

# EMBEDDED SYSTEMS PROGRAMMING 2014-15

Application Tip: Managing Screen Orientation

# ORIENTATIONS

- Portrait



- Landscape



- Reverse portrait



- Reverse landscape



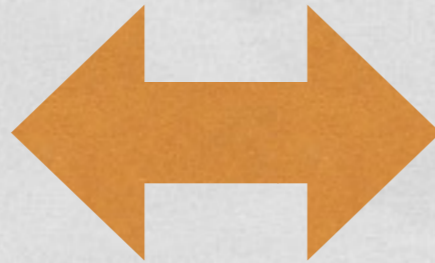
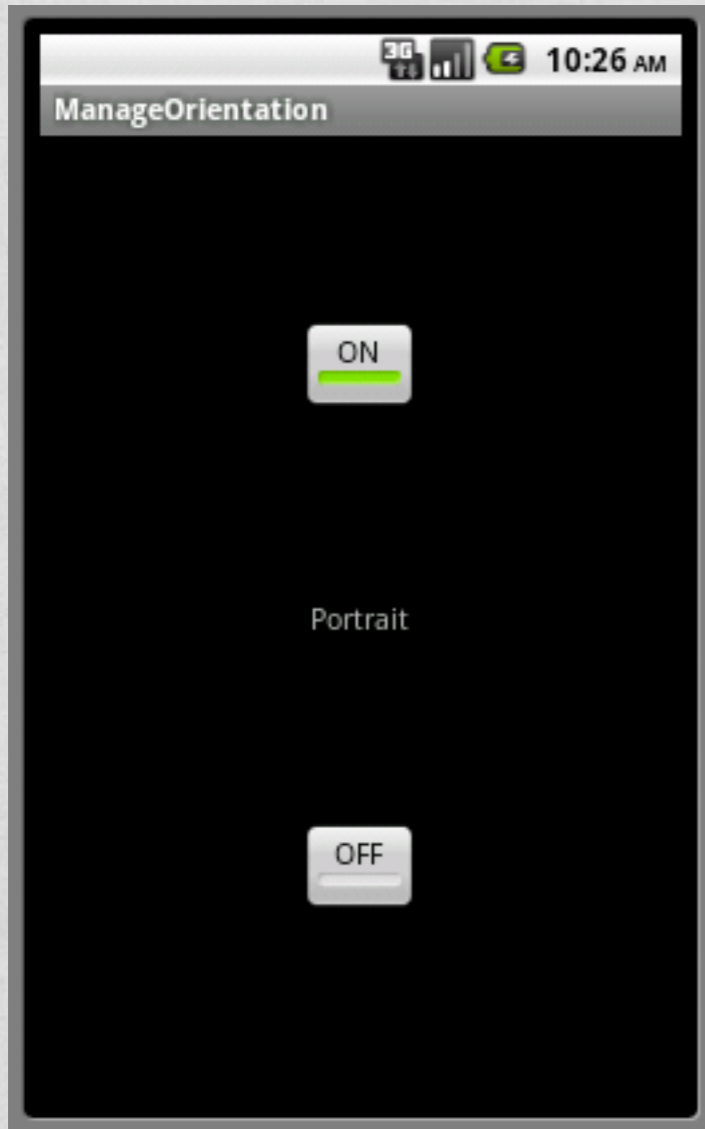
# ON REVERSE PORTRAIT

- **Android:** all four orientations are supported; the application screen is rotated by default
- **iOS:** all four orientations are supported; the application screen is rotated by default, with the exception of the “Reverse Portrait” orientation
- **Windows Phone:** only “Portrait”, “Landscape left” and “Landscape right”; the application screen is rotated only if the `SupportedOrientations` property is set to `PortraitOrLandscape`

# THE PROBLEM

- How to create and manage different UIs for portrait and landscape mode
- Widgets that are common to both UIs must preserve their state when the orientation changes
- UIs are specified declaratively
- Solution: use the facilities provided by the platform's frameworks

