Android

Lectures on University College
for Applied Computer Engineering
http://www.racunarstvo.hr/en/

Miljenko Cvjetko
http://holisticware.net

UI concepts

Layouts

Widgets

Tools

UI concepts
Definition

- 2 possibilities to define UI in Android
  - programatically - in code
    classes like LinearLayout, TextView, Button
  - declarative in xml
    defined in Android xml (axml) files
    represents serialized object graph (tree of xml elements)
- UI is consisted of
  - Views - base class for simple UI elements - Widgets
    TextView, EditText, Button
  - ViewGroups - base class for Layouts
    offer different types of layout architectures

UI concepts

Definition declarative in XML

- decalarative xml markup -
- based on HTML (web authoring)
- quick UI desing
- object graph object graph, xml tree
  each node represents View class (derived)
  node can have multiple attributes - properties
- folder (project folder)
  - res/layout/
    java based - eclipse, Android.Studio
  - Resource/layout/
    Xamarin.Android

UI concepts

Layouts 01

- Frame Layout
- Absolute Layout
- Relative Layout
- Linear Layout
- ListView
- ScrollView
- TableView
- TabHost
- GridView
- Gallery

UI concepts

Layouts 02

- Spinner
- ViewFilpper/ ViewSwitcher

UI concepts

Layouts 03 - Tags

- Linearlayout
  - Orientation = Horizontal | Vertical
  - Gravity
    alignment of all child elements
- ListView
  - divider - between list elements
  - dividerHeight
  - entries - array to populate ListView
  - footerDividersEnabled
  - headerDividersEnabled

UI concepts
Menus

- enable user access application functions and settings
- 3 types
  - Options Menu - user presses hardware Option menu
    2 types
    - IconMenu
    - ExpandedMenu
  - ContextMenu
    floating list of items - trigered by long press on a UI widget
  - SubMenu
    floating list of items - triggered by pressing on a item in a menu in Options men or Context menu

UI concepts

Events

- Listeners
- Handlers
- Focus

UI concepts
Notifications

- need to notify user about some data or event
- types
  - Toast Notifications
    short messages comming from background
  - StatusBar Notifiactions
    reminders comming from background and requesting actions
  - Dialog Notifications
    Activity notifiactions my require user interaction

UI concepts

Notifications - Dialgos

- types
  - AlertDialog
  - ProgressDialog
  - DatePickerDialog
  - TImePickerDialog
  - Custom Dialogs

UI concepts
Lifecycle Activity

- States
  - Created
  - Started (running)
  - Resumed (running)
  - Paused (partially Visible)
  - Stopped (hidden) activity is partially obscured by another activity the other activity that's in the foreground is semi-transparent or doesn't cover the entire screen. The paused activity does not receive user input and cannot execute any code.
  - Destroyed

UI concepts

Lifecycle Activity

- Events
  - Events
  - onCreate
  - onDestroy
  - onStart
  - onStop
  - onPause
  - onResume

UI concepts
Lifecycle Activity

```
//===========================================================================
// Error is just to distinguish it in LogCat 8)

@Override
protected void onDestroy() {
        super.onDestroy();

        Log.e("LIFECYCLE = ", this.getClass().toString() + ".onDestroyed");
        return;
}

@Override
protected void onStart() {
        super.onStart();

        Log.e("LIFECYCLE = ", this.getClass().toString() + ".onStart");
        return;
}

@Override
protected void onStop() {
        super.onStop();

        Log.e("LIFECYCLE = ", this.getClass().toString() + ".onStop");
        return;
}


@Override
protected void onPause() {
        super.onPause();

        Log.e("LIFECYCLE = ", this.getClass().toString() + ".onPause");
        return;
}

@Override
protected void onResume() {
        super.onResume();

        Log.e("LIFECYCLE = ", this.getClass().toString() + ".onResume");
        return;
}
//===========================================================================
```

UI concepts
Lifecycle Activity 03

- onCreate: loading layout (UI) and getting Widget objects

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

//-----------------------------------------------
// loading UI (layouts)
setContentView(R.layout.activity_main);
//-----------------------------------------------

/*
 * not using fragments!!
if (savedInstanceState == null) {
getSupportFragmentManager().beginTransaction()
.add(R.id.container, new PlaceholderFragment()).commit();
}
*/

//-----------------------------------------------
// getting Widget objects
buttonNavigationSimple = (Button) findViewById(R.id.buttonNavigateSimple);
buttonNavigationComplex = (Button) findViewById(R.id.buttonNavigateComplex);
editTextDataToSend = (EditText) findViewById(R.id.editTextDataToSend);
//-----------------------------------------------


// Touch/Click Event programmatically
// not using attribute in xml layout
buttonNavigationComplex.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
navigateComplex(v);
}
});

Log.e("LIFECYCLE = ", this.getClass().toString() + ".onCreate");

return;
}
```

UI concepts

Lifecycle Activity 03

UI concepts

Lifecycle Application 01

- custom Application object (class derived from Application
  - used for global (application wide) objects/variables
- States
  - Started (created)
  - Stopped (terminated)
- Events
  - onCreate
  - onTerminate
  - onLowMemory
  - onConfigurationChanged

UI concepts
Lifecycle Application 02

```
package com.example.demo01;

import android.app.Application;
import android.content.res.Configuration;
import android.util.Log;

public class ApplicationCustom extends Application {

@Override
public void onCreate() {
super.onCreate();

Log.e("LIFECYCLE = ", this.getClass().toString() + ".onCreate");
return;
}


@Override
public void onTerminate() {
// TODO Auto-generated method stub
super.onTerminate();

Log.e("LIFECYCLE = ", this.getClass().toString() + ".onTerminate");
return;
}

@Override
public void onConfigurationChanged(Configuration newConfig) {
// TODO Auto-generated method stub
super.onConfigurationChanged(newConfig);

