



Android Studio

" We know fragments already,
Let's work with Navigation Component"

INNEHÅLLSFÖRTECKNING

01

Översikt

02

Navigation
Graph,
Navigation
Component

03

Constraint
Layout #2

04

Uppgifter
&
Övningar

UPPDATERAD LAB #2 BESKRIVNING

01

ÖVERSIKT

What's wrong with INTENT?



Intent...

Varför är det fel med Intent?

- Inte fel, bättre alternativ för navigering!
- Fragments + komponenter
- Intuitiv design och navigering
- Grafisk design



02

Navigation Graph

Navigation Graph



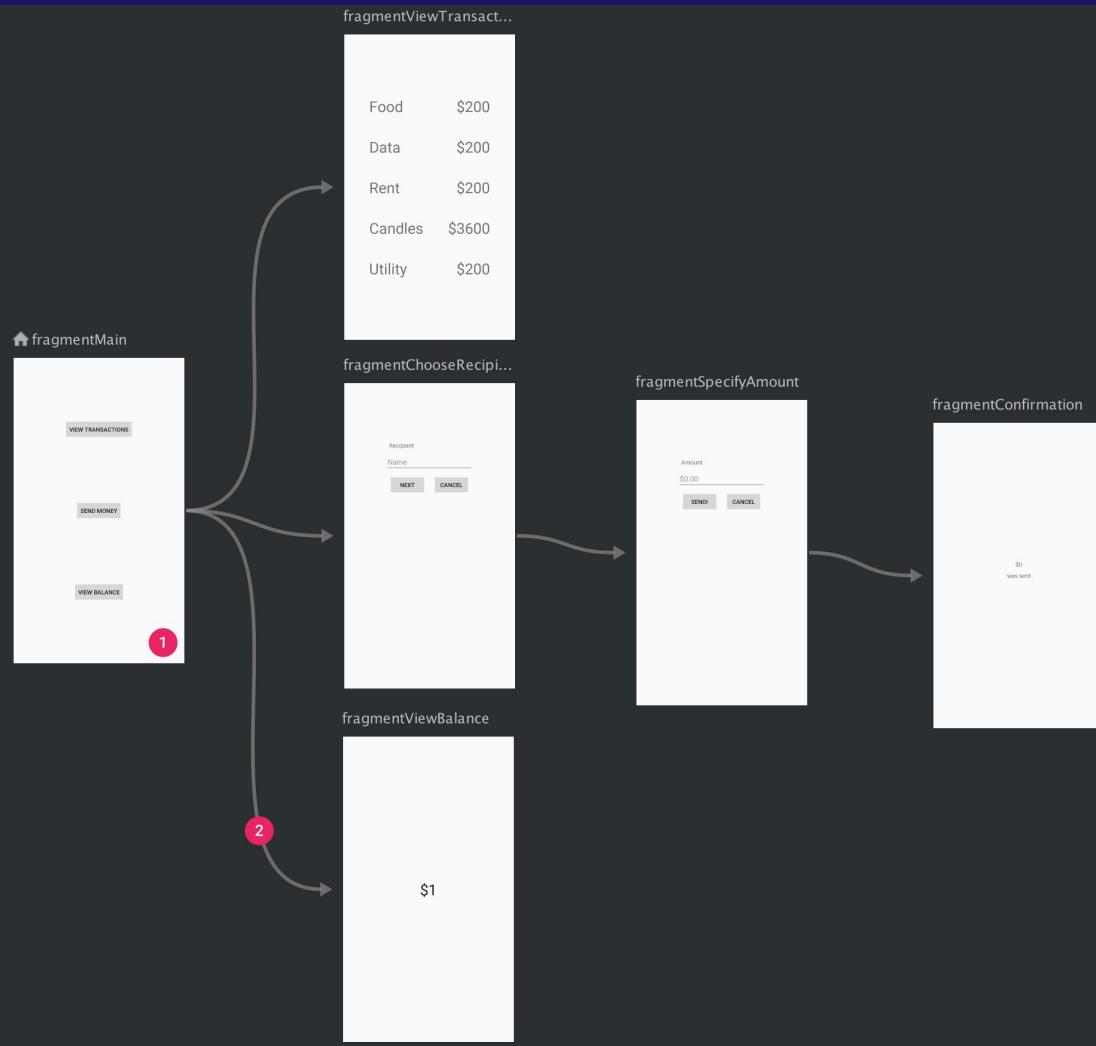


*“A **navigation graph** is a resource file that contains all of your destinations and actions.*

The graph represents all of your app's navigation paths.”

- Android Developers docs

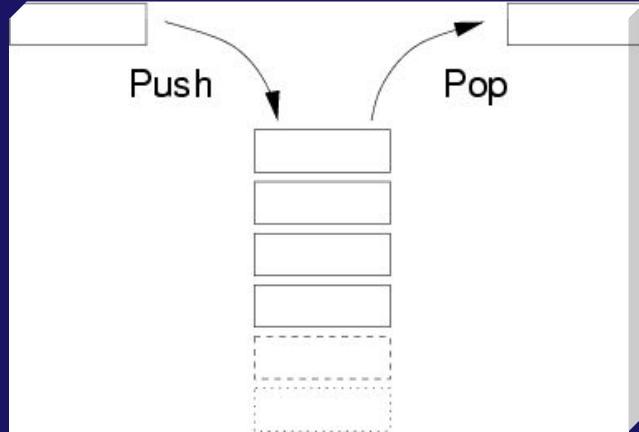
1. HomeScreen - 3d button navigates to 2
2. FragmentViewBalance - current balance



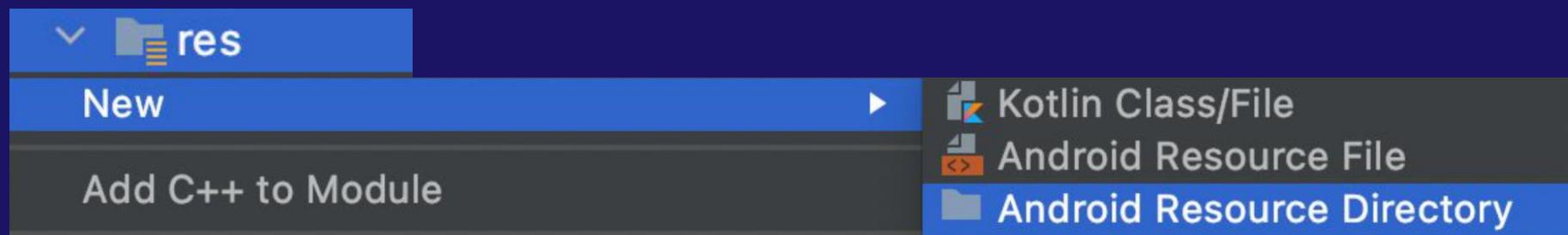
Begränsningar

Trots att det ser ut som att vi bara kan arbeta med 'Drag and Drop' så behöver vi fortfarande pussla ihop navigering själva.

Hursomhelst, så tillkommer navigering och bakåtknapp + STACK!



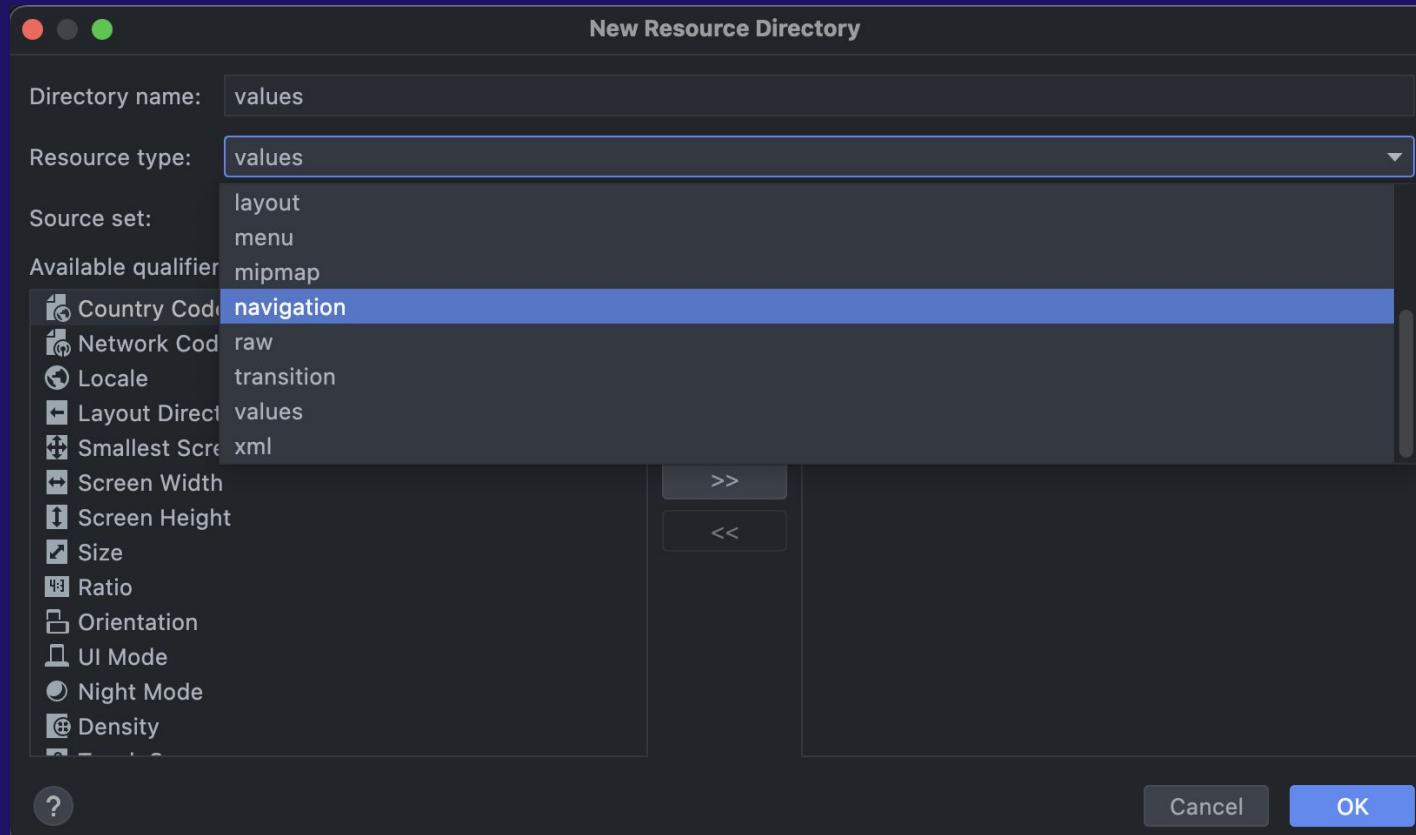
Get Started



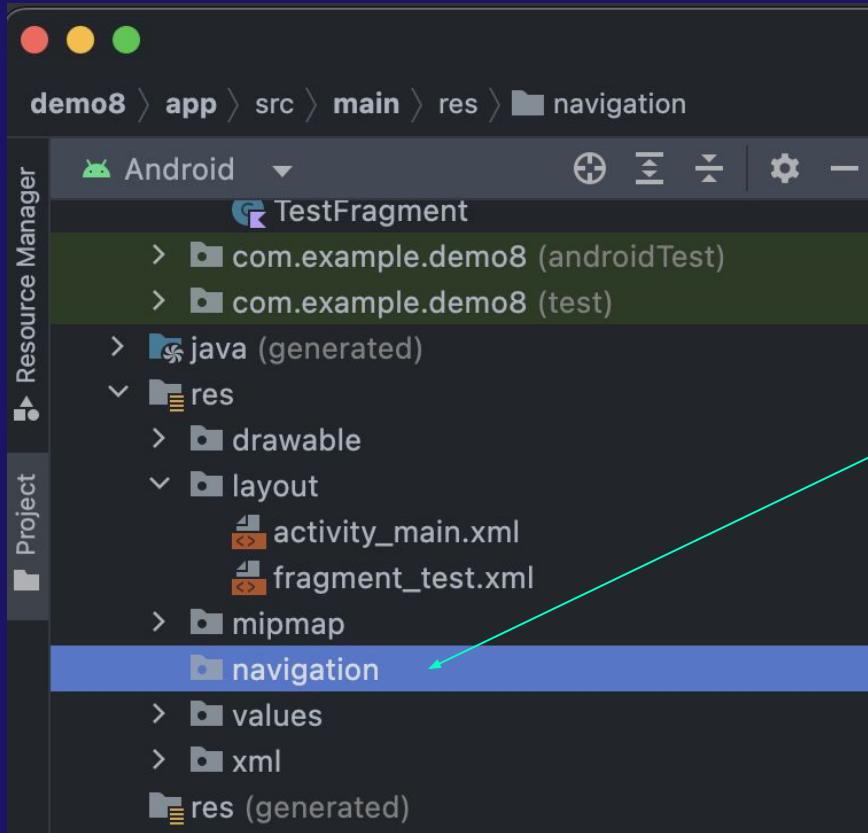
Res folder -> New -> Android Resource Directory

That's right! Vi skapar en helt ny katalog!

Get Started



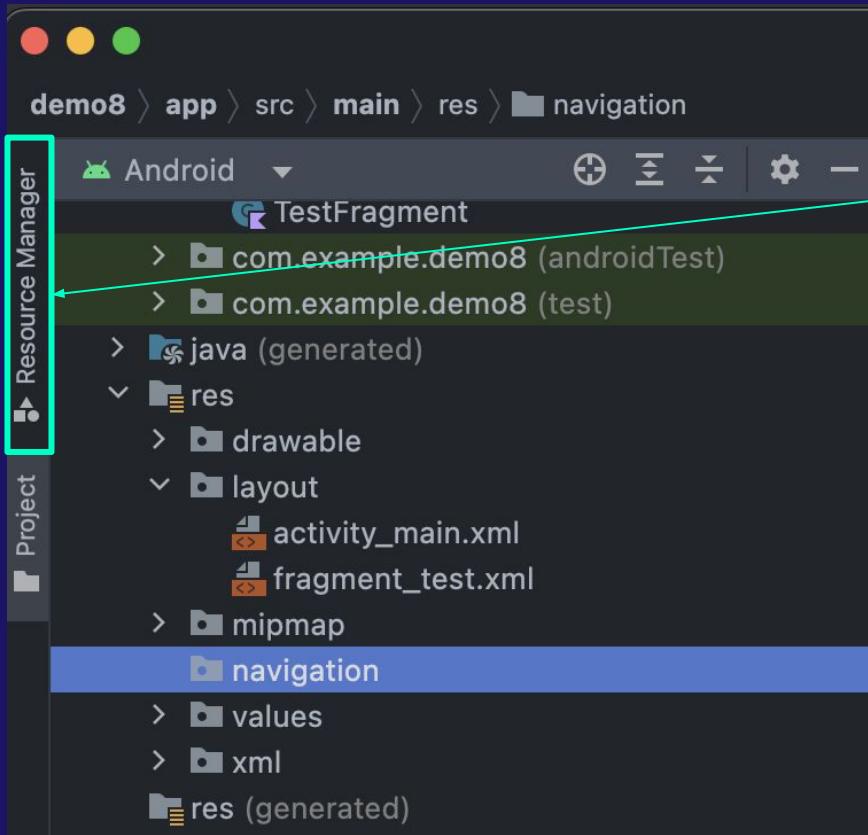
Get Started



Resultat!

