Device Rotation

Introduction

- Some android handsets switch from portrait mode into landscape mode when sliding out the keyboard. Others switch between the two based on the data received from the accelerometer.
- The android platform provides several ways for handling screen rotation in order to allow us properly handle each of the two possible orientations.

Introduction

- By default, the android platform destroys and re-creates the activity each time the screen rotate from portrait to landscape and from landscape to portrait.

Two XML Layout Documents

- We can keep two XML layout files in the same name. The first within the `layout-land` folder. The second within the `layout` folder.

Two XML Layout Documents

```
package com.abelski.samples;

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

public class RotationActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

When testing this code on the emulator pressing Ctrl+F12 switches between the horizontal and vertical orientations.

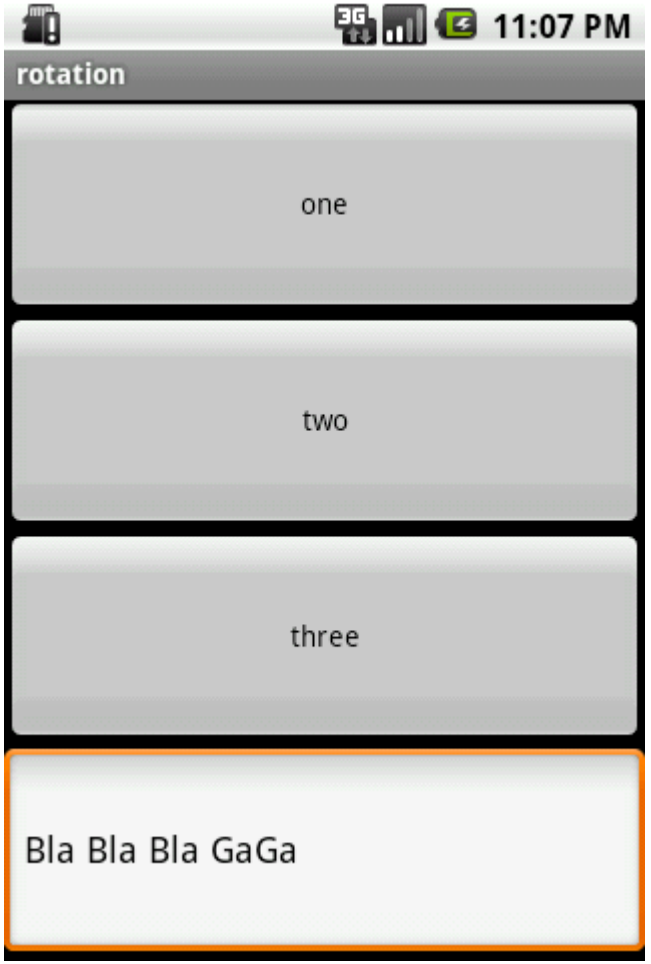
Two XML Layout Documents

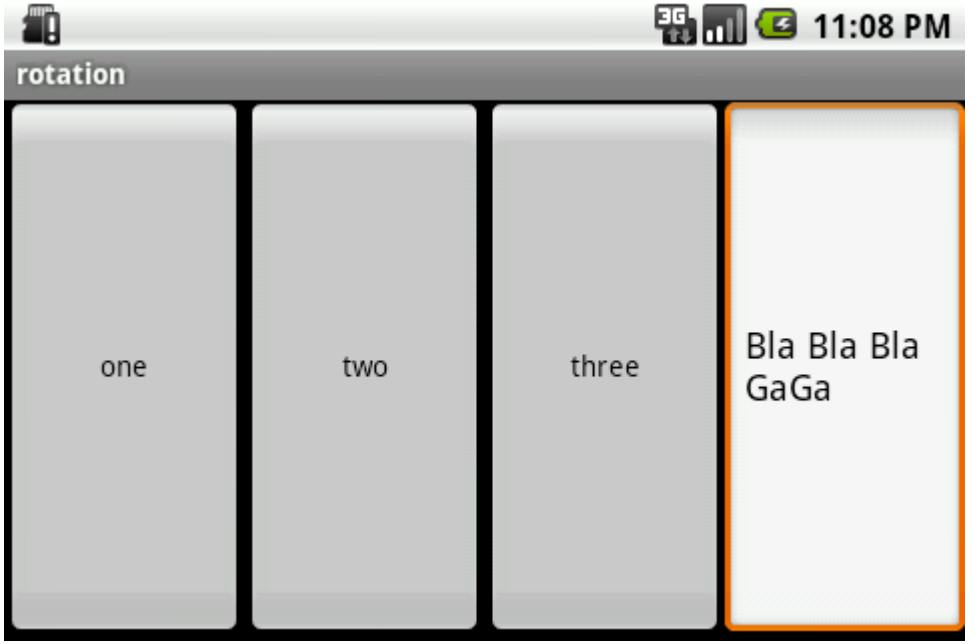
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical">

<Button android:layout_weight="1" android:text="one"
android:id="@+id/Button01" android:layout_width="fill_parent"
android:layout_height="fill_parent"></Button>

<Button android:layout_weight="1" android:text="two"
android:id="@+id/Button02" android:layout_width="fill_parent"
android:layout_height="fill_parent"></Button>

<EditText android:layout_weight="1" android:id="@+id/EditText01"
android:text="Bla Bla Bla" android:layout_height="fill_parent"
android:layout_width="fill_parent"></EditText>

</LinearLayout>
```

Two XML Layout Documents

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

<Button android:layout_weight="1" android:text="one"
android:id="@+id/Button01" android:layout_width="fill_parent"
android:layout_height="fill_parent"></Button>

<Button android:layout_weight="1" android:text="two"
android:id="@+id/Button02" android:layout_width="fill_parent"
android:layout_height="fill_parent"></Button>

<EditText android:layout_weight="1" android:id="@+id/EditText01"
android:text="Bla Bla Bla" android:layout_height="fill_parent"
android:layout_width="fill_parent"></EditText>

</LinearLayout>
```

Two XML Layout Documents





Entering text into the EditText it will be automatically saved when the activity is restarted. The text is part of the EditText state.

The `onSaveInstanceState` Function

- When the user moves from landscape to portrait (and vice versa) a new object is instantiated from our activity class.
- We can restore the activity state by overriding onSaveInstance function and saving it to the Bundle object it gets. Overriding onCreate we will be able to fetch that state from the Bundle object and restore our activity.

The `onSaveInstanceState` Function

```
package com.abelski.android;


public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.i("onsaveinstance","within onCreate "+this.hashCode());
        setContentView(R.layout.activity_main);
        //restore state from the bundle
    }

    @Override
    public void onSaveInstanceState(Bundle bundle)
    {
        super.onSaveInstanceState(bundle);
        Log.i("onsaveinstance","within onSaveInstanceState");
        //save state to bundle
    }
}
```