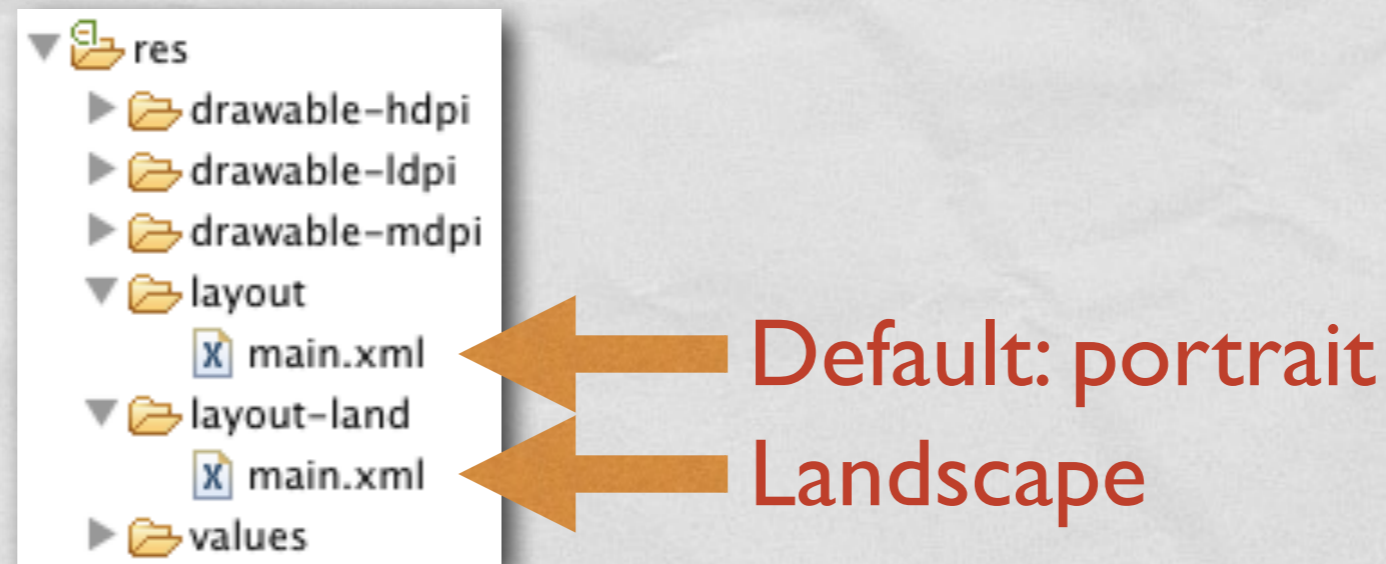
# THE TIP (1/3)



# THE TIP (2/3)

- In Android it is possible to declaratively define **multiple versions of the same UI** that match different orientations and screen sizes
- The different versions must be XML files with the same name but placed in different directories with appropriate **qualifiers** (they are just suffixes in the name of the directories)

# THE TIP (3/3)



- For a full list of qualifiers, look up <http://developer.android.com/guide/topics/resources/providing-resources.html>
- Resources (including layouts) are used automatically by Android as the need arises

# CODE (1/5)

- Source files:
  - `PersistenceActivity.java`
- Other resources:
  - `layout/main.xml` (UI layout, portrait),
  - `layout-land/main.xml` (UI layout, landscape),
  - `values/strings.xml` (UI strings)



# CODE (2/5)

- values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">ManageOrientation</string>
    <string name="hello">Portrait</string>
    <string name="hello_land">Landscape</string>
</resources>
```

# CODE (3/5)

- layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:weightSum="3.0">

    <ToggleButton android:text="ToggleButton"
        android:id="@+id/toggleButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </ToggleButton>

    <TextView android:text="@string/hello"
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1.5"
        android:gravity="center">
    </TextView>

    <ToggleButton android:text="ToggleButton"
        android:id="@+id/toggleButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </ToggleButton>

</LinearLayout>
```

# CODE (4/5)

- layout-land/main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:weightSum="3.0">

    <ToggleButton android:text="ToggleButton"
        android:id="@+id/toggleButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

    </ToggleButton>

    <TextView android:text="@string/hello_land"
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1.5"
        android:gravity="center">

    </TextView>

    <ToggleButton android:text="ToggleButton"
        android:id="@+id/toggleButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

    </ToggleButton>

</LinearLayout>
```

# CODE (5/5)

- MainActivity.java

```
package it.unipd.dei.es1011.manageorientation;

import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        // The ToggleButton's auto-save and auto-restore their instance
        // state using the savedInstanceState Bundle.
        // Since corresponding ToggleButton's have the same name in both
        // layouts, their state is correctly auto-managed: there is no need
        // of writing any custom code
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

LAST MODIFIED: MARCH 26, 2015

COPYRIGHT HOLDER: CARLO FANTOZZI ([FANTOZZI@DEI.UNIPD.IT](mailto:FANTOZZI@DEI.UNIPD.IT))  
LICENSE: [CREATIVE COMMONS ATTRIBUTION SHARE-ALIKE 3.0](https://creativecommons.org/licenses/by-sa/3.0/)