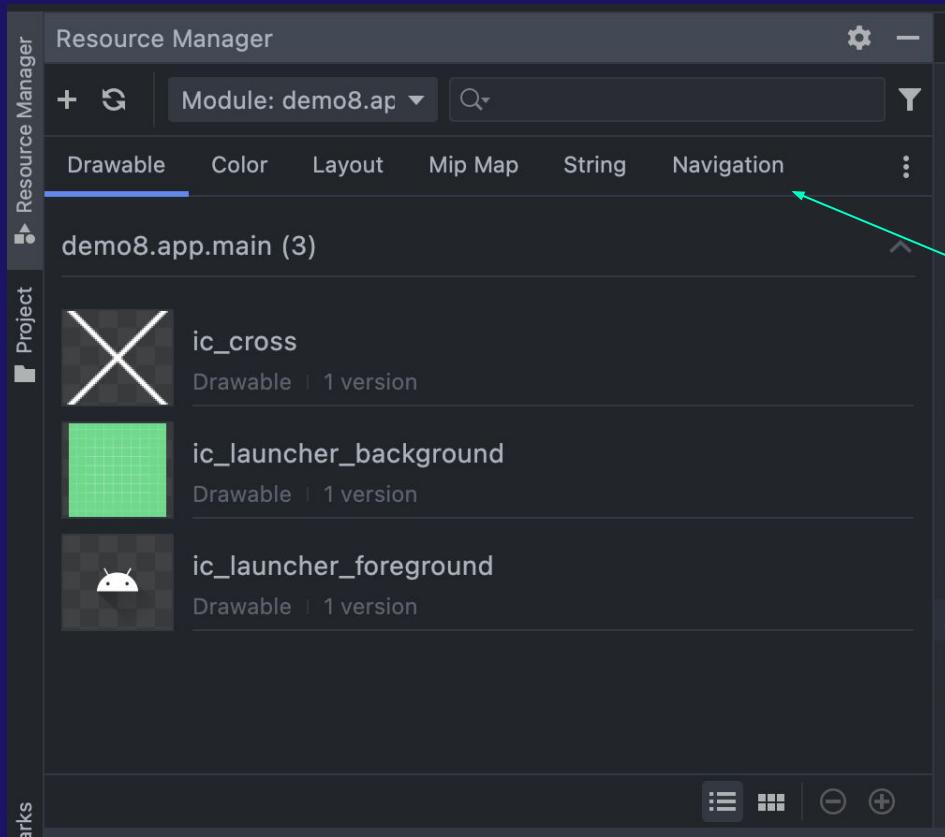
Nu kan vi arbeta inom Navigations mappen!

Get Started



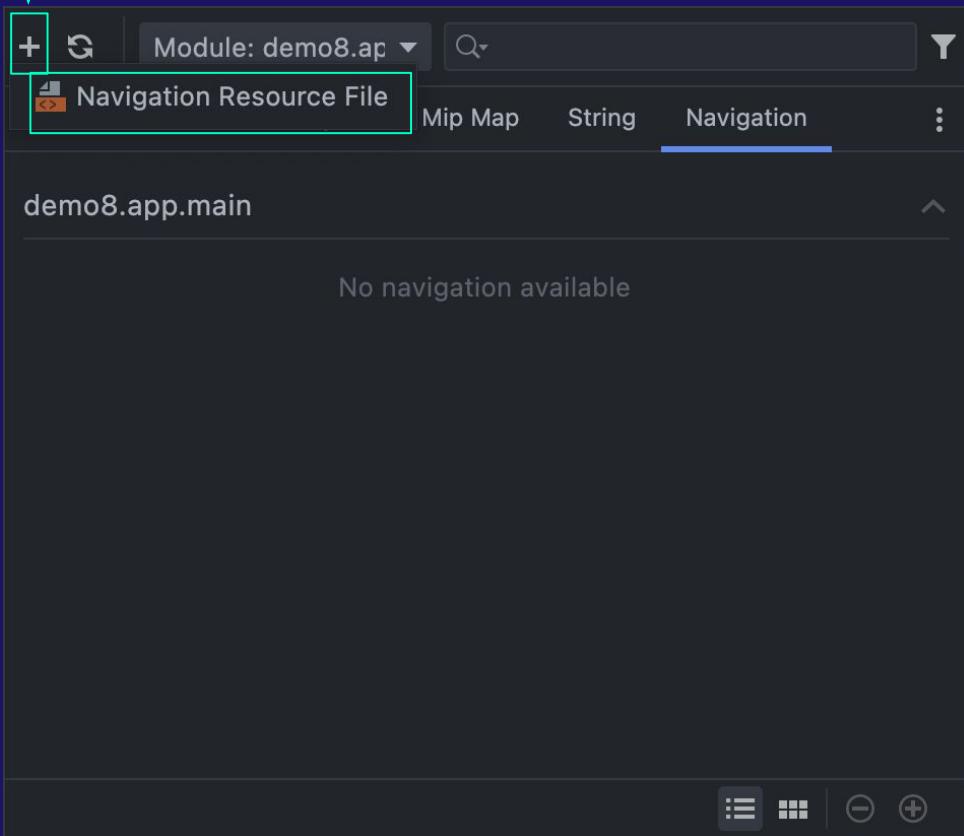
Navigera sedan till **Resource Manager**

Get Started



Navigera in hit!

Get Started



Tryck på +
Navigation Resource File

Get Started



Add Project Dependency

This operation requires the libraries
androidx.navigation:navigation-fragment-ktx:+,
androidx.navigation:navigation-ui-ktx:+.

Problem: Inconsistencies in the existing project
dependencies found.

Version incompatibility between:

- com.google.android.material:material:1.8.0@aar
and:
- androidx.fragment:fragment-ktx:1.5.5@aar

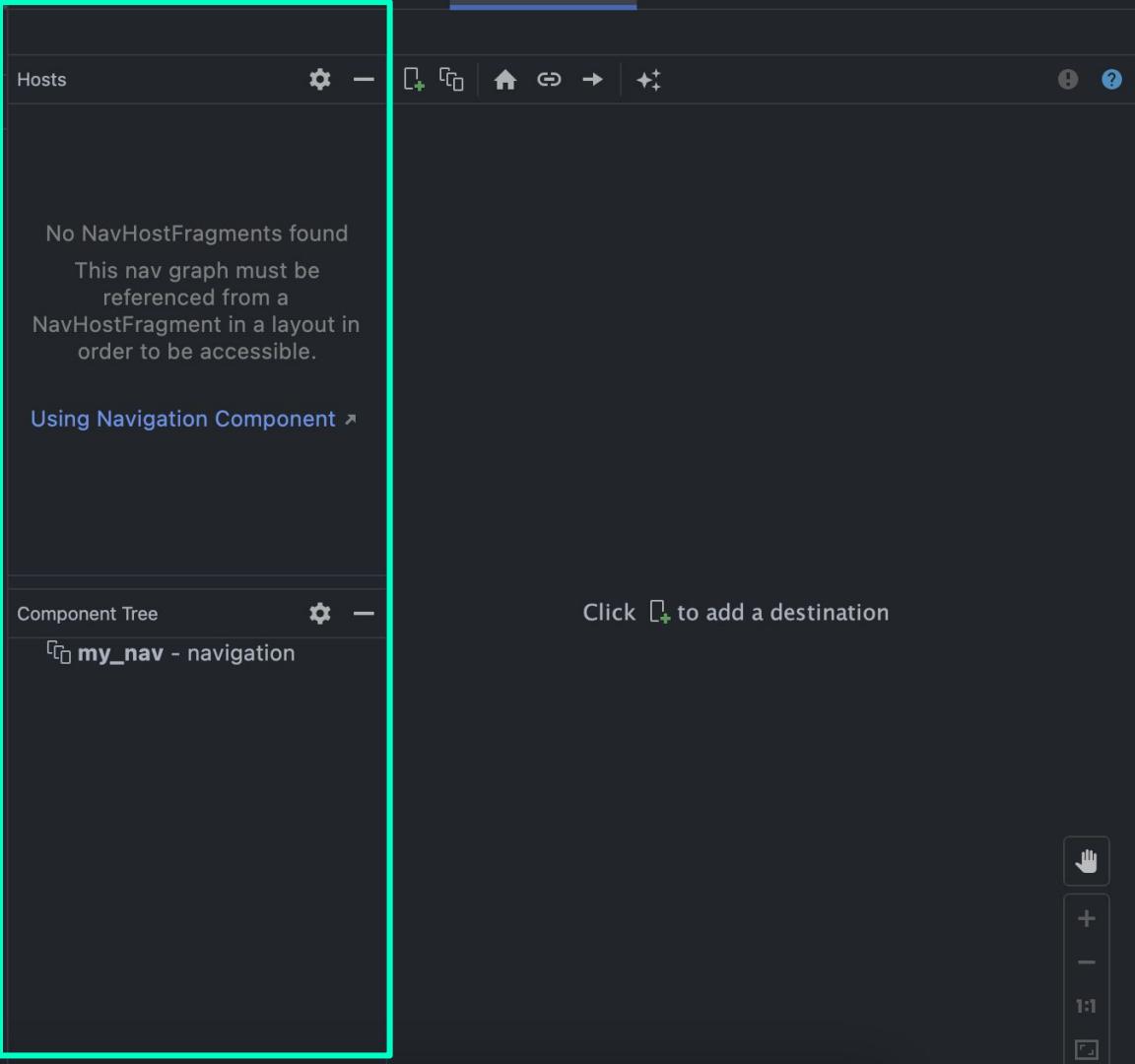
With the dependency:

- androidx.annotation:*:1.1.0

Tryck på OK så kommer den
automatiskt att lägga till
dependencies!

Cancel

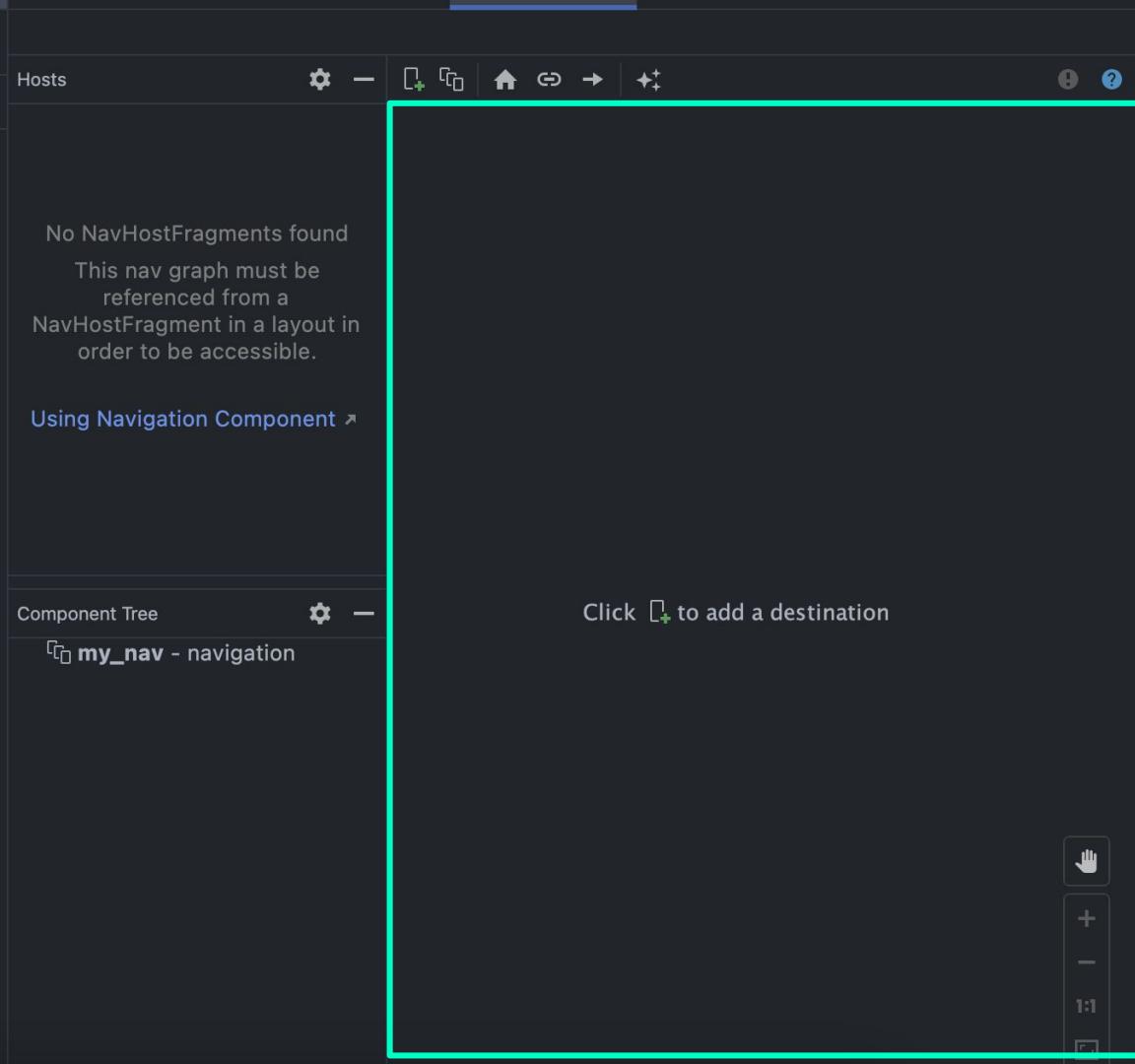
OK



Host location

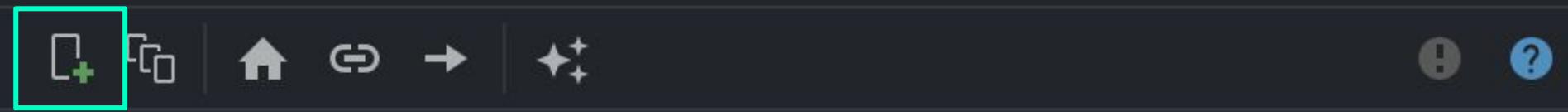
(väldigt likt FragmentContainer)

Komponent hierarki (Tree)



Grafiska navigering (tom just nu)

Här kommer alla våra vyer att visas upp och i vilken ordning man kan navigera inom sin '**STACK**'



Tryck på 'New Destination'

 Search existing destinations

Create new destination



placeholder

Empty destination



fragment_test

Fragment



activity_main

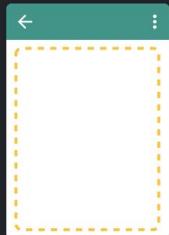
Activity

Här ser vi att vi får valmöjligheten till tidigare redan skapta 'Fragments'!

