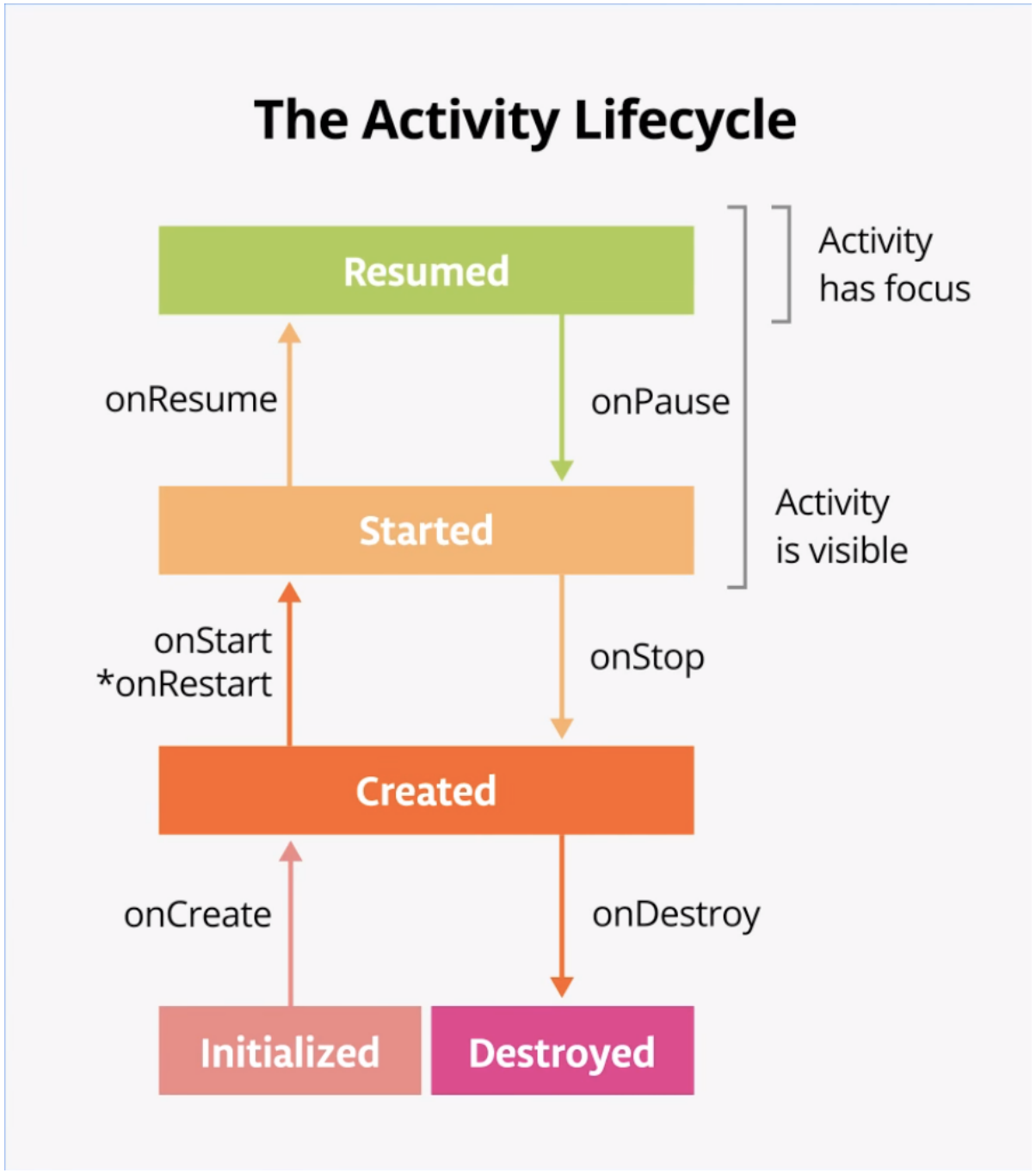
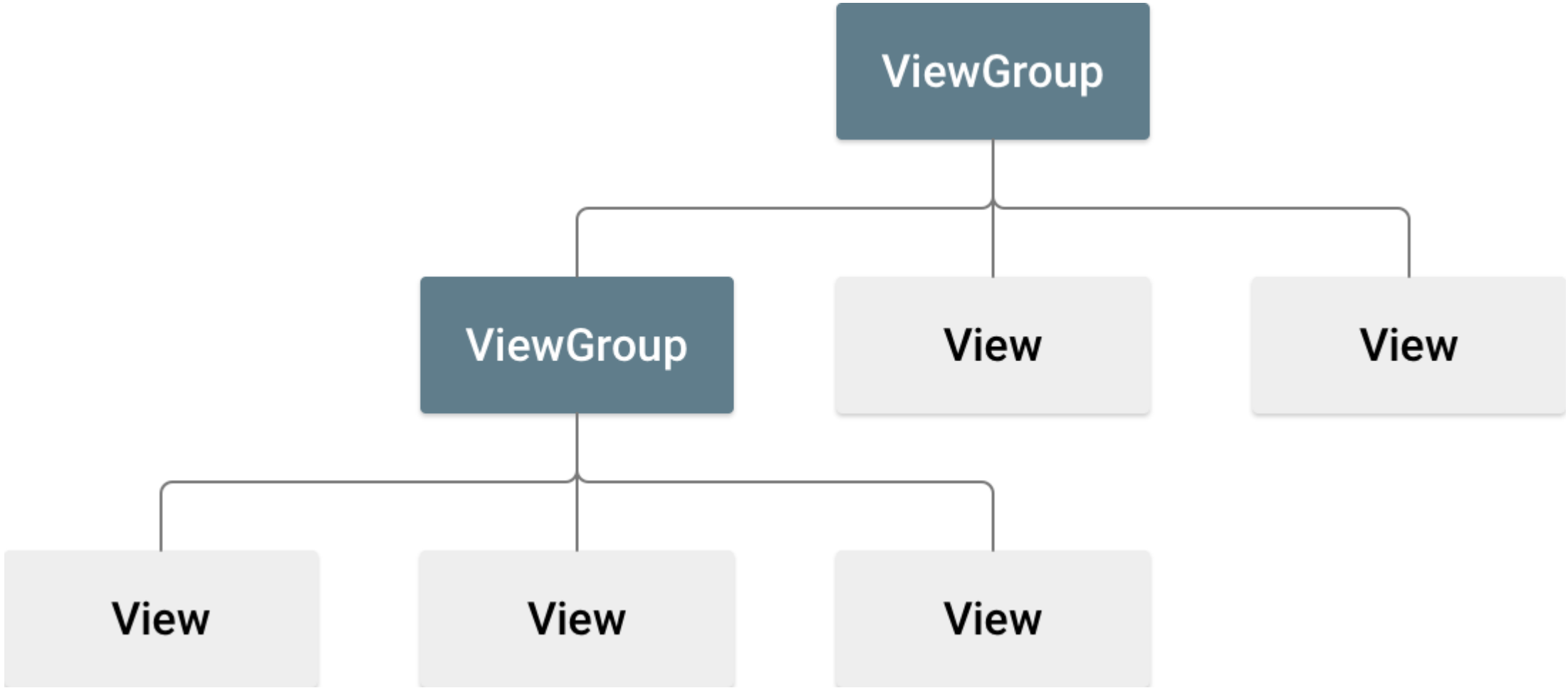
STU FEI
SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

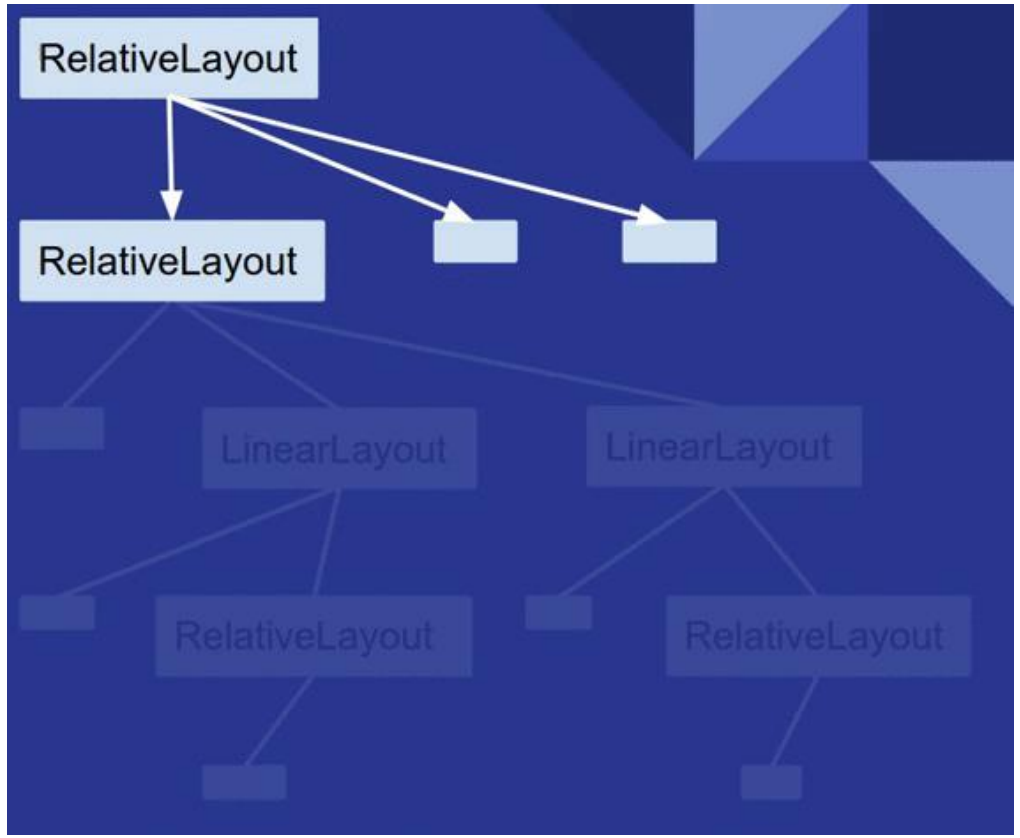
# Mobilné výpočty

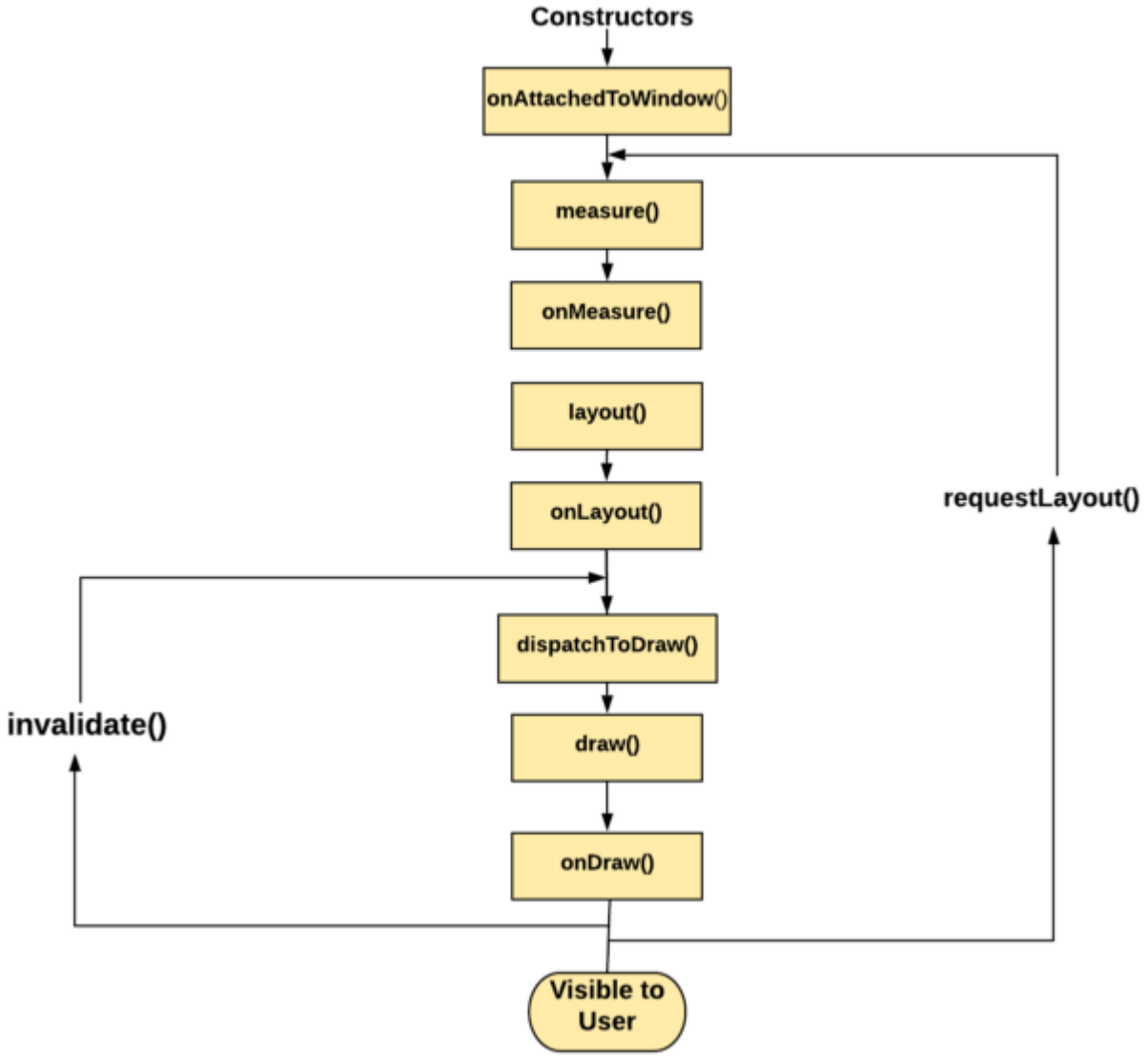
Ing. Maroš Čavojský, PhD.


