



ISEL
INSTITUTO SUPERIOR DE
ENGENHARIA DE LISBOA

Desenvolvimento de Aplicações Móveis Mobile Application Development DAM

Tutorial 2 - Calculator

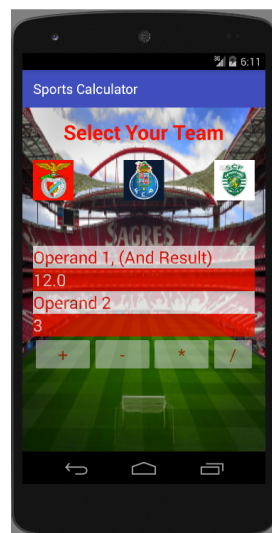
António Teófilo
ADEETC - C.2.4
antonio.teofilo@isел.pt

Pedro Fazenda
CEDET - E.2.16.4
pedro.fazenda@isел.pt

Abstract

This tutorial aims to develop a simple calculator with an interface adjustable to any screen with content related to a theme.

Deadline: 5 April 2020



Thematic Calculator

9 March 2020

Contents

1	Introduction	1
2	Laboratory Work - Thematic Calculator	1
2.1	Create project and set background	1
2.2	ImageButtons	1
2.3	TextViews and Buttons	2
2.4	OnTouch Event - ImageButton	3
2.5	OnClick Event - Button	4
2.6	Alternative Images	5
2.7	Multiple Languages	5
2.8	Different Layouts	6
2.9	More operations	6
3	Laboratory Work - Changes in Hello World from Tutorial 1	6
3.1	Hello World from Tutorial 1	6

List of Figures

1	Application main Layout.	2
2	Layout configuration for FC Porto.	3
3	Layout configuration for Sporting CP.	3
4	Image for button, with text referring the density group.	5

1 Introduction

This work aims to introduce the application development environment for Android devices. The tutorial introduces the anatomy of an Android program. From the application “Hello World” a new application with a more elaborate user interface is built. The main idea is to make a simple calculator with an interface adjustable to any screen with content related to a theme. The suggested theme is soccer but the students are free to choose another theme.

Follows a link that should be consulted during the development of Android applications.

Android Developers: <https://developer.android.com/guide>

2 Laboratory Work - Thematic Calculator

2.1 Create project and set background

1. **Create the project.** Create a new project like in tutorial 1, with: **new project** with **Empty Activity**, with Name **Thematic Calculator**, in package **dam_A6000.thematiccalculator** (replace with your student number).
2. **Get the image.** Search the Web for an image of the “Estádio da Luz” with of at least 500x700 pixels. In an image editing program, change the image resolution to approximately 320x240 and convert the image to **png** format. Copy the image to the drawable folder of the project. Image names should only contain lowercase a-z, 0-9, or underscore or else they will not appear in Pick a Resource window. Images should be placed in the drawables directory (not any other like drawable-v24).
3. **Set the image as the background.** Open the “activity_main.xml” in res / layout folder. Open the “Split” tab in the lower left corner of the editor to switch to the Layout Editor. Select the <constraintLayout> element and in the properties panel (lower right corner) put the name of “**container**” in the attribute “id” of the element. In the “background” attribute, place the image you copied in the previous point, by pressing the right most vertical rectangular/rounded shape to open the Pick a Resource window and select the previous image. Run the program on the Pixel 3 AVD created in the previous tutorial.

2.2 ImageButtons

1. Add a TextView with the text “Select your team” and adjust its attributes to look as in the Figure 1.
2. Look on the Web for an image with the S. L. Benfica emblem, and repeat for Sporting C. P. and F. C. Porto. Images must have a resolution between 350x350 and 500x500 pixels. In an image editor, change the image resolution to 60x60 pixels and convert the image format to PNG. Copy the images to the **drawable** folder of the project.

3. In the Layout Editor (file “content_main.xml”) place three **ImageButton**s below a **TextView** with “Select Your Team” (see Figure 1).

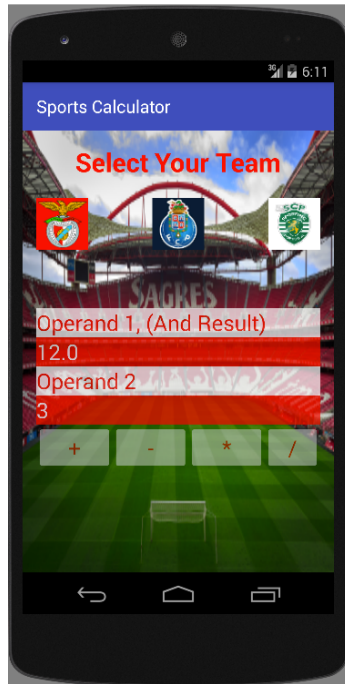


Figure 1: Application main Layout.

