/\*

\* Copyright 2012 The Android Open Source Project

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

package com.example.android.animationsdemo;

import android.app.Fragment;

import android.app.FragmentManager;

import android.content.Intent;

import android.os.Bundle;

import android.support.v13.app.FragmentStatePagerAdapter;

import android.support.v4.app.FragmentActivity;

import android.support.v4.app.NavUtils;

import android.support.v4.view.PagerAdapter;

import android.support.v4.view.ViewPager;

import android.view.Menu;

import android.view.MenuItem;

/\*\*

\* Demonstrates a "screen-slide" animation using a {@link ViewPager}. Because {@link ViewPager}

\* automatically plays such an animation when calling {@link ViewPager#setCurrentItem(int)}, there

\* isn't any animation-specific code in this sample.

\*

\* <p>This sample shows a "next" button that advances the user to the next step in a wizard,

\* animating the current screen out (to the left) and the next screen in (from the right). The

\* reverse animation is played when the user presses the "previous" button.</p>

\*

\* @see ScreenSlidePageFragment

\*/

public class ScreenSlideActivity extends FragmentActivity {

/\*\*

\* The number of pages (wizard steps) to show in this demo.

\*/

private static final int NUM\_PAGES = 5;

/\*\*

\* The pager widget, which handles animation and allows swiping horizontally to access previous

\* and next wizard steps.

\*/

private ViewPager mPager;

/\*\*

\* The pager adapter, which provides the pages to the view pager widget.

\*/

private PagerAdapter mPagerAdapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_screen\_slide);

// Instantiate a ViewPager and a PagerAdapter.

mPager = (ViewPager) findViewById(R.id.pager);

mPagerAdapter = new ScreenSlidePagerAdapter(getFragmentManager());

mPager.setAdapter(mPagerAdapter);

mPager.setOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener() {

@Override

public void onPageSelected(int position) {

// When changing pages, reset the action bar actions since they are dependent

// on which page is currently active. An alternative approach is to have each

// fragment expose actions itself (rather than the activity exposing actions),

// but for simplicity, the activity provides the actions in this sample.

invalidateOptionsMenu();

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

super.onCreateOptionsMenu(menu);

getMenuInflater().inflate(R.menu.activity\_screen\_slide, menu);

menu.findItem(R.id.action\_previous).setEnabled(mPager.getCurrentItem() > 0);

// Add either a "next" or "finish" button to the action bar, depending on which page

// is currently selected.

MenuItem item = menu.add(Menu.NONE, R.id.action\_next, Menu.NONE,

(mPager.getCurrentItem() == mPagerAdapter.getCount() - 1)

? R.string.action\_finish

: R.string.action\_next);

item.setShowAsAction(MenuItem.SHOW\_AS\_ACTION\_IF\_ROOM | MenuItem.SHOW\_AS\_ACTION\_WITH\_TEXT);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case android.R.id.home:

// Navigate "up" the demo structure to the launchpad activity.

// See http://developer.android.com/design/patterns/navigation.html for more.

NavUtils.navigateUpTo(this, new Intent(this, MainActivity.class));

return true;

case R.id.action\_previous:

// Go to the previous step in the wizard. If there is no previous step,

// setCurrentItem will do nothing.

mPager.setCurrentItem(mPager.getCurrentItem() - 1);

return true;

case R.id.action\_next:

// Advance to the next step in the wizard. If there is no next step, setCurrentItem

// will do nothing.

mPager.setCurrentItem(mPager.getCurrentItem() + 1);

return true;

}

return super.onOptionsItemSelected(item);

}

/\*\*

\* A simple pager adapter that represents 5 {@link ScreenSlidePageFragment} objects, in

\* sequence.

\*/

private class ScreenSlidePagerAdapter extends FragmentStatePagerAdapter {

public ScreenSlidePagerAdapter(FragmentManager fm) {

super(fm);

}

@Override

public Fragment getItem(int position) {

return ScreenSlidePageFragment.create(position);

}

@Override

public int getCount() {

return NUM\_PAGES;

}

}

}