part_chronome...
notification_template_part_time
select_dialog_item_material
select_dialog_multichoice_material
New Resource
Cancel
OK
Component Tree
LinearLayout (horizontal)
settingsActivityfragment
0dp
match_parent
[]
[]
2
at.htl.flagquiz.MainActi...
quizFragment


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.flagc
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



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

MainActivity.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

preferences.xml ×

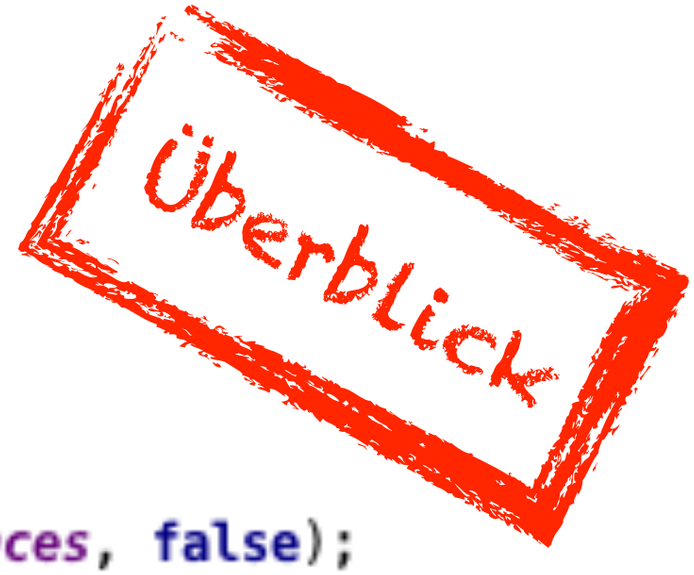
PreferenceScreen

```
<?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




```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

2

3

1 defaultPreferences

```
// 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);
}
```

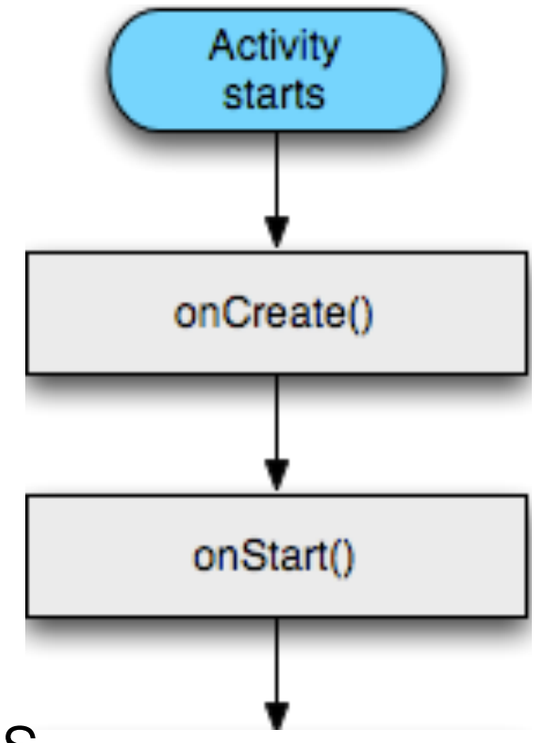
- getResources() gibt die Referenz auf das **Ressourcen-Objekt** (android.content.res.Resource) zurück.