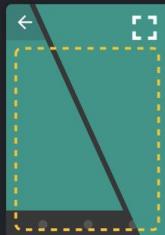
I detta fall kommer vi att välja: **New Destination**



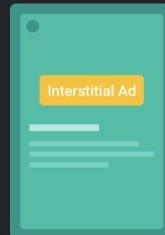
Add a Fragment to Mobile



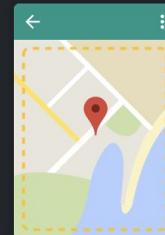
Fragment (Blank)



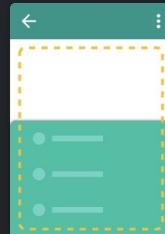
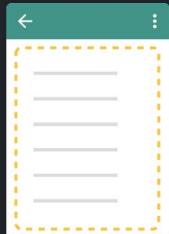
Fullscreen Fragment



Google AdMob Ads Fragment



Google Maps Fragment



Cancel

Previous

Next

Finish



Configure Fragment



Fragment (Blank)

Creates a blank fragment that is compatible back to API level 16

Fragment Name

Fragment Layout Name

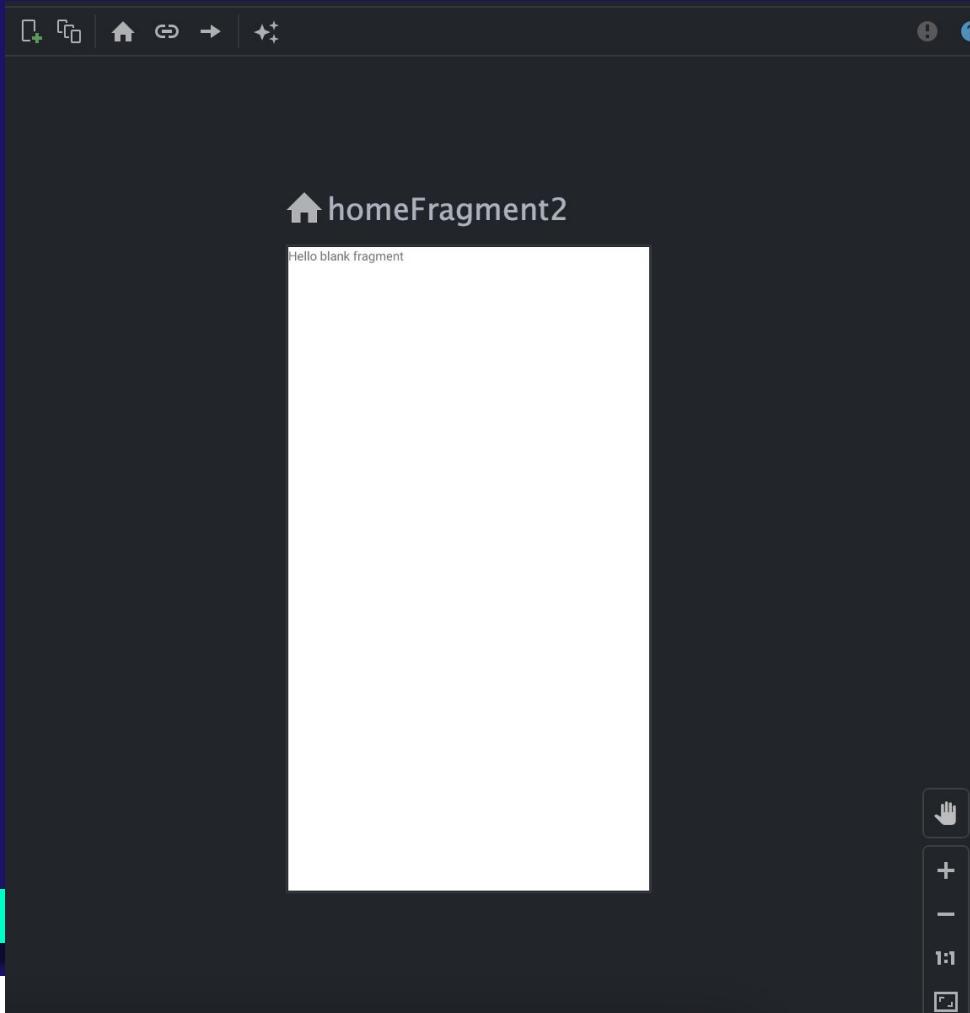
Source Language

Cancel

Previous

Next

Finish

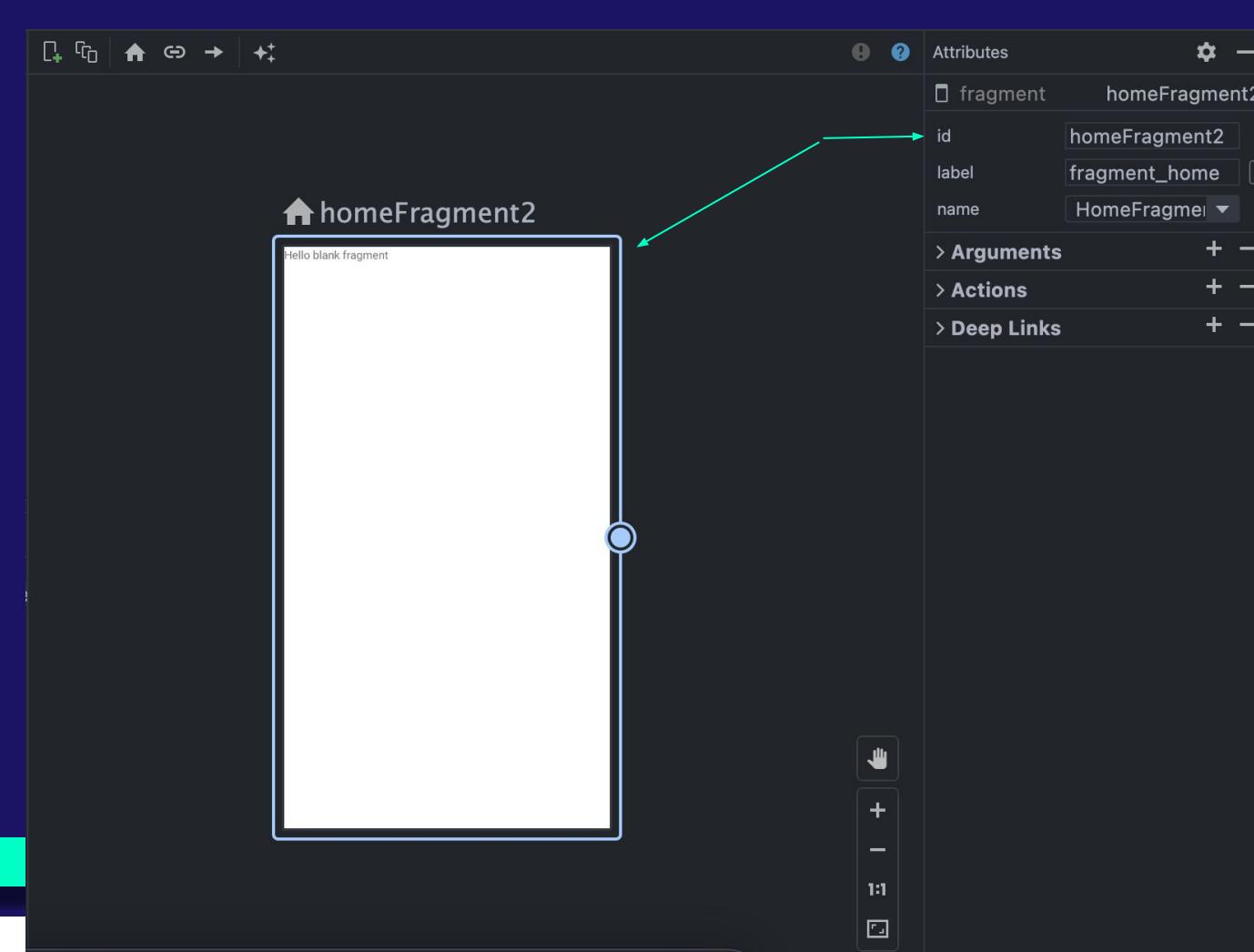


Grattis, du har nu skapat ditt första
fragment med **Navigation Graph!**

Vi har dock tre saker vi behöver se över..

1. **homeFragment2** - namn är fel?
2. Liten text inom fragment
3. Inget att navigera till..





Tryck på **homeFragment2**

Ändra sedan ID't!

Namnkonvention:
Fragment+name

fragmentHome ← Exempel

fragmentMain

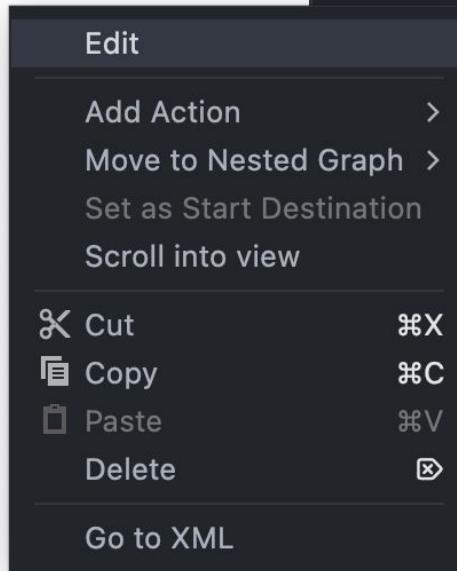
Hello blank fragment

Ser bättre ut!

1. ~~homeFragment2~~ — namn är fel?
2. Liten text inom fragment
3. Inget att navigera till..

fragmentMain

Hello blank fragment



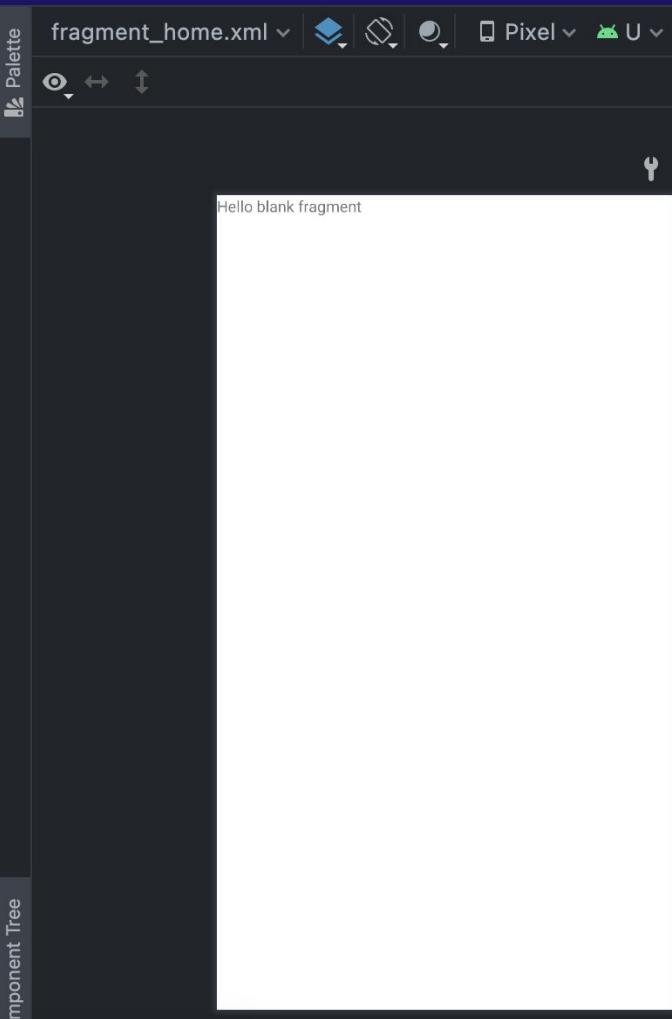
Högerklicka på vy'n → **EDIT**

Detta kommer automatiskt att ta dig till
fragmentets XML fil!

```
1 <?xml version="1.0" encoding="utf-8"?> ✓
2 <FrameLayout xmlns:android="http://schemas.android.com/apk
3     xmlns:tools="http://schemas.android.com/tools"
4         android:layout_width="match_parent"
5         android:layout_height="match_parent"
6         tools:context=".HomeFragment">
7
8     <!-- TODO: Update blank fragment layout -->
9     <TextView
10         android:layout_width="match_parent"
11         android:layout_height="match_parent"
12         android:text="Hello blank fragment" />
13
14 </FrameLayout>
```

*Detta ser ut som när vi hade vårt tidigare
fragment!*

*Låt oss centrera texten och lägga till en
bakgrundsfärg!*



```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    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"
    tools:context=".HomeFragment"
    android:background="@color/purple_200"
    >

    <androidx.appcompat.widget.LinearLayoutCompat
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAlignment="center"
            android:textStyle="bold"
            android:textSize="50sp"
            android:text="1" />

    </androidx.appcompat.widget.LinearLayoutCompat >

</FrameLayout>
```

fragmentMain

1

Ett enkelt fragment, men nu kan vi se att den helt och tydligt urskiljer sig från andra vyer.

Q▼ Search existing destinations

Create new destination



placeholder

Empty destination



fragment_test

Fragment

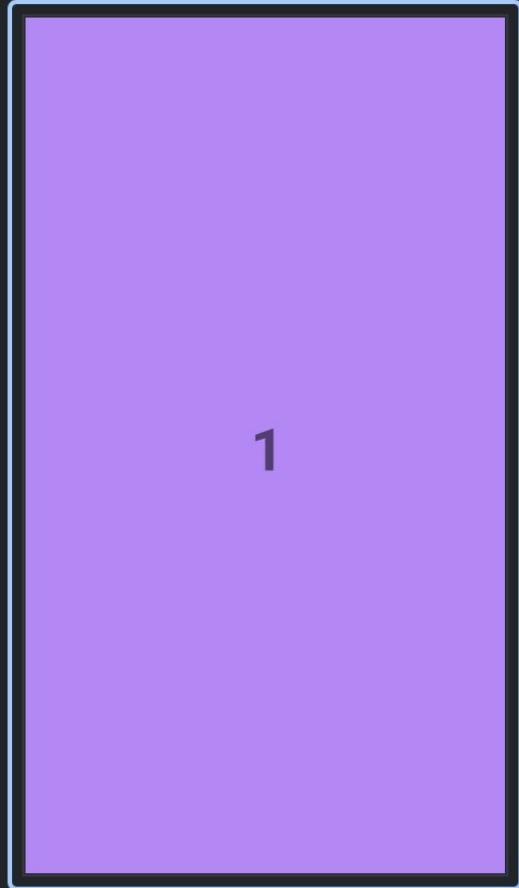


activity_main

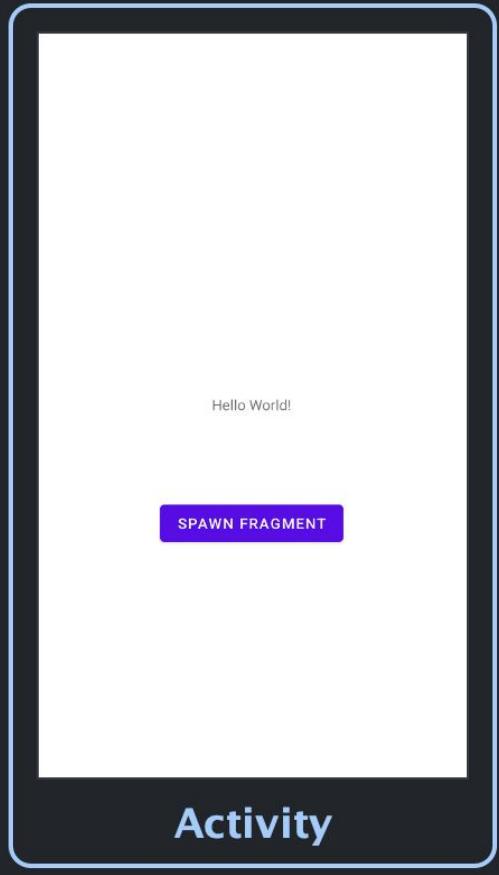
Activity

Låt oss dra in denna aktivitet nu!

fragmentMain



mainActivity2



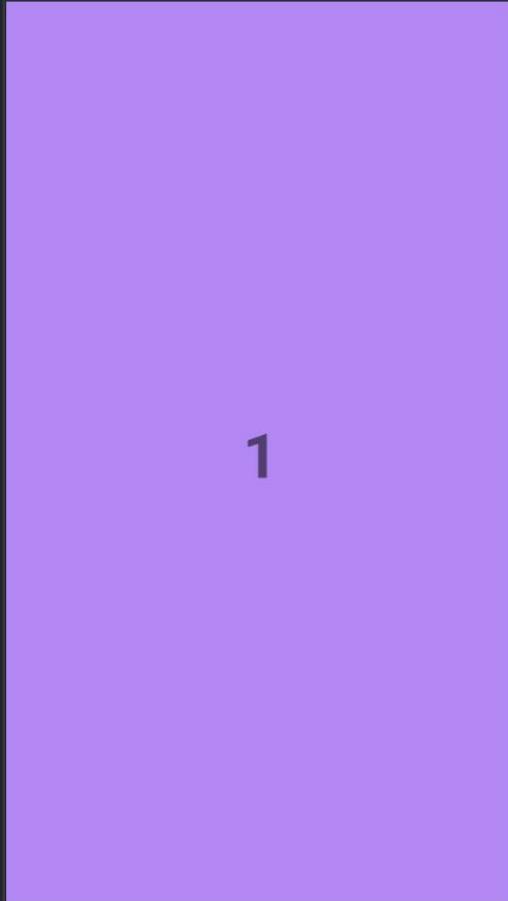
Trycker vi på **COMMAND + A**
För att automatiskt välja alla vyer,
då ser vi att vår aktivitet faktiskt
markeras som en 'ACTIVITY'

Vi har också problemet igen med id
namnet på vår aktivitet.

