

Platform-Based Development: Android Programming – Advanced UI

BS UNI studies, Spring 2019/2020

Dr Veljko Pejović
Veljko.Pejovic@fri.uni-lj.si



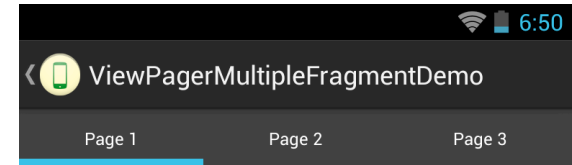
Fragments

- Main purpose:
 - Support more dynamic and flexible UI designs
 - Support UI element reusability
- Just like an Activity, a Fragment has
 - Layout
 - Lifecycle
- Unlike an Activity, a Fragment:
 - Needs a parent Activity
 - Need not be the only Fragment on the screen
 - Does not talk directly to other Fragments



Example Usages

- Passing information among different screens:
 - Fragments are tied to the **common Context**, i.e. Activity
 - Storing information within the common context allows simple information sharing



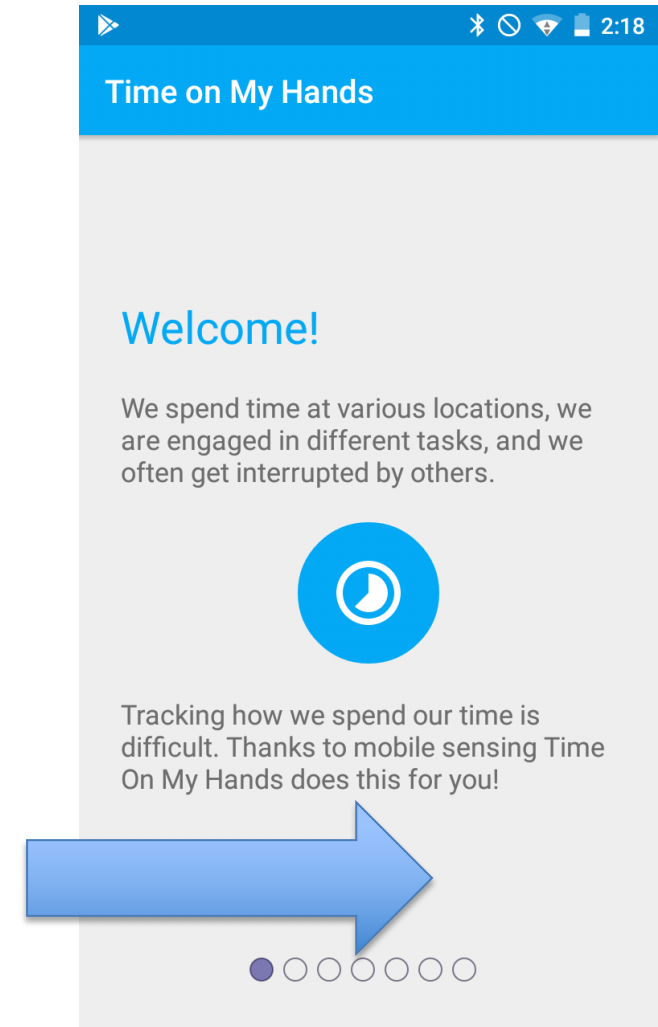
Fragment A1

Next Fragment



Example Usages

- Flexible and advanced UI metaphors
 - Tabs
 - Swipe navigation



Associating Fragment to Activity

- In an Activity's XML layout file:

```
<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>
```



Associating Fragment to Activity

- Or programmatically in Java using `FragmentManager`:

```
FragmentManager fragmentManager = getSupportFragmentManager();  
FragmentTransaction fragmentTransaction =  
    fragmentManager.beginTransaction();
```

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

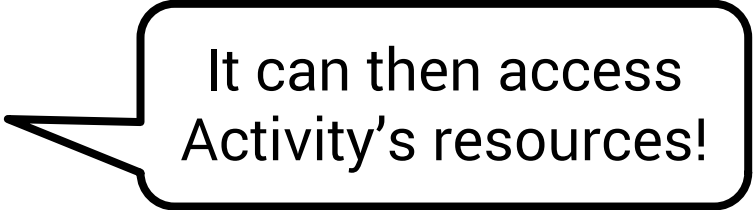
fragment_container is a ViewGroup where the fragment screen will be shown (often `FrameLayout`)



Communicating between Fragment and Activity

- Fragment can access the underlying Context (Activity)

- getContext() or getActivity()



It can then access Activity's resources!