- 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

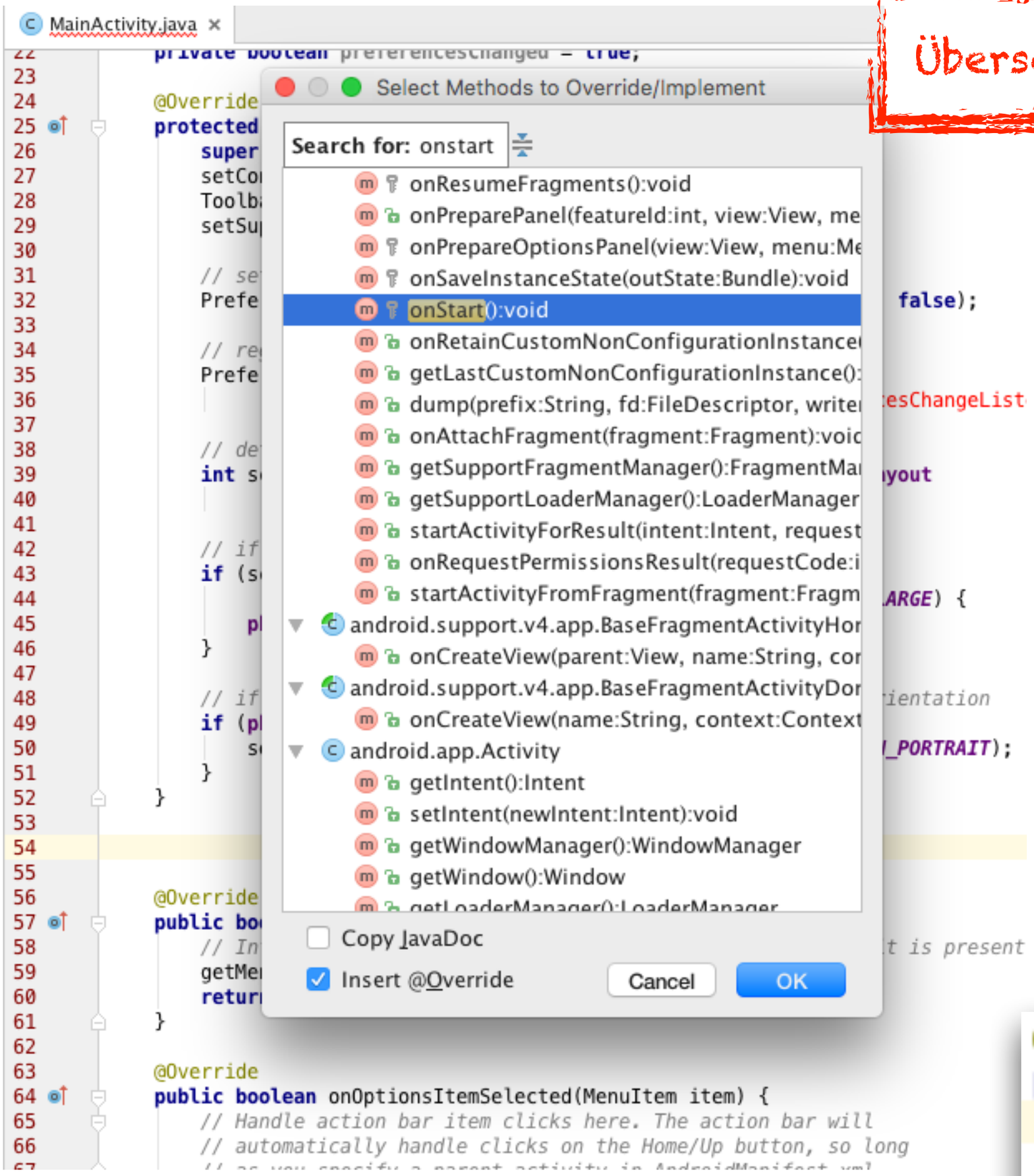
- Zuerst wird mit einem Bitoperator & (AND) die Bildschirmgröße ermittelt. Anschließend wird ermittelt, ob die Bildschirmgröße LARGE oder XLARGE ist. Falls ja, wird die Portrait-Orientierung (Hochformat) erzwungen

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



- 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()

```java
// 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)

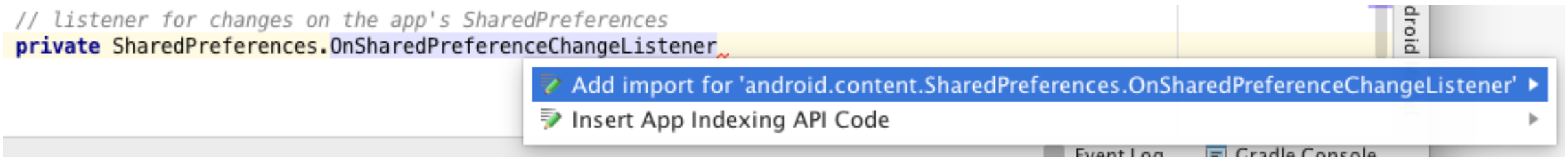
onOptionsItemSelected()

```
// 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



<alt> + <Eingabe> zum Erstellen des Imports

```
// 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

MainActivityFragment.java