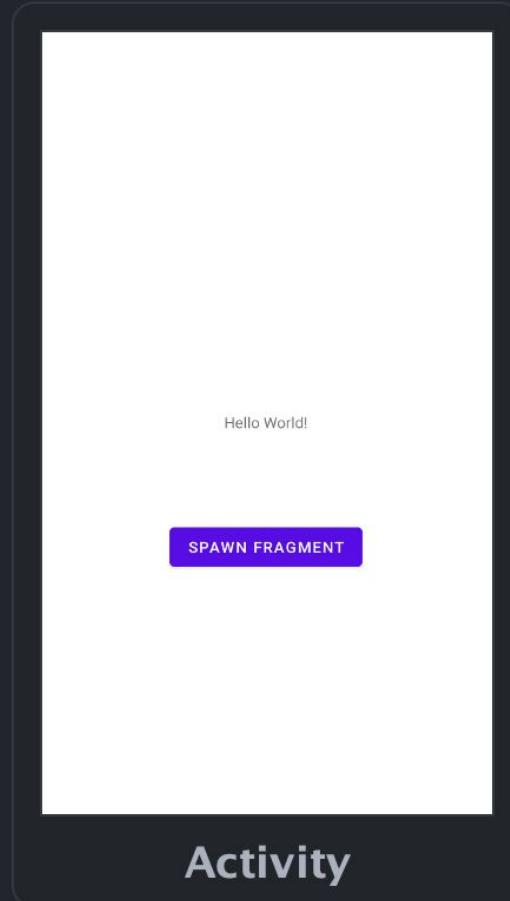
Låt oss fixa detta!

Döp om den till:
activityMain

fragmentMain



activityMain



Looking good!

Next up: connect!

fragmentMain

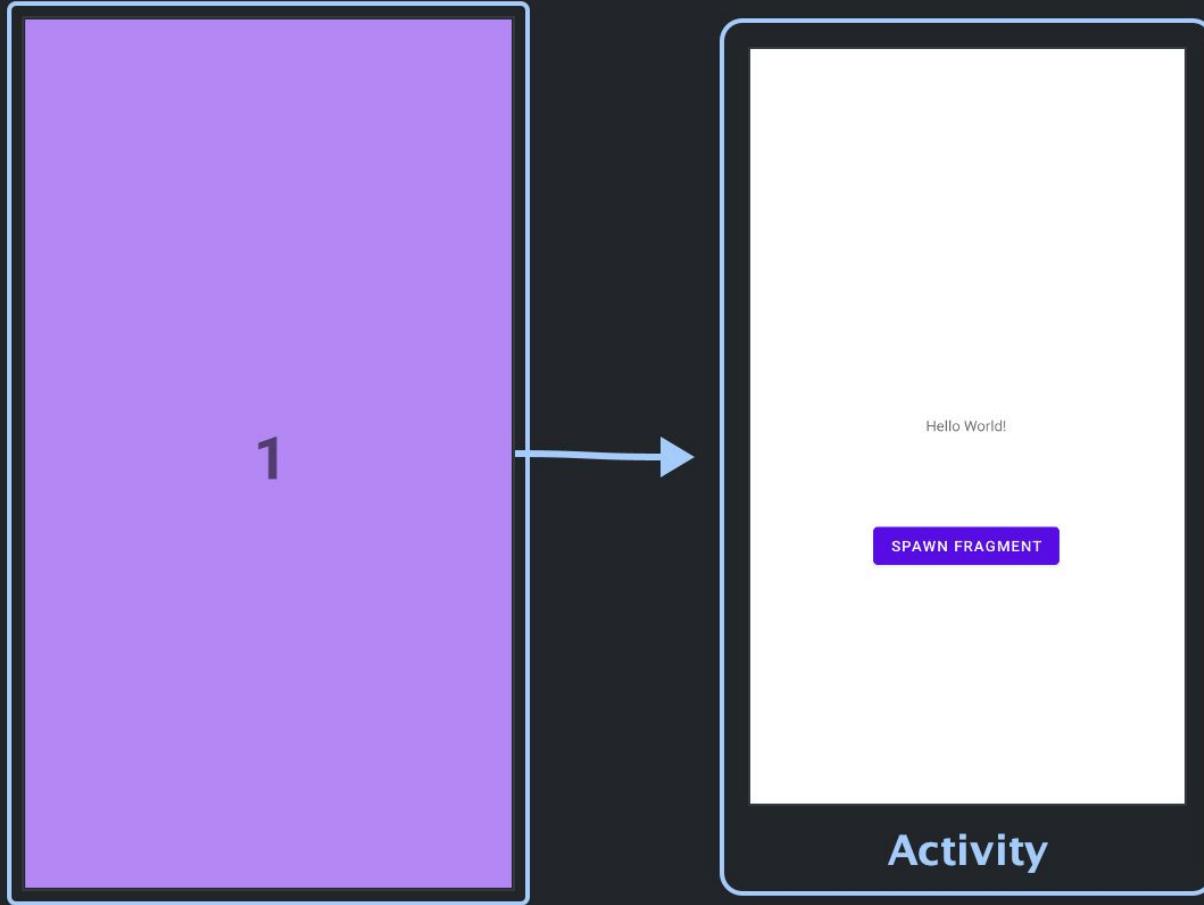
1



Drag and drop via cirkeln!

fragmentMain

activityMain

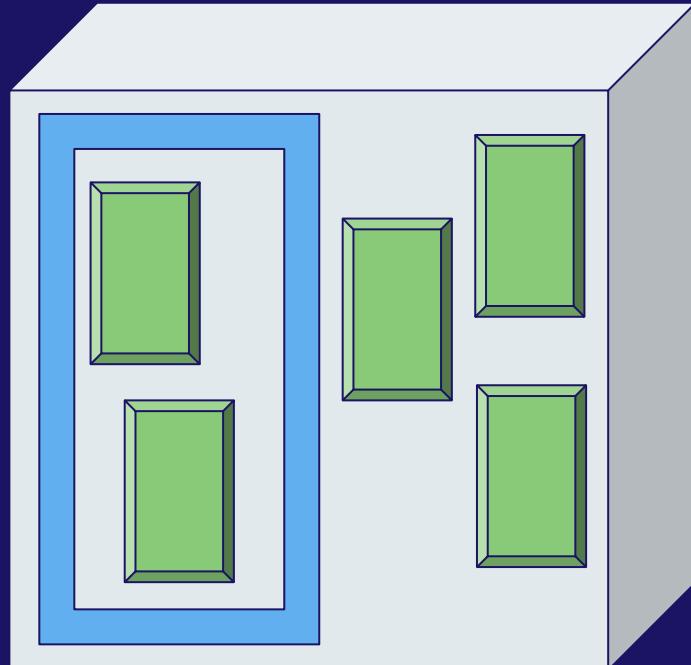


Voila!

Nu har vi en koppling!

Hursomhelst, så ser vi att ActivityMain INTE har förmågan att skapa kopplingar!

Varför....?

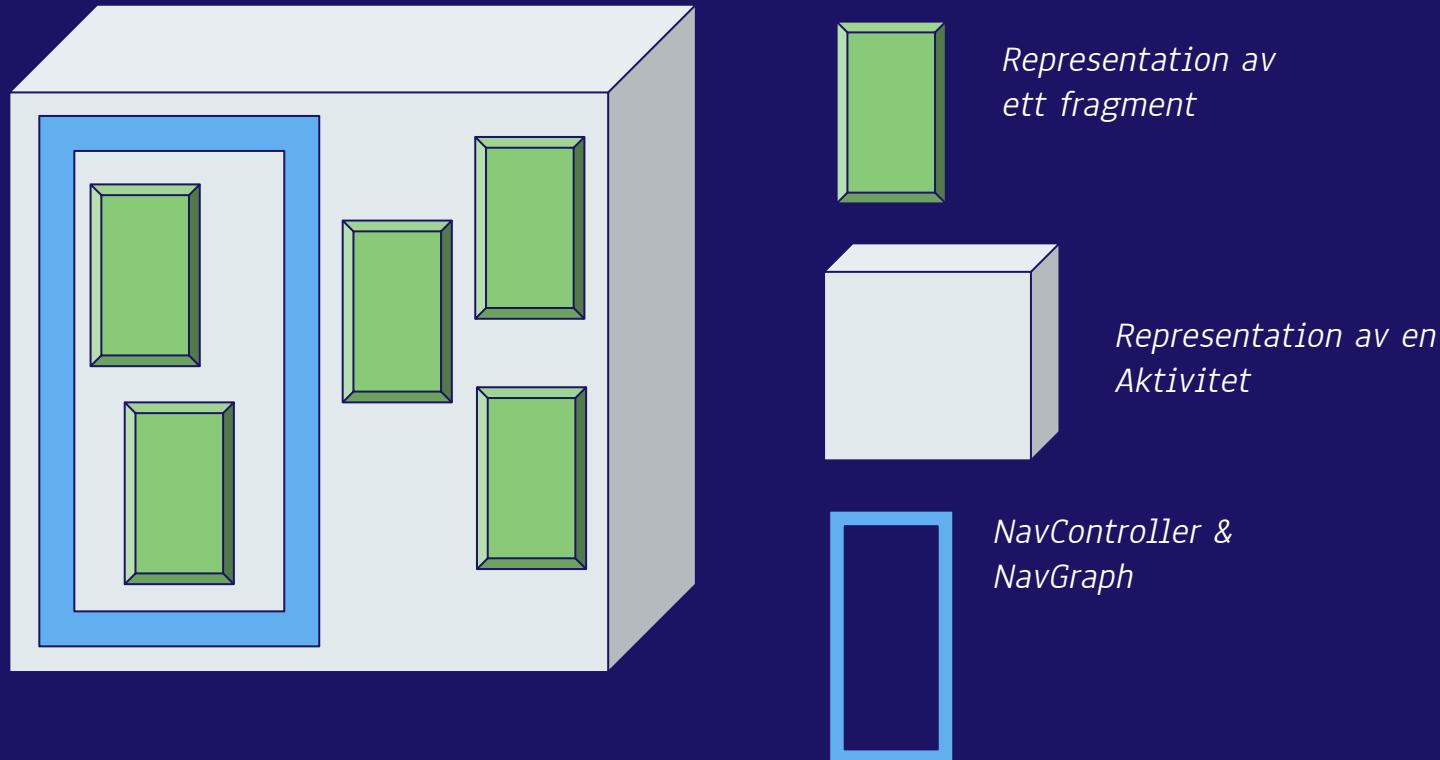


*Representation av
ett fragment*

*Representation av en
Aktivitet*

*NavController &
NavGraph*

När du använder dig av `navigate()` så är denna navController + Graph ej aktiv!



fragmentMain

1

← Återgå till hur det såg ut från början

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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"
    tools:context=".AboutFragment" >

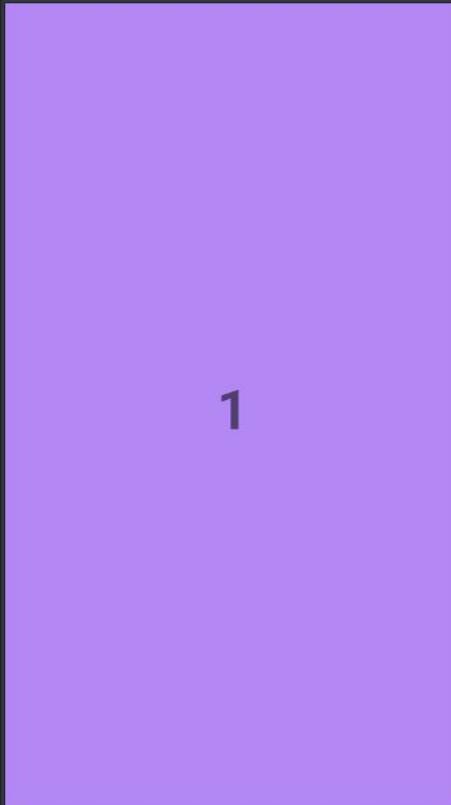
    <androidx.appcompat.widget.LinearLayoutCompat
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAlignment="center"
            android:textStyle="bold"
            android:textSize="50sp"
            android:text="2" />

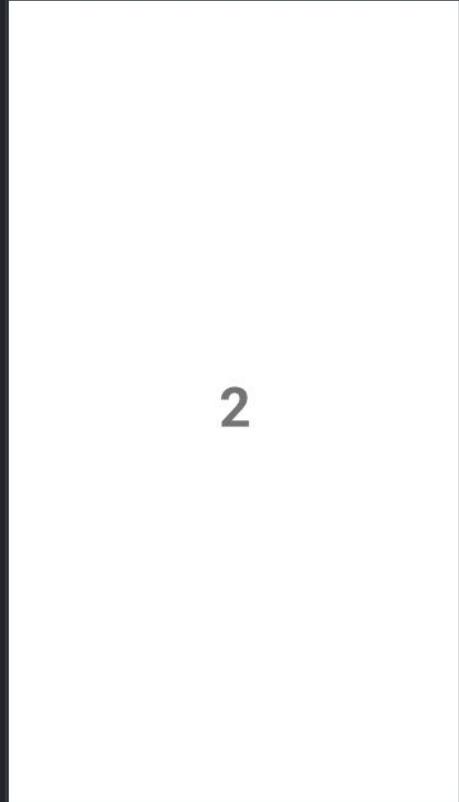
    </androidx.appcompat.widget.LinearLayoutCompat >

