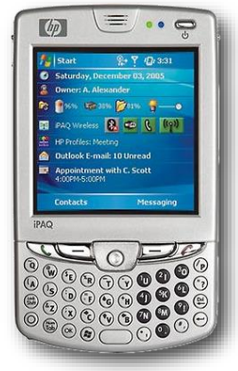
EMBEDDED SYSTEMS PROGRAMMING 2017-18

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)

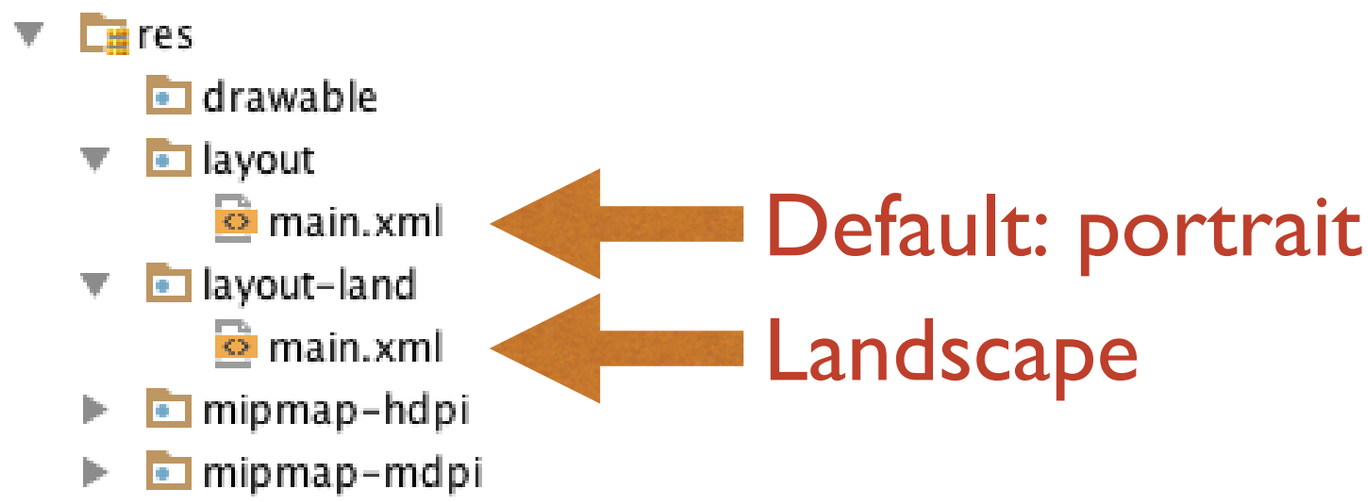

6:00
ManageOrientation
OFF
Portrait
OFF
6:00
ManageOrientation
OFF
Landscape
OFF

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:
  - **`MainActivity.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

```
<resources>
    <string name="app_name">ManageOrientation</string>
    <string name="hello">Portrait</string>
    <string name="hello_land">Landscape</string>
</resources>
```

CODE (3/7)

- `layout/main.xml` (1/2)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:gravity="center"
    android:weightSum="3.0"
    tools:context="it.unipd.dei.esp1516.manageorientation.MainActivity">

...
```

CODE (4/7)

- `layout/main.xml` (2/2)

```
...
    <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="0dp"
        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 (5/7)

- `layout-land/main.xml` (1/2)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:gravity="center"
    android:weightSum="3.0"
    tools:context="it.unipd.dei.esp1516.manageorientation.MainActivity">

...
```

CODE (6/7)

- `layout-land/main.xml` (2/2)

```
...
    <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="0dp"
        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 (7/7)

- MainActivity.java

```java
package it.unipd.dei.esp1516.manageorientation;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity
{
    /** Called when the activity is first created. */
    @Override
    protected 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
        // to write any custom code
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

LAST MODIFIED: MARCH 28, 2018