In this chapter you will learn the Android UI mechanism
What is Context

- Interface to global information about an application environment
- Allows access to application-specific resources and classes
View Class

- The “View” class is the basic building block
- ViewGroup has the function “addView()”
View Groups

- Determines the layout in which the children will be placed
- Popular Layouts:
  - RelativeLayout
  - LinearLayout
Creating a layout
Place UI Elements (Widgets)

- Widgets are UI elements (Buttons, EditText, etc..)
- Can be done using XML or Graphic Layout
- It is mandatory to define the “android:layout_width” and “android:layout_height”
- Optional values:
  - “match_parent”
  - “wrap_content”
  - Size in “dp” → “100dp”
Connecting a layout with an Activity

In the `onCreate()` method add:

```java
setContentView(R.layout.new_layout);
```

```java
public class MyFirstActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```
Communication with widgets

Initialize the UI

public class TestUIActivity extends Activity {

    Button b_startActivityButton;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        b_startActivityButton = (Button) findViewById(R.id.b_startActivityButton);
        b_startActivityButton.SetText("first text");
    }
}


Communication with widgets

Set listener

public class TestUIActivity extends Activity implements OnClickListener {

    Button b_startActivityButton;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        b_startActivityButton = (Button) findViewById(R.id.b_startActivityButton);
        b_startActivityButton.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
    }

    }

}
Exceptions

- Meet the enemy

Tour – How to trace exception
What is Intent?

- Standard Android way to communicate between entities
- Intent is also object containing data
- Example: initiating a phone call:

```java
String uri = "tel:052-3331111";
Intent intent = new Intent(Intent.ACTION_CALL);
intent.setData(Uri.parse(uri));
startActivity(intent);
```

Tour – Phone call and Security
How to call activity

Intent intent = new Intent(getApplicationContext(), MyNewActivity.class);

startActivity(intent);
Ex2 – new activity

Now you do it!

Ex

- Create a new project
- Create 2 more activities
- Create a log when activities are created
- Create a log when starting new activity
Back Stack

- Application can have multiple activities
- App can also call other app activity (Send SMS)
- Android maintain the activities in a stack (Back Stack)
- Starting activity push the previous activity to the stack
- Pressing “Back” pop the last activity from the stack
- Don’t count on Android – the activities can be destroyed if there is not enough memory

⚠️ Android no guarantee policy
Back Stack

Finish will close the Activity without storing it in the back stack
Multiple tasks

Conclusion – Don’t count on the operating system
Activity Lifecycle

Tour – Logs in methods
Support Multiple Screens

- The purpose: Supporting many screen sizes and densities in one .apk file
- “Screen size”: Actual size (Diagonal)
  - small, normal, large, and extra large
- “Screen density”: DPI (Dots per inch)
  - low, medium, high, and extra high
- “Resolution” – The physical pixels
Screen sizes and Densities

DPI = $\sqrt{x^2 + y^2}/d$ (x and y in pixels)

Example: (Samsung Galaxy S, DIP = $\sqrt{480*480 + 800*800} / 4 = 233$)
Units

- Do not use pixels ("200px")
- Use "dp" (Density DPI)
- For text – use "sp" (Scale Independent)
  - Take in account the user settings

```java
DisplayMetrics met = new DisplayMetrics();
getWindowManager().getDefaultDisplay().getMetrics(met);
float densityDIP = met.densityDpi;
```

Always consider the user default settings (Fonts/Colors/Size)
Icons

- For best practice create all size icons (xxhdpi, xhdpi, hdpi, mdpi)
- Using only one can lower the performance (Scaling)
  - xxhdpi – 114x114
  - xhdpi – 96x96
  - hdpi – 72x72
  - mdpi – 48x48

⚠️ Check your application on different emulator size
Density support

Bad support:

Good support:
Set the orientation by
android:orientation="horizontal/vertical"

Gravity – controls the content placement

LayoutGravity – controls the parent placement

WeightSum – control distribution

Margin – controls the spacing
Ex3 – Sheep order form

- Create a new project
- Main activity will be the order form
- After pressing the “Make order” it will pass to the “MakeOrder” activity
- Log when the order is being sent
- Advanced: Use also slider for input
Relative Layout
RelativeLayout

- android:layout_alignParentRight/Top/Bottom/Left="true"
- android:layout_below/above/toLeftOf/toRightOf
- android:layout_centerVertical/Horizontal

Tour – RelativeLayout
Add view elements to Activity

- View can be added directly by using “setContentView()”
- View can be added directly to layout
- The view content can be changed dynamically

Button myButton = new Button(context);
setContentView(myButton);

Demonstrate
UI Widgets

In this chapter you will learn to use the Android UI components and control them
Step 1 – Declare the element in the xml file (in res/layout):

```xml
<Button
    android:id="@+id/b_exitButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

Step 2 – Add specific element parameters (in res/layout):

```xml
android:text="hello"
```

Step 3 – Declare the element in the activity class (in our package name):

```java
public class MyNewActivity extends Activity {
    Button myNewButton;
}
```

It is critical to add the ‘layout_width’ and ‘layout_height’
Using UI element

- **Step 4** – Initialize the element in the ‘onCreate’ method

  ```java
  @Override
  public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
      myNewButton = (Button) findViewById(R.id.);
  }
  ```

- **Step 5** – Create listeners for the element (if needed):

  ```java
  b.setOnClickListener(this);
  ```

- **Step 6** – Get/Set element properties (if needed):

  ```java
  myNewButton.setText(“press me”);
  ```
EditText et_myEditText = (EditText) findViewById(R.id.et_myET);
et.getText().addTextChangedListener(new TextWatcher()...
et.getText().toString();
Tip from the Sheep

Use the “EditText” component as less as possible, generally users say it is very annoying to type data on mobile devices (especially on landscape...)

32
Button b_myButton = (Button) findViewById(R.id.b_myButton);
b.setOnClickListener(this);

@Override
public void onClick(View v) {
    int viewId = v.getId();
    if (viewId == R.id.b_myButton) {
        // My button was pressed
    }
}
Ex4 – Sheep order form + Validation

- Use the previous exercise
- The “Make order” button will be disabled if the “with food” is not selected
- You should find out how to make the enable/disable (Check the Android developers)
- Advanced – Add more controls to the UI
In this chapter you will learn to use the Android UI components and control them
Resource files

- Located under /res directory
- Let us use different configurations

```
res
  drawable-hdpi
    icon.png
  drawable-ldpi
  drawable-mdpi
layout
  main.xml
values
  strings.xml
```
- Auto generated when saving resource
- Let us use different configurations
Android Resources

- Auto generated when saving resource
- Let us use different configurations
- Android has it’s own R.java resources

From XML:

```xml
<ImageView
    android:id="@+id/imageView1"
    android:src="@android:drawable/ic_dialog_alert" />
```

From Code:

```java
getResources().getDrawable(android.R.drawable.ic_dialog_alert);
```

- [http://androiddrawables.com/](http://androiddrawables.com/)
Tip from the Sheep

Don’t even think of solving errors in the java code before you handled all the errors in the resources files!
Images

- Put your image resource in res/drawable
  - Png
  - Jpg
  - Gif

```xml
<ImageView
    android:id="@+id/imageView1"
    android:layout_width="180dp"
    android:layout_height="144dp"
    android:layout_gravity="center_horizontal"
    android:src="@drawable/sheep1"/>
```
We will need to create a different xml file for each configuration:

Use the same element ids in the both files
Ex5 – Sheep order form – landscape mode

- Use the previous exercise
- Add support to landscape mode
- Advanced – Add also sheep image and change image when it is clicked