

Sunny / Cloudy Android Application

Building this Android application will demonstrate several concepts covered in today's lesson. You'll use IF statements to determine the outcome of the application, as well as use the GUI builder tool to create a functional user interface.

To get started, create a new Android project with the following criteria:

Project name: **Sunny_Cloudy**

Create new project in workspace should be selected.

Build Target: **Android 2.3.3**

Application Name: **Sunny_Cloudy**

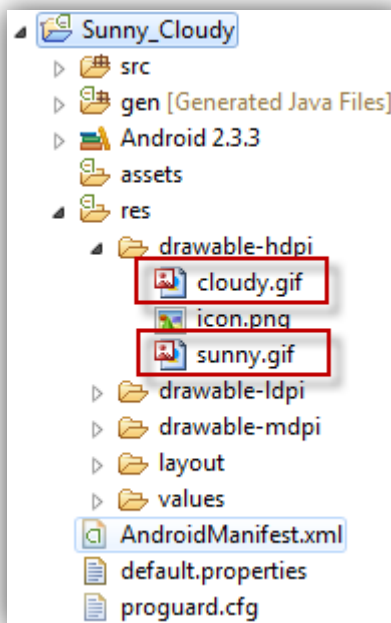
Package name: **lastname.firstname.Sunny_Cloudy** (substitute lastname.firstname with your last name and first name – no spaces, apostrophes or dashes.)

Create Activity should be checked, and **SunnyCloudyActivity** should be the text for it.

Min SDK Version: **10**

Click **Finish**.

Extract the contents of sunny_cloudy_resources.zip file, and place the two images (sunny.gif and cloudy.gif) in your **res/drawable-hdpi** folder. We'll use these two images in our code, and in the GUI itself.



Open the main.xml file, located in the **res/layout** folder of your project for editing.

Highlight **“Hello World, SunnyCloudyActivity!”**, then **right-click**, and select **Delete**.

Click on the empty display to select the **LinearLayout**. Verify that the orientation property for the **LinearLayout** is set to **“vertical”**. In order to do this, you’ll need to use the Properties panel in Eclipse.

Drag out a **RadioGroup** onto the **LinearLayout**. Inside the **RadioGroup**, you’ll find 3 **RadioButtons** already created for you. If they are not created (earlier versions of the Android SDK do not provide this functionality), drag 2 **RadioButtons** into the **RadioGroup**.

Right-click on the **third RadioButton** (if they were created for you automatically), and **Delete** it.

Select the **first RadioButton**, and modify its properties:

ID: **@+id/rdSunny**

Text: **Sunny**

Checked: **false**

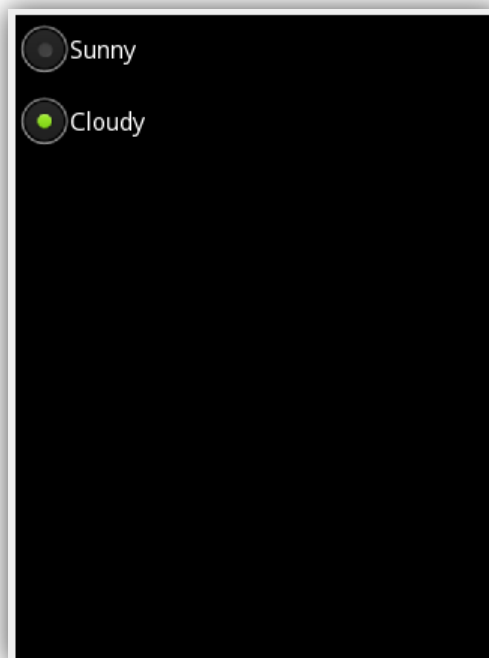
Select the second **RadioButton** and modify its properties:

ID: **@+id/rdCloudy**

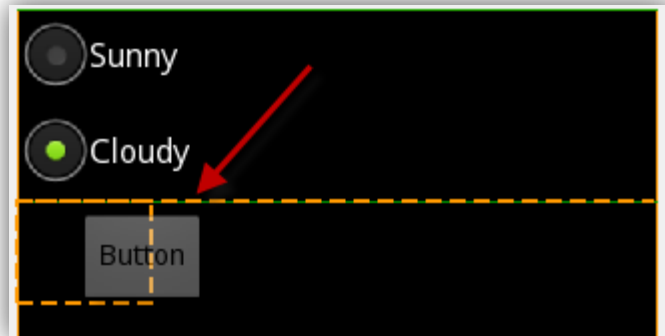
Text: **Cloudy**

Checked: **true**

Your screen should look like this:



Drag out a **Button** from the **Form Widgets** section of the Palette toolbar, and drop it **below** the **RadioGroup**.