The Activity Lifecycle
Resumed
Started
Created
Initialized
Destroyed
onResume
onPause
onStart
*onRestart
onStop
onCreate
onDestroy
Activity has focus
Activity is visible

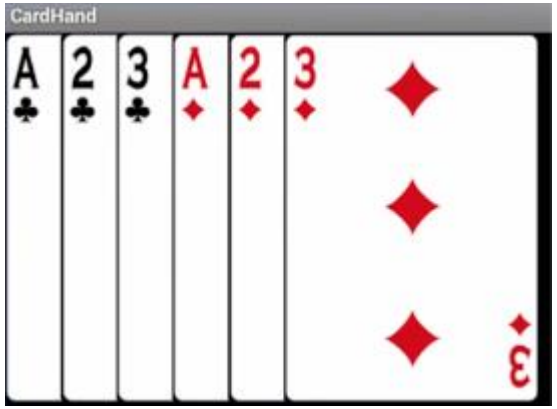

ViewGroup
ViewGroup
View
View
View
View
View

# View / ViewGroup



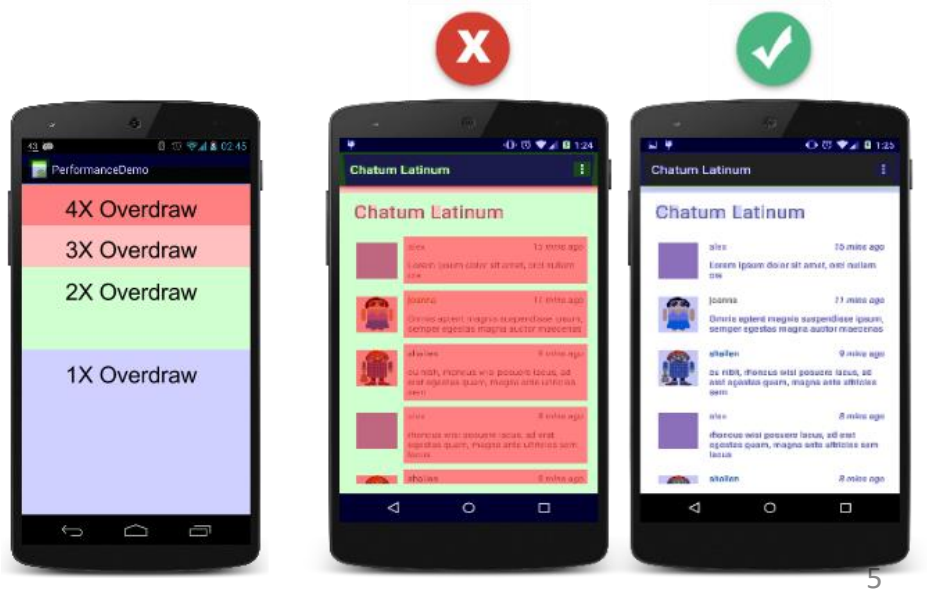
# Chyby pri vytváraní UI




- Prekresľovanie – Overdraw
- Komplexné vnorené layouty – LinearLayout/RelativeLayout .... ConstraintLayout
- Double taxation – komplikované layouty vyžadujú viac-násobné layout-measure
  - RelativeLayout
  - LinearLayout
  - GridLayout

Vykreslenie za 16ms pre 60 fps

***1000ms / 60 frames =***
**16.666 ms / frame**



# View

- základný stavebný prvok

# ViewGroup

- podtrieda triedy View
- základný stavebný prvok pre rozloženia (neviditeľné prvky)

# LinearLayout

- pre riadkové/stĺpcové rozloženie

# ConstrainLayout

- pre efektívne rozloženie

# RelativeLayout

- pre relatívne rozloženie

# LinearLayout

- pre riadkové/stĺpcové rozloženie

Android Developers > Docs > Referencie

## LinearLayout

```
open class LinearLayout : ViewGroup
```

kotlin.Any
↳ android.view.View
↳ android.view.ViewGroup
↳ android.widget.LinearLayout

# ConstrainLayout

- pre efektívne rozloženie

Android Developers > Docs > Reference

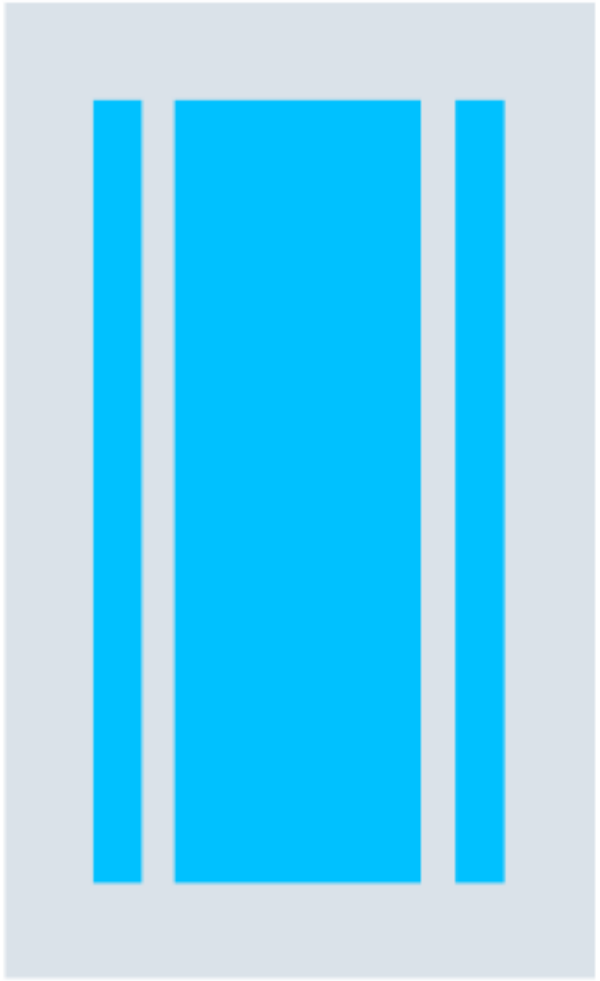
## ConstraintLayout

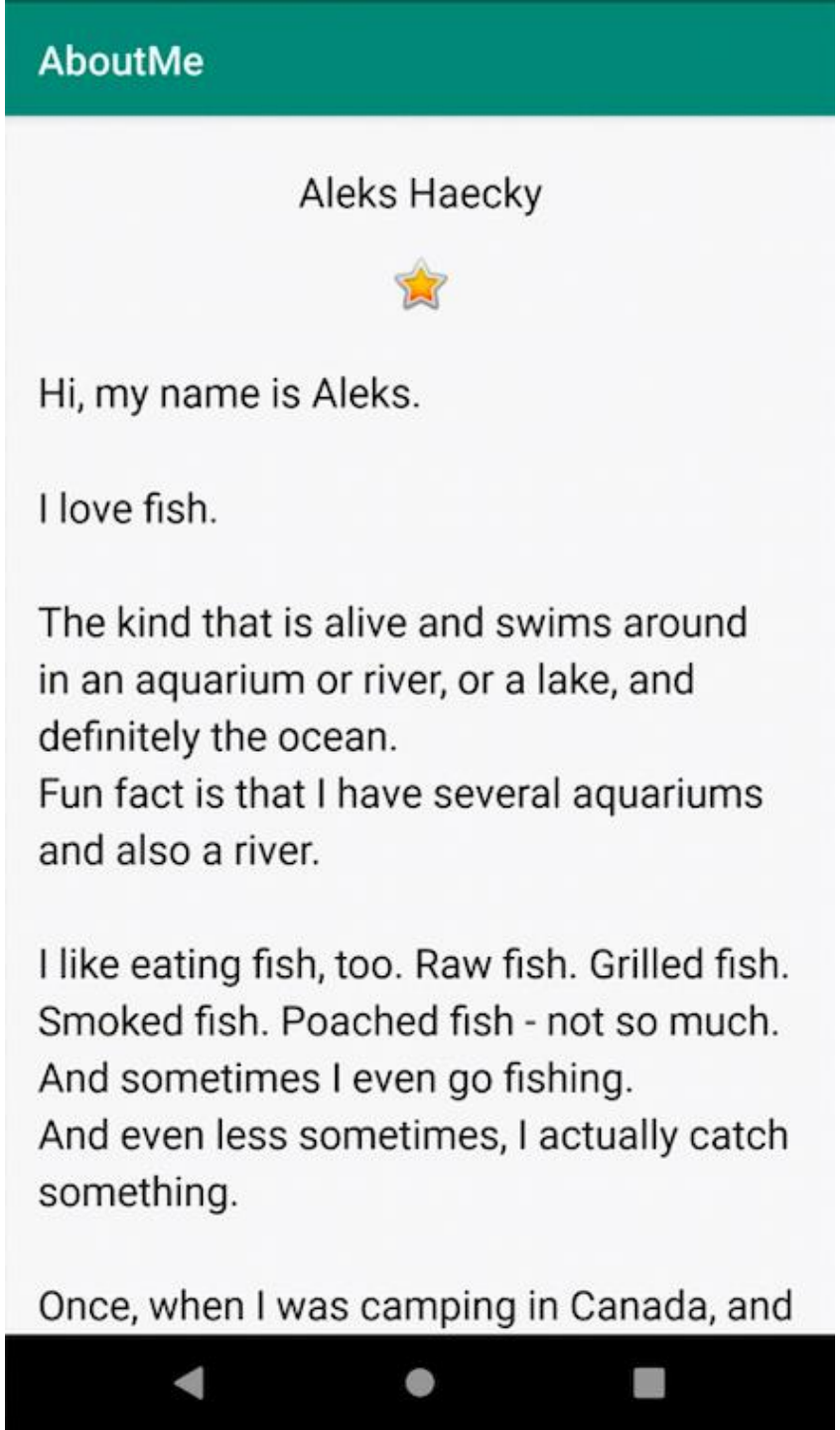
```
public class ConstraintLayout
extends ViewGroup
```

java.lang.Object
↳ ViewGroup
↳ androidx.constraintlayout.widget.ConstraintLayout

## Vertical LinearLayout

## Horizontal LinearLayout






# Vertial LinearLayout

# ScrollView

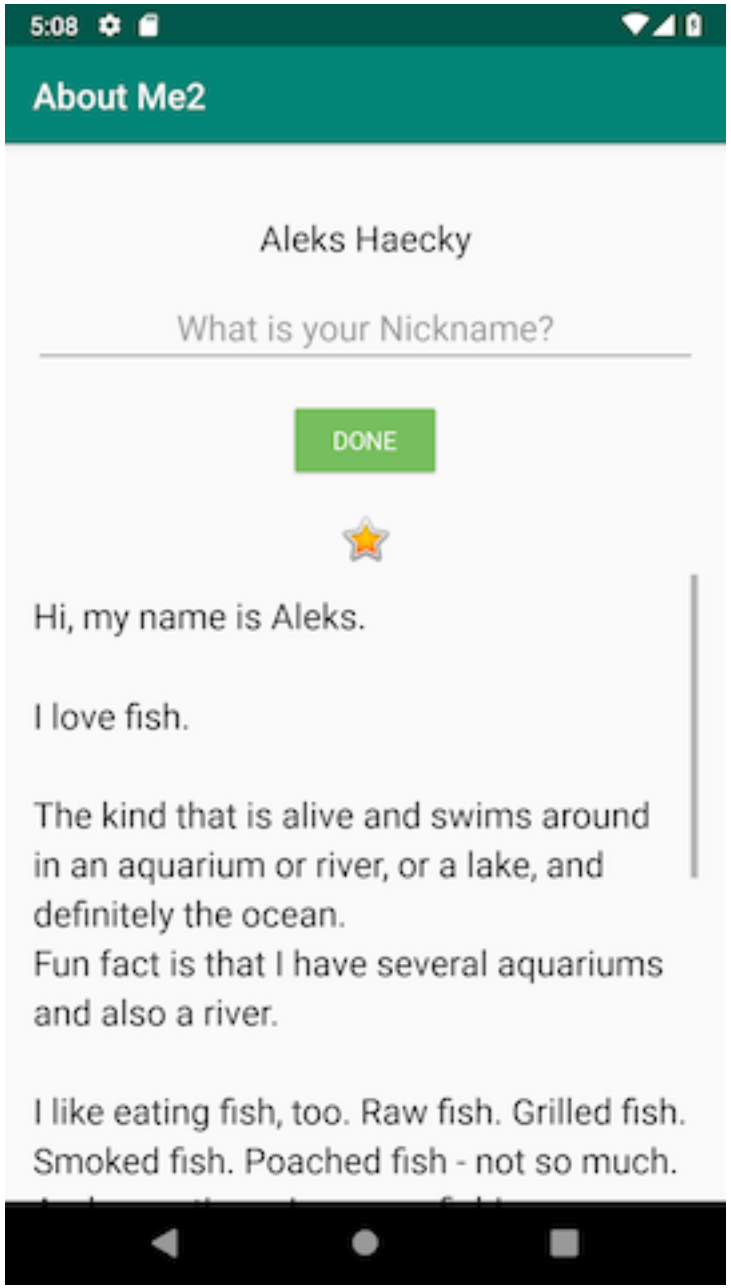
- umožňuje scrolovať obsah
- môže obsahovať najviac jeden prvok

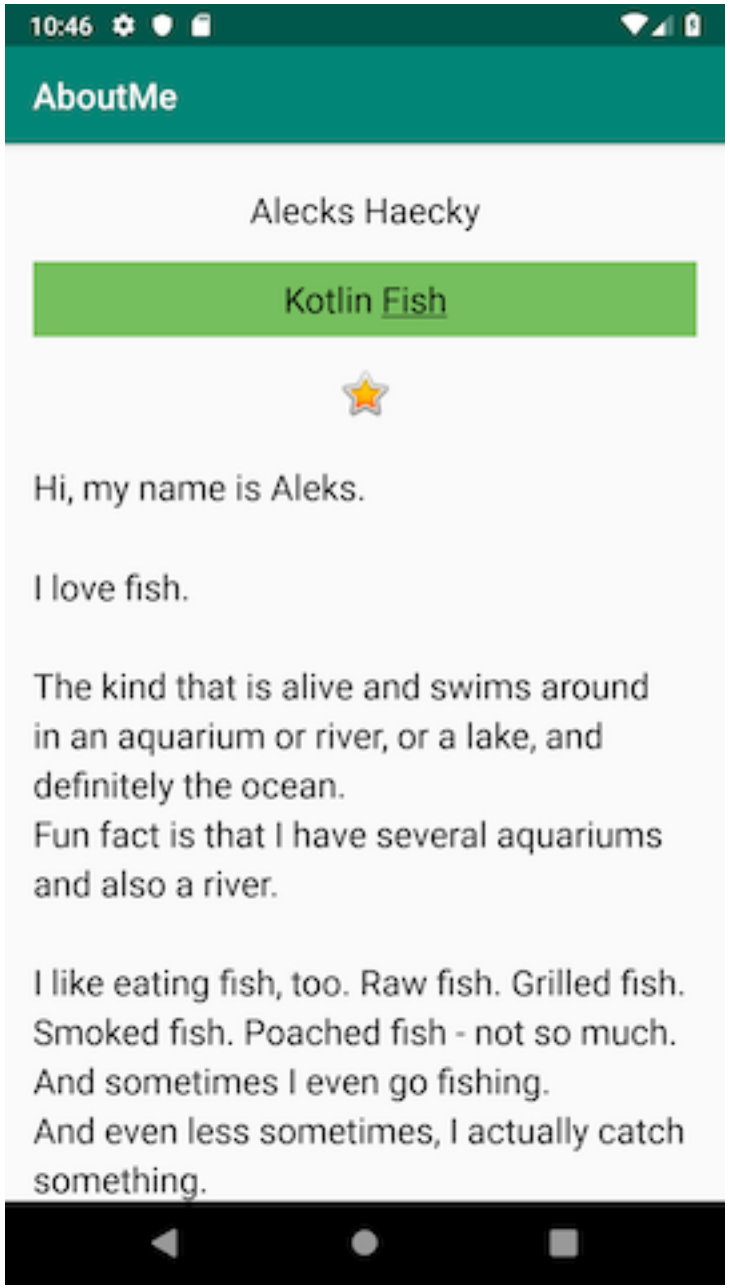
Android Developers > Docs > Referencie

## ScrollView

```
open class ScrollView : FrameLayout
```

kotlin.Any
↳ android.view.View
↳ android.view.ViewGroup
↳ android.widget.FrameLayout
↳ android.widget.ScrollView


5:08
About Me2
Aleks Haecky
What is your Nickname?
DONE
Hi, my name is Aleks.
I love fish.
The kind that is alive and swims around in an aquarium or river, or a lake, and definitely the ocean.
Fun fact is that I have several aquariums and also a river.
I like eating fish, too. Raw fish. Grilled fish. Smoked fish. Poached fish - not so much.


