
CIS 470

Mobile App Development

Lecture 9

Wenbing Zhao

Department of Electrical Engineering and Computer Science

Cleveland State University

wenbing@ieee.org

Displaying Pictures and Menus with Views

- ImageSwitcher, GridView, ImageView
- Display options menus and context menus
- WebView: display web content

ImageSwitcher

- ImageSwitcher enables the display of pictures with some animation
- Using Android Studio, create a new Android project and name it ImageSwitcher
- Modify the activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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="com.wenbing.imageswitcher.MainActivity">
    <Button
        android:text="View Windows"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button2"
    />
    <ImageSwitcher
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/button2"
        android:id="@+id/imageSwitcher">
    </ImageSwitcher>
    <Button
        android:text="View Butterfly"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
    />
</RelativeLayout>
```

ImageSwitcher

- Find on the Web one windows image (named windows.png/jpg) and one butterfly image (named butterfly.png/jpg). Add two images to your res/mipmap/hdpi folder
- Modify MainActivity.java

```
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.support.v7.app.ActionBar;  
import android.view.View;  
import android.view.animation.AnimationUtils;  
import android.widget.Button;  
import android.widget.ImageSwitcher;  
import android.widget.ImageView;  
import android.widget.Toast;  
import android.widget.ViewSwitcher;  
  