</FrameLayout>
```

fragmentMain



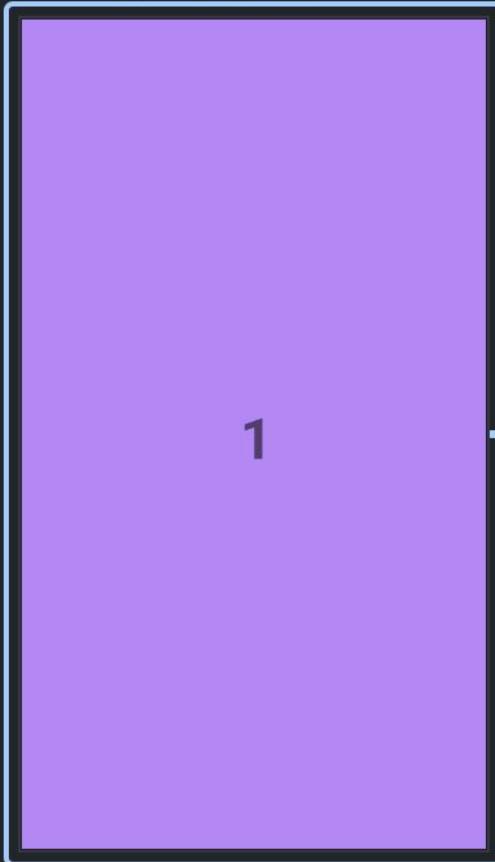
fragmentAbout



Drag 'n drop!



fragmentMain

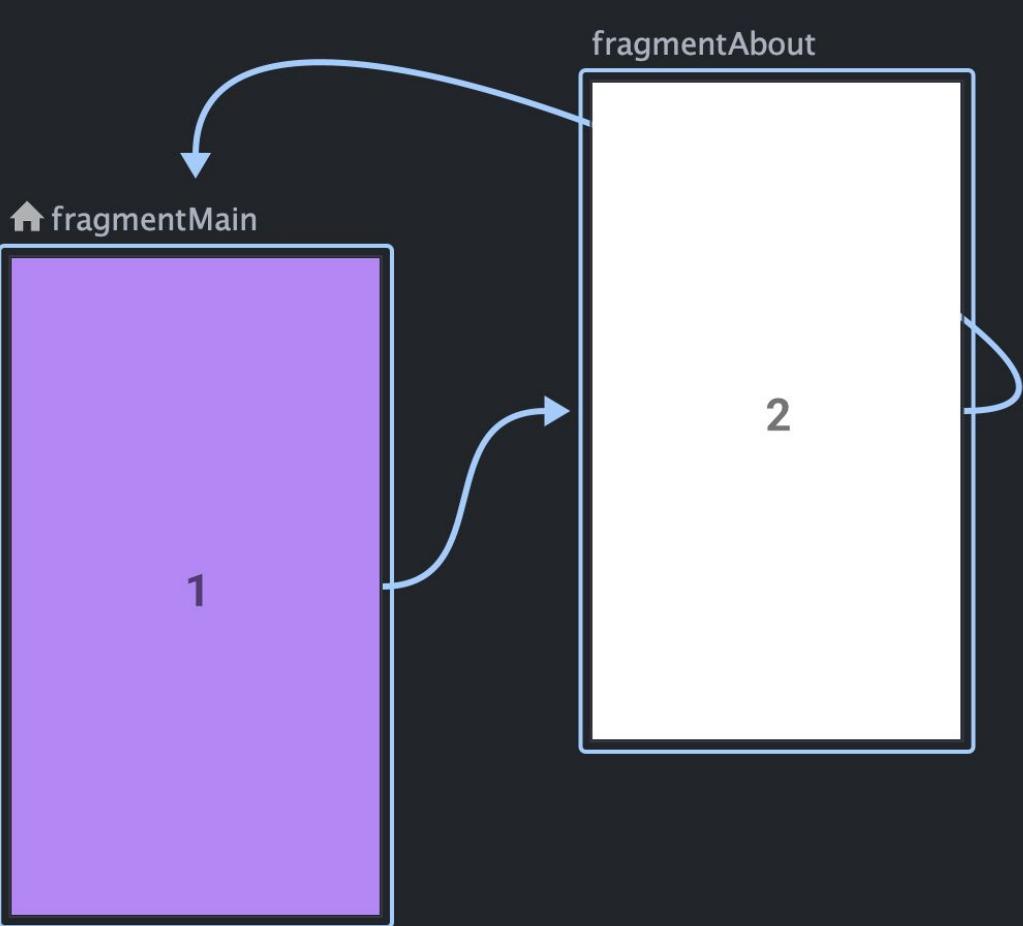


fragmentAbout



1

2



Vi kan sedan koppla samman
dessa då dem båda är fragment
som 'hostas' av en aktivitet!



Problem

Hur skapar vi en 'navGraph' med 'navigation component'?

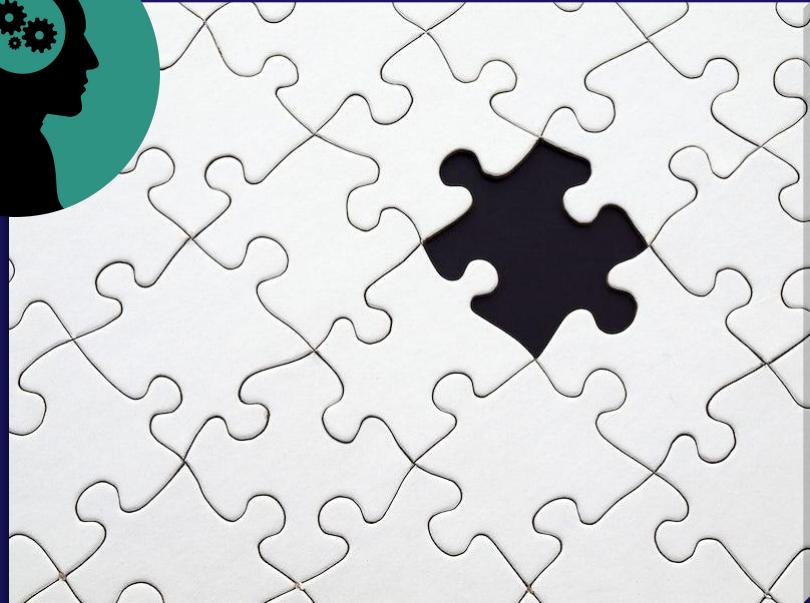


Solution

Skapa 'Resource Folder' för 'Navigation'. Inuti NavGraph så kan du senare skapa fragment och koppla ihop dem!



Frågor?



How to: Navigate



res

drawable

layout

activity_main.xml

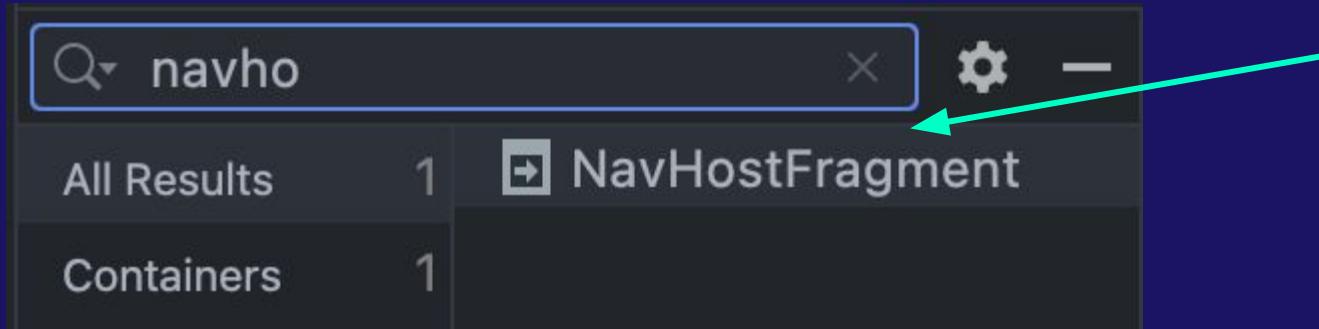
fragment_about.xml

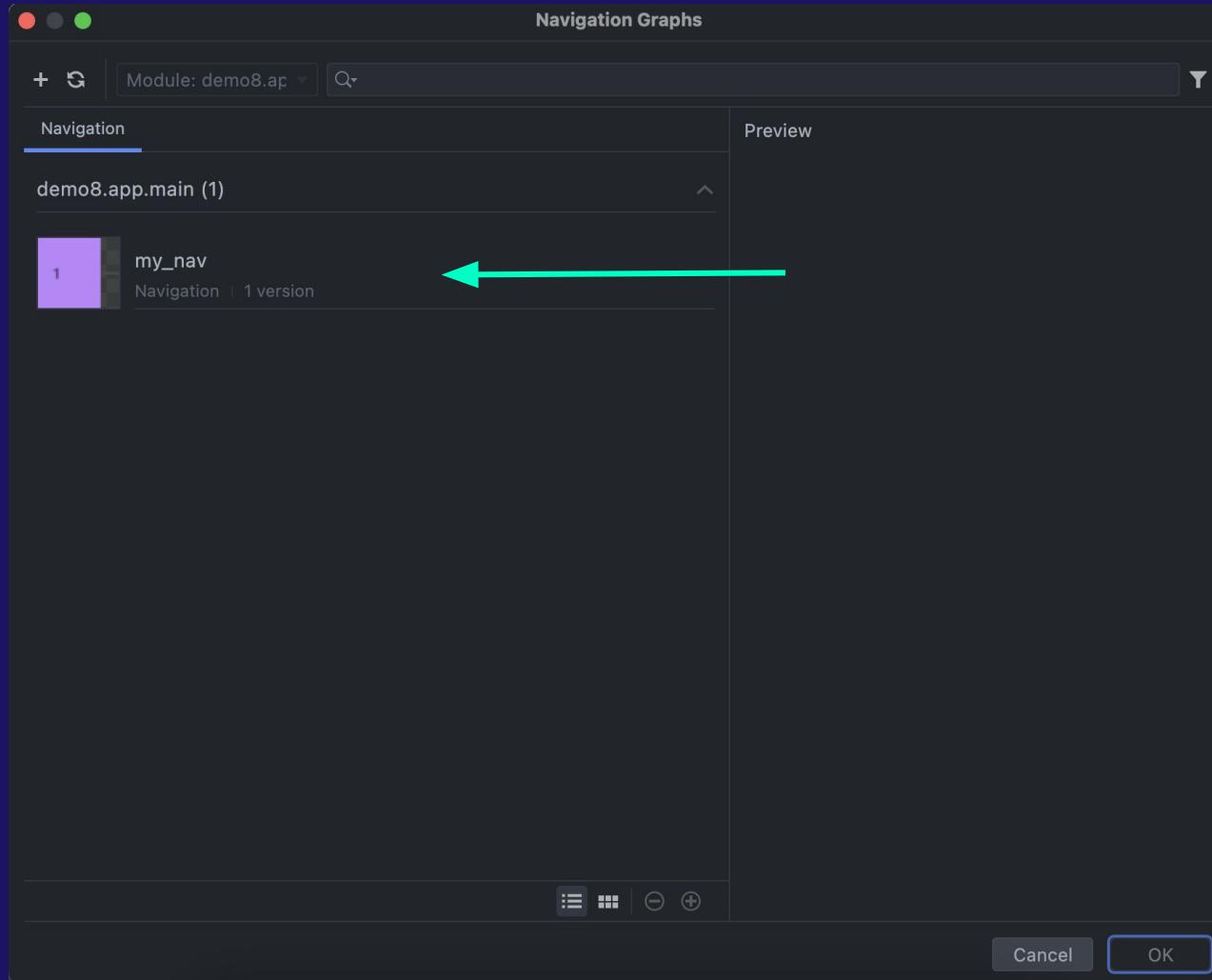
fragment_home.xml

fragment_test.xml



res
> drawable
< layout
activity_main.xml
fragment_about.xml
fragment_home.xml
fragment_test.xml



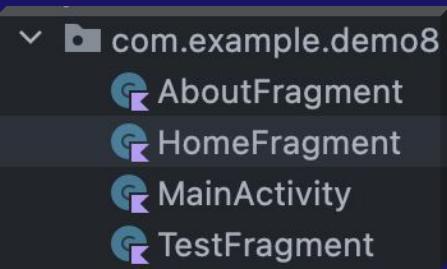


1

Hello World!

SPAWN FRAGMENT

Nu ser vi att den befinner sig inom vår applikation!



Om vi gör samma sak för vårt andra fragment så kan vi navigera fram och tillbaka genom att klicka!

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

    view.findViewById<FrameLayout>(R.id.fragment_home).setOnClickListener () {
        Navigation.findNavController (view).navigate(R.id.action_main_to_fragmentAbout )
    }

    return view
}
```



Problem

Hur skapar vi en 'navGraph' med 'navigation component'?

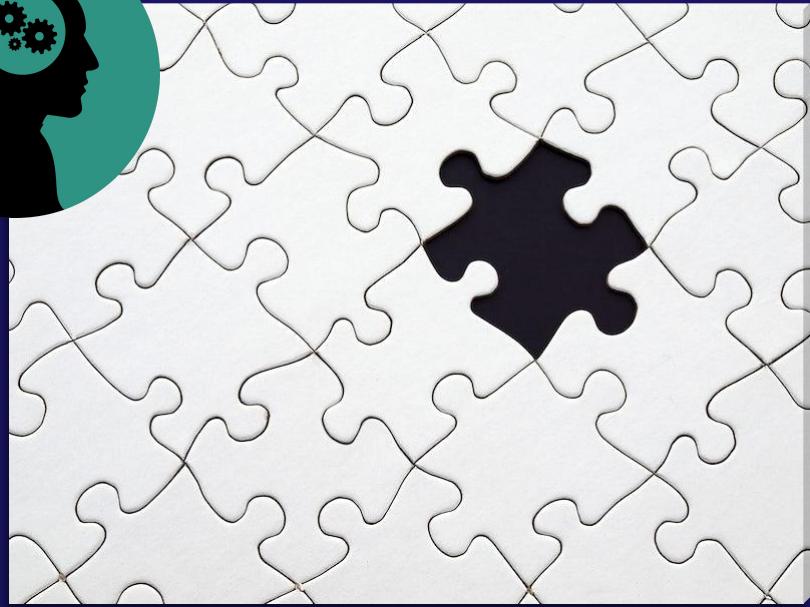


Solution

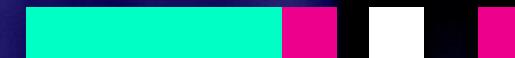
Skapa 'Resource Folder' för 'Navigation'. Inuti NavGraph så kan du senare skapa fragment och koppla ihop dem!



Frågor?

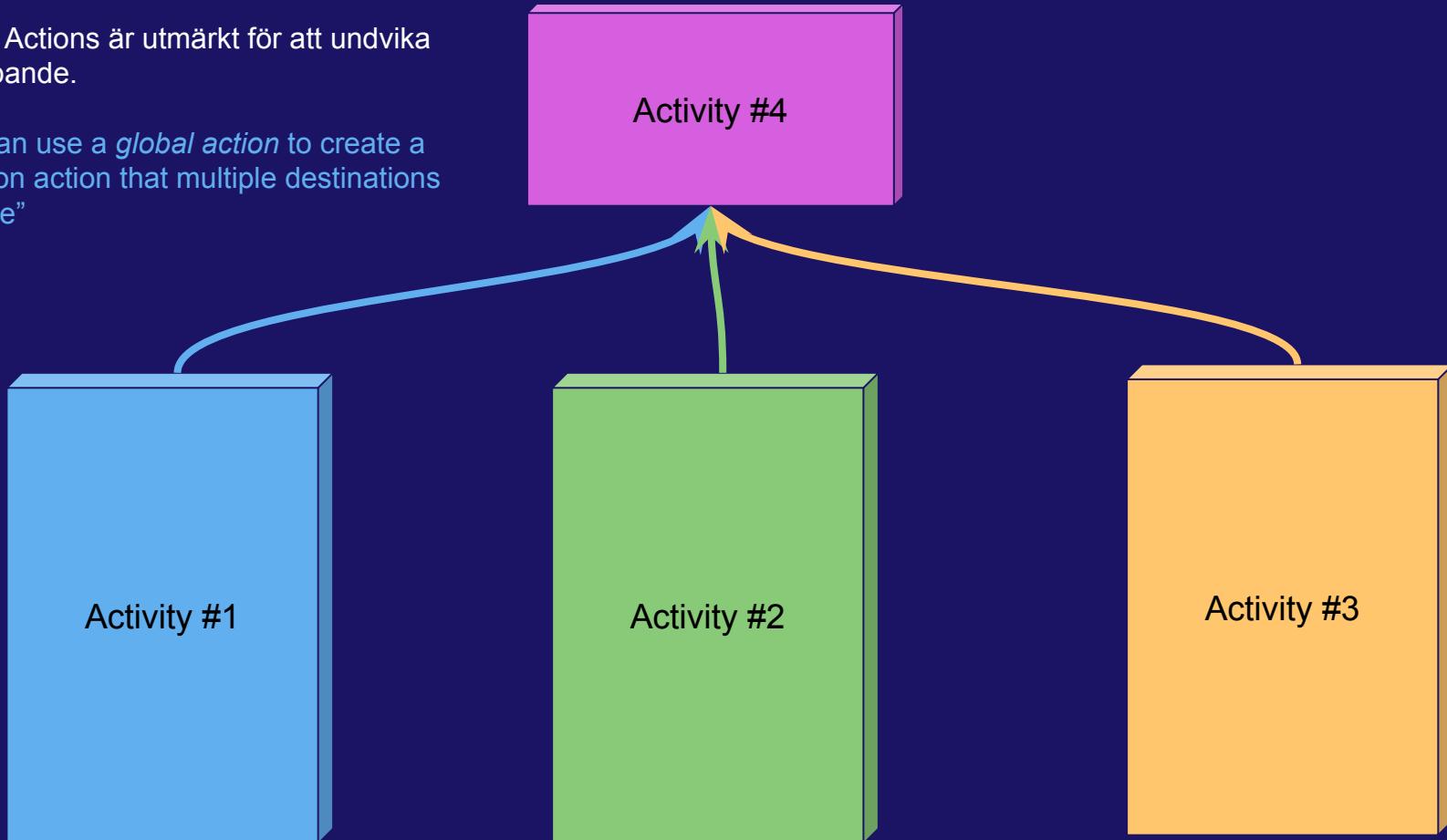


Global Actions



Global Actions är utmärkt för att undvika upprepande.