10:46
AboutMe
Alecks Haecky
Kotlin Fish
Hi, my name is Aleks.
I love fish.
The kind that is alive and swims around in an aquarium or river, or a lake, and definitely the ocean.
Fun fact is that I have several aquariums and also a river.
I like eating fish, too. Raw fish. Grilled fish. Smoked fish. Poached fish - not so much.
And sometimes I even go fishing.
And even less sometimes, I actually catch something.

# EditText

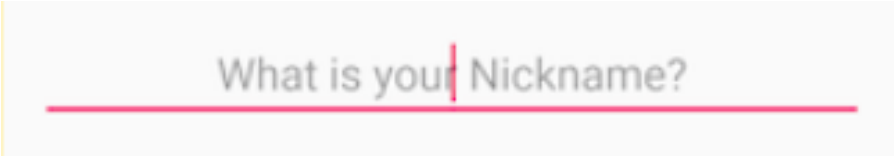
- umožňuje zadávať text

Android Developers > Docs > Referencie

## EditText

```
open class EditText : TextView
```

kotlin.Any
↳ android.view.View
↳ android.widget.TextView
↳ android.widget.EditText




```
<EditText
      android:id="@+id/plain_text_input"
      android:layout_height="wrap_content"
      android:layout_width="match_parent"
      android:inputType="text"/>
```

# TextView

- umožňuje zobrazovať text

```
<TextView
    android:id="@+id/text_view_id"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="@string/hello" />
```

Android Developers > Docs > Referencie

## TextView

---

```
open class TextView : View, ViewTreeObserver.OnPreDrawListener
```

kotlin.Any
↳ android.view.View
↳ android.widget.TextView

# Button

- umožňuje potvrdiť / vykonať akciu

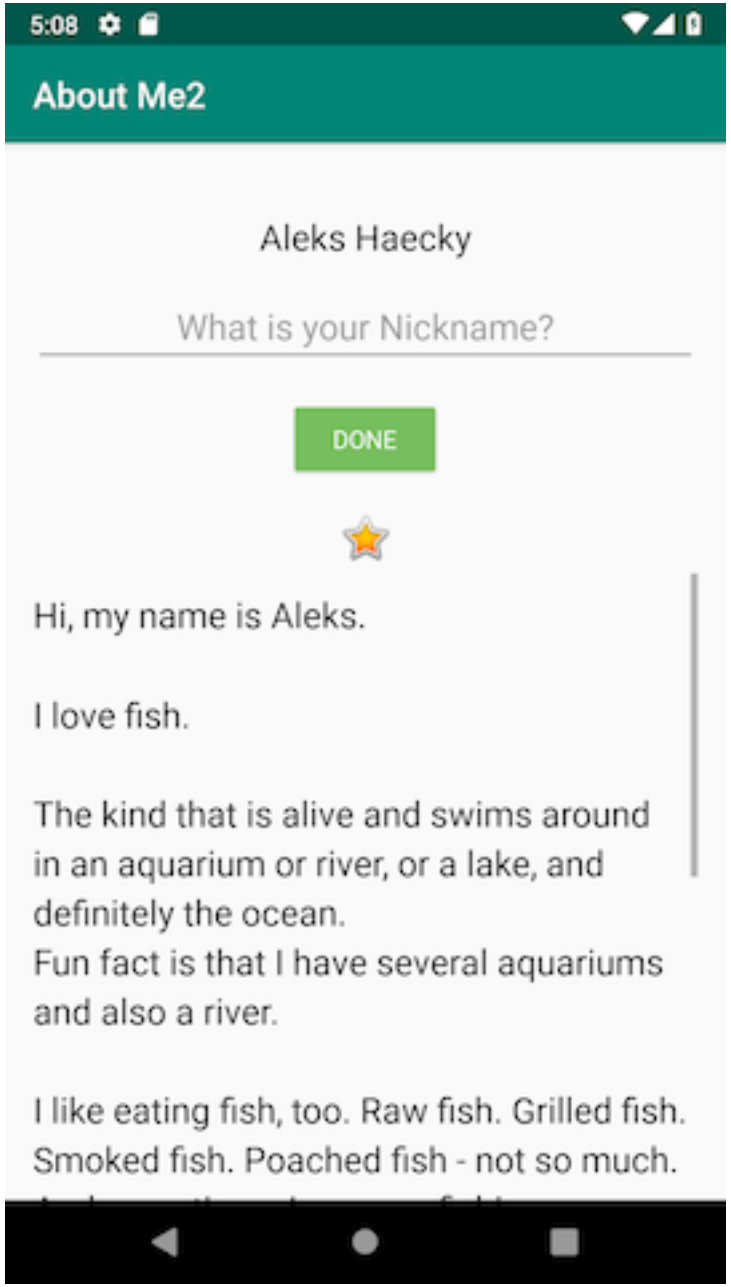

Android Developers > Docs > Referencie

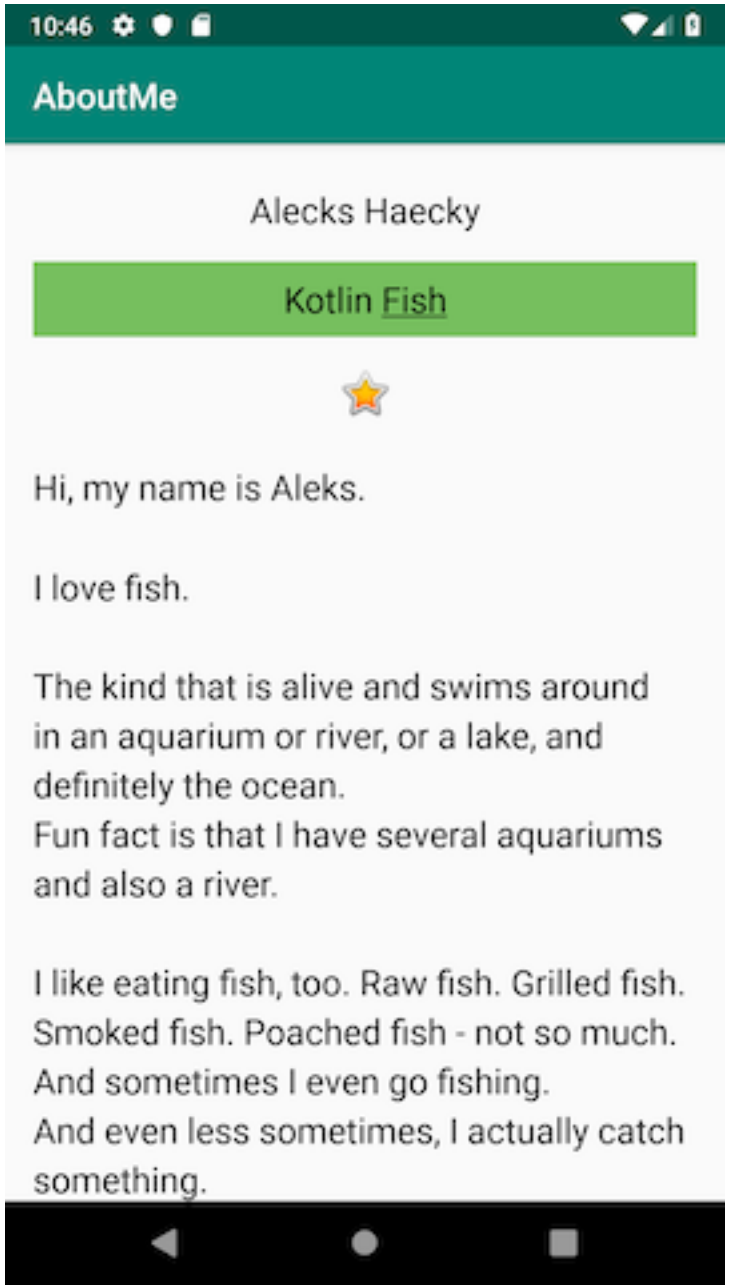
## Button

```
open class Button : TextView
```

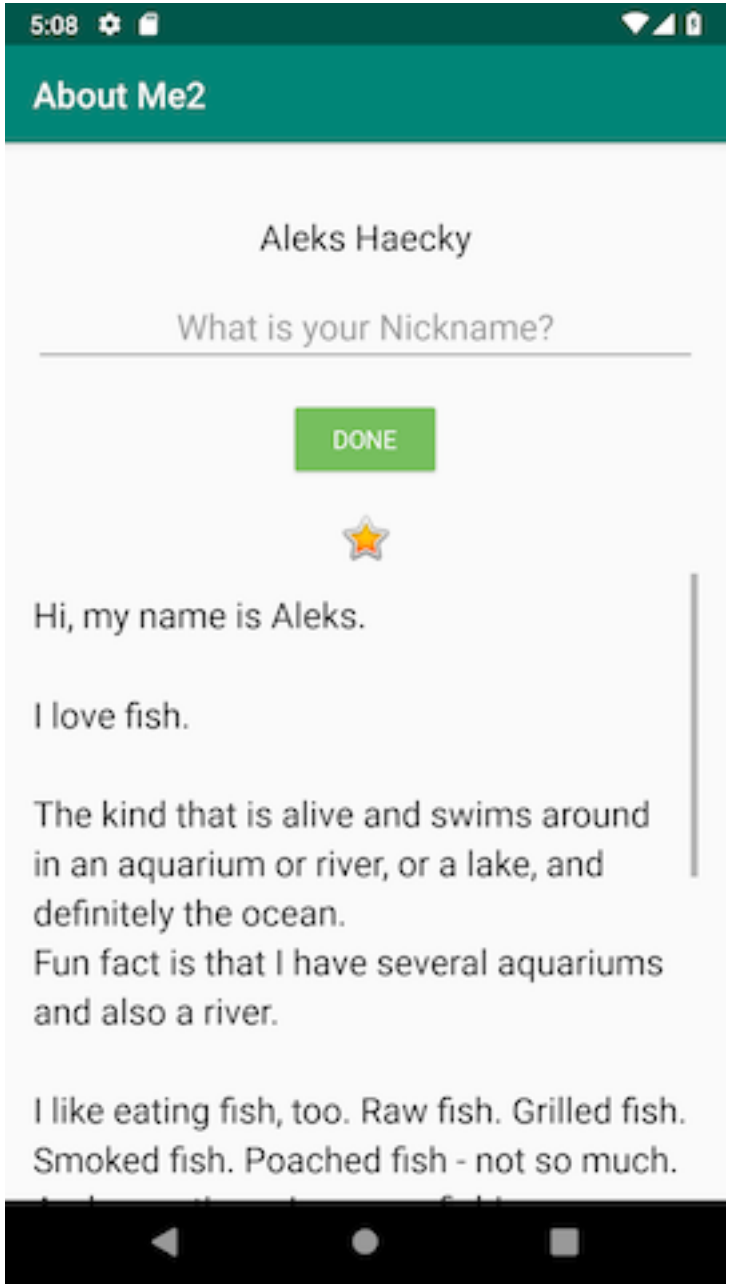
kotlin.Any
↳ android.view.View
↳ android.widget.TextView
↳ android.widget.Button

```
<Button
      android:id="@+id/button_id"
      android:layout_height="wrap_content"
      android:layout_width="wrap_content"
      android:text="@string/self_destruct" />
```


5:08
About Me2
Aleks Haecky
What is your Nickname?
DONE
Hi, my name is Aleks.
I love fish.
The kind that is alive and swims around in an aquarium or river, or a lake, and definitely the ocean.
Fun fact is that I have several aquariums and also a river.
I like eating fish, too. Raw fish. Grilled fish. Smoked fish. Poached fish - not so much.


10:46
AboutMe
Alecks Haecky
Kotlin Fish
Hi, my name is Aleks.
I love fish.
The kind that is alive and swims around in an aquarium or river, or a lake, and definitely the ocean.
Fun fact is that I have several aquariums and also a river.
I like eating fish, too. Raw fish. Grilled fish. Smoked fish. Poached fish - not so much. And sometimes I even go fishing.
And even less sometimes, I actually catch something.

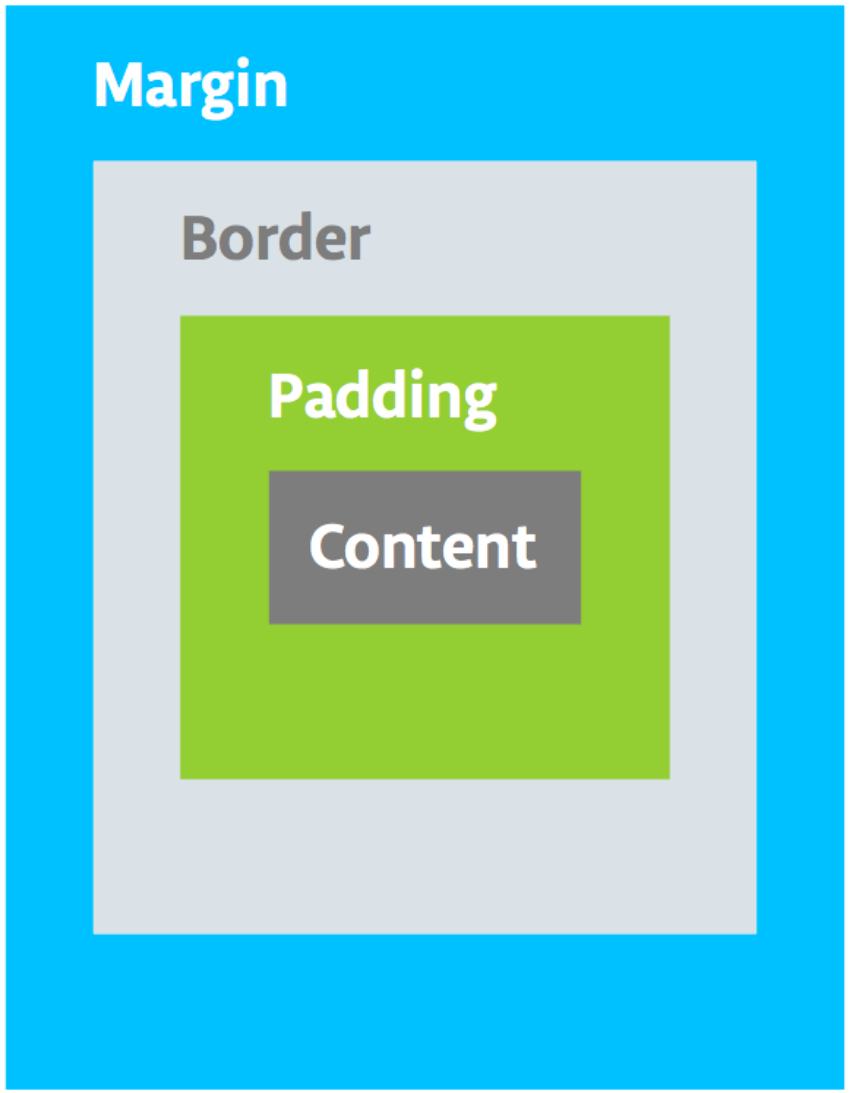

ScrollView
LinearLayout
TextView
EditText
TextView
Button
ImageView
TextView
visible
scroll to