```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
```

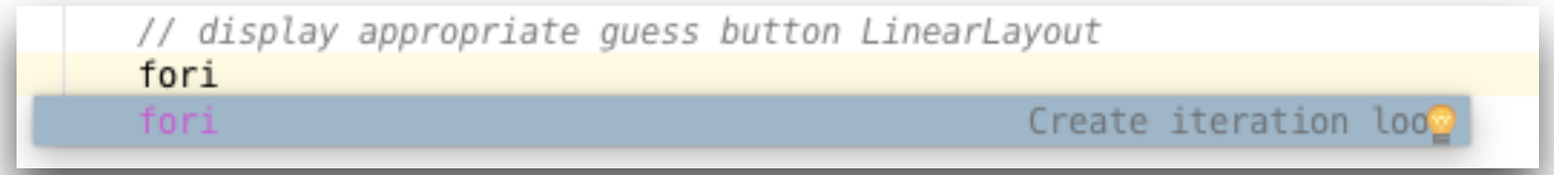
fileNameList List<String>
quizCountriesList List<String>
regionsSet Set<String>
guessLinearLayouts LinearLayout[]

⬇️⬇️⬇️<Enter>

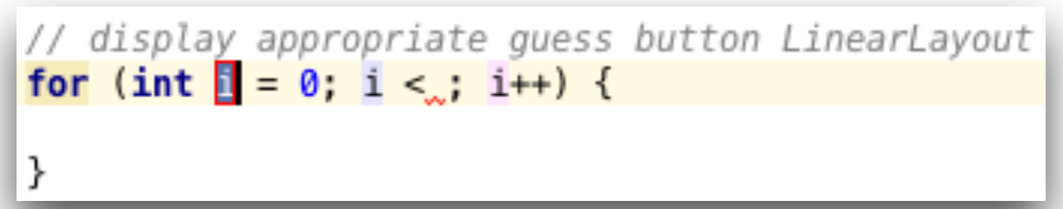
```
// hide all quests button LinearLayout
for (LinearLayout layout : guessLinearLayouts) {

}
```

layout<Enter>

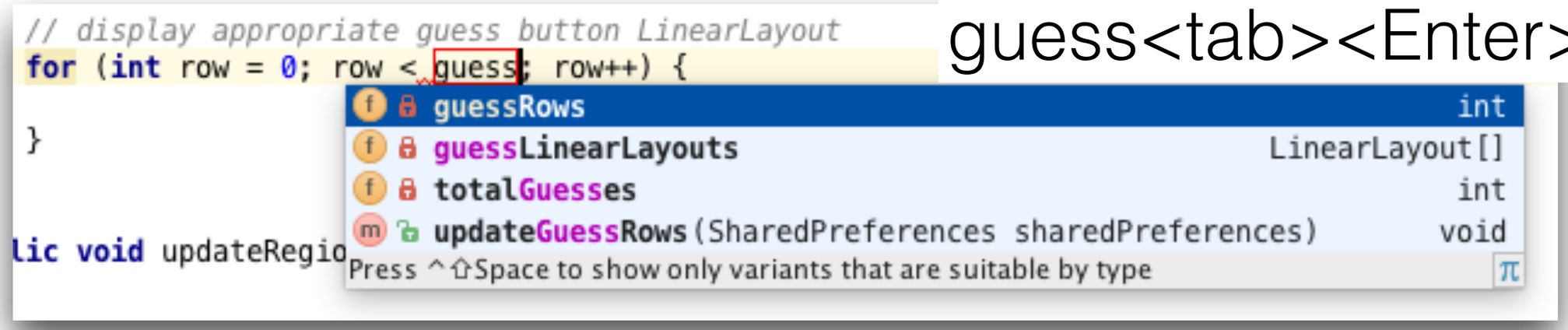
Live Template fori



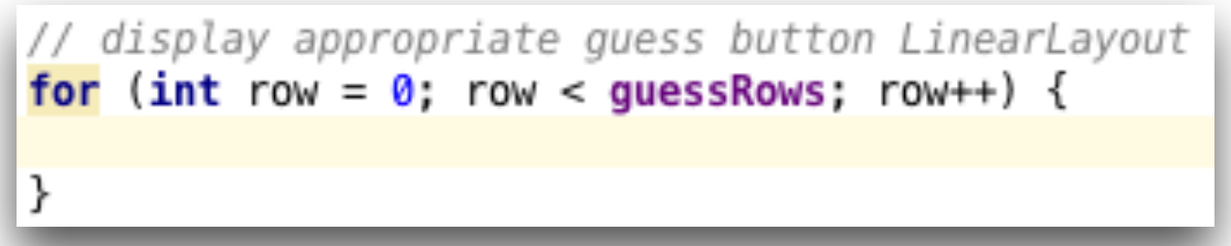
fori<tab>



row<Enter>



guess<tab><Enter>



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()

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

resetQuiz() 1