4. In each **ImageButton** place an image of a club in the “srcCompat” attribute, give a unique value to the “id” attribute of each, as well as a “contentDescriptor”. Set the **scaleType** to adjust the image. Do not forget to send strings to “strings.xml”. Run the program on the Pixel 3 AVD or on the mobile device.

2.3 TextViews and Buttons

1. Construct the rest of the layout of Figure 1. The remaining elements are represented by the `<TextView>` tag. For elements that receive input from the user you must activate the “editable” property (they become `EditText` elements). Do not forget to assign a different “id” for each one. Later, we will manipulate the values of these elements in Java and for that reason we will need the “id”. Use the “background” and “textColor” attributes to set the text and element colors. These colors should be defined in the “colors.xml” file.
2. For some elements it is necessary to put text. Make new strings in the file “strings.xml” with the content and use them in the respective attribute. Also define the “Input-Type” attribute as the decimal, of the element that receives the final result or the first operand.
3. Run the program and check if the user interface is close to the one in Figure 1. Small changes in the layout of the elements are accepted.

2.4 OnTouch Event - ImageButton

1. Search the Web for an image of the “Estádio do Dragão” stadium and the stadium of Alvalade XXI. Repeat points 2 and 3, of section “Background Image”, for these images.
2. **ImageButton** elements with club badges serve to make minor changes to the layout according to the club. When the user clicks or touches the Porto emblem the interface should be as in Figure 2 and when the user clicks the Sporting emblem the interface is as in Figure 3. To activate the **OnTouch** event in Benfica **ImageButton** use the code in Listing 1. Should be placed inside the **OnCreate** (...) method of the “MainActivity” class.

```
ImageButton slbIcon = (ImageButton) findViewById(R.id.slbIcon);

slbIcon.setOnClickListener(new View.OnClickListener() {

    @Override
    public boolean onTouch(View arg0, MotionEvent arg1) {
        View container = (View) findViewById(R.id.container);
        container.setBackgroundResource(R.drawable.estadiodaluz);
        return false;
    }
});
```

Listing 1: setOnClickListener method.

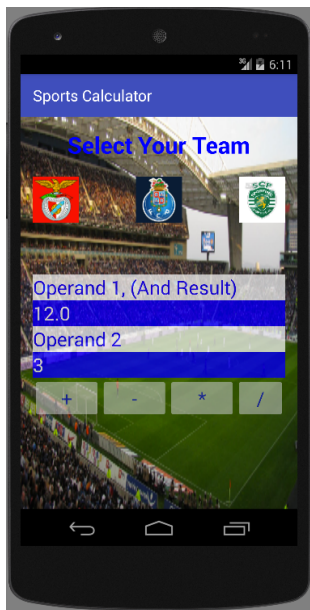


Figure 2: Layout configuration for FC Porto.

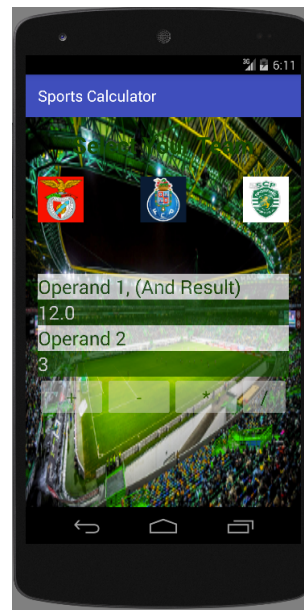


Figure 3: Layout configuration for Sporting CP.

3. Repeat the above for the ImageButton elements of Sporting and Porto.
4. Run the program and make sure that when you click on the emblem the background image changes.
5. To change colors when an **ImageButton** is clicked, first create the colors for each context (club) in the “colors.xml” file. Then, as we will have several things to do

inside the **OnTouch** (...) method, created in point 15, create the private method “**setContext** (...)” in the “MainActivity” class, to be called in the **OnTouch** method. Copy and complete the code of the “**setContext** (...)” method shown in Listing 2.

```
private void setContext (int id) {
    TextView title;
    int fc = 0, bc = 0;

    View container = (View) findViewById(R.id.container);

    if (id == R.id.fcpIcon) {
        container.setBackgroundResource(R.drawable.estadiododragao);
        fc = R.color.fcpf;
        bc = R.color.fcpb;
    }
    else if (id == R.id.slbIcon) {
        ...
        title = (TextView) findViewById(R.id.title);
        title.setTextColor(ContextCompat.getColor(getBaseContext(), fc));

        title = (TextView) findViewById(R.id.operator1);
        title.setTextColor(ContextCompat.getColor(getBaseContext(), fc));
        title.setBackgroundColor(ContextCompat.getColor(getBaseContext(), bc));
        ...
    }
}
```

Listing 2: setContext method.