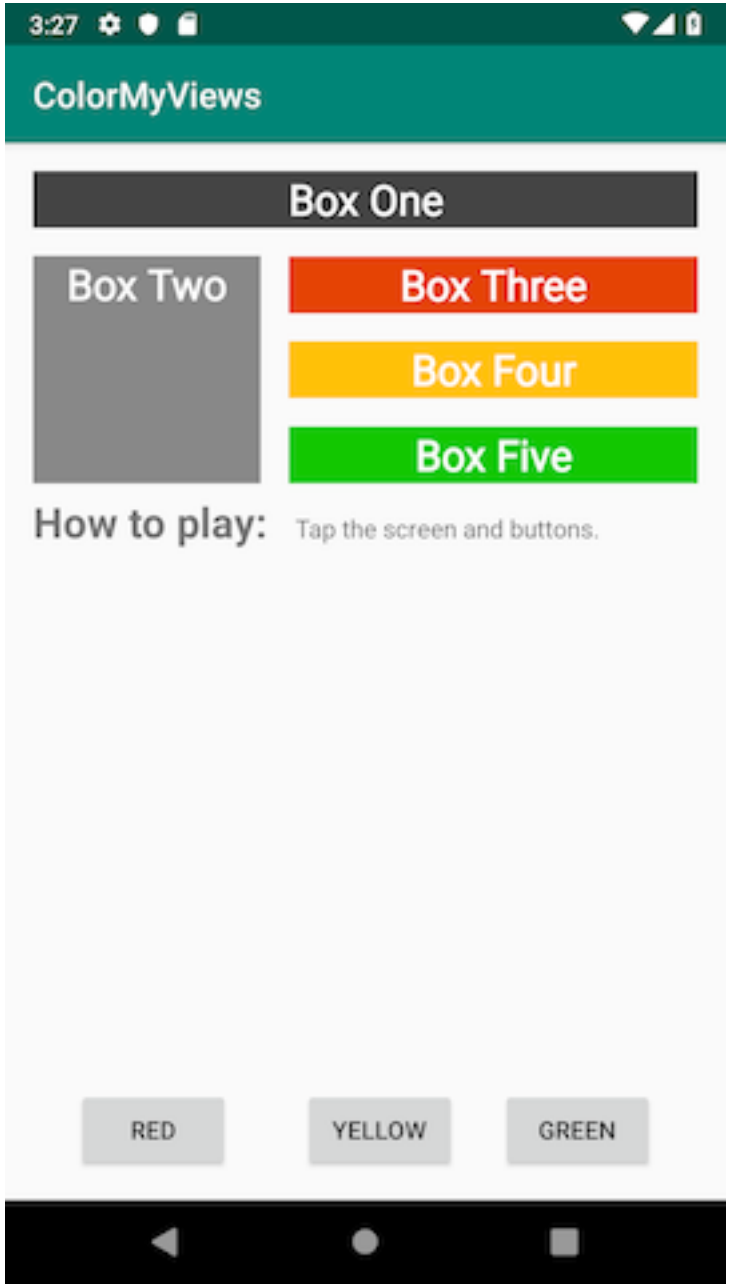




```
android:visibility="gone"
```

- visible = viditeľné
- invisible = neviditeľné (zaberá miesto)
- gone = neviditeľné (nezaberá miesto)


Margin
Border
Padding
Content




# ConstraintLayout

- pre efektívne rozloženie

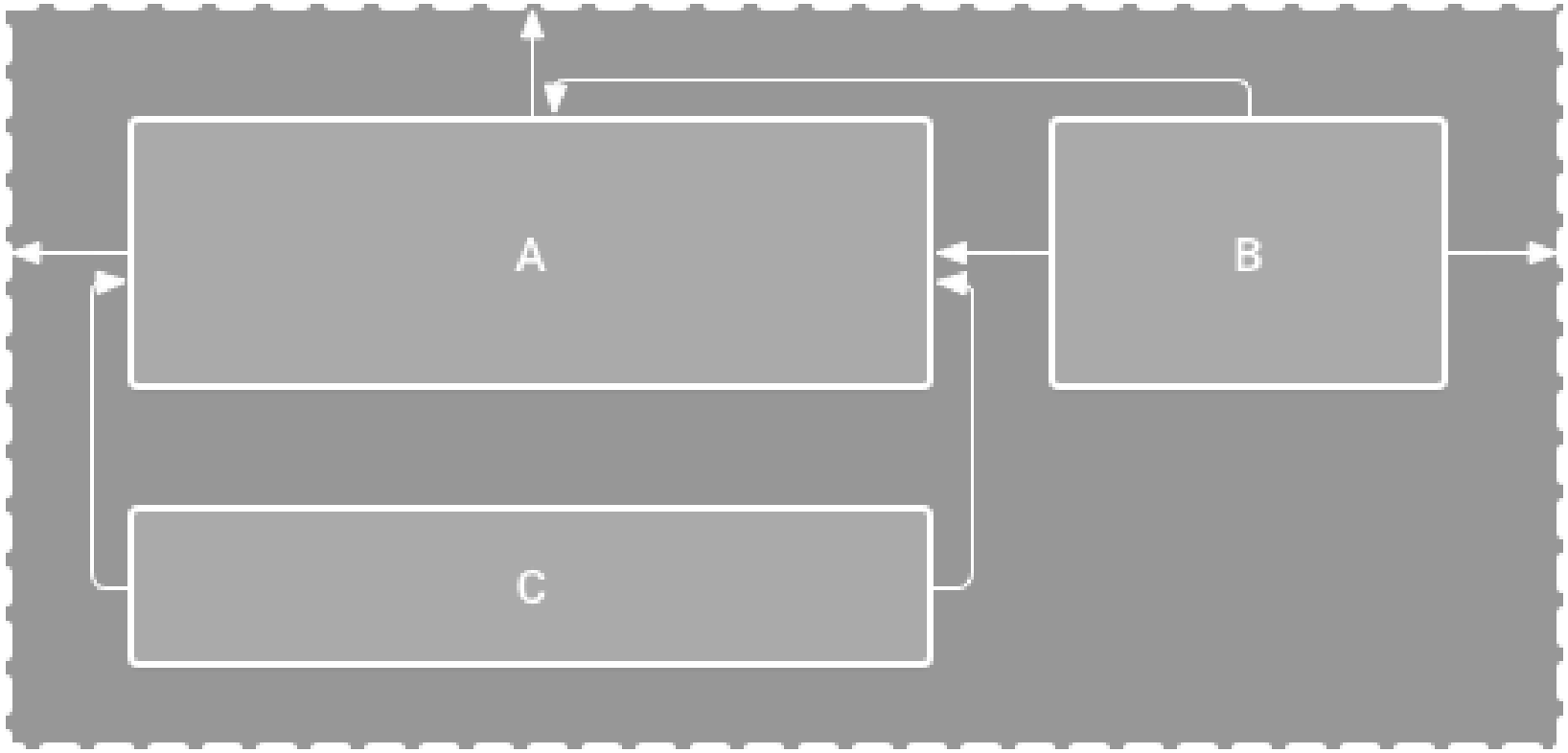
Android Developers > Docs > Reference

## ConstraintLayout

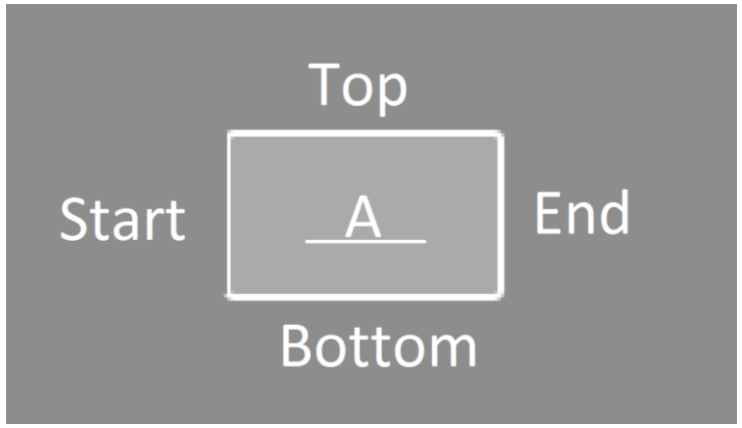
```
public class ConstraintLayout
extends ViewGroup
```

java.lang.Object
↳ ViewGroup
↳ androidx.constraintlayout.widget.ConstraintLayout

# ConstraintLayout



- layout_constraintLeft_toLeftOf
- layout_constraintLeft_toRightOf
- layout_constraintRight_toLeftOf
- layout_constraintRight_toRightOf




- layout_constraintStart_toEndOf
- layout_constraintStart_toStartOf

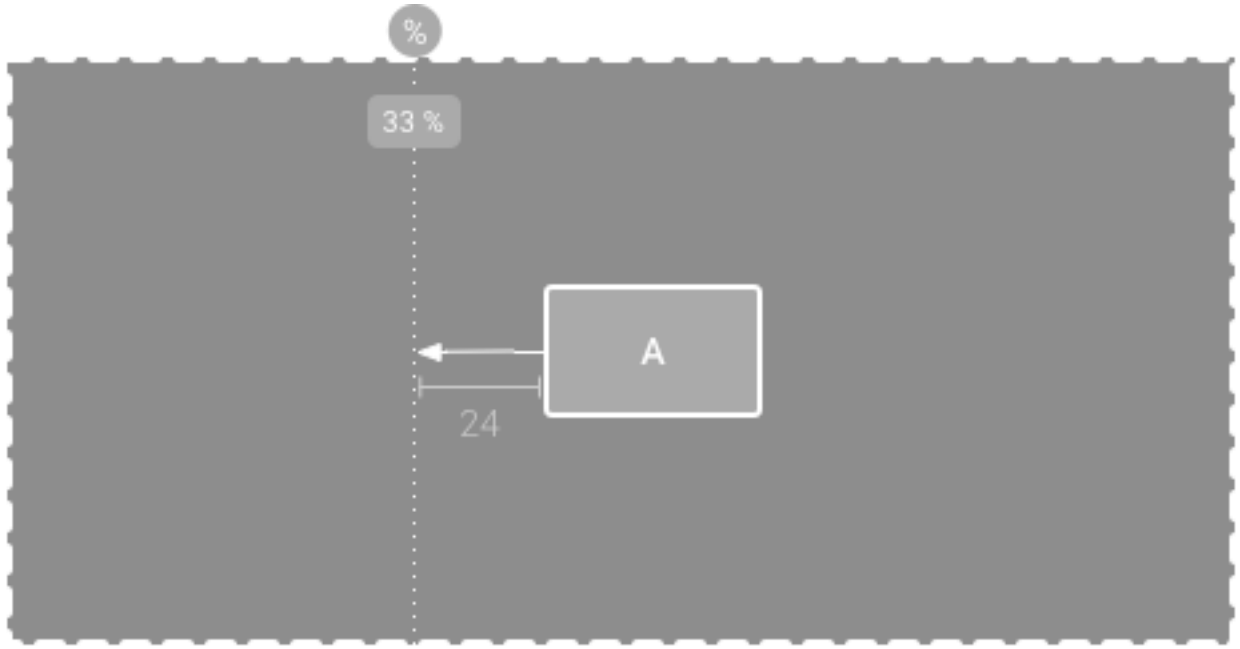
- layout_constraintEnd_toStartOf
- layout_constraintEnd_toEndOf

- layout_constraintTop_toTopOf
- layout_constraintTop_toBottomOf

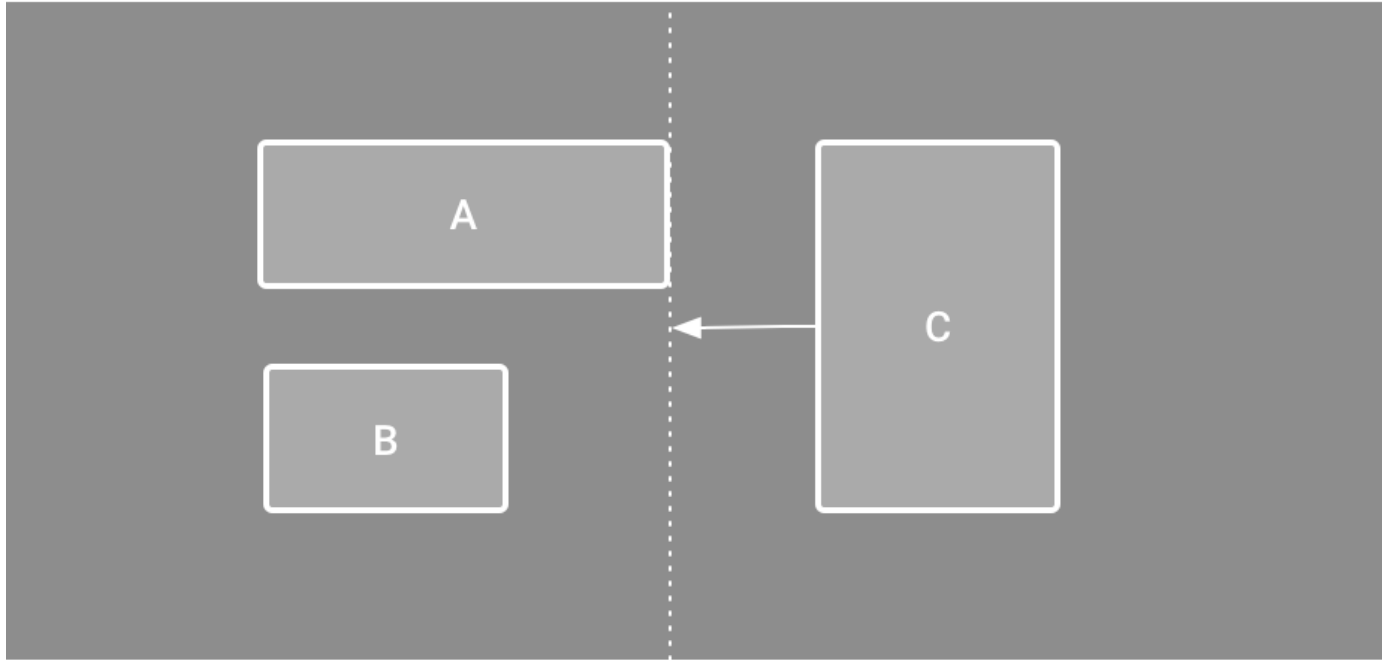
- layout_constraintBottom_toTopOf
- layout_constraintBottom_toBottomOf