public class MainActivity extends AppCompatActivity {  
    private ImageSwitcher imgSwitcher;  
    private Button btnViewWindows,btnViewButterfly;
```

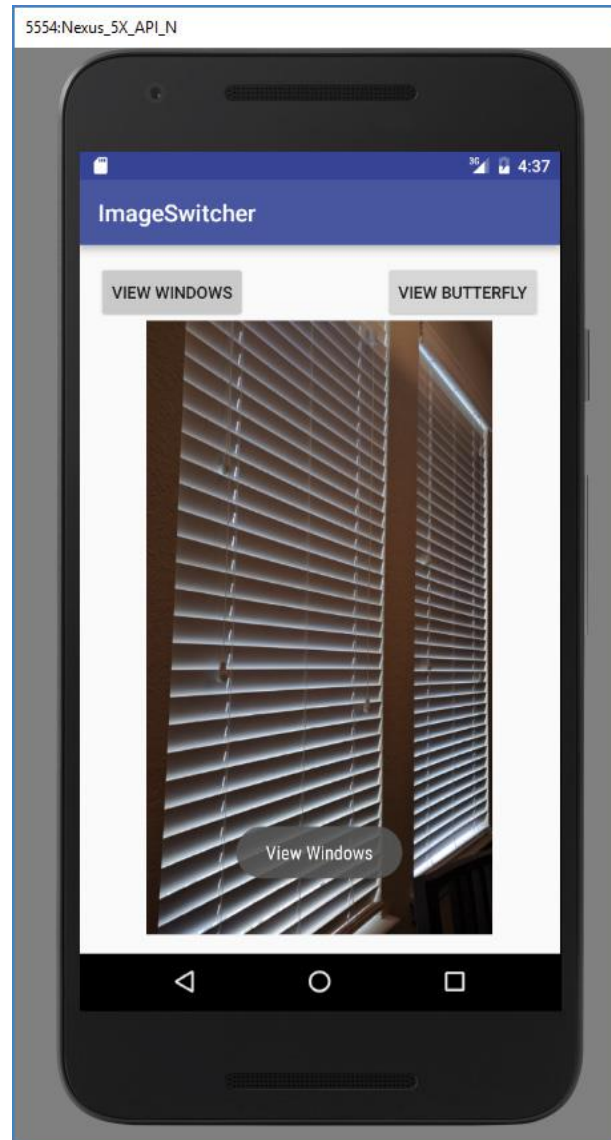
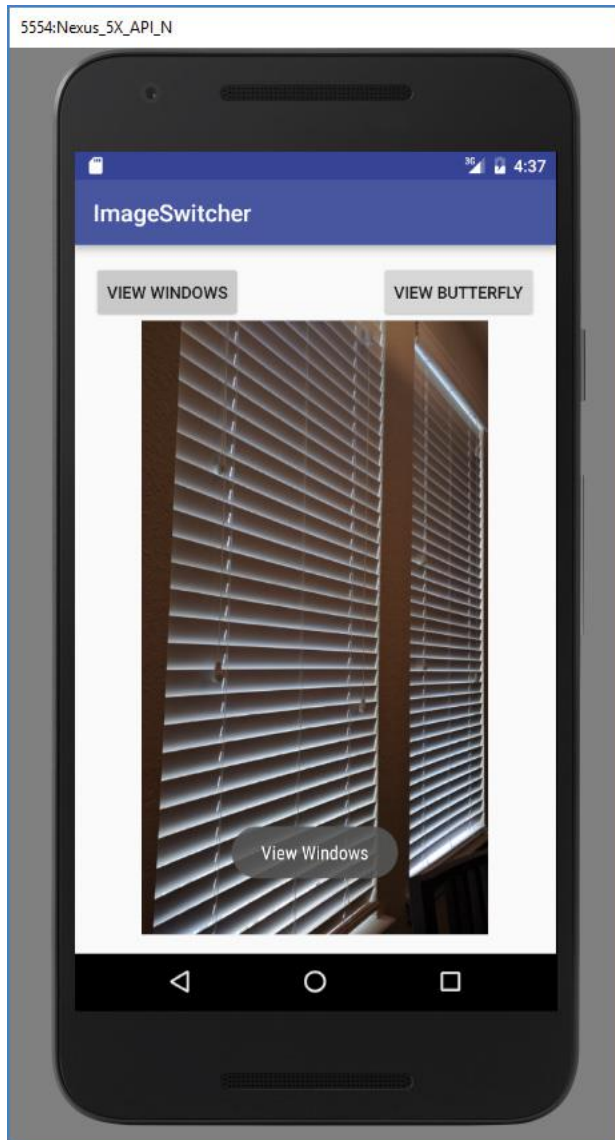
@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    imgSwitcher = (ImageSwitcher) findViewById(R.id.imageSwitcher);  
    imgSwitcher.setInAnimation(AnimationUtils.loadAnimation(this,  
        android.R.anim.fade_in));  
    imgSwitcher.setOutAnimation(AnimationUtils.loadAnimation(this,  
        android.R.anim.fade_out));  
    btnViewWindows = (Button) findViewById(R.id.button2);  
    btnViewButterfly = (Button) findViewById(R.id.button);  
    imgSwitcher.setFactory(new ViewSwitcher.ViewFactory() {  
        @Override  
        public View makeView() {  
            ImageView myView = new ImageView(getApplicationContext());  
            myView.setScaleType(ImageView.ScaleType.FIT_CENTER);  
            myView.setLayoutParams(new ImageSwitcher.LayoutParams(  
                ActionBar.LayoutParams.WRAP_CONTENT,  
                ActionBar.LayoutParams.WRAP_CONTENT));  
            return myView;  
        }  
    });
```

```

    btnViewWindows.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(getApplicationContext()
                , "View Windows", Toast.LENGTH_LONG).show();
            imgSwitcher.setImageResource(R.mipmap.windows);
        }
    });
    btnViewButterfly.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(getApplicationContext(), "View Butterfly"
                , Toast.LENGTH_LONG).show();
            imgSwitcher.setImageResource(R.mipmap.butterfly);
        }
    });
}
}

```



GridView

- The GridView shows items in a two-dimensional scrolling grid
- Use the GridView together with an ImageView to display a series of images
- Create a new app and name it Grid
- Add the windows and butterfly images to the res/mipmap folder
- Modify activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/activity_main"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="com.wenbing.grid.MainActivity">
  <GridView
    android:id="@+id/gridview"
    android:layout_width="384dp"
    android:layout_height="511dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

</android.support.constraint.ConstraintLayout>
```

GridView

■ Modify MainActivity.java:

```
import android.content.Context;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.AnimationUtils;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.GridView;
import android.widget.ImageSwitcher;
import android.widget.ImageView;
import android.widget.Toast;
import android.widget.ViewSwitcher;
```

```
public class MainActivity extends AppCompatActivity {
    private ImageSwitcher imgSwitcher;
```

```
    //---the images to display---
```

```
    Integer[] imageIDs = {
        R.mipmap.butterfly,
        R.mipmap.windows,
        R.mipmap.ic_launcher
    };
}
```

GridView

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    GridView gridView = (GridView) findViewById(R.id.gridview);  
    gridView.setAdapter(new ImageAdapter(this));  
    gridView.setOnItemClickListener(new AdapterView.OnItemClickListener()  
    {  
        public void onItemClick(AdapterView parent, View v, int position, long id) {  
            imgSwitcher.setImageResource(imageIds[position]);  
            Toast.makeText(getBaseContext(),  
                "pic" + (position + 1) + " selected",  
                Toast.LENGTH_SHORT).show();  
        }  
    });  
}
```