- Activity can access fragments via FragmentManager

- findFragmentById() or findFragmentByTag()

```
ExampleFragment fragment =  
    (ExampleFragment) getSupportFragmentManager()  
        .findFragmentById(R.id.example_fragment);
```



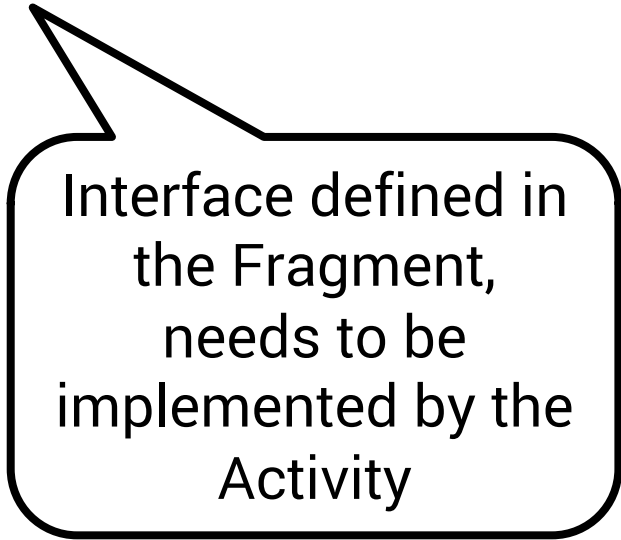
Communicating between Fragment and Activity

- Listeners
 - Fragment A has an **interface** that the underlying Activity implements
 - The interface contains a **method** that is called when the monitored data is modified
 - The Activity can now record/process this data
 - The Activity can use this data in another Fragment
- Shared ViewModel
 - ViewModels are a part of Android Jetpack library



Listener Communication Example

```
public static class FragmentA extends ListFragment {  
    ...  
    // Container Activity must implement this interface  
    public interface OnArticleSelectedListener {  
        public void onArticleSelected(Uri articleUri);  
    }  
    ...  
}
```



Interface defined in the Fragment, needs to be implemented by the Activity

Listener Communication Example

```
public static class FragmentA extends ListFragment {  
    OnArticleSelectedListener listener;  
    ...  
    @Override  
    public void onAttach(Context context) {  
        super.onAttach(context);  
        try {  
            listener = (OnArticleSelectedListener) context;  
        } catch (ClassCastException e) {  
            throw new ClassCastException(context.toString() +  
" must implement OnArticleSelectedListener");  
        }  
    }  
    ...  
}
```

The Fragment “refuses” an Activity that does not implement the Interface



Listener Communication Example

```
public static class FragmentA extends ListFragment {
    OnArticleSelectedListener listener;
    ...
    @Override
    public void onItemClick(ListView l, View v,
                            int position, long id) {
        // Append the clicked item's row ID with
        //the content provider Uri
        Uri noteUri = ContentUris
            .withAppendedId(ArticleColumns.CONTENT_URI, id);
        // Send the event and Uri to the host activity
        listener.onArticleSelected(noteUri);
    }
    ...
}
```