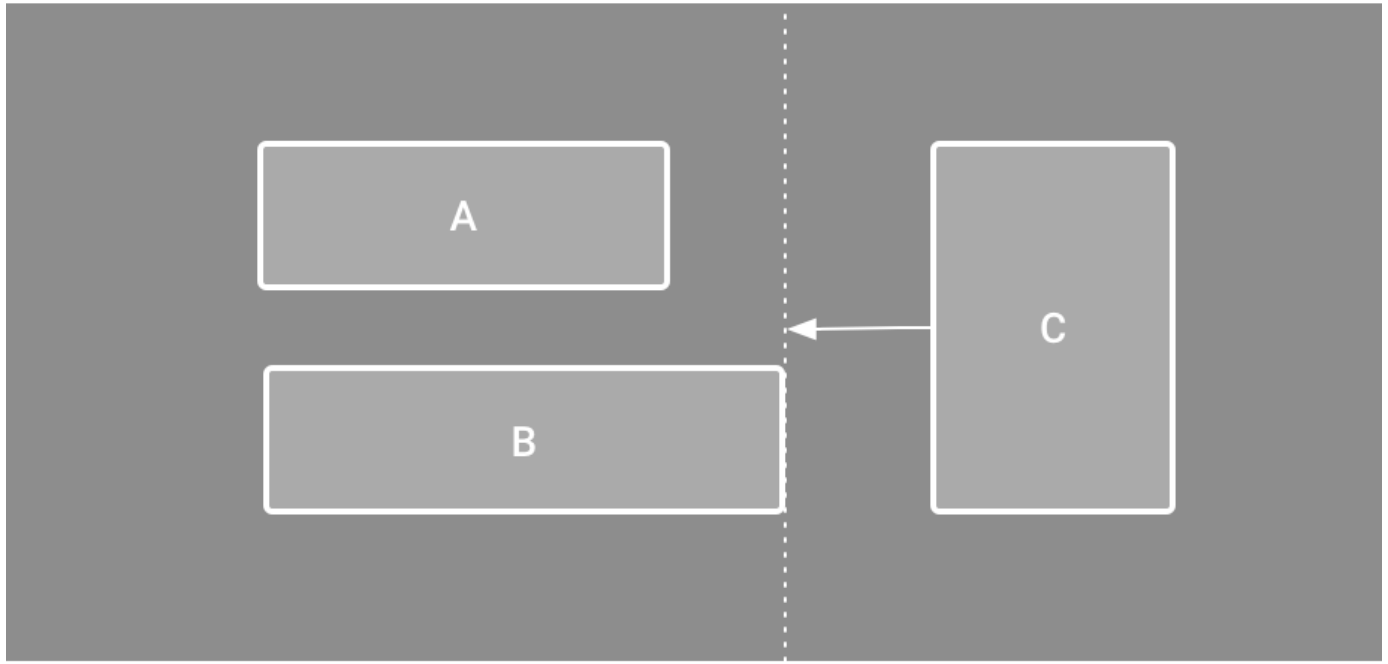
- layout_constraintBaseline_toBaselineOf

# Guideline

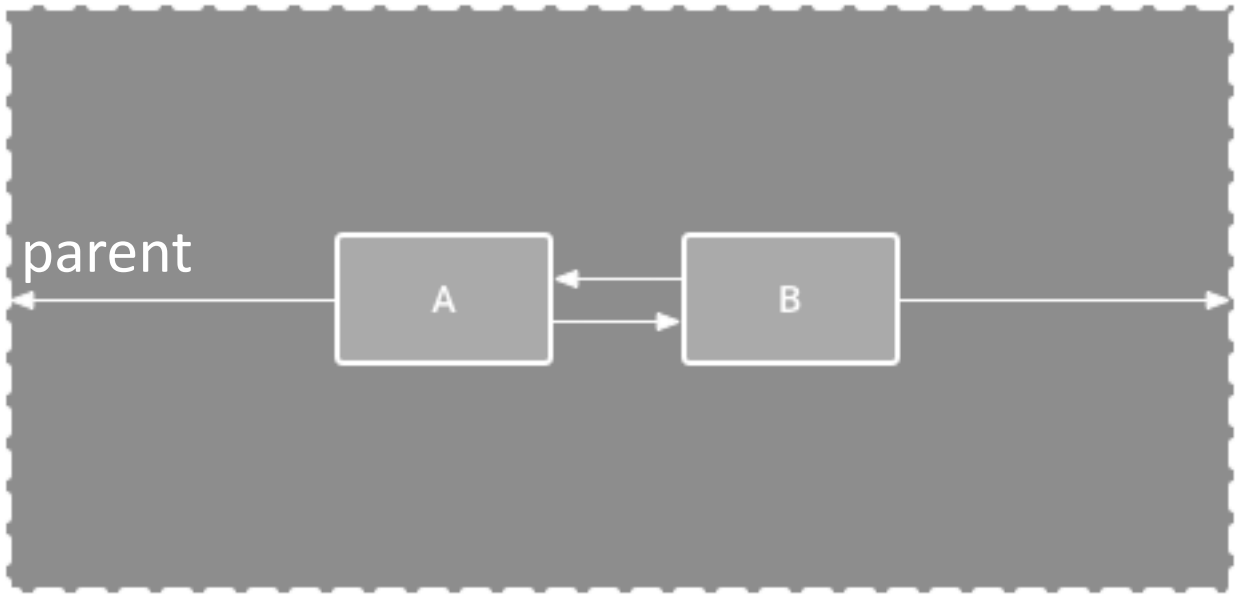


# Barier

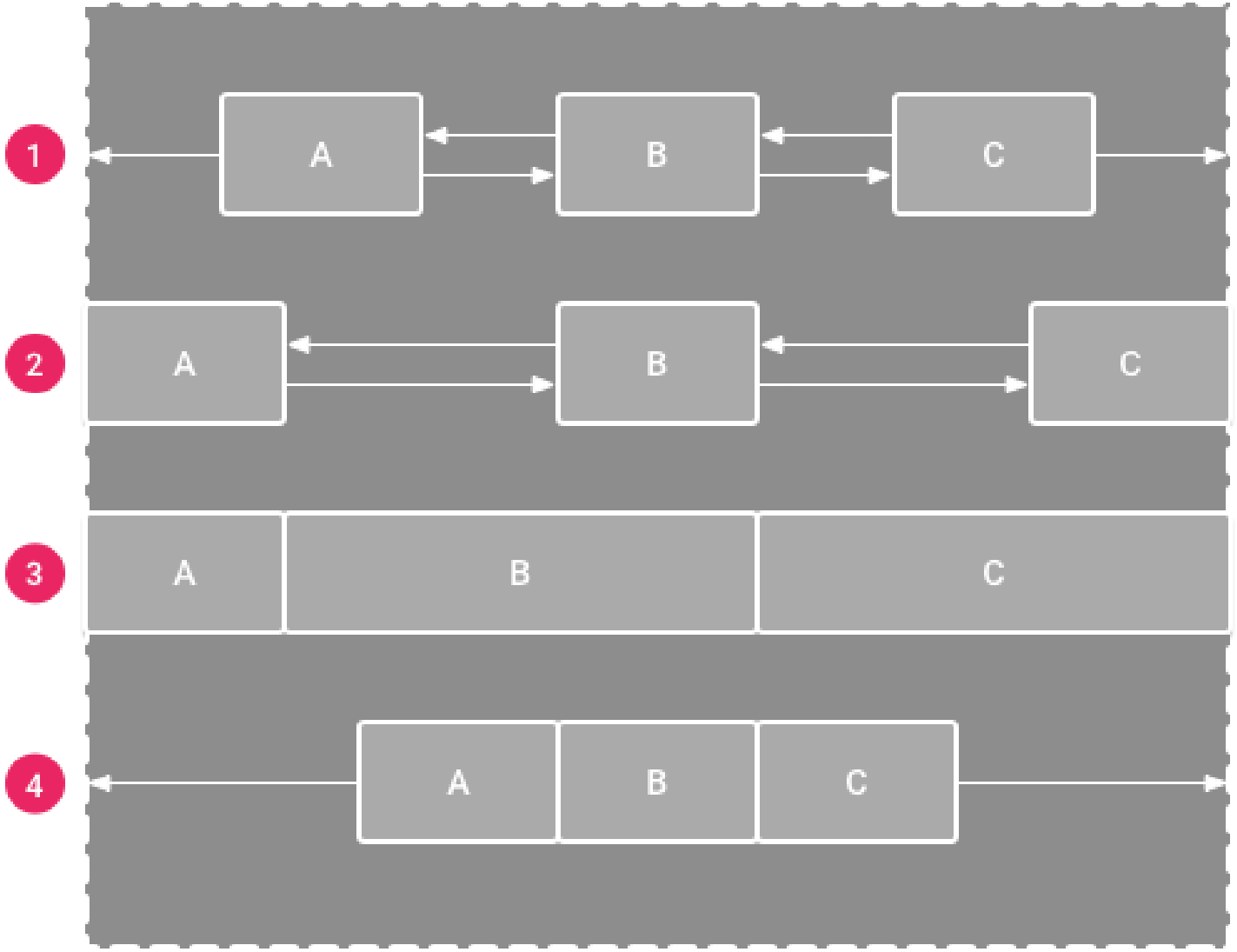


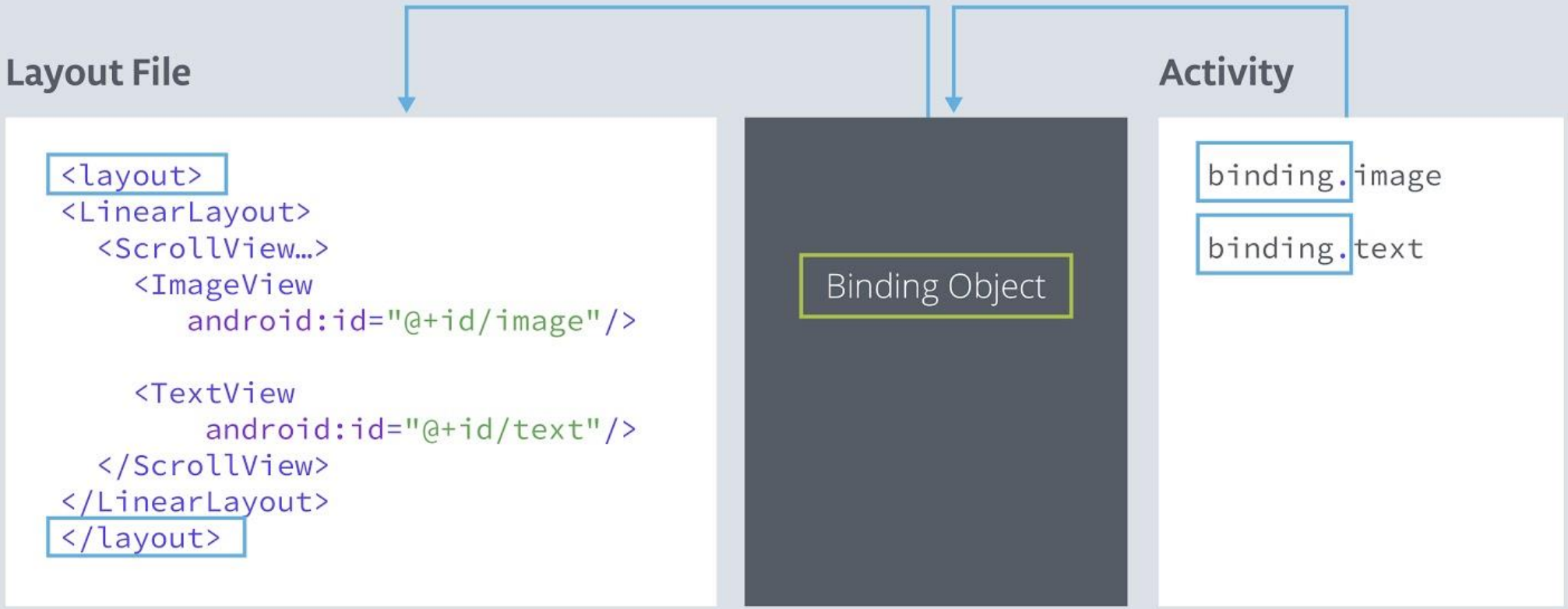
# Chain



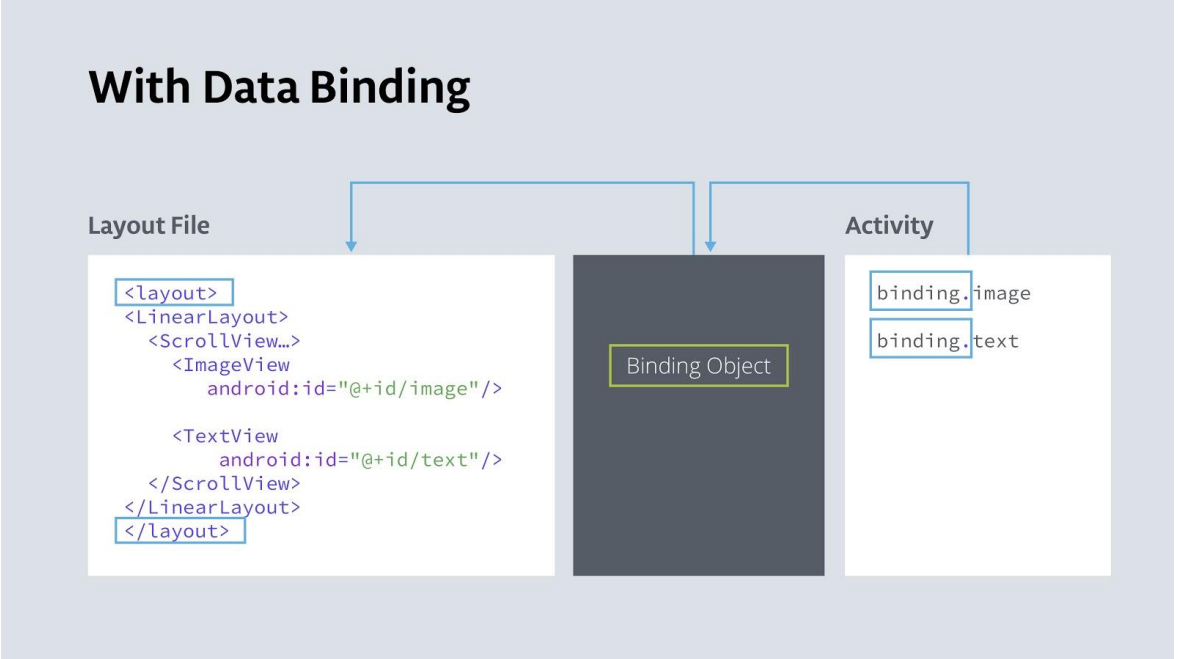
1. SPREAD
   - rovnomerne rozložené
   - predvolený štýl
2. SPREAD INSIDE
   - prvý a posledný prilepené
   - stredné rovnomerne rozložené
3. WEIGHTED
   - SPREAD alebo SPREAD INSIDE
   - nastaviť váhy pre šírku
4. PACKED
   - prilepené k sebe