```
// 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);
    }
```

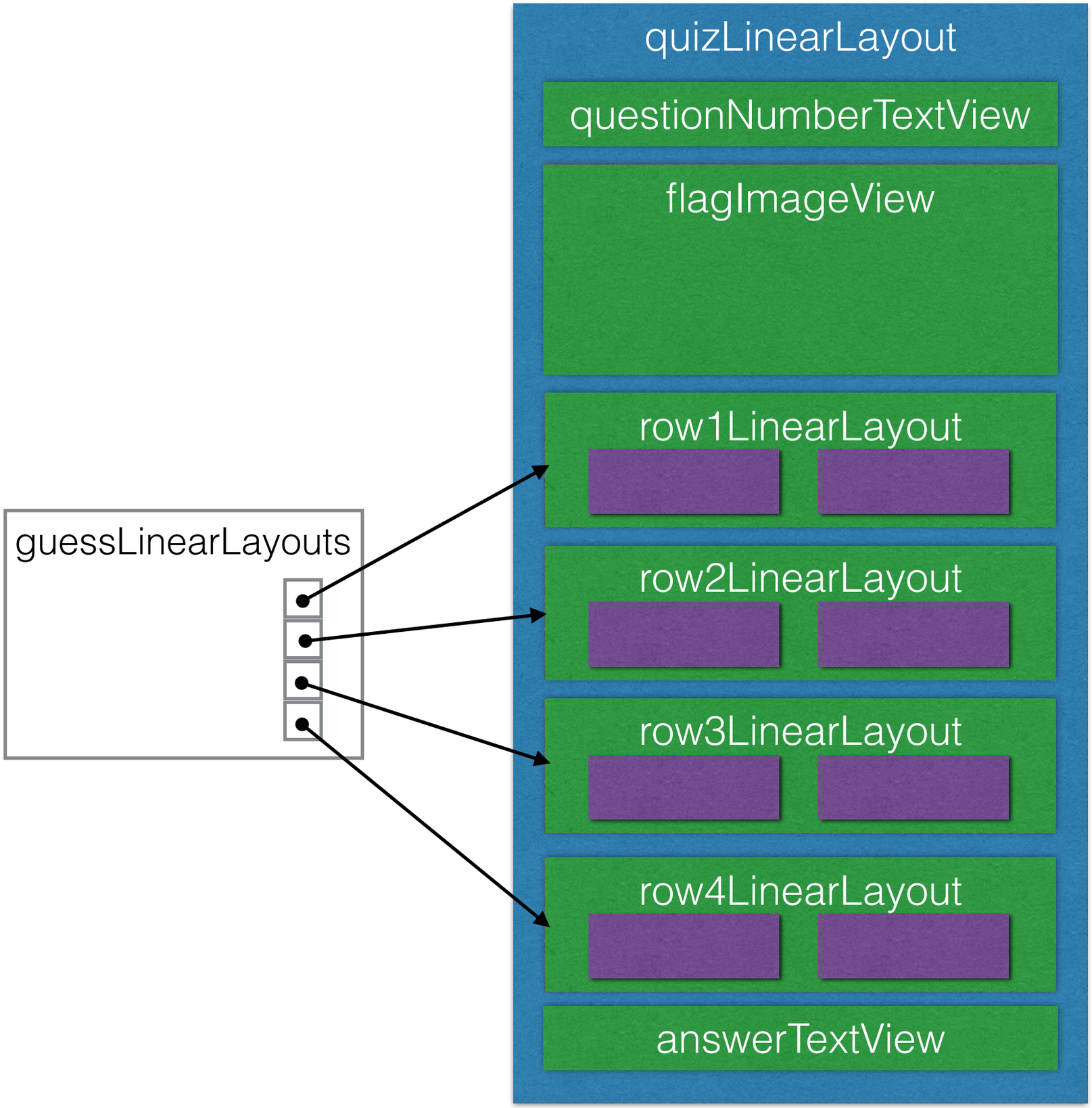
Es werden nur die Images der ausgewählten Regionen in die fileNameList geladen

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
}
```



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.)