The Fragment sets the data value for the Activity



Example

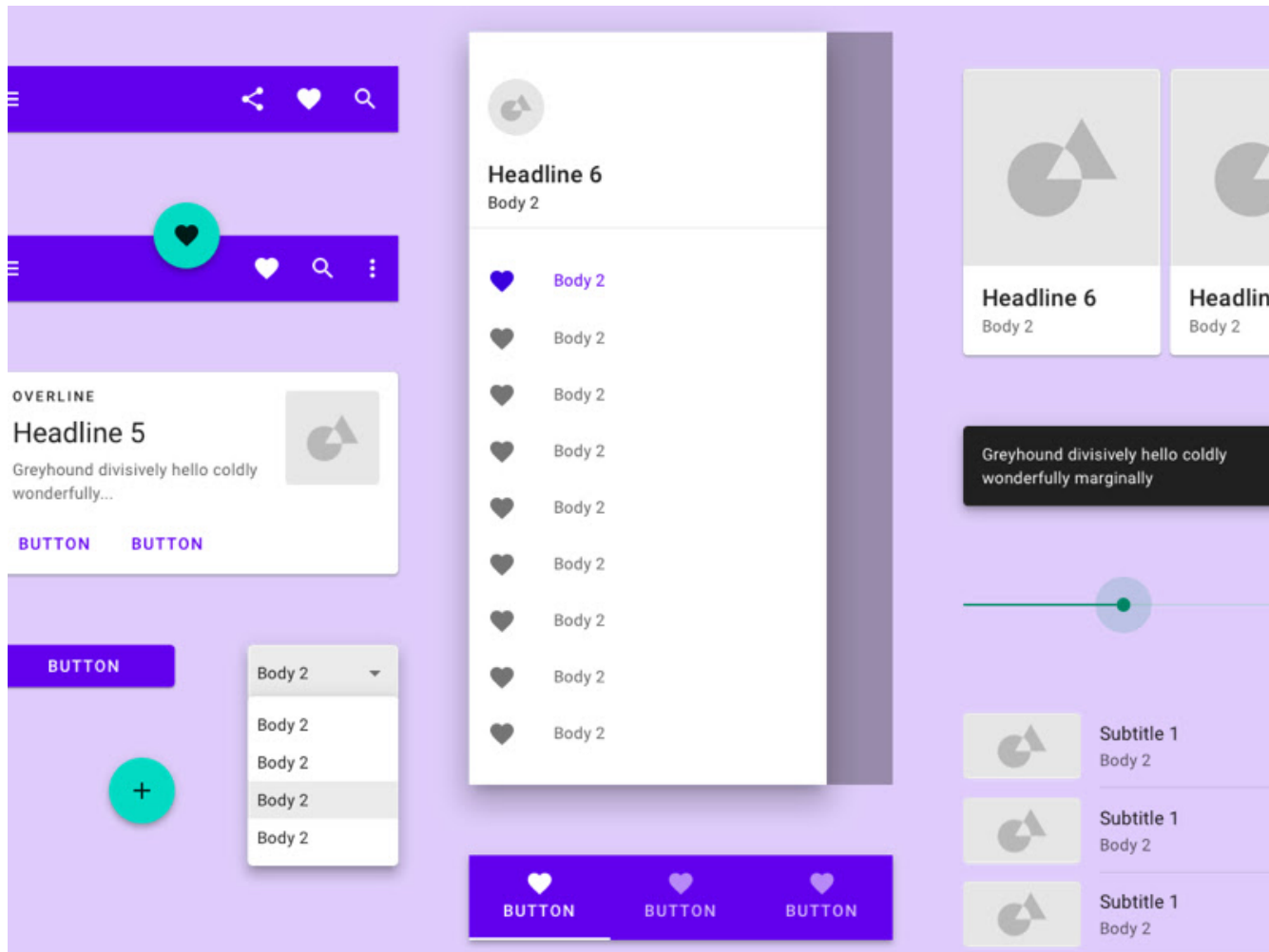


Material Design

- “...an adaptable system of guidelines, components, and tools that support the best practices of user interface design.”
- <https://material.io/design/>