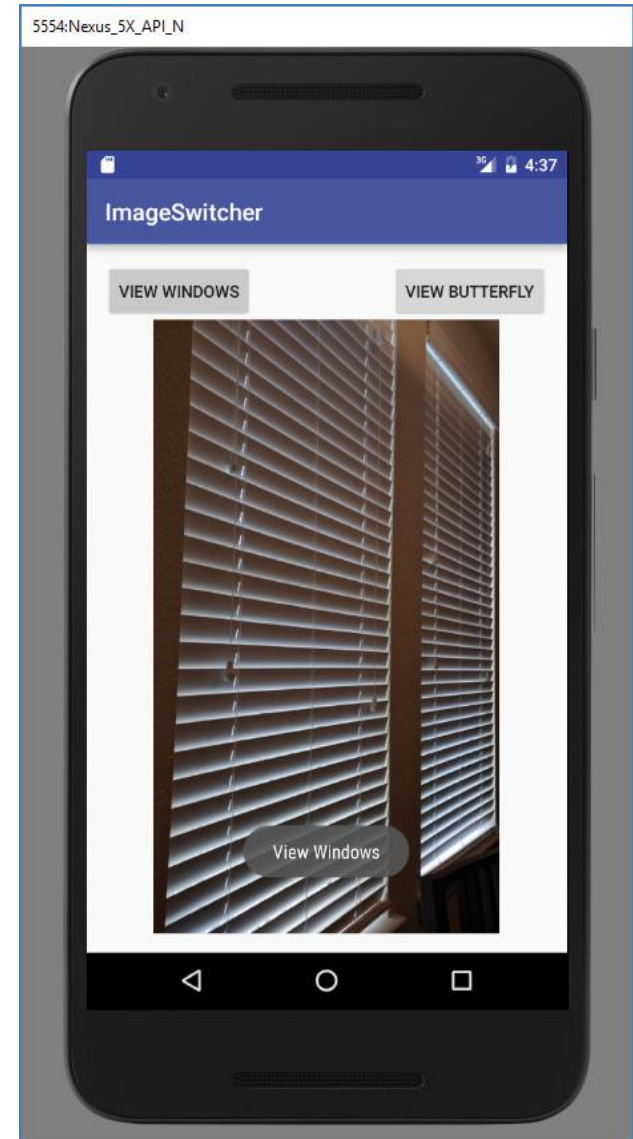
```

public class ImageAdapter extends BaseAdapter
{
    private Context context;
    public ImageAdapter(Context c) { context = c; }
    //---returns the number of images---
    public int getCount() { return imageIds.length; }
    //---returns the item---
    public Object getItem(int position) { return position; }
    //---returns the ID of an item---
    public long getItemId(int position) { return position; }
    //---returns an ImageView view---
    public View getView(int position, View convertView, ViewGroup parent) {
        ImageView imageView;
        if (convertView == null) {
            imageView = new ImageView(context);
            imageView.setLayoutParams(new GridView.LayoutParams(150, 150));
            imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            imageView.setPadding(5, 5, 5, 5);
        } else {
            imageView = (ImageView) convertView;
        }
        imageView.setImageResource(imageIds[position]);
        return imageView;
    }
}

```

GridView

- Exercise (**required**): Use ImageSwitcher such that when an image in the grid view is pushed, the full image is shown in a common area that does not overlap the grid



Using menus with views

- Options menu: displays information related to the current activity
- Context menu: displays information related to a particular view on an activity
 - tap and hold a view with a context menu to activate it
- WebView: embed a web browser in your activity

Using menus with views

- Create a new Android project named Menus
- Modify activity_main.xml:

