

A seagull is shown in flight, wings spread wide, against a dramatic sky with orange and blue clouds. The seagull is white with dark wingtips and a yellow beak. The sky is a mix of deep blue and vibrant orange, with scattered white clouds.

globalsyn
Taking People To The Next Level

globalsyn



globsyn 
finishing school

Basic Android UI Design



Basic Android UI Design

♣ Topics to be covered in this session

- Using Common UI Elements
- Android View Hierarchies
- Defining Screen Layouts: Using XML

UI In Android

Basic Android UI elements are used to create functional and intuitive user interfaces. The elements are classified as:

- ♣ View
- ♣ View Groups
- ♣ Layouts

Each element of an android user interface are arranged on screen by means of a variety of layout managers derived from View Group

Android UI Terminologies

- Views – They are the base class for all visual interface elements. All UI controls, including the layout classes are derived from Views
- View Groups – View Groups are extension of view class that contain multiple child
- Activities – It represents the window or screen being displayed. They are the android equivalent of forms.
- For creating graphical applications , the standard views need to be extended and modified

Introducing Views

- 1 All visual components in Android descends from the view class and are referred to as Views
- 2 View Group – Special kinds of views that contains other view
- 3 Layout – It is a special type of View Group

Approaches of Developing UI in Android

- 1 User Interface can be developed using coding in Java
- 2 User Interface can also be designed using XML
- 3 The preferred and conventional approach of designing UI is through XML
- 4 The IDEs facilitates this approach

XML Based Layout in Android

- XML Layout Files are stored under the resources (/res/) directory within a folder called layout

Developing Application in Android

- Every Android Application consists of one or more activities
- Each activity is developed using class:
`android.app.activity`
- UI is designed inside activity
- Users interact with activity

Important Methods of Activity

- 1 onCreate(Bundle b) - Where the activity is initialized, Create components, layout interface
- 2 setContentView(View v) - Sets the content of this activity to a specific view. Views can be components or layout objects that contain components or nested layouts . This method is always called by the OS when a new activity is started

Approach of Developing UI

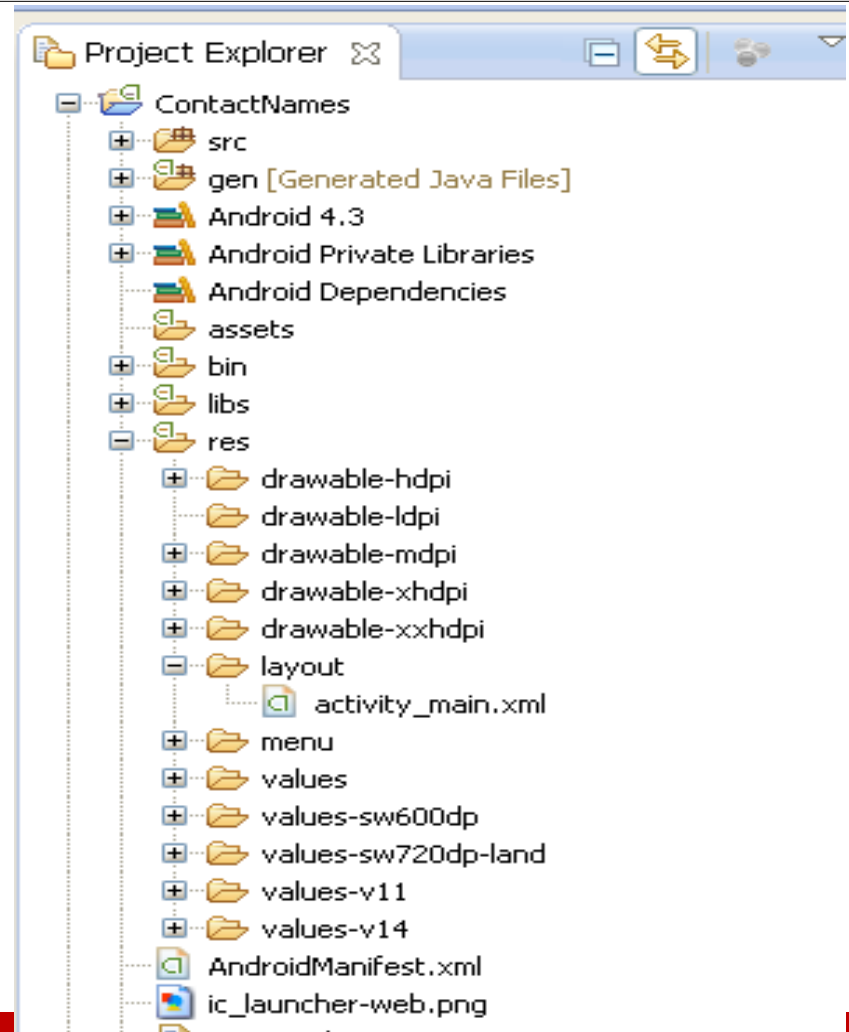
- 1 First Step: Decide a Layout
- 2 Second Step: Place UI components in Layout
- 3 Third Step : Add interactivity using Android event handling

Layout

- Android adopts a revolutionary approach to develop UI
- Layout are designed using XML
- XML layout files are stored under the resources (/res/) directory within a folder called Layout

Types of Layout

- Linear
- Horizontal
- Vertical
- Relative
- Grid
- Table



Code for Using Layout

```
protected void onCreate (Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}
```


Creating XML Layout in Eclipse

- 1 New activity creation process creates a layout for that activity
- 2 XML layout file can be created in the following way:
- 3 File - > New -> Other....->Android ->Android XML Layout File

Linear Layout

```
Orientation
Layout_width
Layout_height

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

android:layout_width="match_parent"

android:layout_height="match_parent"
    android:orientation="vertical" >

</LinearLayout>
```

Placing UI Components in Layout

- Every UI component has corresponding tag
- UI component tag must be used inside a View Group
- “Linear Layout” is example of View Group

Placing UI Components in Layout (Code)

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

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="It is a label"/>

</LinearLayout>
```



Taking People To The Next Level ...