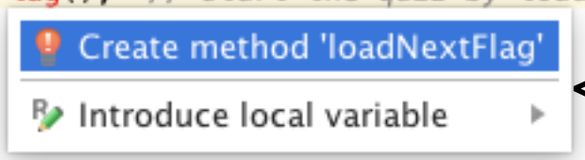
quizCountriesList = {ArrayList@4204} size = 10
- 0 = "Europe-Norway"
- 1 = "Africa-Madagascar"
- 2 = "Africa-Cape_Verde"
- 3 = "Africa-Ivory_Coast"
- 4 = "Africa-Togo"
- 5 = "Asia-India"
- 6 = "Asia-Iraq"
- 7 = "Africa-Rwanda"
- 8 = "Africa-Kenya"
- 9 = "Africa-Tanzania"

Anzahl der Elemente = FLAGS_IN_QUIZ

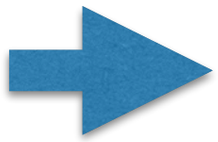
- nextImage = "Europe-Norway"
- region = "Europe"
- correctAnswer = "Europe-Norway"
- correctAnswers = 0

loadNextFlag() 1

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



<Alt><Enter>

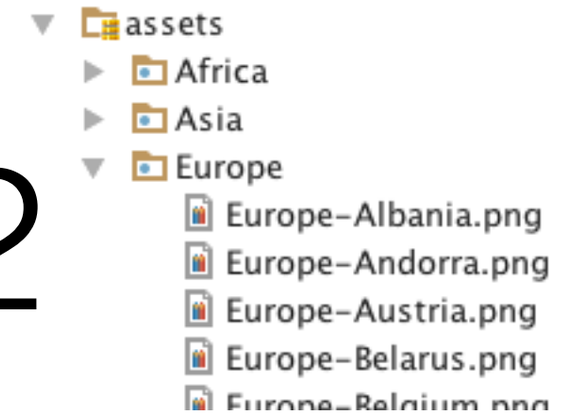


```
private void loadNextFlag() {
    
}
```

```
// 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




```
// 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()

```
// 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
<?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>
```