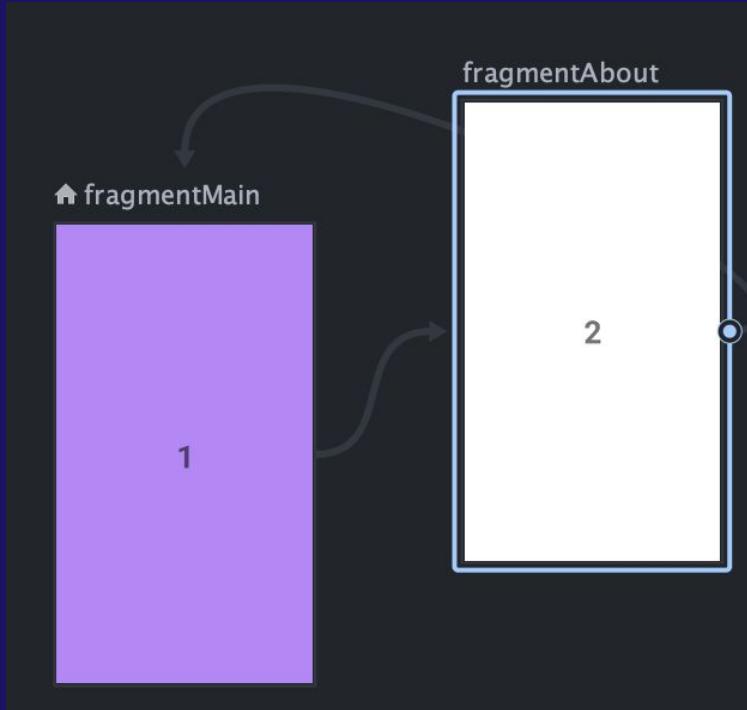
“You can use a *global action* to create a common action that multiple destinations can use”



Create a global action

To create a global action, do the following:

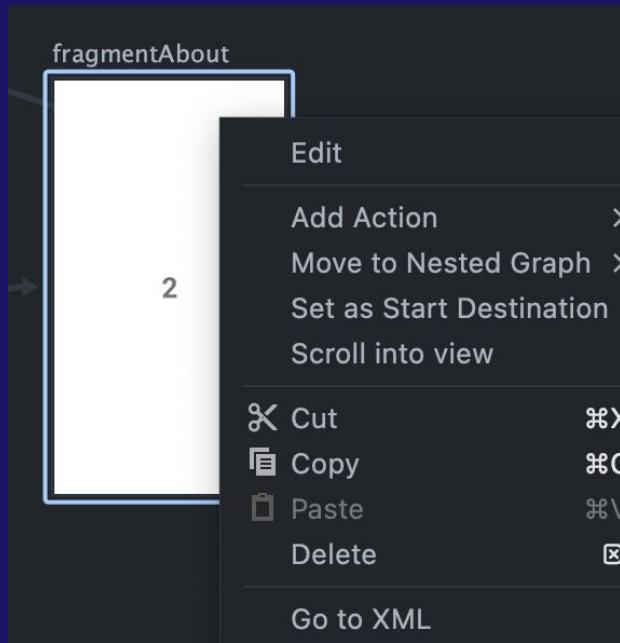
1. From the **Graph Editor**, click on a destination to highlight it.



Create a global action

To create a global action, do the following:

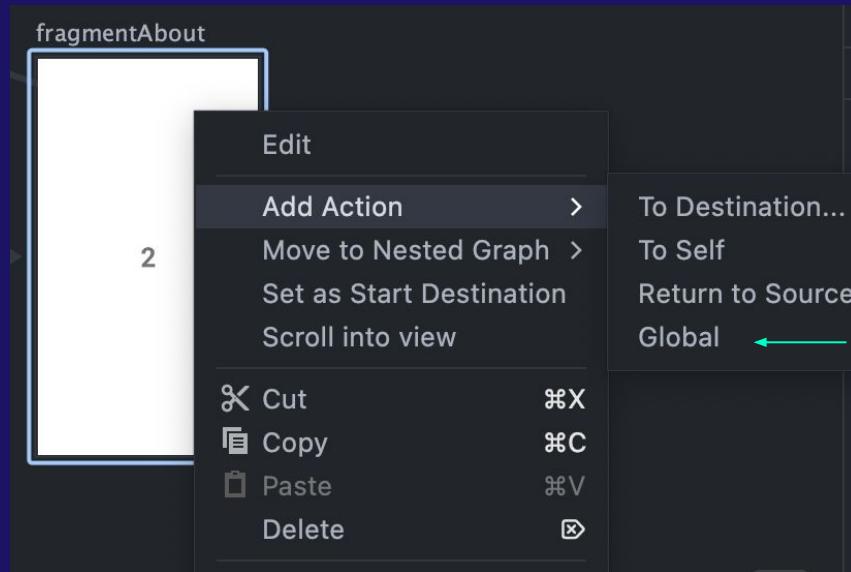
1. From the **Graph Editor**, click on a destination to highlight it.
2. Right-click on the destination to display the context menu.



Create a global action

To create a global action, do the following:

1. From the **Graph Editor**, click on a destination to highlight it.
2. Right-click on the destination to display the context menu.
3. Select **Add Action > Global**. An arrow (→) appears to the left of the destination.



Create a global action

To create a global action, do the following:

1. From the **Graph Editor**, click on a destination to highlight it.
2. Right-click on the destination to display the context menu.
3. Select **Add Action > Global**. An arrow (→) appears to the left of the destination.
4. Click the **Text** tab to navigate to the XML text view. The XML for the global action looks similar to the following:

```
<fragment
    android:id="@+id/fragmentAbout"
    android:name="com.example.demo8.AboutFragment"
    android:label="fragment_about"
    tools:layout="@layout/fragment_about" >
    <action
        android:id="@+id/action_fragmentAbout_to_fragmentMain"
        app:destination="@+id/fragmentMain" />
</fragment><action android:id="@+id/action_global_fragmentAbout"
    app:destination="@+id/fragmentAbout" />
```

Use a global action

To use a global action in your code, pass the resource ID of the global action to the `navigate()` method for each UI element, as shown in the following example:

```
viewTransactionButton.setOnClickListener { view ->
    view.findNavController().navigate(R.id.action_global_mainFragment)
}
```



Problem

Tänk om vi har flertalet knappar som leder till samma destination?

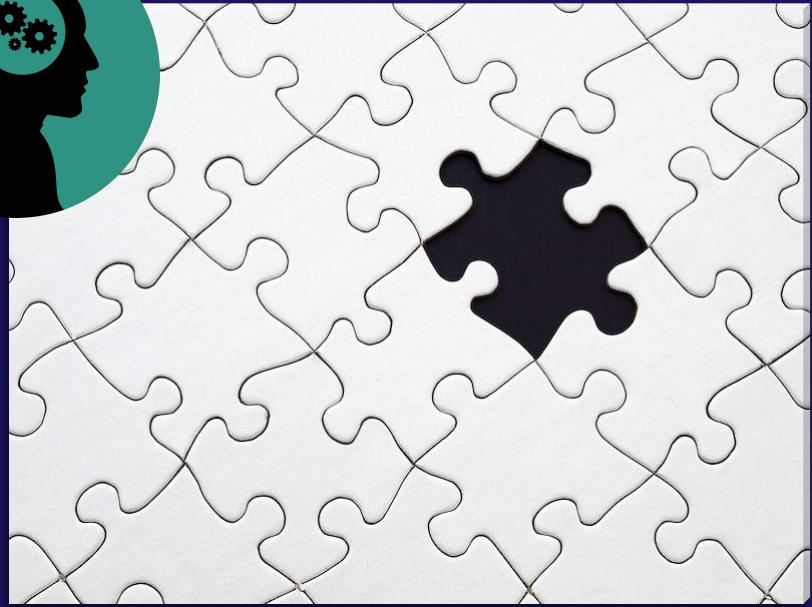
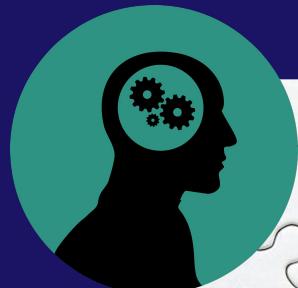


Solution

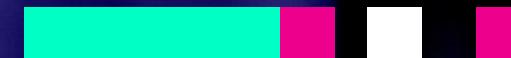
'Global Actions', tillåter oss att arbeta med globala lösningar inom Navigerings komponenten.



Frågor?



Tips



Navigation and the back stack

Android maintains a [back stack](#) that contains the destinations you've visited. The first destination of your app is placed on the stack when the user opens the app. Each call to the [navigate\(\)](#) method puts another destination on top of the stack. Tapping **Up** or **Back** calls the [NavController.navigateUp\(\)](#) and [NavController.popBackStack\(\)](#) methods, respectively, to remove (or *pop*) the top destination off of the stack.

[NavController.popBackStack\(\)](#) returns a boolean indicating whether it successfully popped back to another destination. The most common case when this returns false is when you manually pop the start destination of your graph.

Animation

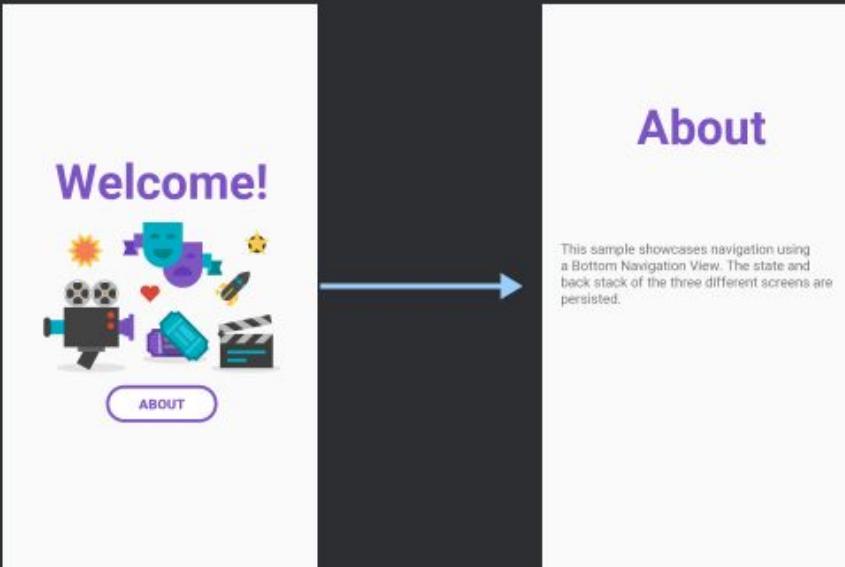
Destinations

HOST titleScreen aboutScreen

No NavHostFragments found

GRAPH

titleScreen - Start
 aboutScreen



The diagram shows a navigation graph with two nodes: 'titleScreen' and 'aboutScreen'. A directed edge connects 'titleScreen' to 'aboutScreen', indicating a transition from the title screen to the about screen.

Attributes

→ action_title_to_about action

id action

destination

Animations

enterAnim	<input type="text" value="@anim/fade_in"/>
exitAnim	<input type="text" value="@anim/fade_out"/>
popEnterAnim	<input type="text" value="@anim/fade_in"/>
popExitAnim	<input type="text" value="@anim/fade_out"/>

Argument Default Values

Pop Behavior

popUpTo	<input type="text" value="none"/>
popUpToInclusive	<input type="checkbox"/>

Launch Options

launchSingleTop	<input type="checkbox"/>
-----------------	--------------------------

03

Constraints #2

?

QUIZ

TIME

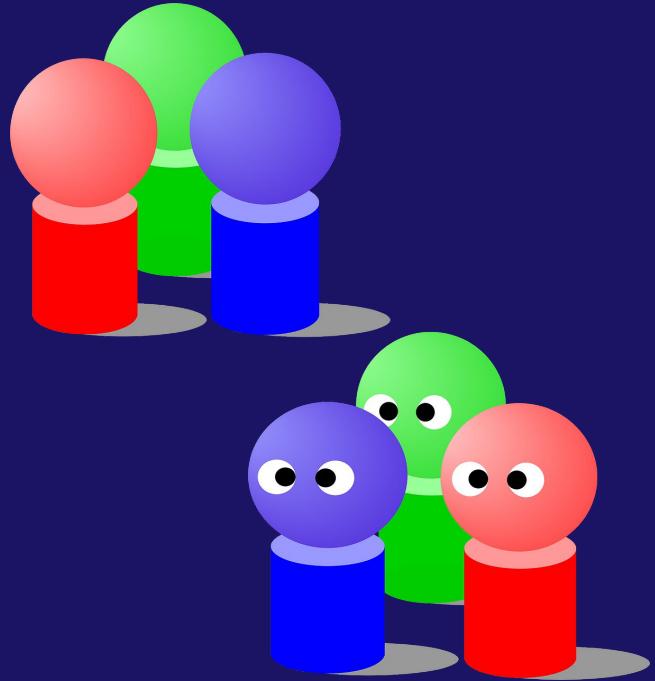
Grupp 1

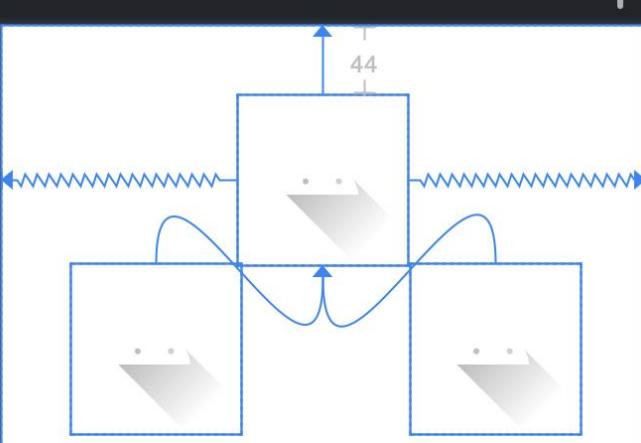
Grupp 2

Grupp 3

Ni får ett poäng för varje rätt svar
Det finns totalt FYRA frågor

READY?





Fråga - Vad tror ni kommer hända om vi kör denna app?

