



In-class Project

A Simple Swiping App



Overview

In this in-classroom project, you will learn how to create a simple app allowing swiping gestures with the use of ViewPager2.



AndroidVersionsSwipes

In this project you will learn how to work to `ViewPager2` to implement basic swipe gestures.

- Download the `AndroidVersionsSwipes` app starter project from eLC (in the `and Sample Apps` folder) unzip it in your `AndroidStudioProjects` folder on your laptop.
- The project you downloaded is only a partial implementation and you will have to complete it today.
- A suitable `Fragment` class for displaying information about Android versions, `AndroidVersionFragment`, is already included; examine its `newInstance` method.



Modify the `activity_main` layout

- Open the `activity_main` layout and remove the placeholder `TextView` with `Hello world!`.
- In its place, add a `ViewPager2`: in the Palette panel with views, open the `Containers` group and drag a `ViewPager2` onto your layout.
- Set its id to `viewpager`, if not already set.
- Set its `layout_width` and `layout_height` to `0dp` (`match_constraint`) and then add four constraints on binding the *start* to *start* of the *parent*, *end* to *end* of the *parent* and then do the same for the *top* to *top* of the *parent* and *bottom* to *top* and *bottom* of the *parent*, respectively.



Implement `FragmentManager`

- In the Android project view on the left, right click on the package `edu.uga.cs.androidversionsswipes`, select `New->Java class` and create a new class called `AndroidVersionsPagerAdapter`.
- In the editor's tab with the new class, add the

```
extends FragmentManager
```

clause. The new class will be a special `FragmentManager` class, which will work with our `ViewPager2`.



Implement `FragmentManagerAdapter`

- Add the following constructor to the new class:

```
public AndroidVersionsPagerAdapter(  
    FragmentManager fragmentManager,  
    Lifecycle lifecycle ) {  
    super( fragmentManager, lifecycle );  
}
```

- We will create an instance of this adapter in the `MainActivity` class and set it to work with a `ViewPager2`.



Implement `FragmentManager`

- Now, add the `createFragment` method:

```
@Override  
  
public Fragment createFragment(int position) {  
    return AndroidVersionFragment  
        .newInstance( position );  
}
```

- Android will call this method every time the user swipes left or right; the supplied `position` parameter indicates the page position to display while swiping.



Implement `FragmentManager`

- Finally, add the `getItemCount` method:

```
@Override  
  
public int getItemCount() {  
    return AndroidVersionFragment  
        .getNumberOfVersions();  
}
```

- Android will call this method when it wants to know how many pages will be available while swiping.
- Add the necessary import statements.



Finalize the MainActivity Class

- Open the MainActivity class
- Right after the setContentView call, add the following code:

```
ViewPager2 pager = findViewById( R.id.viewpager );  
AndroidVersionsPagerAdapter avpAdapter = new  
    AndroidVersionsPagerAdapter(  
        getSupportFragmentManager(), getLifecycle() );  
pager.setOrientation(  
    ViewPager2.ORIENTATION_HORIZONTAL );  
pager.setAdapter( avpAdapter );
```

- Make sure that the ViewPager2's XML id is correct.
- Add the necessary import statements.



Build and Run the App

- Build and run the app in the Android Emulator.
- You should be able to swipe left and swipe right and move through the consecutive versions of the Android system.
- If something is not working properly, correct any errors.
- Capture 2 screenshots of 2 different Android Versions, as shown in your app running in the emulator, and upload the images to the `Android Versions Swipes` assignment folder on eLC (in the `In-class` group of assignments).
- The assignment folder will close today at 11:59pm.