Log.e("LIFECYCLE = ", this.getClass().toString() + ".onConfigurationChanged");
return;
}

@Override
public void onLowMemory() {
// TODO Auto-generated method stub
super.onLowMemory();
}
}
```

UI concepts

Lifecycle Application 03

- custom Application object (class derived from Application
  - must be activated in AcdroidManifest.xml by adding name attribute

```
<application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme"
        android:name="com.example.sample01basic01lifecycle.App
licationCustom"
        >
        ...
        </application>
```

UI concepts

Navigation 01 Events

- Event Listeners defined
  - programmatically with Event Listeners
  - declaratively in XML
    Button.onClick

UI concepts

Navigation 02 Events subscription programmatically

after getting widget objects (Button) - subsribe to Event Listener (handler)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

//-----------------------------------------------
// loading UI (layouts)
setContentView(R.layout.activity_main);
//-----------------------------------------------

//-----------------------------------------------
// getting Widget objects
buttonNavigationSimple = (Button) findViewById(R.id.buttonNavigateSimple);
buttonNavigationComplex = (Button) findViewById(R.id.buttonNavigateComplex);
editTextDataToSend = (EditText) findViewById(R.id.editTextDataToSend);
//-----------------------------------------------


// Touch/Click Event programmatically
// not using attribute in xml layout
buttonNavigationComplex.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
navigateComplex(v);
}
});

return;
}
```

UI concepts

Navigation 03 Events declaratively in XML

Button - android:onClick attribute

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <Button
        android:id="@+id/buttonNavigateSimple"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editTextDataToSend"
        android:layout_below="@+id/editTextDataToSend"
        android:layout_marginTop="16dp"
        android:text="Navigation Simple"
        android:onClick="navigateSimple"/>

</RelativeLayout>
```

UI concepts

Navigation 04 EventHandling

basic concept: Intent object + startActivity() method. Intents also used for passing data

```
public void navigateSimple(View v)
{
// data_class_level = "Class";
//-----------------------------------------------
Intent intent = new Intent(this, HelperActivity.class);
this.startActivity(intent);
//-----------------------------------------------
}

// object for passing data on class/object level (public, or with methods)
public static String data_class_level = "";

public void navigateComplex(View v)
{
String data =
//"data sent from main activity"
this.editTextDataToSend.getText().toString().concat(" - Intent")
;

//-----------------------------------------------
// Intent for passing data
Intent intent = new Intent(this, HelperActivity.class);
intent.putExtra("Data", data);
intent.putExtra("Broj", 1.0);
this.startActivity(intent);
//-----------------------------------------------

//-----------------------------------------------
// class object for passing data
data_class_level =
// ""
this.editTextDataToSend.getText().toString().concat(" - Class")
;
//-----------------------------------------------
}
```

UI concepts

Navigation 05 sending/passing data

Data can be passed

- through Android Intents
- through classes and objects (public and/or static objects in Activities)

UI concepts

Navigation 05 sending/passing data - Intents

```
//-----------------------------------------------
// Intent for passing data
Intent intent = new Intent(this, HelperActivity.class);
intent.putExtra("Data", data);
intent.putExtra("Broj", 1.0);
this.startActivity(intent);
//-----------------------------------------------
```

UI concepts

Navigation 06 sending/passing data - Classes

```
        //-----------------------------------------------
        // class object for passing data
        data_class_level =
        // ""
        this.editTextDataToSend.getText().toString(
).concat(" - Class")
        ;
        //-----------------------------------------------
```

UI concepts

Navigation 07 receiving/extracting data

Data from calling Activity can be received / extracted in onCreate ethod of the called Activity

- through Android Intents
- through classes and objects (public and/or static objects in Activities)

UI concepts

Navigation 08 receiving/extracting data

```
@Override
protected void onCreate(Bundle savedInstanceState) {

        //-------------------------------------------------
        super.onCreate(savedInstanceState);
        //-------------------------------------------------

        //-------------------------------------------------
        // loading UI (layouts)
        setContentView(R.layout.activity_helper);
        //-------------------------------------------------

        //-------------------------------------------------
        // getting Widget objects
        Button buttonBack = (Button) findViewById(R.id.buttonBack);
        TextView textViewDataFromClass = (TextView) findViewById(R.id.textViewDataFromClass);
        TextView textViewDataFromIntent = (TextView) findViewById(R.id.textViewDataFromIntent);
        //-------------------------------------------------

        //-------------------------------------------------
        // extracting data from intent
        String message1 = this.getIntent().getStringExtra("Data");
        //-------------------------------------------------

        //-------------------------------------------------
        // extracting data from class
        String message2 = MainActivity.data_class_level;
        //-------------------------------------------------

        //-------------------------------------------------
        // working with Widget objects
        if (message1 != null)
        {
        textViewDataFromClass.setText(message1);
        }
        textViewDataFromIntent.setText(message2);
        //-------------------------------------------------

        Log.e("LIFECYCLE = ", this.getClass().toString() + ".onCreate");
        return;
}
```

UI concepts

Navigation 09 receiving/extracting data - Intents

```
        //-----------------------------------------------
        // extracting data from intent
        String message1 = this.getIntent().getStringExtra("Data");
        //-----------------------------------------------
```

UI concepts

Navigation 10 receiving/extracting data - Classes

```
        //-----------------------------------------------
        // extracting data from class
        String message2 = MainActivity.data_class_level;
        //-----------------------------------------------
```

UI concepts

Links

- http://developer.android.com/guide/topics/ui/index.html
- http://www.androidpatterns.com/
- http://www.androidviews.net/
- Sample https://github.com/moljac/Samples.XamarinAndroid/tree/master/T