Material Design



Material Design in Android

- Design library
 - Include as a dependency in your module

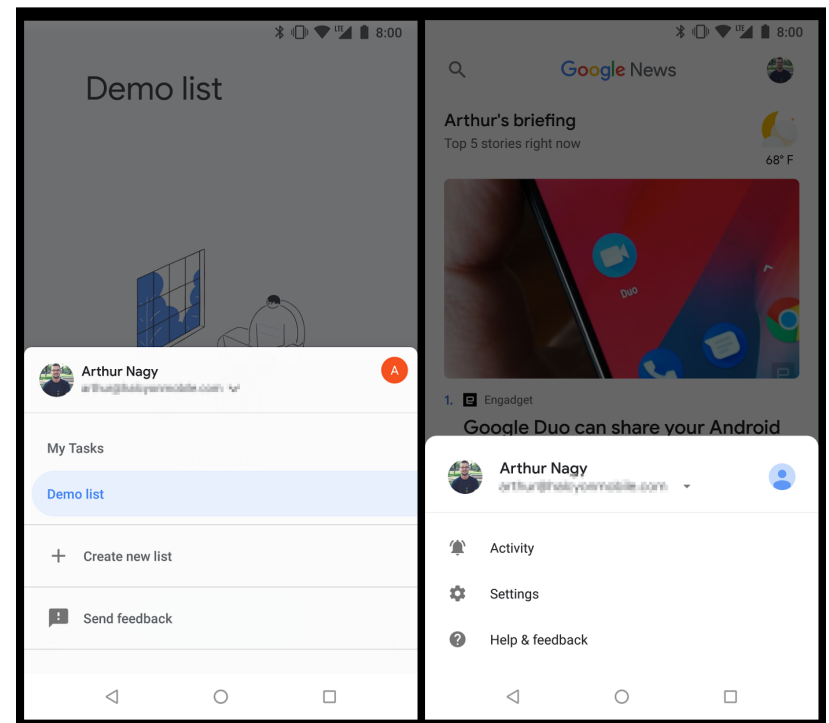
```
dependencies {  
    // ...  
    implementation 'com.google.android.material:material:<version>'  
    // ...  
}
```

Used to be a part of
the support library:
`com.android.support:design`



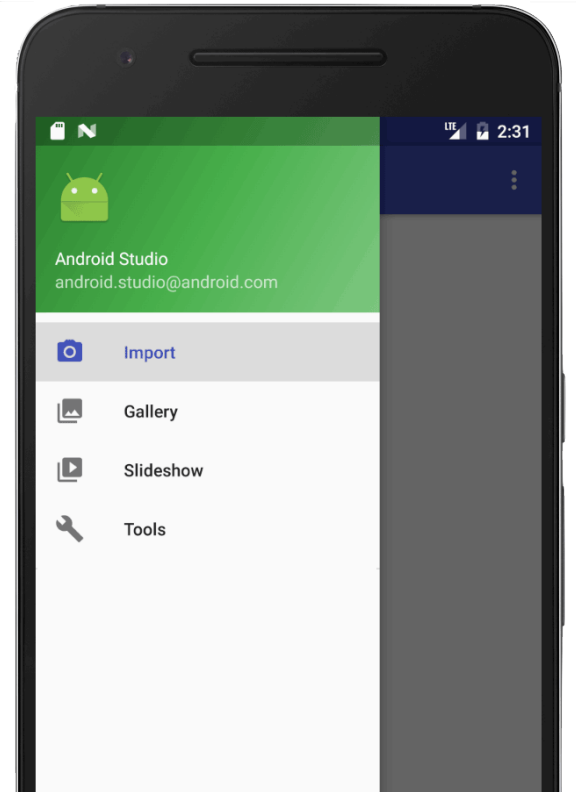
Material Design in Android

- Design library components
 - CoordinatorLayout
 - BottomSheet
 - FloatingActionButton
 - Tabs
 - CardView
 - Navigation Drawer
 - ...