+

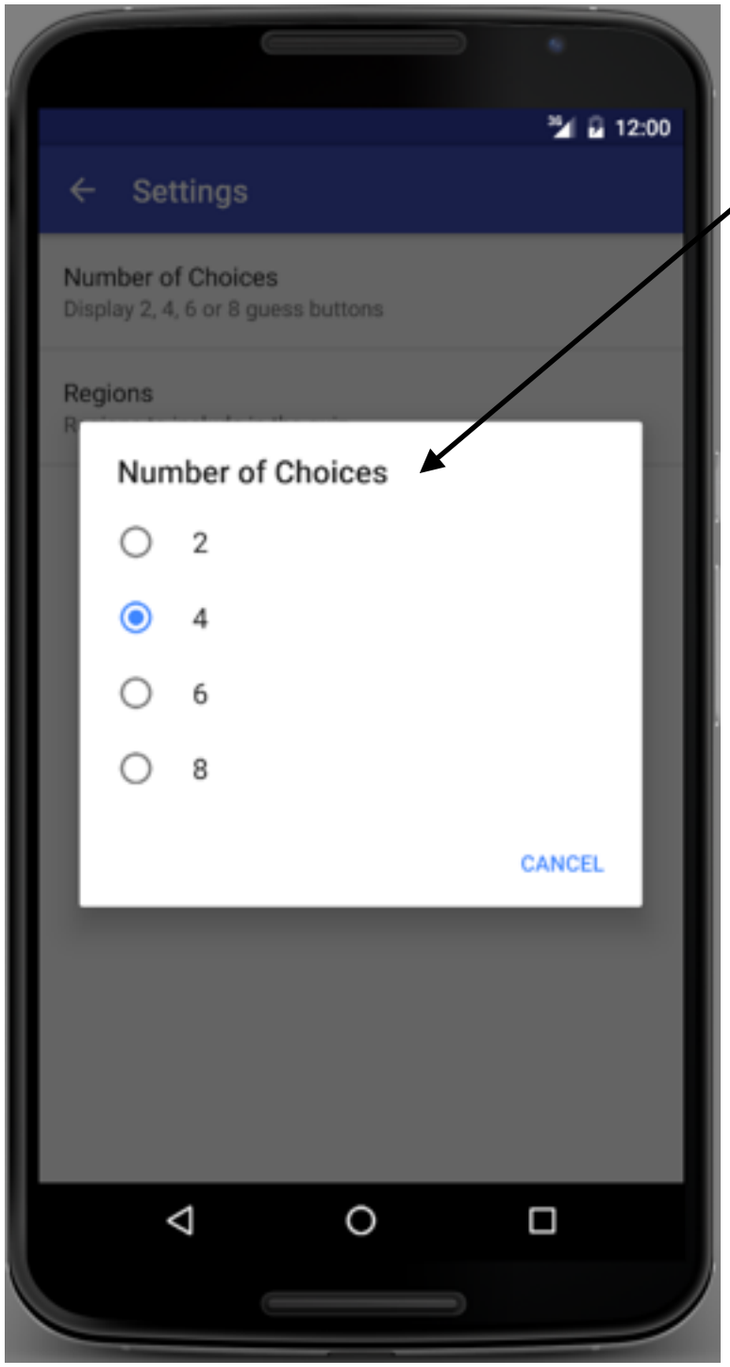
```java
public class SettingsActivityFragment extends PreferenceFragment {
    // creates preferences GUI from preferences.xml file in res/xml
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.preferences);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

+

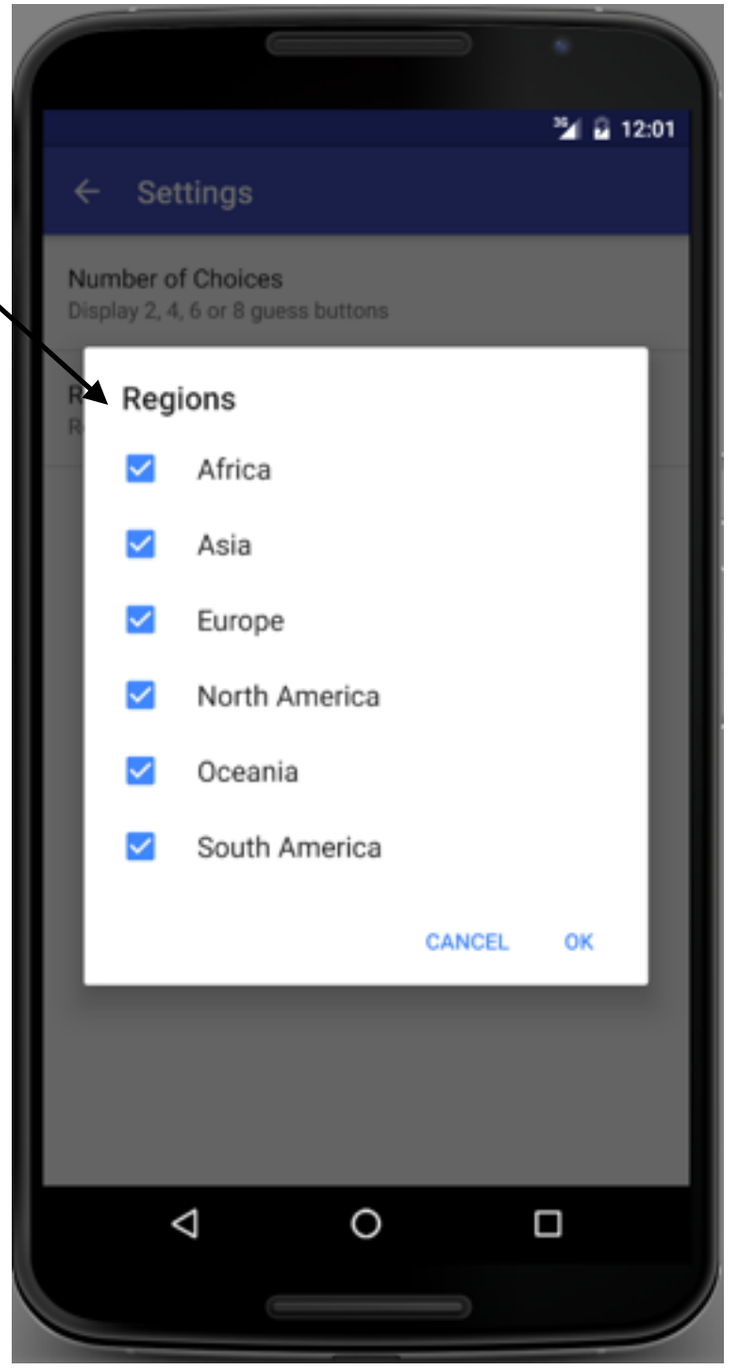
arrays.xml

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

=


12:00
Settings
Number of Choices
Display 2, 4, 6 or 8 guess buttons
Regions
Number of Choices
2
4
6
8
CANCEL


12:00
Settings
Number of Choices
Display 2, 4, 6 or 8 guess buttons
Regions
Regions to include in the quiz


12:01
Settings
Number of Choices
Display 2, 4, 6 or 8 guess buttons
Regions
Africa
Asia
Europe
North America
Oceania
South America
CANCEL
OK

AndroidManifest.xml