# With Data Binding

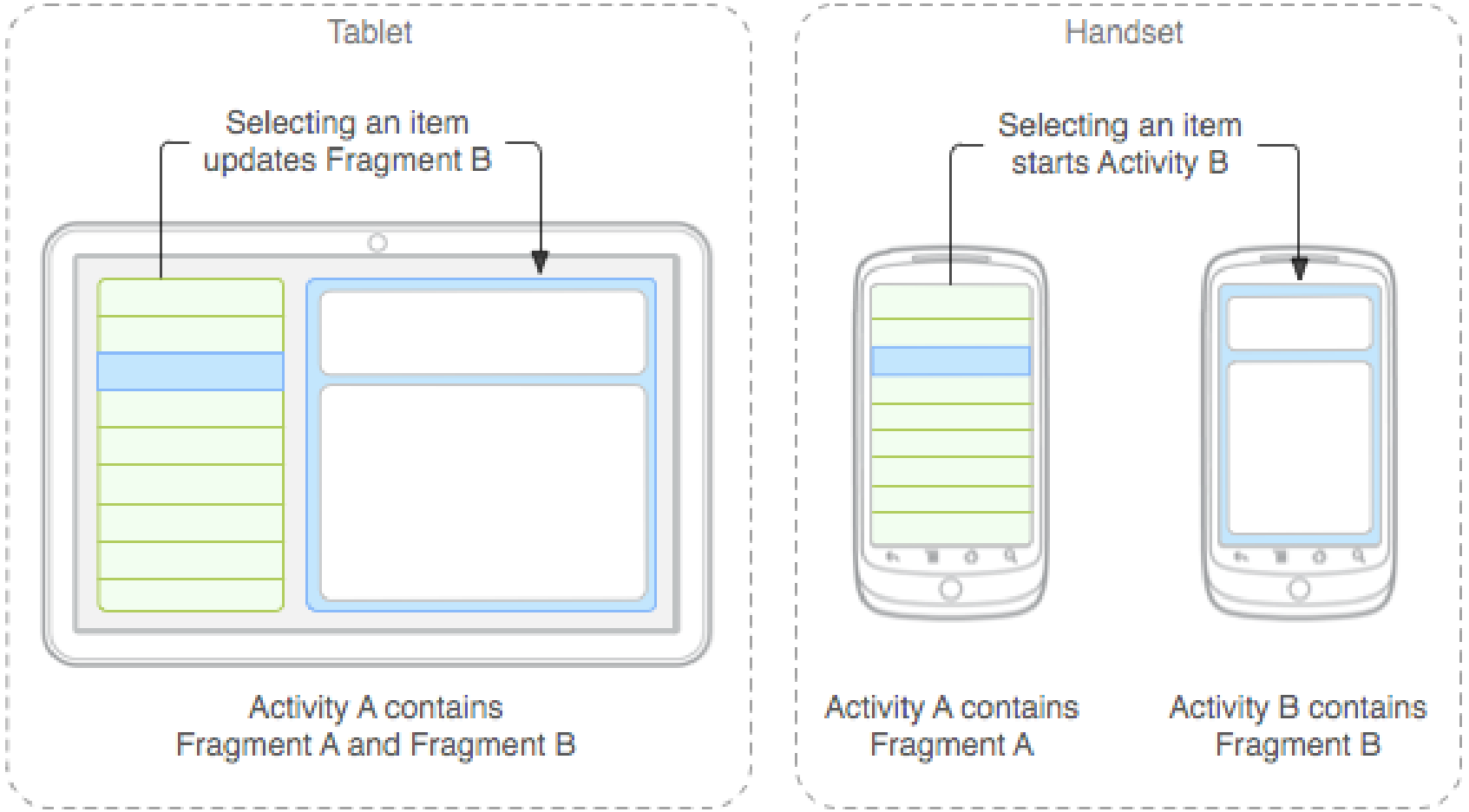


- kód je jednoduchší a na jednom mieste
- dáta a UI je oddelené
- všetky elementy sú nájdené  len raz a to pri prvom štarte, a bez následných opakovaní počas behu aplikácie
- type safety pre prístup k Views (kontrola kompilátorom pred spustením)

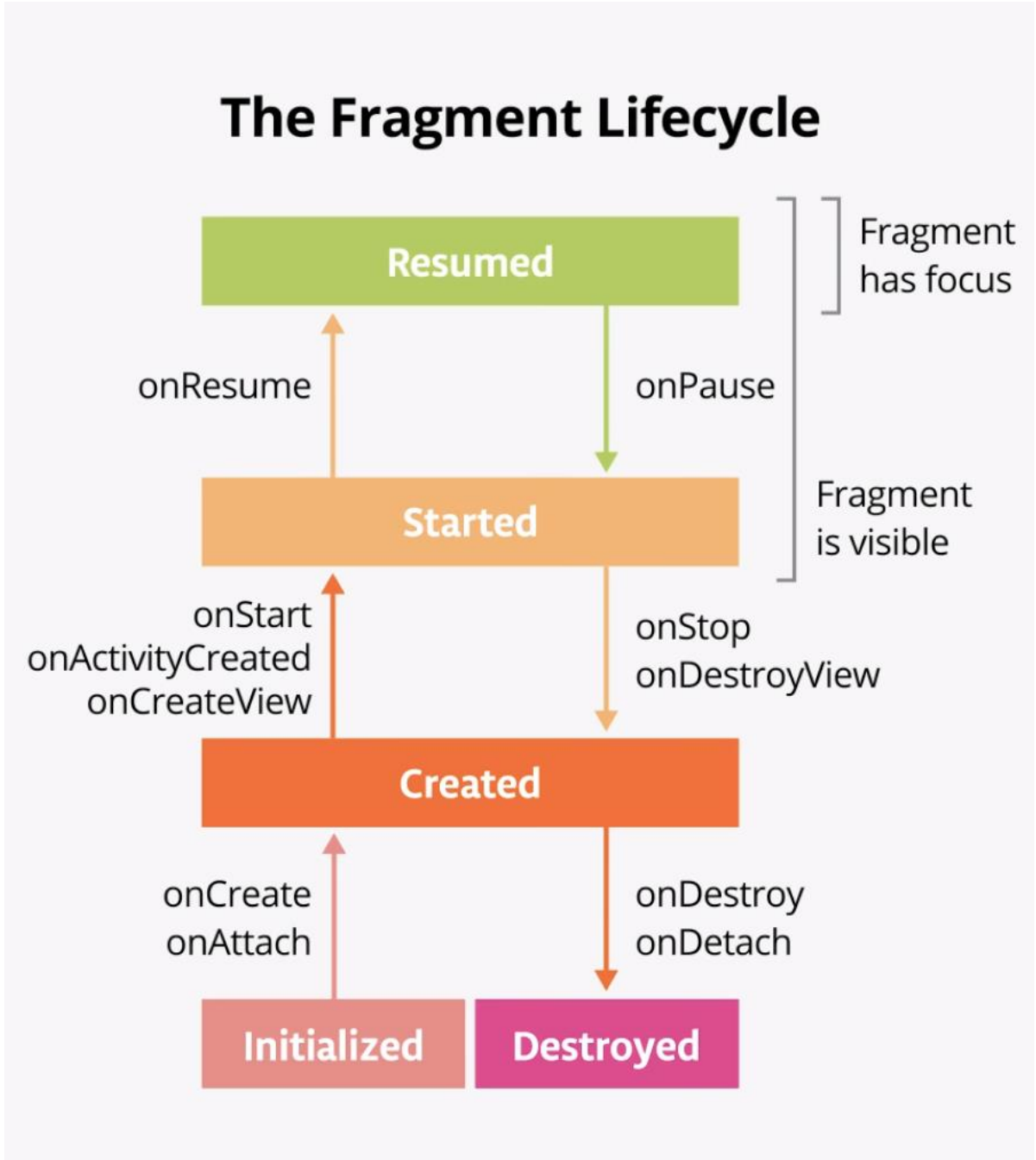


```
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <data>
        <variable
            name="myName"
            type="com.example.android.aboutme.MyName" />
    </data>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:paddingStart="16dp"
        android:paddingEnd="16dp">

        <TextView
            android:id="@+id/name_text"
            style="@style/NameStyle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@={myName.name}"
            android:textAlignment="center" />

        <TextView
            android:id="@+id/nickname_text"
            style="@style/NameStyle"
            android:text="@={myName.nickname}"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textAlignment="center"
            android:visibility="gone" />

    </LinearLayout>
</layout>
```

# Fragmenty


Tablet
Selecting an item updates Fragment B
Activity A contains Fragment A and Fragment B
Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B

# Fragmenty


The Fragment Lifecycle
Resumed
Started
Created
Initialized
Destroyed
onResume
onPause
onStart
onActivityCreated
onCreateView
onStop
onDestroyView
onCreate
onAttach
onDestroy
onDetach
Fragment has focus
Fragment is visible

# Fragmenty

```
class ExampleFragment : Fragment() {

    override fun onCreateView(
            inflater: LayoutInflater,
            container: ViewGroup?,
            savedInstanceState: Bundle?
    ): View {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.example_fragment, container, false)
    }
}
```

# Fragmenty

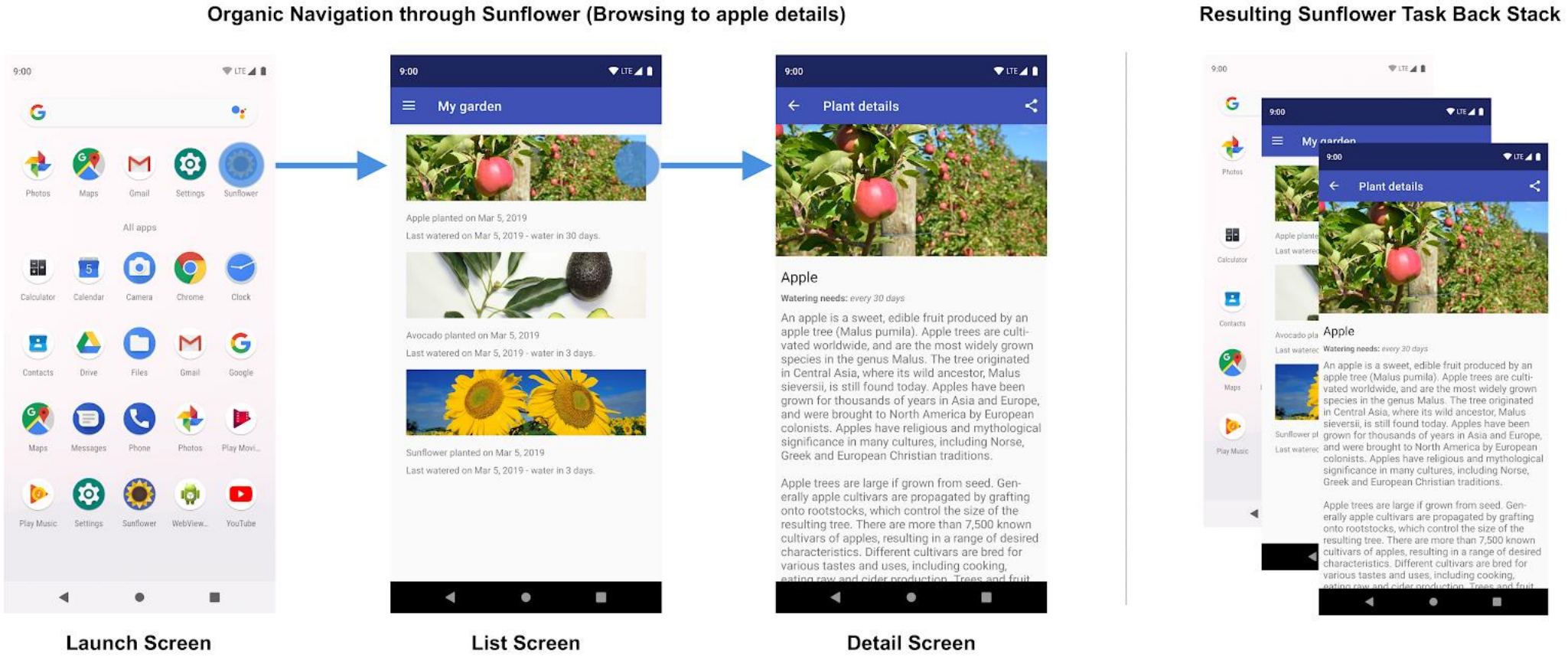
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.example.news.ArticleListFragment"
            android:id="@+id/list"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="match_parent" />
    <fragment android:name="com.example.news.ArticleReaderFragment"
            android:id="@+id/viewer"
            android:layout_weight="2"
            android:layout_width="0dp"
            android:layout_height="match_parent" />
</LinearLayout>
```

# Fragmenty

```
val fragmentManager = supportFragmentManager
val fragmentTransaction = fragmentManager.beginTransaction()
```

```
val fragment = ExampleFragment()
fragmentTransaction.add(R.id.fragment_container, fragment)
fragmentTransaction.commit()
```

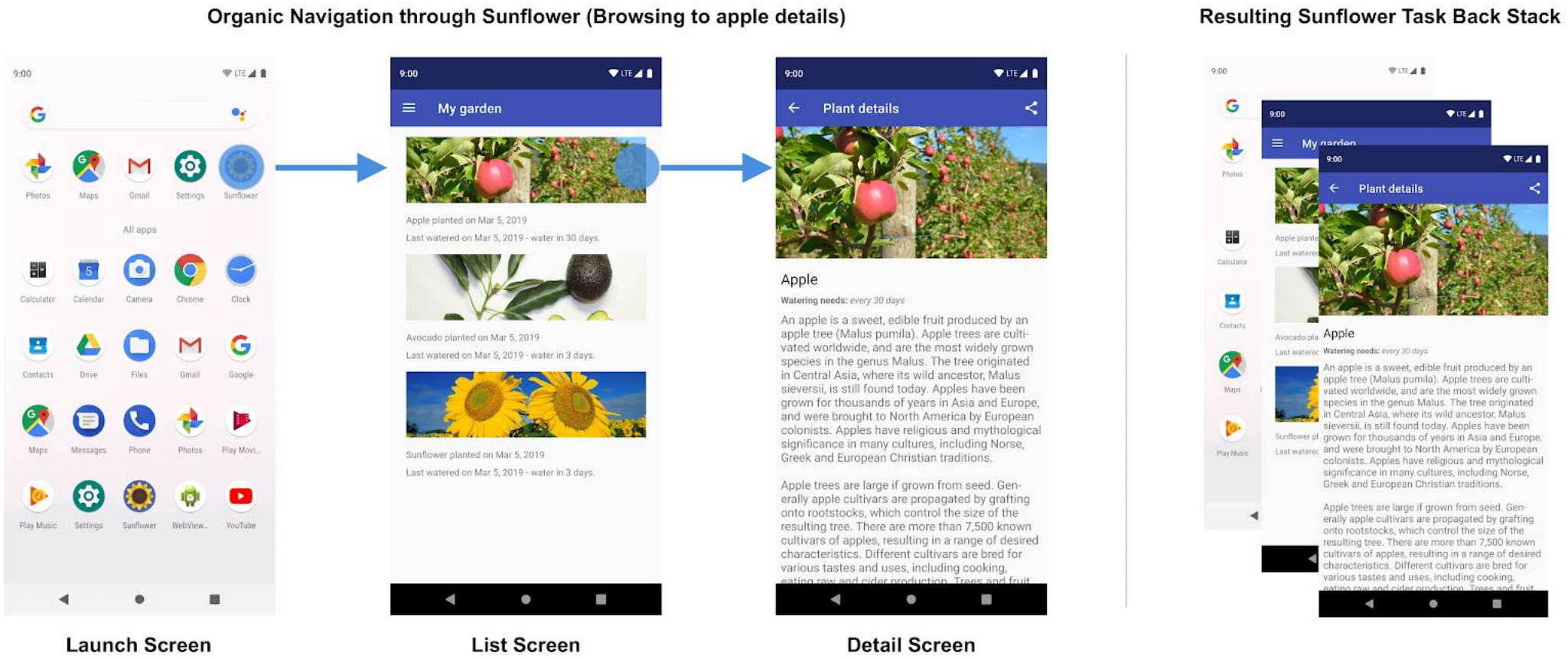
# Fragmenty


Organic Navigation through Sunflower (Browsing to apple details)
Resulting Sunflower Task Back Stack
9:00
Photos
Maps
Gmail
Settings
Sunflower
All apps
Calculator
Calendar
Camera
Chrome
Clock
Contacts
Drive
Files
Gmail
Google
Maps
Messages
Phone
Photos
Play Music
Settings
Sunflower
WebView...
YouTube
My garden
Apple planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 30 days.
Avocado planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 3 days.
Sunflower planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 3 days.
Plant details
Apple
Watering needs: every 30 days
An apple is a sweet, edible fruit produced by an apple tree (Malus pumila). Apple trees are cultivated worldwide, and are the most widely grown species in the genus Malus. The tree originated in Central Asia, where its wild ancestor, Malus sieversii, is still found today. Apples have been grown for thousands of years in Asia and Europe, and were brought to North America by European colonists. Apples have religious and mythological significance in many cultures, including Norse, Greek and European Christian traditions.
Apple trees are large if grown from seed. Generally apple cultivars are propagated by grafting onto rootstocks, which control the size of the resulting tree. There are more than 7,500 known cultivars of apples, resulting in a range of desired characteristics. Different cultivars are bred for various tastes and uses, including cooking,
Launch Screen
List Screen
Detail Screen

# Fragmenty