- 1 - Appen kraschar
- 2 - Komponenterna kommer se ut precis som de gör på bilden
- 3 - Komponent längst till vänster & höger kommer falla upp till position 0 på höjden
- 4 - Komponent längst till vänster & höger kommer falla till vänster men under den övre komponenten



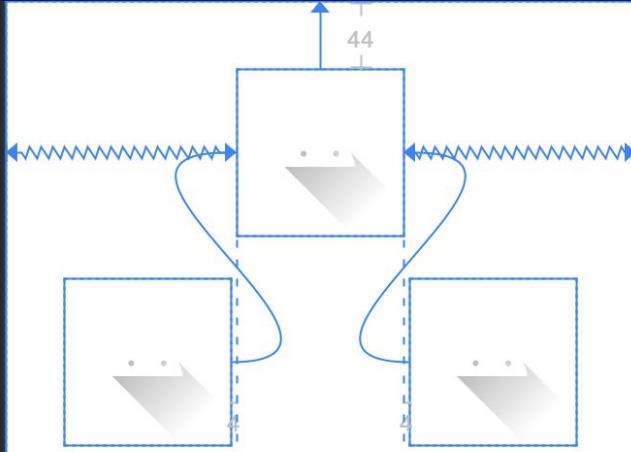
demo9



RESULT

Fråga - Vad tror ni kommer hända om vi kör denna app?

- 1 - Appen kraschar
- 2 - Komponenterna kommer se ut precis som de gör på bilden
- 3 - Komponent längst till vänster & höger komma falla upp till position 0 på höjden
- 4 - Komponent längst till vänster & höger komma falla till vänster men under den övre komponenten



Fråga - Vad tror ni kommer hända om vi kör denna app?

- 1 - Komponenten till vänster och höger, förflyttar sig till mitten (under övre komponent)
- 2 - Komponenten till vänster och höger, contreras horisontellt.
- 3 - Komponenten till vänster och höger, förflyttas högst upp och centreras.
- 4 - Komponenten till vänster och höger, förflyttar sig upp till samma höjd som den övre komponent.

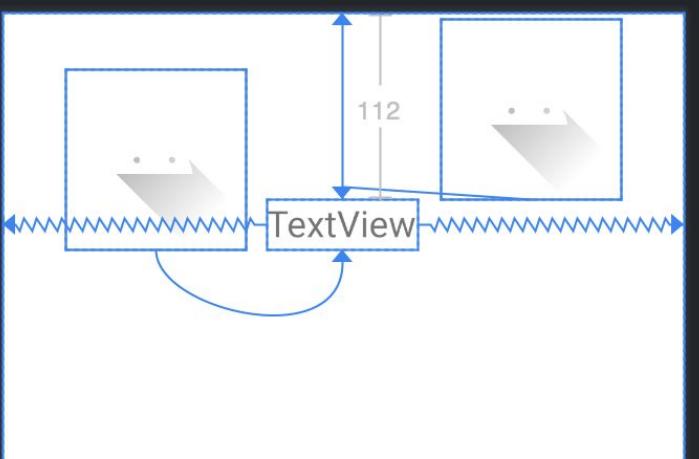


demo9

RESULT

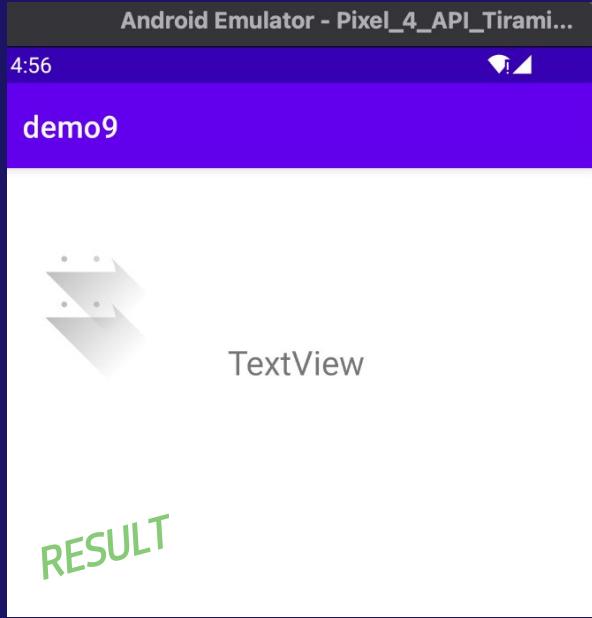
Fråga - Vad tror ni kommer hända om vi kör denna app?

- 1 - Komponenten till vänster och höger, förflyttar sig till mitten (under övre komponent)
- 2 - Komponenten till vänster och höger, contreras horisontellt.
- 3 - Komponenten till vänster och höger, förflyttas högst upp och centreras.
- 4 - Komponenten till vänster och höger, förflyttar sig upp till samma höjd som den övre komponent.



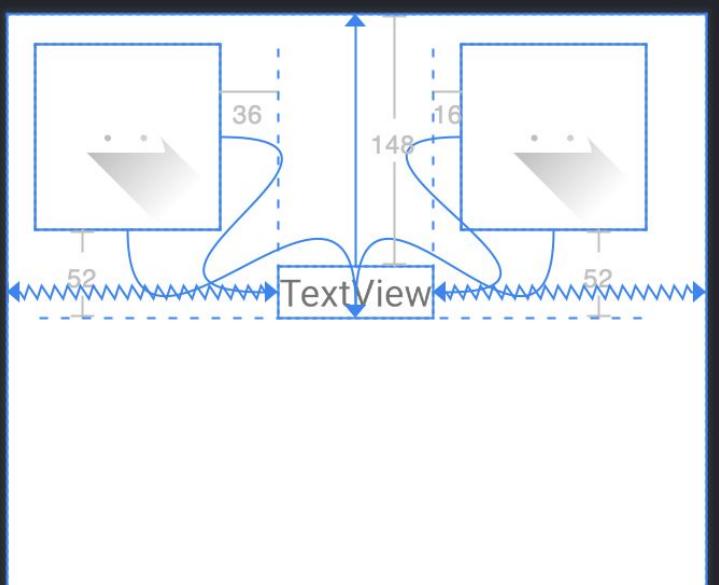
Fråga - Vad tror ni kommer hända om vi kör denna app?

- 1 - Kommer se ut precis som på bilden
- 2 - Komponenten till vänster förflyttas under texten, komponenten till höger förflyttas över texten
- 3 - Bilderna förflyttas till vänster om texten med förhållande på höjd av texten
- 4 - Bilderna förflyttas till höger om texten med förhållande på höjd av texten



Fråga - Vad tror ni kommer hända om vi kör denna app?

- 1 - Kommer se ut precis som på bilden
- 2 - Komponenten till vänster förflyttas under texten, komponenten till höger förflyttas över texten
- 3 - Bilderna förflyttas till vänster om texten med förhållande på höjd av texten**
- 4 - Bilderna förflyttas till höger om texten med förhållande på höjd av texten



Fråga - Vad tror ni kommer hända om vi kör denna app?

- 1 - Kommer se ut precis som på bilden
- 2 - Bilderna förhåller sig till under texten med samma höjd
- 3 - Bilderna förhåller sig till över texten med samma höjd
- 4 - Bilderna hoppar längst ner på sidan

4:59



demo9



TextView

RESULT

Fråga - Vad tror ni kommer hänta om vi kör denna app?

- 1 - Kommer se ut precis som på bilden
- 2 - Bilderna förhåller sig till under texten med samma höjd
- 3 - Bilderna förhåller sig till över texten med samma höjd
- 4 - Bilderna hoppar längst ner på sidan

It's tough, that's ok!

Låt oss titta mer på hur ConstraintLayout fungerar och hur vi bäst kan arbeta med Constraints!

Parent position

Constrain the side of a view to the corresponding edge of the layout.

In figure 7, the left side of the view is connected to the left edge of the parent layout. You can define the distance from the edge with margin.

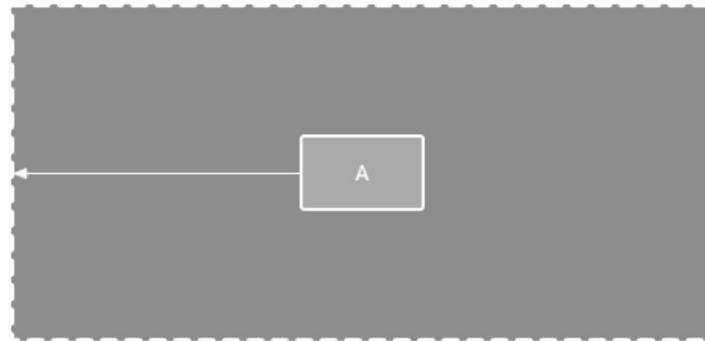


Figure 7. A horizontal constraint to the parent

Order position

Define the order of appearance for two views, either vertically or horizontally.

In figure 8, B is constrained to always be to the right of A, and C is constrained below A. However, these constraints do not imply alignment, so B can still move up and down.

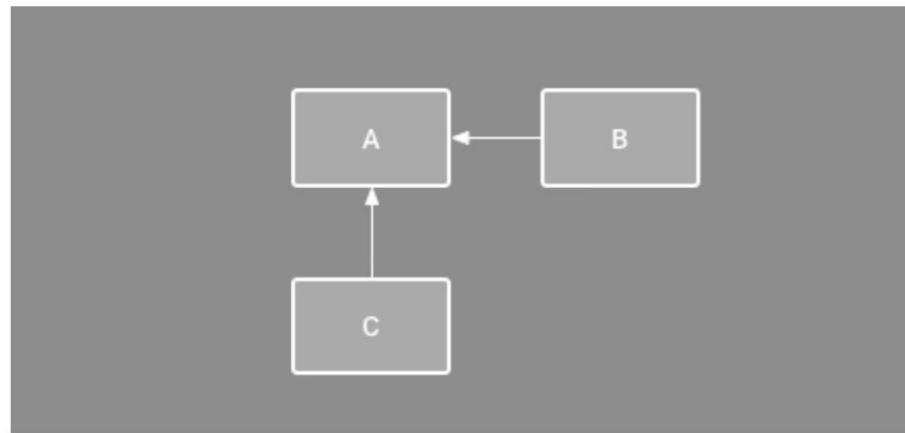


Figure 8. A horizontal and vertical constraint

Alignment

Align the edge of a view to the same edge of another view.

In figure 9, the left side of B is aligned to the left side of A.
If you want to align the view centers, create a constraint on both sides.

You can offset the alignment by dragging the view inward from the constraint. For example, figure 10 shows B with a 24dp offset alignment. The offset is defined by the constrained view's margin.

You can also select all the views you want to align, and then click **Align**  in the toolbar to select the alignment type.

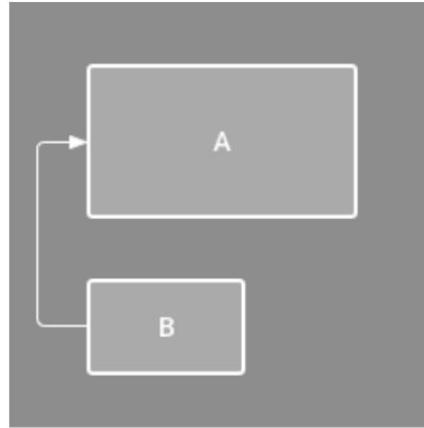


Figure 9. A horizontal alignment constraint

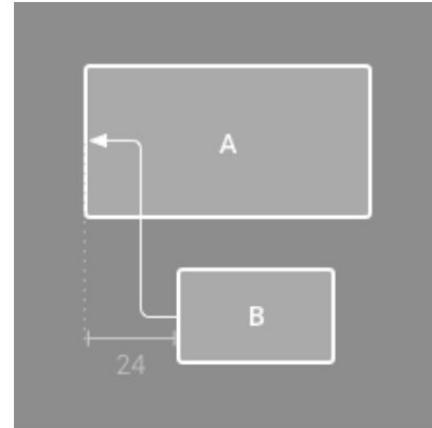


Figure 10. An offset horizontal alignment constraint

Baseline alignment

Align the text baseline of a view to the text baseline of another view.

In figure 11, the first line of B is aligned with the text in A.

To create a baseline constraint, right-click the text view you want to constrain and then click **Show Baseline**. Then click on the text baseline and drag the line to another baseline.

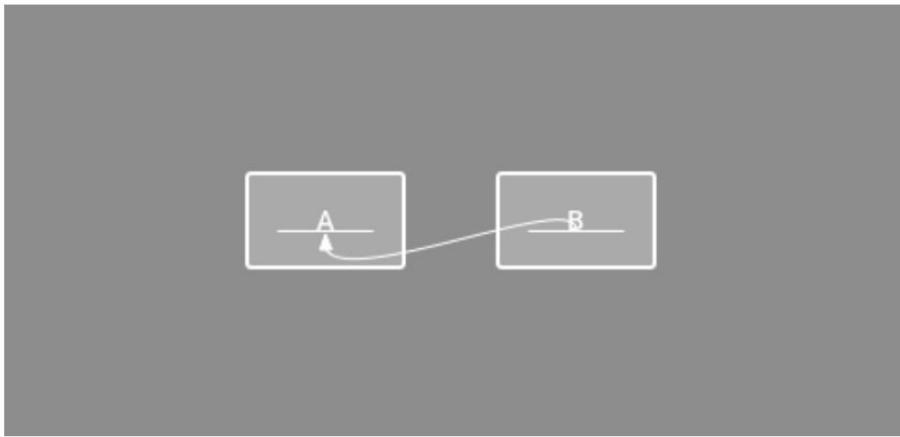


Figure 11. A baseline alignment constraint

Constrain to a guideline

You can add a vertical or horizontal guideline to which you can constrain views, and the guideline will be invisible to app users. You can position the guideline within the layout based on either dp units or percent, relative to the layout's edge.

To create a guideline, click **Guidelines**  in the toolbar, and then click either **Add Vertical Guideline** or **Add Horizontal Guideline**.

Drag the dotted line to reposition it and click the circle at the edge of the guideline to toggle the measurement mode.

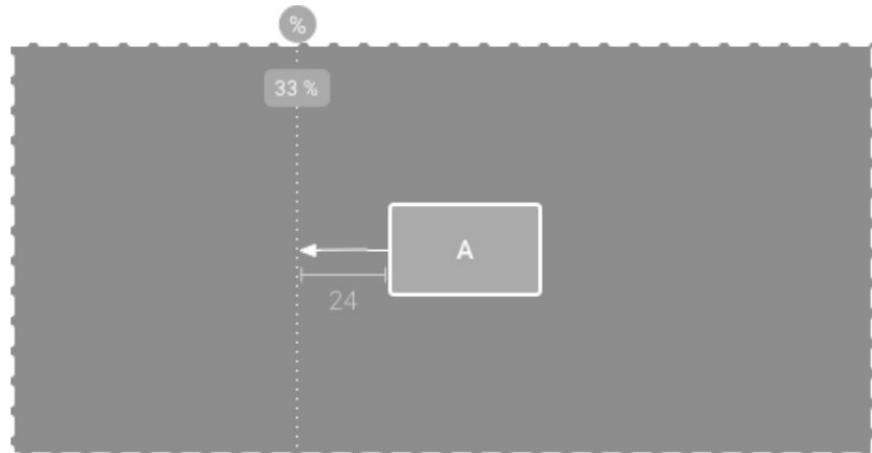


Figure 12. A view constrained to a guideline

Constrain to a barrier

Similar to a guideline, a barrier is an invisible line that you can constrain views to. Except a barrier does not define its own position; instead, the barrier position moves based on the position of views contained within it. This is useful when you want to constrain a view to the a set of views rather than to one specific view.

For example, figure 13 shows view C is constrained to the right side of a barrier. The barrier is set to the "end" (or the right side in a left-to-right layout) of both view A and view B. So the barrier moves depending on whether the right side of view A or view B is is farthest right.

To create a barrier, follow these steps:

1. Click **Guidelines**  in the toolbar, and then click **Add Vertical Barrier** or **Add Horizontal Barrier**.
2. In the **Component Tree** window, select the views you want inside the barrier and drag them into the barrier component.
3. Select the barrier from the **Component Tree**, open the **Attributes**  window, and then set the **barrierDirection**.