```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>
```

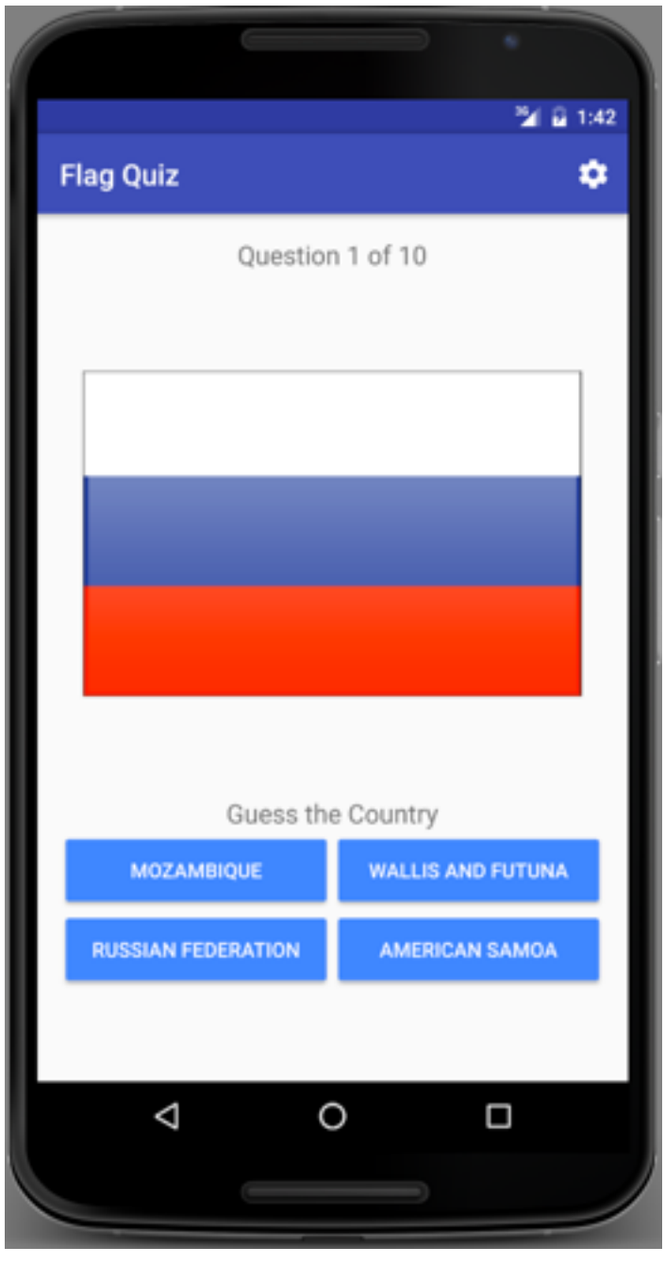
Beim Öffnen der App (neuer Intent) wird nicht jedesmal eine neue Instanz erstellt, sondern - falls vorhanden - erhält die bestehende Instanz den Intent

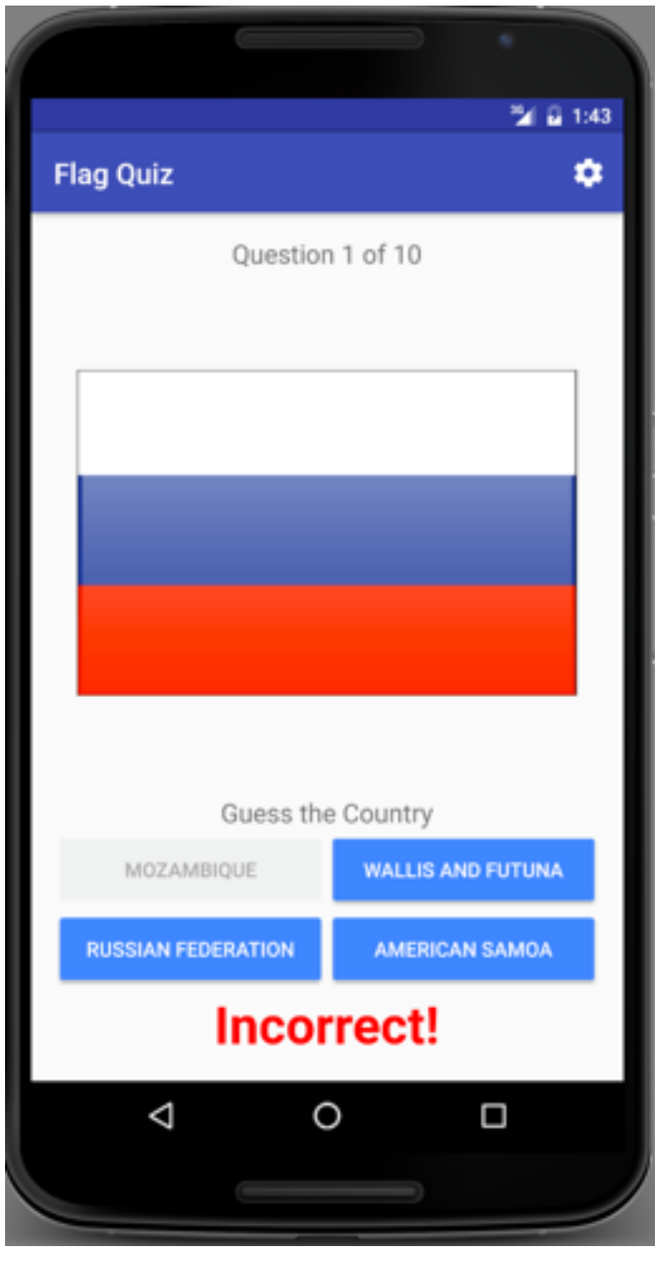
https://developer.android.com/guide/topics/manifest/activity-element.html#lmode

