**Android Navigation. Анимация переходов между пунктами назначения.**

В первой части этого урока мы анимируем переходы между экранами в нашем приложении. Для анимации в андроид-приложениях традиционно использовался Animation Framework.  Он позволяет анимировать view-компоненты экрана, графические ресурсы, создавать покадровые анимации. Реализовать анимацию можно как с помощью специальных xml-ресурсов, так и непосредственно в коде.

**Переходы между экранами с Animation Framework**

**Подготовим проект**

В этом уроке будем использовать уже готовый проект из прошлого урока.

**Добавим ресурсы анимации**

В папке res создайте папку anim и добавьте в нее такие файлы:

slide\_in\_left.xml



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <?xml version="1.0" encoding="utf-8"?>  <set xmlns:android="http://schemas.android.com/apk/res/android">      <translate android:fromXDelta="-50%p" android:toXDelta="0"          android:duration="@android:integer/config\_mediumAnimTime"/>      <alpha android:fromAlpha="0.0" android:toAlpha="1.0"          android:duration="@android:integer/config\_mediumAnimTime" />  </set> |

slide\_in\_right.xml



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <?xml version="1.0" encoding="utf-8"?>  <set xmlns:android="http://schemas.android.com/apk/res/android">      <translate android:fromXDelta="0" android:toXDelta="50%p"          android:duration="@android:integer/config\_mediumAnimTime"/>      <alpha android:fromAlpha="1.0" android:toAlpha="0.0"          android:duration="@android:integer/config\_mediumAnimTime" />  </set> |

slide\_out\_left.xml



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <?xml version="1.0" encoding="utf-8"?>  <set xmlns:android="http://schemas.android.com/apk/res/android">      <translate android:fromXDelta="0" android:toXDelta="-50%p"          android:duration="@android:integer/config\_mediumAnimTime"/>      <alpha android:fromAlpha="1.0" android:toAlpha="0.0"          android:duration="@android:integer/config\_mediumAnimTime" />  </set> |

slide\_out\_right.xml



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <?xml version="1.0" encoding="utf-8"?>  <set xmlns:android="http://schemas.android.com/apk/res/android">      <translate android:fromXDelta="0" android:toXDelta="50%p"          android:duration="@android:integer/config\_mediumAnimTime"/>      <alpha android:fromAlpha="1.0" android:toAlpha="0.0"          android:duration="@android:integer/config\_mediumAnimTime" />  </set> |

Как видите, во всех четырех файлах похожий код – они описывают перемещение из одной точки координат в другую с помощи директивы <translate> и изменение прозрачности от нуля до единицы при помощи <alpha>. Длительность для каждого действия можно задать в миллисекундах в свойстве duration – здесь ссылка на стандартную переменную из внутренних ресурсов android. Строго говоря, добавленные нами ресурсы тоже являются стандартными ресурсами системы android, просто взяты они из открытых исходников системы. К ним также можно обращаться и напрямую из кода, не создавая в папке res/anim вашего проекта. Однако документация не рекомендует такой подход, поскольку в разных версиях наборы ресурсов могут меняться. Поэтому рекомендуется копировать их из исходников android, доступных на гитхабе.

По [ссылке](https://android.googlesource.com/platform/frameworks/base/+/HEAD/core/res/res/anim/) вы найдете очень много различных ресурсов для анимации. В качестве домашнего задания я рекомендую изучить их, скопировать интересующие ресурсы в ваш проект и поэкспериментировать с ними.

**Применим анимацию к переходу между экранами**

Откройте файл графа навигации res/navigation/nav\_graph.xml. На вкладке дизайна выделите стрелку перехода между экранами. В панели атрибутов справа откройте раздел Animations.

У перехода есть четыре свойства, которые можно анимировать:

* enterAnim – вход в пункт назначения
* exitAnim – выход из пункта назначения
* popEnterAnim — вход в пункт назначения с помощью pop action
* popExitAnim — выход из пункта назначения с помощью pop action

Pop action – это действие, при котором экраны — пункты назначения – как бы выталкиваются из стека переходов назад в процессе навигации между экранами.

Для тех, кто не знает: стек переходов назад, или back stack – это своего рода история переходов по экранам, из которой их можно открыть путем нажатия кнопки «Назад» на устройстве.

В панели атрибутов перехода в разделе Animations напротив каждого свойства нажмите кнопку «Pick a resource» и выберите созданную нами ранее анимацию для каждого свойства.

В графе навигации вы должны получить такой код в секции action первого фрагмента:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <action      android:id="@+id/action\_fragmentOne\_to\_fragmentTwo"      app:destination="@id/fragmentTwo"      app:enterAnim="@anim/slide\_in\_right"      app:exitAnim="@anim/slide\_out\_left"      app:popEnterAnim="@anim/slide\_in\_left"      app:popExitAnim="@anim/slide\_out\_right"/> |

Запустите приложение и выполните переход с одного экрана на другой, а затем обратно. Как видите, все переходы сопровождаются плавной анимацией, которая делает поведение приложения более логичным. Улучшает, как говорится, пользовательский опыт.

На досуге поработайте с анимациями – поменяйте их местами, выберите другие анимации или создайте свои собственные.

А мы движемся дальше и перейдем к анимированию вьюшек при помощи Transition Framework.