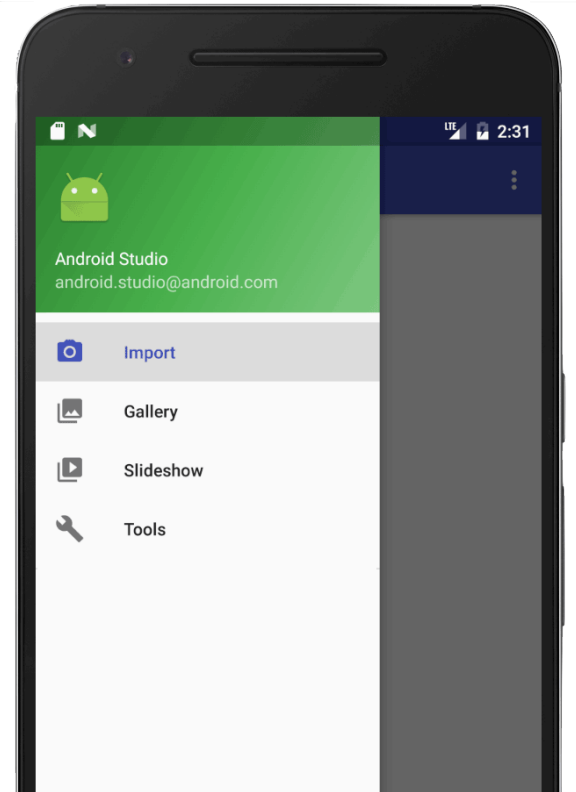
Navigation Drawer

- Pullout panel with a range of menu options and a banner-like title view
 - Highly popular navigation item (Gmail, Play Store, Google Drive)
 - Consists of:
 - DrawerLayout
 - NavigationView
 - ActionBarDrawerToggle



Navigation Drawer

- DrawerLayout contains
 - The **main layout** shown when the drawer is pulled in
 - The **NavigationView** shown when the drawer is pulled out
 - NavigationView should be populated with menu items



Navigation Drawer

- DrawerLayout

```
<androidx.drawerlayout.widget.DrawerLayout ...>
```

```
<include
```

```
    layout="@layout/some_main_layout"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent" />
```

```
<com.google.android.material.navigation.NavigationView
```

```
    android:id="@+id/nav_view"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="match_parent"
```

```
    android:layout_gravity="start"
```

```
    android:fitsSystemWindows="true"
```

```
    app:headerLayout="@layout/nav_header"
```

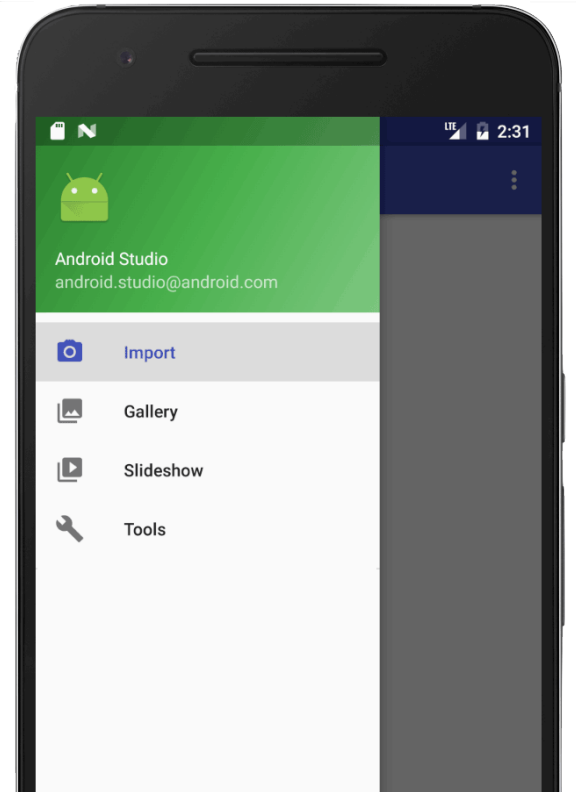
```
    app:menu="@menu/navigation_menu" />
```

```
</androidx.drawerlayout.widget.DrawerLayout>
```



Navigation Drawer

- ActionBarDrawerToggle
 - Defines how the ActionBar and the DrawerLayout will interact
 - Associated with a DrawerLayout
 - Detects when a drawer is opened or closed



Example



Fragment Navigation with Navigation Drawer

- The first child of the DrawerLayout should be a placeholder for your fragments
 - Usually a FrameLayout
- NavigationView's `onNavigationItemSelectedListener` should
 - **Create a new Fragment** as needed
 - **Call a FragmentManager** that will `beginTransaction` to replace the content of the placeholder with this new fragment
 - Don't forget to `commit()` the transaction

This is the old-school way!



Navigation Component

- Navigation Component (part of Android X)
 - Solves the issues of
 - Complex navigation – many Fragments
 - Different interaction methods (drawer, bottom navigation, tabs, etc.)
 - Backstack
 - Move towards single-Activity/multiple-Fragment apps
 - Define:
 - Navigation graph
 - Navigation host fragment
 - Navigation controller



Example