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 . . .

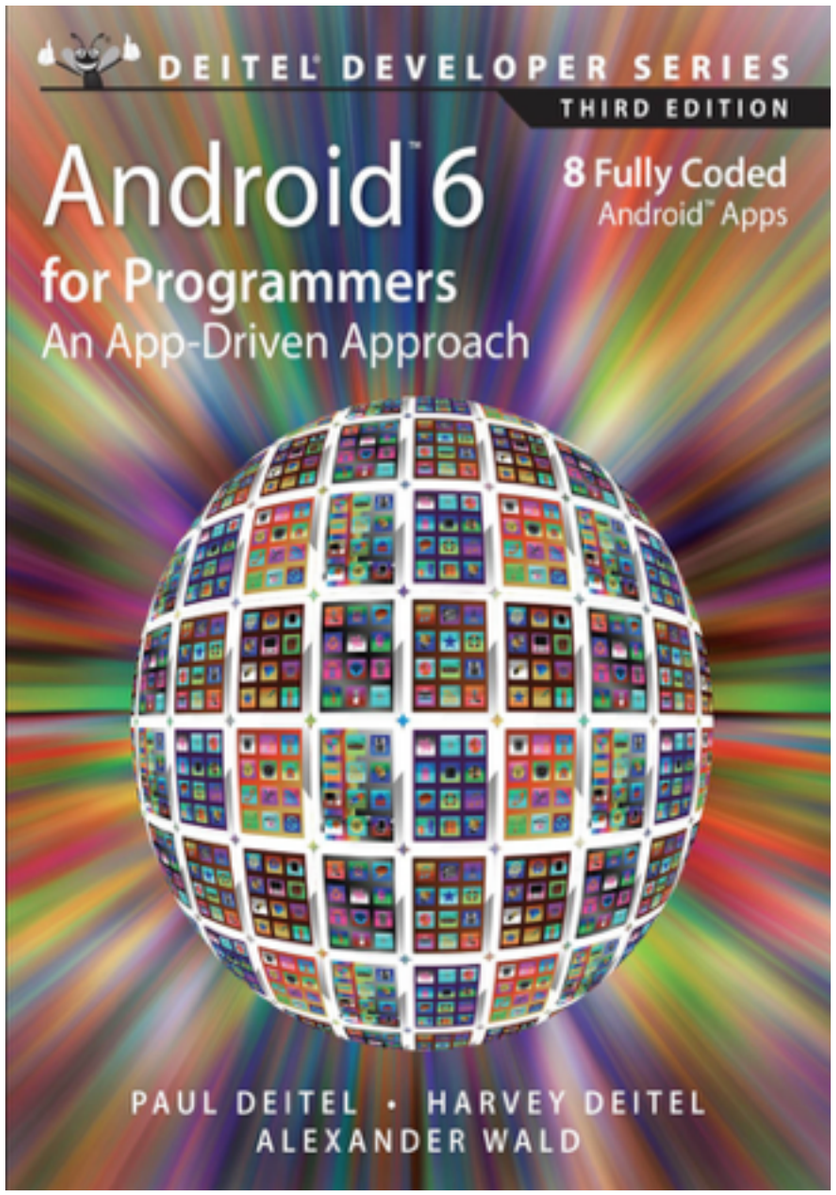

1:42
Flag Quiz
Question 1 of 10
Guess the Country
MOZAMBIQUE
WALLIS AND FUTUNA
RUSSIAN FEDERATION
AMERICAN SAMOA


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



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