```
val newFragment = ExampleFragment()
val transaction = supportFragmentManager.beginTransaction()
transaction.replace(R.id.fragment_container, newFragment)
transaction.addToBackStack(null)
transaction.commit()
```

# Fragmenty


Organic Navigation through Sunflower (Browsing to apple details)
Resulting Sunflower Task Back Stack
9:00
Photos
Maps
Gmail
Settings
Sunflower
All apps
Calculator
Calendar
Camera
Chrome
Clock
Contacts
Drive
Files
Gmail
Google
Maps
Messages
Phone
Photos
Play Music
Settings
Sunflower
WebView...
YouTube
My garden
Apple planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 30 days.
Avocado planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 3 days.
Sunflower planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 3 days.
Plant details
Apple
Watering needs: every 30 days
An apple is a sweet, edible fruit produced by an apple tree (Malus pumila). Apple trees are cultivated worldwide, and are the most widely grown species in the genus Malus. The tree originated in Central Asia, where its wild ancestor, Malus sieversii, is still found today. Apples have been grown for thousands of years in Asia and Europe, and were brought to North America by European colonists. Apples have religious and mythological significance in many cultures, including Norse, Greek and European Christian traditions.
Apple trees are large if grown from seed. Generally apple cultivars are propagated by grafting onto rootstocks, which control the size of the resulting tree. There are more than 7,500 known cultivars of apples, resulting in a range of desired characteristics. Different cultivars are bred for various tastes and uses, including cooking,
Launch Screen
List Screen
Detail Screen

# Intenty

- Implicitné

```
val webIntent: Intent = Uri.parse("http://www.android.com").let { webpage ->
    Intent(Intent.ACTION_VIEW, webpage)
}
```

```
val callIntent: Intent = Uri.parse("tel:5551234").let { number ->
    Intent(Intent.ACTION_DIAL, number)
}
```

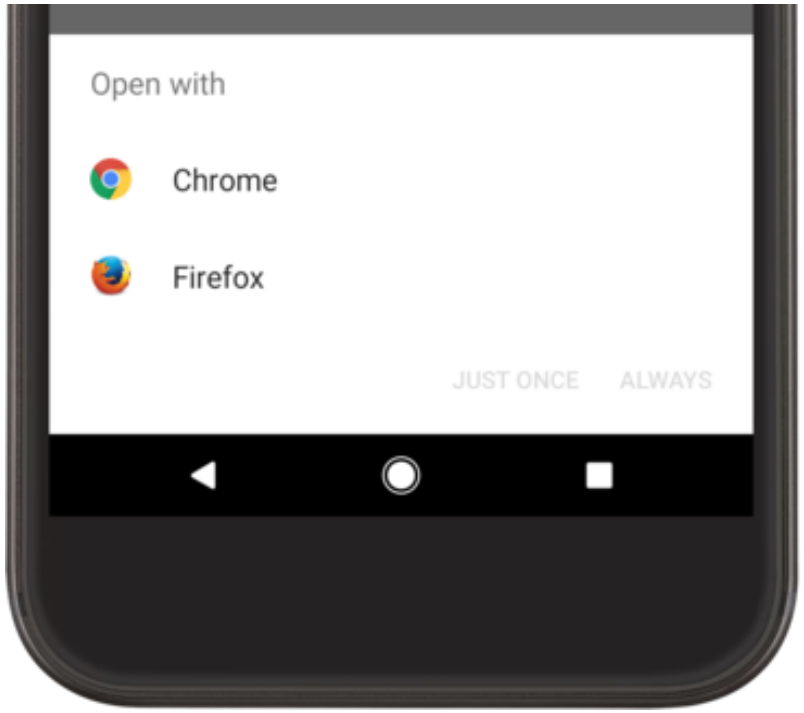
- Explicitné

```
val explicitIntent: Intent = Intent(context,MojaActivita::class.java)
```

# Intenty – implicitné

```
// Build the intent
val location = Uri.parse("geo:0,0?q=1600+Amphitheatre+Parkway,+Mountain+View,+California")
val mapIntent = Intent(Intent.ACTION_VIEW, location)

// Verify it resolves
val activities: List<ResolveInfo> = packageManager.queryIntentActivities(mapIntent, 0)
val isIntentSafe: Boolean = activities.isNotEmpty()

// Start an activity if it's safe
if (isIntentSafe) {
    startActivity(mapIntent)
}
```

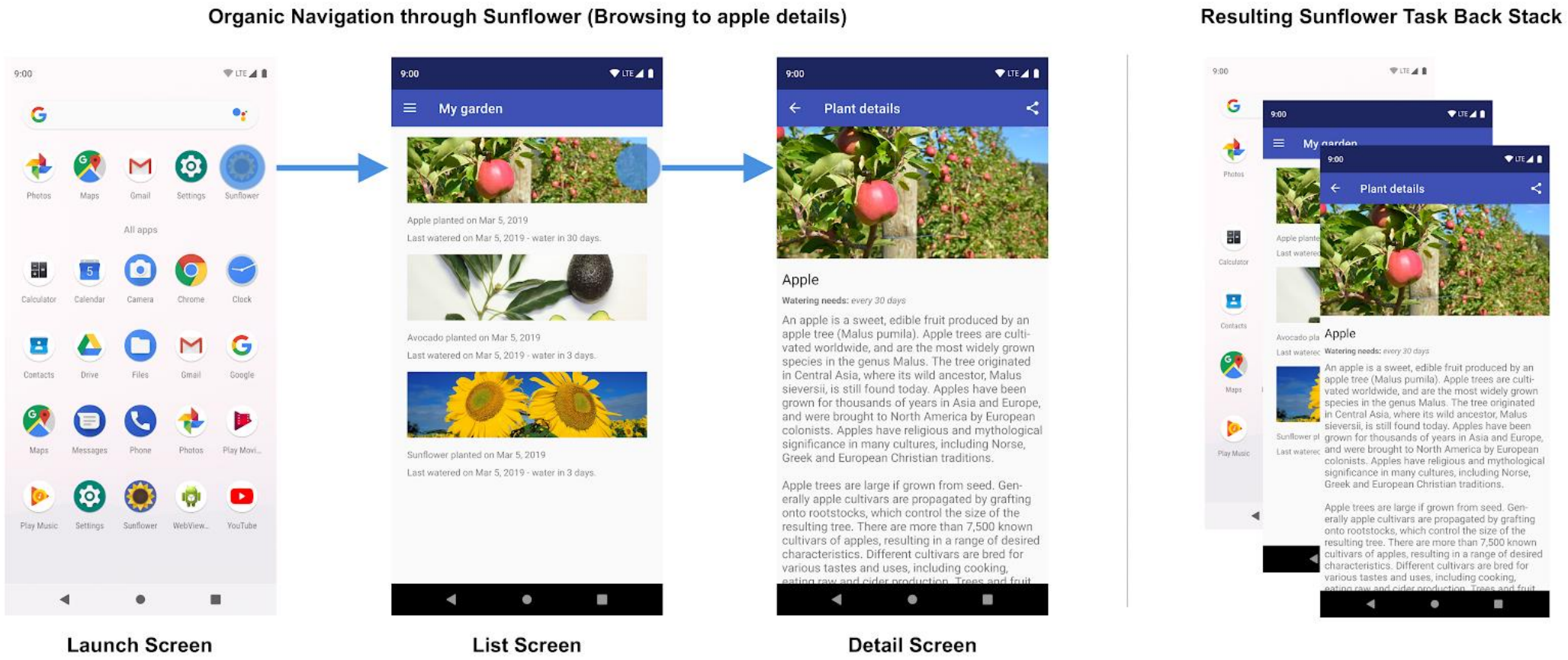
# Intenty (komunikácia)

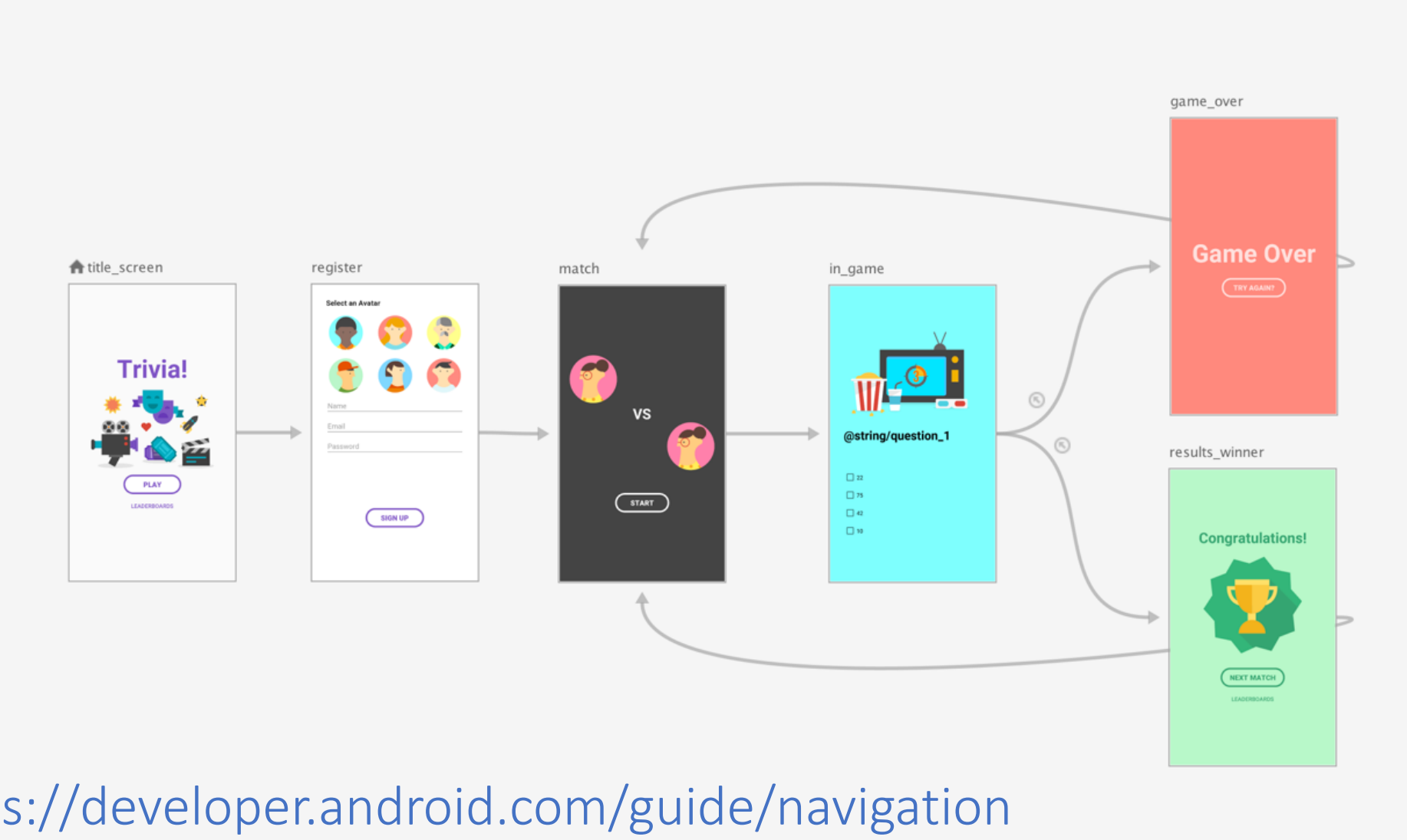


# Intenty (komunikácia)

```
val intent = Intent(Intent.ACTION_SEND)
...

// Always use string resources for UI text.
// This says something like "Share this photo with"
val title = resources.getString(R.string.chooser_title)
// Create intent to show chooser
val chooser = Intent.createChooser(intent, title)