2.5 OnClick Event - Button

1. To enable the **OnClick** event for each **button** on the calculator, use the following lines of code in Listing 3.

```
Button b = (Button) this.findViewById(R.id.bsoma);
b.setOnClickListener(this);
```

Listing 3: set onClickListener code.

2. When a **button** is clicked the following method must be executed:

```
public void onClick(View v) {
    String sNum1 = number1EditText.getText().toString();
    String sNum2 = number2EditText.getText().toString();
    double num1 = getDouble(sNum1);
    double num2 = getDouble(sNum2);
    Button b = (Button)v;
    double value = 0;

    if (b.getId() == R.id.bsoma) {
        ...
    }
    else if (b.getId() == R.id.bsub) {
        ...
    }
    ...
    number1EditText.setText(Double.toString(value));
}
```

Listing 4: onClick method.

3. Create a new Java class, “Calculator” within the same package with the methods needed for the calculator operations presented in the application.
4. Create an instance of the “Calculator” class in the “MainActivity” class and on the “**OnClick** (...)” method use the methods of the “Calculator” class to perform the calculator operations.

2.6 Alternative Images

1. **Alternative background images.** Use background images (stadiums of Benfica, Sporting and Porto) with at least 1920x1080 pixels. Generate the alternative images for all groups of densities so that in the other groups the image be represented by approximately 640x360 dp. Use the Android Drawable Importer (ADI). Run the program on the AVD Pixel 3, AVD Pixel C and on your device.
2. **Alternative club emblem images.** Do the same for the images of the clubs that are used on the buttons. The buttons should occupy 0.5 inches of display space in all generalized groups.
3. **Alternative app image icon.** Create or search in the Web for an image with a minimum dimensions of 200x200 pixels to replace the application icon. Generate alternative images (use ADI or New Image Asset, see section 3.1) so that in each density group the icon is represented by 48x48 dp. Change the manifest file to use that resource.

In one set of generated images (background or emblem) put text in each image to inform from which group it belongs using the dpis (see figure 4) or the class (l, m, h, x, xx, xxx) (see slides)). Thus, it will be possible to see which group is being used.



Figure 4: Image for button, with text referring the density group.

2.7 Multiple Languages

1. Create a “values” directory within the “res” directory for the Spanish language. Within this directory create a file “strings.xml” with the strings in Spanish. Run the program on the AVD Pixel 3 and your device. Change the language on the devices and verify the changes in the application.
2. Repeat the previous point for the Portuguese language.

2.8 Different Layouts

1. In the visual editor (file “activity_main.xml”) the “Orientation for Preview” icon, in the upper left corner, has a dropdown menu that allows you to create alternate layouts for landscape, tablets or other. Create a layout variation in landscape mode for “activity_main.xml”. Organize the interface differently to take full advantage of the width space. Run the program on the AVD Pixel 3 and Pixel C and in your device and on them change the presentation mode to “landscape”.
2. Create another layout version for devices where the minimum size is 720dp. Run the program on the AVD Pixel C.
3. Create another layout version for devices with a minimum height of 600dp. Create the Nexus 7 AVD based on the Nexus 7 smartphone (2012). Run the program on the Nexus 7 AVD.

2.9 More operations

1. **Two more mathematical operations.** Add two more mathematical operations (two more buttons) to the calculator of your choice (e.g., the operation to calculate the exponential or the square root). Do not forget to change the alternate layout as well.
2. **Memory operations.** Add one operation (“Set Mem”) (one button) to save the value of the first operand to memory and another one (“Get Mem”) (another button) to set the value from memory as the value of the first operand.

3 Laboratory Work - Changes in Hello World from Tutorial 1

3.1 Hello World from Tutorial 1

1. **Change the app icon launcher.** Over the `res` directory select with the mouse New → Image Asset, in that window leave the Icon Type as Launcher Icons (Adaptive and Legacy), the Name as `ic_launcher2` and change the background colour to blue. You can also change the foreground with a different Clip Art draw if you want. Finish and set the manifest file to use this new icon. Run the application and check that the icon is now blue. You can check by pressing the main Android square button (Overview/“recent apps” button) and in the installed apps list.
2. **Change the layout to only use constraints.** Change to app layout so to use only the two TextViews, the ImageView and the AnalogClock view. The Views should be positioned using only constraints (no margins or other views). The ImageView should be centred at about 38% of the screen and the AnalogClock at 77% (change these values, if needed), using two Horizontal Guidelines in percentage mode (click on its outside arrow to switch mode).

3. **Add the landscape layout.** Add the landscape layout, in a way that the ImageView and the AnalogClock are vertically centred on the space below “My First app” TextView and horizontally centred in their half part of the screen, using a vertical Guideline in percentage mode.