Modify the button's properties:

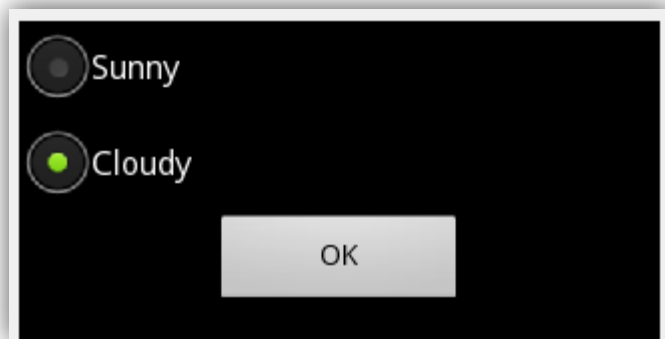
ID: **@+id/btnOK**

Text: **OK**

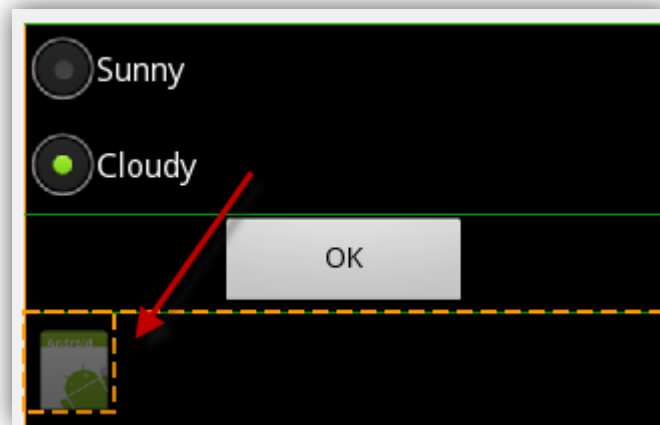
Width: **125dp**

Layout_Gravity: **center_horizontal**

Your screen should look like this:



Locate the **ImageView** widget, located under the **Images & Media** folder of the Palette toolbar. Drag and drop the ImageView below the button.



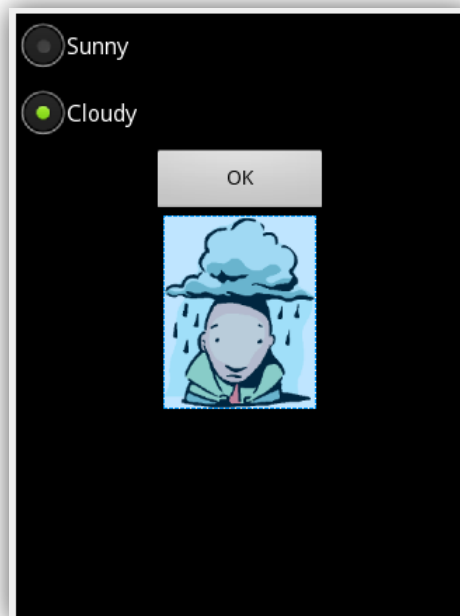
Modify the ImageView's properties:

ID: `@+id/imgMain`

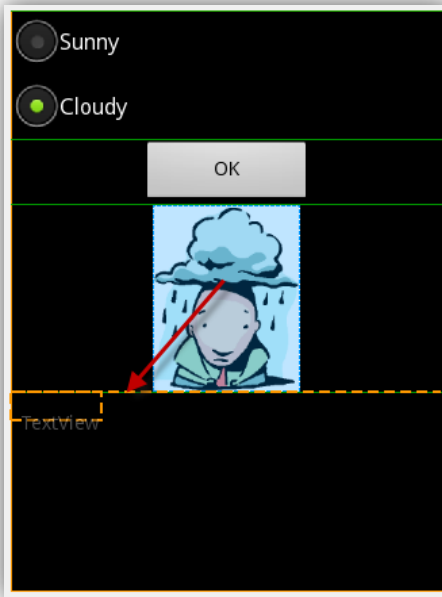
Layout_Gravity: `center_horizontal`

Src: `@drawable/cloudy`

Your screen should look like this when you're done:



Now, drag out a **TextView** widget from the **Form Widgets** section, and drop it under the ImageView.