// Verify the intent will resolve to at least one activity
if (intent.resolveActivity(packageManager) != null) {
    startActivity(chooser)
}
```

# Navigácia

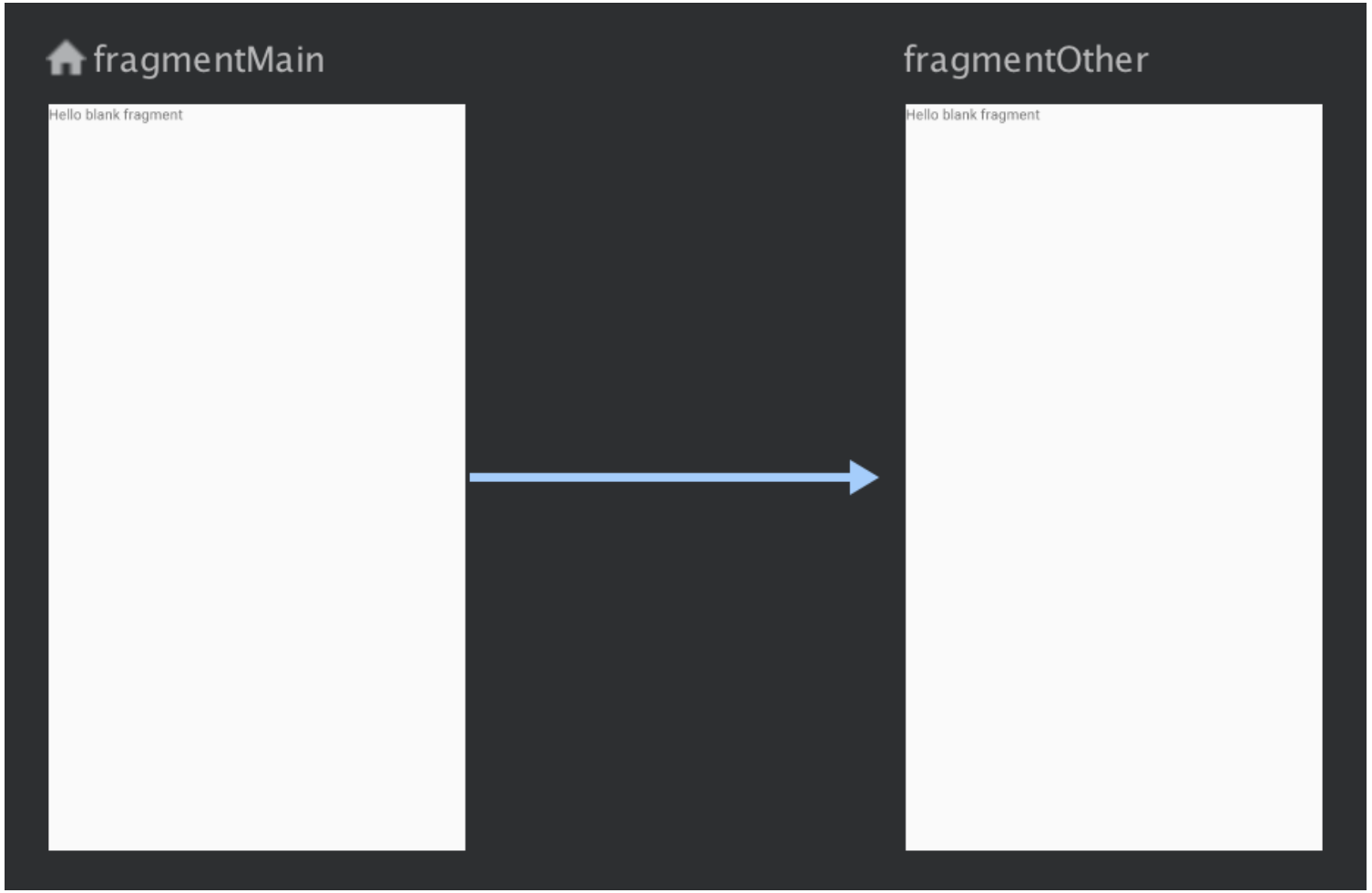

Organic Navigation through Sunflower (Browsing to apple details)
Resulting Sunflower Task Back Stack
9:00
Photos
Maps
Gmail
Settings
Sunflower
All apps
Calculator
Calendar
Camera
Chrome
Clock
Contacts
Drive
Files
Gmail
Google
Maps
Messages
Phone
Photos
Play Movi...
Play Music
Settings
Sunflower
WebView...
YouTube
My garden
Apple planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 30 days.
Avocado planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 3 days.
Sunflower planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 3 days.
Plant details
Apple
Watering needs: every 30 days
An apple is a sweet, edible fruit produced by an apple tree (Malus pumila). Apple trees are cultivated worldwide, and are the most widely grown species in the genus Malus. The tree originated in Central Asia, where its wild ancestor, Malus sieversii, is still found today. Apples have been grown for thousands of years in Asia and Europe, and were brought to North America by European colonists. Apples have religious and mythological significance in many cultures, including Norse, Greek and European Christian traditions.
Apple trees are large if grown from seed. Generally apple cultivars are propagated by grafting onto rootstocks, which control the size of the resulting tree. There are more than 7,500 known cultivars of apples, resulting in a range of desired characteristics. Different cultivars are bred for various tastes and uses, including cooking,
Launch Screen
List Screen
Detail Screen

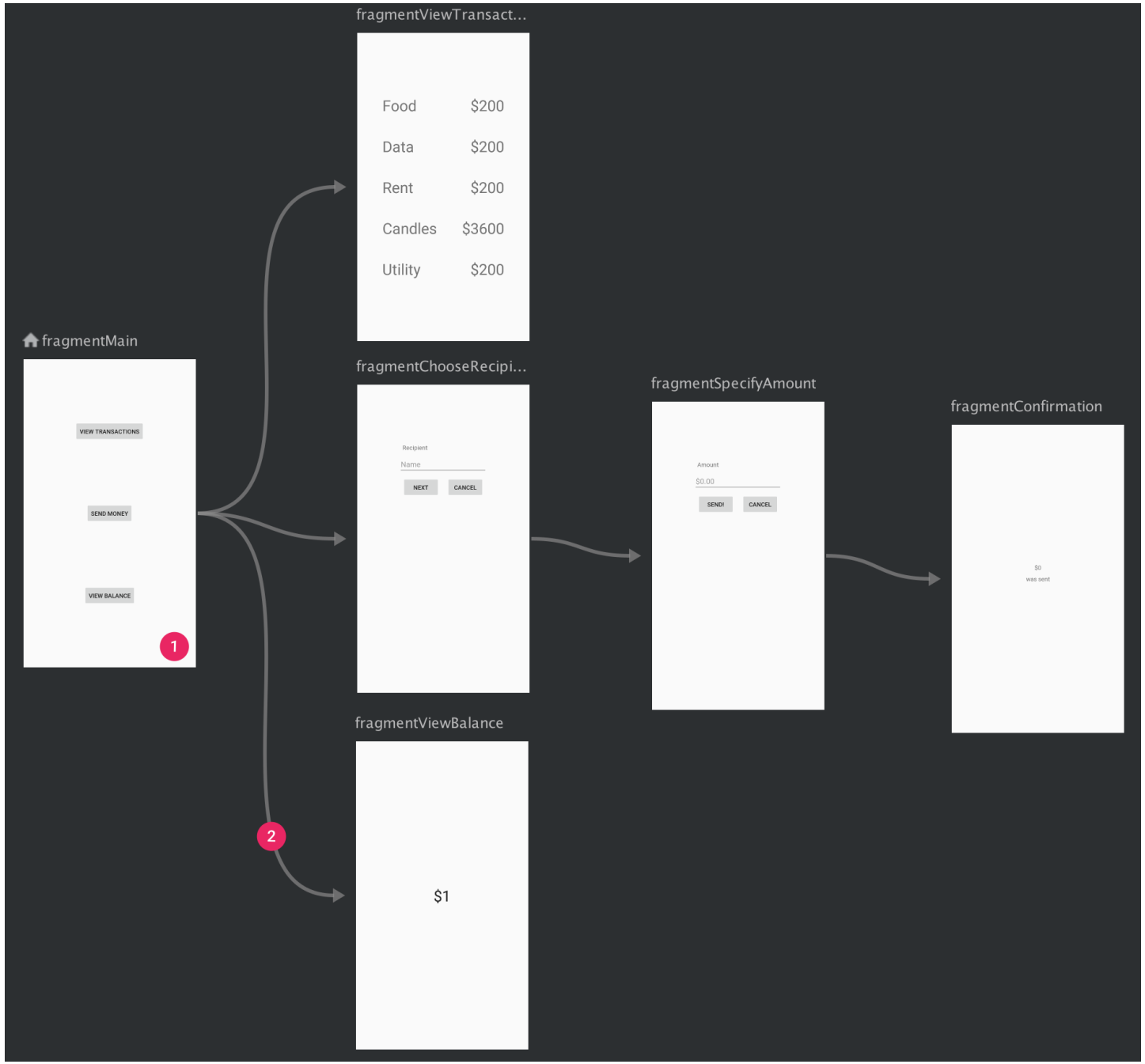



https://developer.android.com/guide/navigation

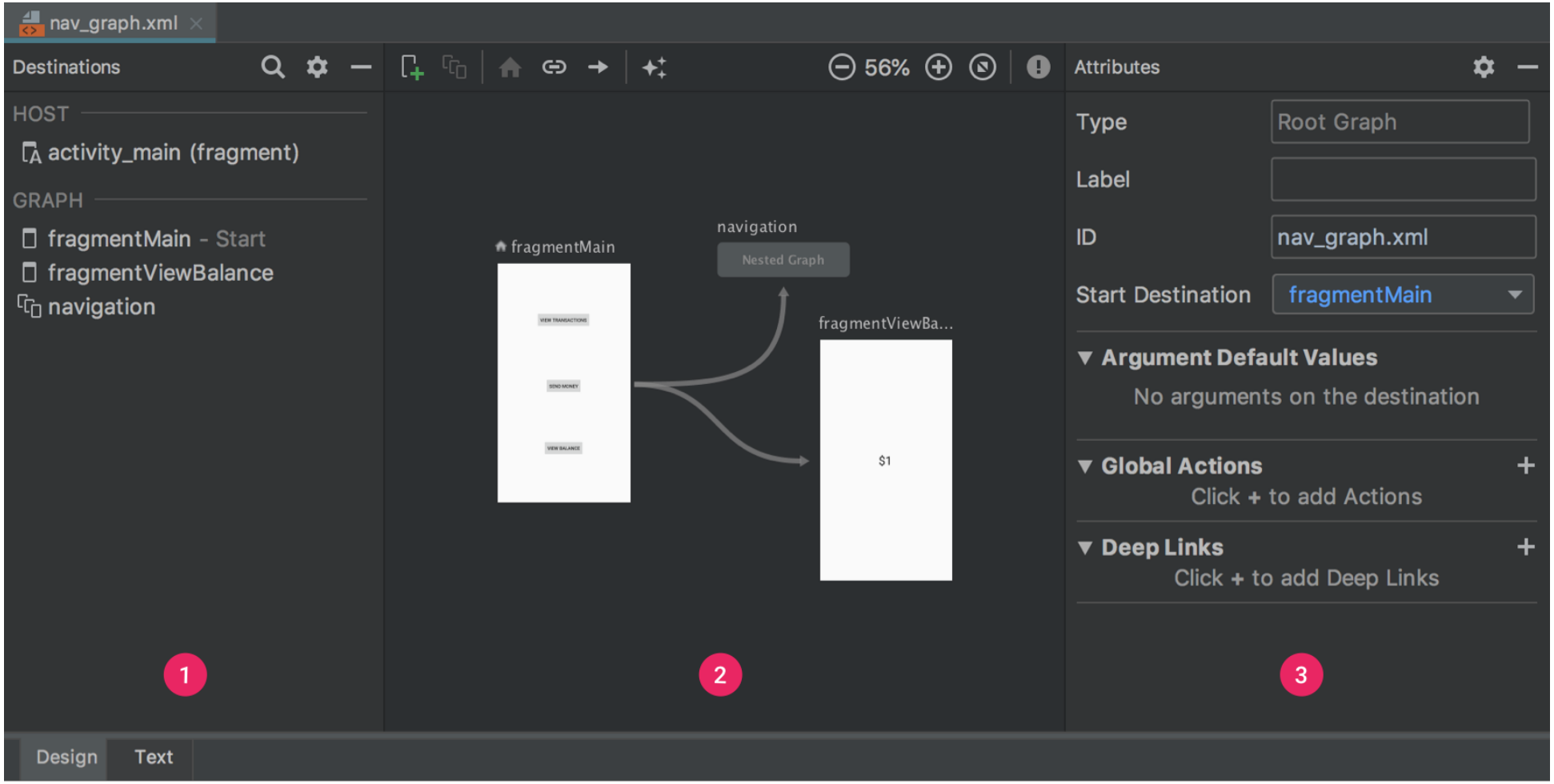
https://codelabs.developers.google.com/codelabs/android-navigation

# Navigácia


fragmentMain
Hello blank fragment
fragmentOther
Hello blank fragment


fragmentViewTransact...
Food $200
Data $200
Rent $200
Candles $3600
Utility $200
fragmentMain
VIEW TRANSACTIONS
SEND MONEY
VIEW BALANCE
1
fragmentChooseRecipi...
Recipient
Name
NEXT
CANCEL
fragmentSpecifyAmount
Amount
$0.00
SEND!
CANCEL
fragmentConfirmation
$0
was sent
2
fragmentViewBalance
$1

# Navigácia


nav_graph.xml
Destinations
HOST
activity_main (fragment)
GRAPH
fragmentMain - Start
fragmentViewBalance
navigation
56%
fragmentMain
navigation
Nested Graph
fragmentViewBa...
$1
Attributes
Type
Root Graph
Label
ID
nav_graph.xml
Start Destination
fragmentMain
Argument Default Values
No arguments on the destination
Global Actions
Click + to add Actions
Deep Links
Click + to add Deep Links
1
2
3
Design
Text

# Navigácia

```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:android="http://schemas.android.com/apk/res/android"
    app:startDestination="@id/blankFragment">
    <fragment
        android:id="@+id/blankFragment"
        android:name="com.example.cashdog.cashdog.BlankFragment"
        android:label="fragment_blank"
        tools:layout="@layout/fragment_blank" >
        <action
            android:id="@+id/action_blankFragment_to_blankFragment2"
            app:destination="@id/blankFragment2" />
    </fragment>
    <fragment
        android:id="@+id/blankFragment2"
        android:name="com.example.cashdog.cashdog.BlankFragment2"
        android:label="fragment_blank_fragment2"
        tools:layout="@layout/fragment_blank_fragment2" />
</navigation>
```

# Navigácia

```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:app="http://schemas.android.com/apk/res-auto"
            xmlns:tools="http://schemas.android.com/tools"
            xmlns:android="http://schemas.android.com/apk/res/android"
            android:id="@+id/main_nav"
            app:startDestination="@id/mainFragment">

  ...

  <action android:id="@+id/action_global_mainFragment"
          app:destination="@id/mainFragment"/>

</navigation>
```

# Navigácia

- `Fragment.findNavController()`
- `View.findNavController()`
- `Activity.findNavController(viewId: Int)`

```
viewTransactionsButton.setOnClickListener { view ->
    view.findNavController().navigate(R.id.viewTransactionsAction)
}
```


https://developer.android.com/guide/navigation

https://codelabs.developers.google.com/codelabs/android-navigation

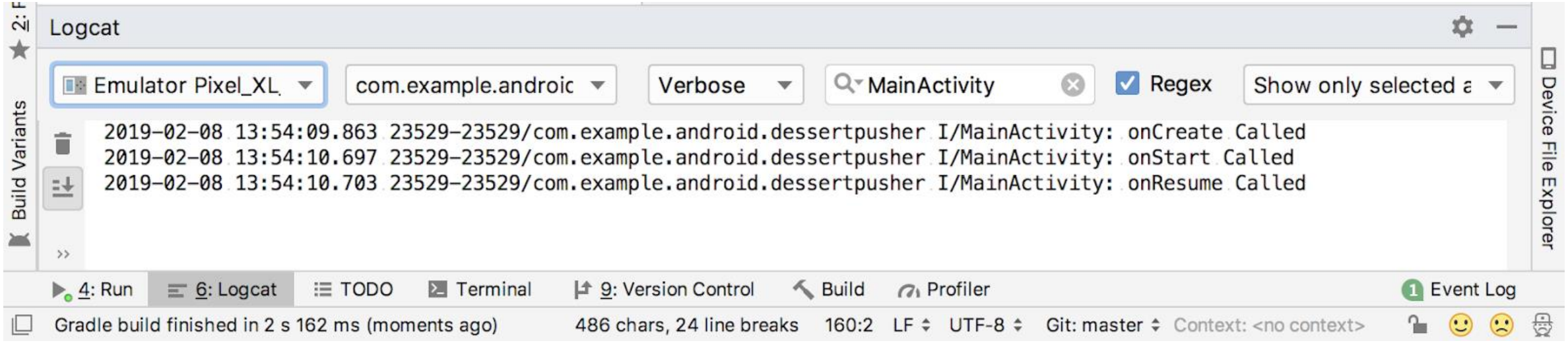
# Logovanie

- `Log.i("MainActivity", “moja informacia")`
- `Log.e("MainActivity", “moja chyba")`
- `Log.w("MainActivity", “moje upozornenie“)`
- `Log.d("MainActivity", “moje ladenie")`

# Logovanie


Build Variants
evice File Explorer
4: Ru
6: Logcat
TODO
Terminal
9: Version Control
Build
Profiler
Event Log
Gradle
6 ms (moments ago)
2:1
LF
UTF-8
Git: master
Context: <no context>

# Logovanie


Logcat
Emulator Pixel_XL
com.example.androic
Verbose
MainActivity
Regex
Show only selected a
2019-02-08 13:54:09.863 23529-23529/com.example.android.dessertpusher I/MainActivity: onCreate Called
2019-02-08 13:54:10.697 23529-23529/com.example.android.dessertpusher I/MainActivity: onStart Called
2019-02-08 13:54:10.703 23529-23529/com.example.android.dessertpusher I/MainActivity: onResume Called
Build Variants
Device File Explorer
4: Run
6: Logcat
TODO
Terminal
9: Version Control
Build
Profiler
Event Log
Gradle build finished in 2 s 162 ms (moments ago)
486 chars, 24 line breaks
160:2
LF
UTF-8
Git: master
Context: <no context>

# Terminal


```
adb shell am kill com.example.android.aplikacia
```

# Mobilné výpočty

Ing. Maroš Čavojský, PhD.

maros.cavojsky@stuba.sk
C606