ISEL / ADEETC

Master in Communication and Multimedia Network Engineering

Interactive Multimedia Applications

Tutorial 2

Interactive Multimedia Applications

Thematic Calculator

Part 1

Rui Jesus

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.

Below is a link that you should consult during the development of this work.

Android Developers
http://developer.android.com/training/index.html

Note: this tutorial should be done in class and the code of the resulting Android projects must be delivered through the Moodle platform until the 11th of March.

Laboratory Work

Background Image

1. Create a new project in **Android Studio** repeating the steps already taken in the first tutorial.

2. Search the Web for an image of the "Estádio da Luz" with a resolution of at least 500x700 pixels. In an image editing program convert the image to **png** format.

3. In **Android Studio**, copy the image to the **drawable** folder of the project.

4. Look in the project to open the file "*content_main.xml*". Select the "Design" 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.

5. Run the program on the Nexus 5 AVD created in the previous tutorial.

ImageButton

6. Look on the Web for an image with the S. L. Benfica emblem, another emblem of Sporting C. P. and another with the emblem of F. C. Porto. Images must have a resolution between 350x350 and 500x500 pixels.

7. 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 in Android Studio.

8. In the Layout Editor (file "content_main.xml") place three **ImageButtons** at the top (see Figure 1).

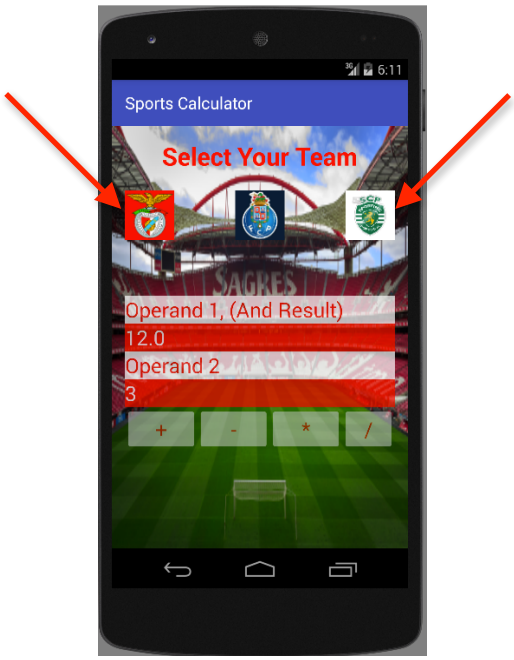


Figure 1. Main Layout of the application.

9. In each ImageButton place an image of a club in the "src" attribute and give a unique value to the "id" attribute of each. Do not forget to create a string in the file "strings.xml" with a short description for each emblem. Place its string in the "contentDescriptor" attribute of each ImageButton.

10. Run the program on the Nexus 5 AVD or on the mobile device.

Calculator - Graphical Interface

11. 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 EditView 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.

12. 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 "InputType" attribute as the decimal, of the element that receives the final result or the first operand.

13. 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.

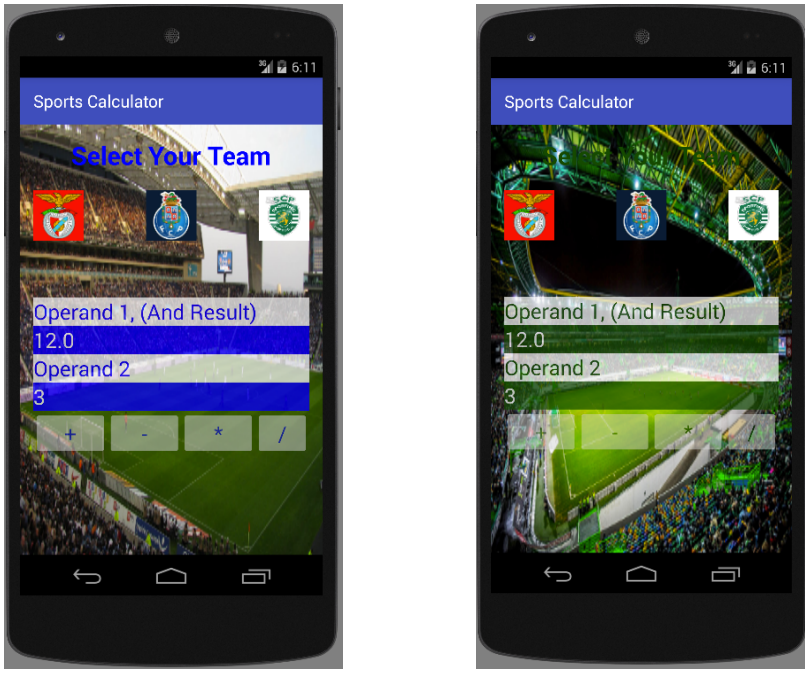
OnTouch Event - ImageButton

14. Search the Web for an image of the "Estádio do Dragão" stadium and the stadium of Alvalade XXI. Repeat points 2 and 3 for these images.

15. **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 2a and when the user clicks the Sporting emblem the interface is as in Figure 2b. To activate the **OnTouch** event in Benfica **ImageButton** use the code below. Should be placed inside the **OnCreate** (...) method of the "MainActivity" class.

```
ImageButton slbIcon = (ImageButton)findViewById(R.id.slbIcon);
slbIcon.setOnTouchListener(new View.OnTouchListener() {
            @Override
        public boolean onTouch(View arg0, MotionEvent arg1) {
            View container = (View)
findViewById(R.id.container);

container.setBackgroundResource(R.drawable.estadiodaluz);
            return false;
        }
});
```



a) b)

Figure 2. Layouts of the Application: at right the configuration of Porto and at left the configuration of Sporting.

16. Repeat the above for the ImageButton elements of Sporting and Porto.

17. Run the program and make sure that when you click on the emblem the background image changes.

18. 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 below.

```
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.setTextColors(ContextCompat.getColor(getBaseContext(),fc));

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

OnClick Event - Button

19. To enable the **OnClick** event for each **button** on the calculator, use the following lines of code:

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

20. 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)
    {
       …
    }
    …
    number1EditText1.setText(Double.toString(value));
}
```

21. Create a new Java class, "Calculator" within the same package with the methods needed for the calculator operations presented in the application.

22. 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.