Set the TextView's properties as follows:

ID: **@+id/txtOut**

TextStyle: **bold**

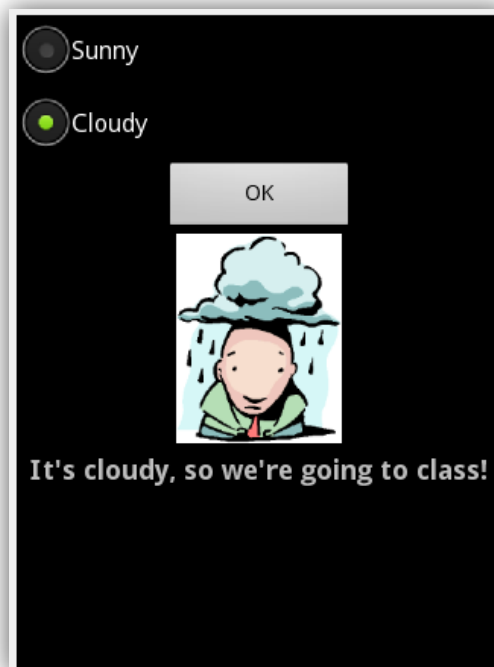
TextSize: **18sp**

Text: **It's cloudy, so we're going to class!**

Layout Gravity: **center_horizontal**

Layout Margin Top: **7dp**

Save the main.xml file, and **close** it. The final layout should look like this:



Here is also the resulting XML for the GUI above:

```
<?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">

    <RadioGroup android:id="@+id/radioGroup1"
        android:layout_width="wrap_content" android:layout_height="wrap_content">
        <RadioButton android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:checked="false"
            android:text="Sunny" android:id="@+id/rdSunny"></RadioButton>
        <RadioButton android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:id="@+id/rdCloudy"
            android:text="Cloudy" android:checked="true"></RadioButton>
    </RadioGroup>

    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:id="@+id/btnOK"
        android:text="OK" android:width="125dp" android:layout_gravity="center_horizontal"></Button>

    <ImageView android:layout_height="wrap_content" android:id="@+id/imgMain"
        android:layout_gravity="center_horizontal" android:src="@drawable/cloudy"
        android:layout_width="wrap_content"></ImageView>

    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:id="@+id/txtOut"
        android:text="It's cloudy, so we're going to class!"
        android:textStyle="bold" android:textSize="18sp"
        android:layout_gravity="center_horizontal" android:layout_marginTop="7dp"></TextView>

</LinearLayout>
```

Open the SunnyCloudyActivity.java file, and modify the contents on the onCreate event with the highlighted code below:

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    final Button btnOK = (Button) findViewById(R.id.btnOK);

    btnOK.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            final RadioButton rdCloudy = (RadioButton) findViewById(R.id.rdCloudy);
            final ImageView imgMain = (ImageView) findViewById(R.id.imgMain);
            final TextView txtOut = (TextView) findViewById(R.id.txtOut);

            if (rdCloudy.isChecked()) {
                // Cloudy
                imgMain.setImageResource(R.drawable.cloudy);
                txtOut.setText("It's cloudy, so we're going to class!");
            }
            else { //Sunny
                imgMain.setImageResource(R.drawable.sunny);
                txtOut.setText("It's sunny, so we're going to the beach!");
            }
        }
    });
}
```

Review the code to make sure you understand what is happening and why. Review the video along with this lesson.

Save the SunnyCloudyActivity.java file, and run your application. You should be able to switch between Sunny and Cloudy radio buttons, and receive the correct output on the screen.