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.wenbing.menus.MainActivity">
  <Button
    android:text="Button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:layout_editor_absoluteX="148dp"
    tools:layout_editor_absoluteY="102dp"
    android:id="@+id/button" />
  <WebView
    android:layout_width="384dp"
    android:layout_height="511dp"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="0dp"
    app:layout_constraintLeft_toLeftOf="@+id/activity_main"
    tools:layout_constraintLeft_creator="0"
    app:layout_constraintTop_toTopOf="@+id/activity_main"
    tools:layout_constraintTop_creator="0"
    app:layout_constraintRight_toRightOf="@+id/activity_main"
    tools:layout_constraintRight_creator="0"
    app:layout_constraintBottom_toBottomOf="@+id/activity_main"
    tools:layout_constraintBottom_creator="0"
    android:id="@+id/webview" />
</android.support.constraint.ConstraintLayout>
```


Using menus with views

■ Modify MainActivity.java

```
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.Toast;  
import android.view.ContextMenu;  
import android.widget.Button;  
import android.view.View;  
import android.webkit.WebSettings;  
import android.webkit.WebView;
```

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Button btn = (Button) findViewById(R.id.button);  
        btn.setOnCreateContextMenuListener(this);  
  
        WebView wv = (WebView) findViewById(R.id.webview);  
        WebSettings webSettings = wv.getSettings();  
        webSettings.setBuiltInZoomControls(true);  
        wv.loadUrl(  
            "http://chart.apis.google.com/chart" +  
            "?chs=300x225" +  
            "&cht=v" +  
            "&chco=FF6342,ADDE63,63C6DE" +  
            "&chd=t:100,80,60,30,30,30,10" +  
            "&chdl=A|B|C");  
    }  
}
```

Using menus with views

■ Modify MainActivity.java

@Override

```
public void onCreateContextMenu(ContextMenu menu, View view, ContextMenu.ContextMenuInfo menuInfo) {  
    super.onCreateContextMenu(menu, view, menuInfo);  
    createMenu(menu);  
}
```

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);  
    createMenu(menu);  
    return true;  
}
```

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    return menuChoice(item);  
}
```

```
private void createMenu(Menu menu) {  
    MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");  
    {  
        mnu1.setAlphabeticShortcut('a');  
    }  
    MenuItem mnu2 = menu.add(0, 1, 1, "Item 2");  
    {  
        mnu2.setAlphabeticShortcut('b');  
    }  
    MenuItem mnu3 = menu.add(0, 2, 2, "Item 3");  
    {  
        mnu3.setAlphabeticShortcut('c');  
    }  
    MenuItem mnu4 = menu.add(0, 3, 3, "Item 4");  
    {  
        mnu4.setAlphabeticShortcut('d');  
    }  
    menu.add(0, 4, 4, "Item 5");  
    menu.add(0, 5, 5, "Item 6");  
    menu.add(0, 6, 6, "Item 7");  
}
```

The onCreateOptionsMenu() method is called when the Menu button is pressed

When a menu item is selected, the onOptionsItemSelected() method is called

Using menus with views

■ Modify MainActivity.java

```
private boolean menuChoice(MenuItem item) {  
    switch (item.getItemId()) {  
        case 0:  
            Toast.makeText(this, "You clicked on Item 1",  
                Toast.LENGTH_LONG).show();  
            return true;  
        case 1:  
            Toast.makeText(this, "You clicked on Item 2",  
                Toast.LENGTH_LONG).show();  
            return true;  
        case 2:  
            Toast.makeText(this, "You clicked on Item 3",  
                Toast.LENGTH_LONG).show();  
            return true;  
        case 3:  
            Toast.makeText(this, "You clicked on Item 4",  
                Toast.LENGTH_LONG).show();  
            return true;  
        case 4:  
            Toast.makeText(this, "You clicked on Item 5",  
                Toast.LENGTH_LONG).show();  
            return true;  
        case 5:  
            Toast.makeText(this, "You clicked on Item 6",  
                Toast.LENGTH_LONG).show();  
            return true;  
        case 6:  
            Toast.makeText(this, "You clicked on Item 7",  
                Toast.LENGTH_LONG).show();  
            return true;  
    }  
    return false;  
}
```

Using menus with views

■ Modify AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.wenbing.menus">
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Using menus with views

- Exercise (required):
 - ❑ Add a EditText for the user to enter a URL
 - ❑ Place the button on the right of the EditText and rename to “submit”
 - ❑ Add another button beneath the EditText and name it to “bookmark”
 - ❑ On clicking the submit button, the URL should be displayed in the WebView
 - ❑ On clicking the bookmark button, the current URL should be saved as a bookmark
 - ❑ The menu for the app now should display a list of saved bookmarks
 - ❑ On selecting a saved bookmark, the corresponding webpage should be displayed