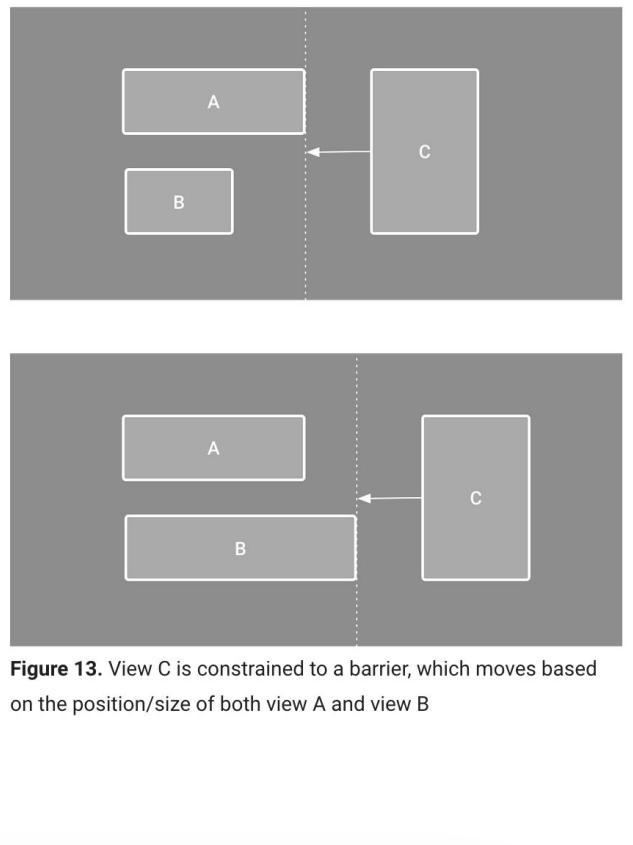


Figure 13. View C is constrained to a barrier, which moves based on the position/size of both view A and view B

Set size as a ratio

You can set the view size to a ratio such as 16:9 if at least one of the view dimensions is set to "match constraints" (0dp). To enable the ratio, click **Toggle Aspect Ratio Constraint** (callout 1 in figure 14), and then enter the **width:height** ratio in the input that appears.

If both the width and height are set to match constraints, you can click **Toggle Aspect Ratio Constraint** to select which dimension is based on a ratio of the other. The view inspector indicates which is set as a ratio by connecting the corresponding edges with a solid line.

For example, if you set both sides to "match constraints", click **Toggle Aspect Ratio Constraint** twice to set the width be a ratio of the height. Now the entire size is dictated by the height of the view (which can be defined in any way) as shown in figure 15.

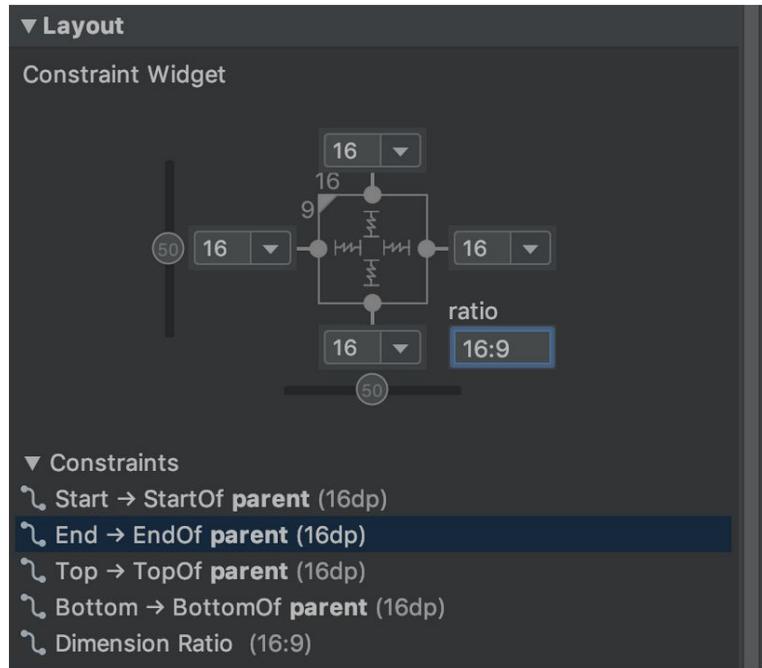
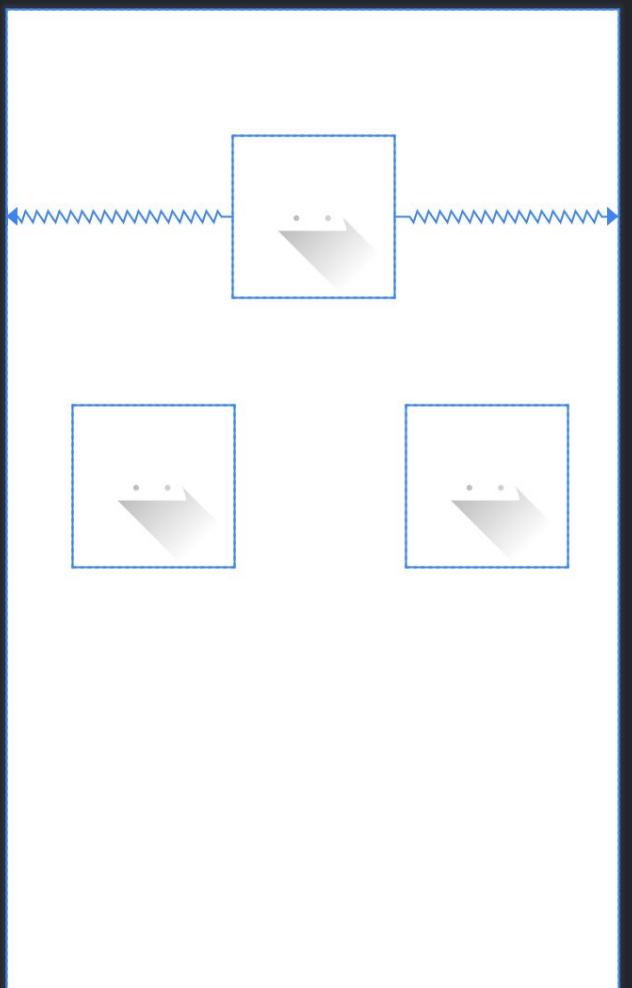


Figure 15. The view is set to a 16:9 aspect with the width based on a ratio of the height.



"Ta följande bild - våra komponenter hamnar utanför vy'n väldigt ofta och även på ställen vi aldrig satt dem på.

Även efter att jag använt mig av constraints.."

DEMONSTRATE:

- + Components fallings outside of view on Rotation
- + Solution





Problem

Hur får vi komponenter till att ej hamn utanför vår vy?



Solution

Padding är den största orsaken till varför komponenten faller utanför vår vy, försök förhålla padding till väldigt små tal.



Frågor?



04

Uppgifter

&

Eget Arbete

Uppgifter

Välkommen till första uppgiften!

Uppgifterna är till för att testa dina färdigheter och kunskaper för att både öva och repetera på det vi har arbetat med under föreläsningarna.

Dessa är **INTE** obligatoriska.
Men är starkt rekommenderat att arbeta med.



MINNS DU?

// Vad är en Navigation Graph?

// Kan man tillämpa Aktiviteter inom
Navigation Graph?

Vilka **begränsningar** finns?

// Varför använder man 'Global Action'
när det går att navigera utan?

// Vad är den vanligaste orsaken för
att komponenter **faller utanför vy'n**?

```
1 // -Uppgift #1- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5 Skapa ett nytt projekt!
```

```
6 Döp projektet till: Lektion_9_uppgifter
```

```
7
```

```
8 Skapa en ny 'Naigation Resource Folder'
```

```
9 Skapa 3 fragments via Navigation Graph.
```

```
10 First, Second & Third fragment.
```

```
11 Drag och släpp kopplingen från pilarna så
```

```
12 att det går att navigera mellan:
```

```
13
```

```
14 1 → 2 → 3 → 1
```

```
15 */
```

```
16
```

```
17 // HINT & Examples
```

```
18 hint(" Slide #12
```

```
19
```

```
20 Glöm inte: Navigation Host
```

```
21 Glöm inte: namnge komponenterna till något
```

```
22 relevant. T.ex FirstFragment.
```

```
23 Glöm inte: ID
```

```
    ")
```



Kom igång enkelt med uppgift #1

Navigering sker inom Uppgift #2

```
1           // -Uppgift #2- //
2
3 /* INSTRUCTIONS
4
5     Tillämpa funktionalitet för navigering!
6
7     Skapa sedan en fragment som agerar som
8     en hub, eller som ett: 'home'
9 */
10
11 // HINT & Examples
12 hint(" Slide #46 ")
13
14
15
16
17
18
19
20
21
22
23
```



Nu är det dags att prova sig på en ny navigering!

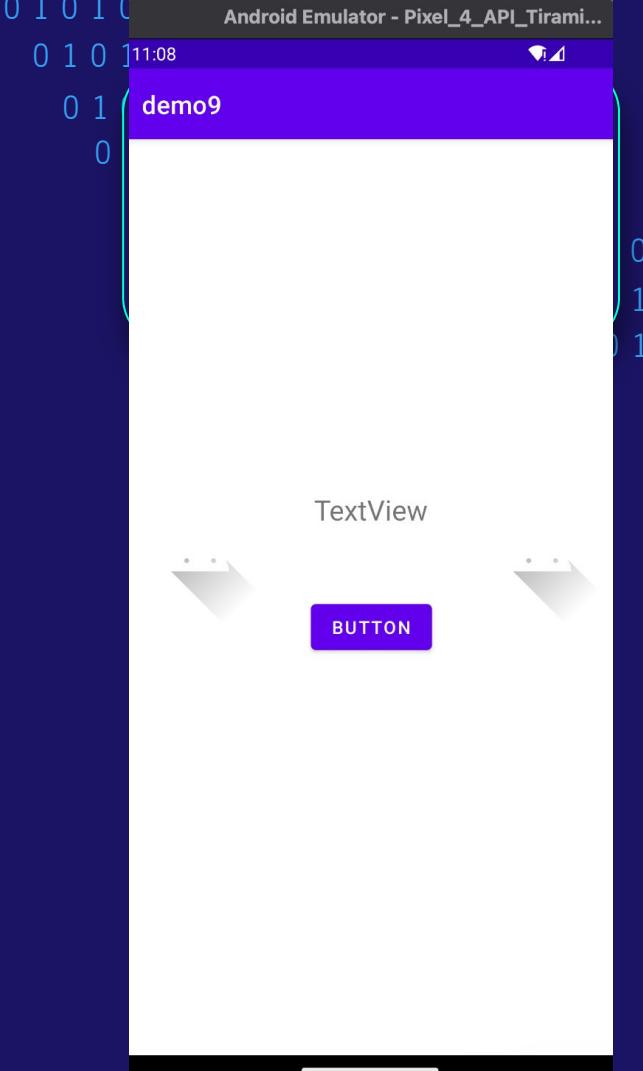
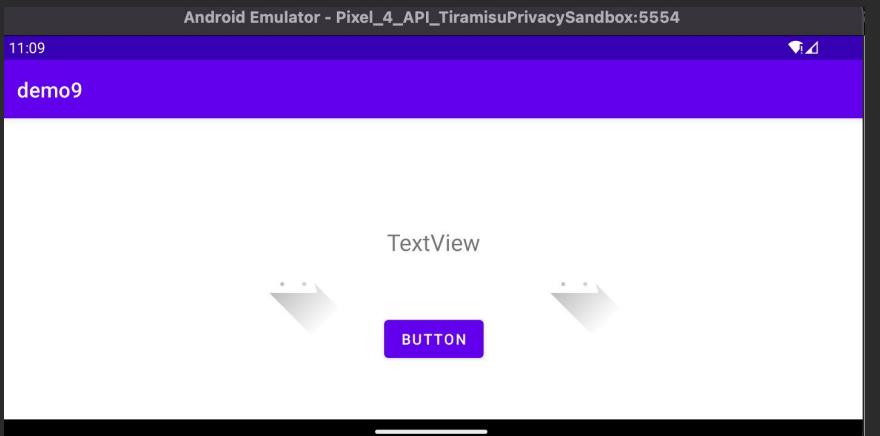
Home fragmentet skall vi arbeta med på uppgift #3

```
1          // -Uppgift #3- //
2          The Tough Nut
3
4  /* INSTRUCTIONS
5
6      Alla fragment kan ta sig till 'Home'
7      Via en Global Action
8
9      Inom home finns en knapp som heter 'GO BACK'
10     och när vi trycker, hoppar vi ett steg
11     tillbaka inom vår 'navigation stack'
12
13 */
14
15 // HINT & Examples
16 hint(" Guide för Global Actions: Slide #54")
17
18
19
20
21
22
23
```

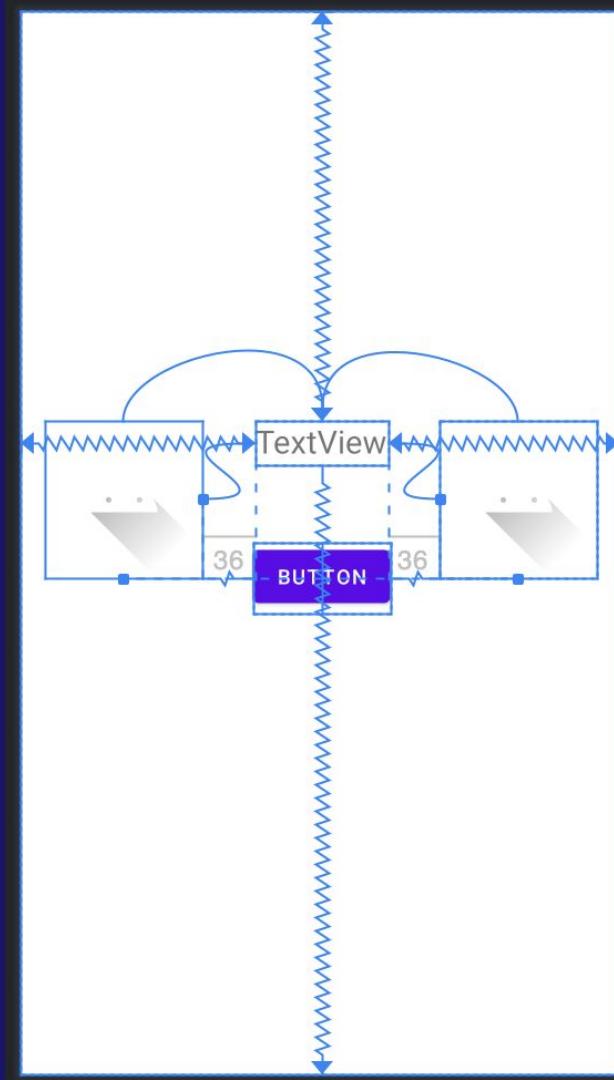


Arrays är en *datastruktur*, storlek kan variera. Detta är viktigt när vi försöker komma åt 'sista' värdet i datastrukturen.

```
1 // -Uppgift #4- //
2     The Tough Nut
3
4 /* INSTRUCTIONS
5
6     Med ConstraintLayout,
7     efterlikna bilden till höger:
8 */
9
10 // HINT & Examples
11 hint("Landscape ska se ut som bilden nedan: ")
12
13
14
15
16
17
18
19
20
21
22
23
```



```
1 // -Uppgift #4- //
2 Need a hint?
3
4 Here's the solution →
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```



THANKS !

Do you have any questions?
kristoffer.johansson@sti.se

sti.learning.nu/

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

*You can also contact me VIA Teams (quicker response)
Du kan också kontakta mig VIA Teams (Snabbare svar)*