Veikko Tapaninen (veikko.tapaninen@oamk.fi)

LAB 2

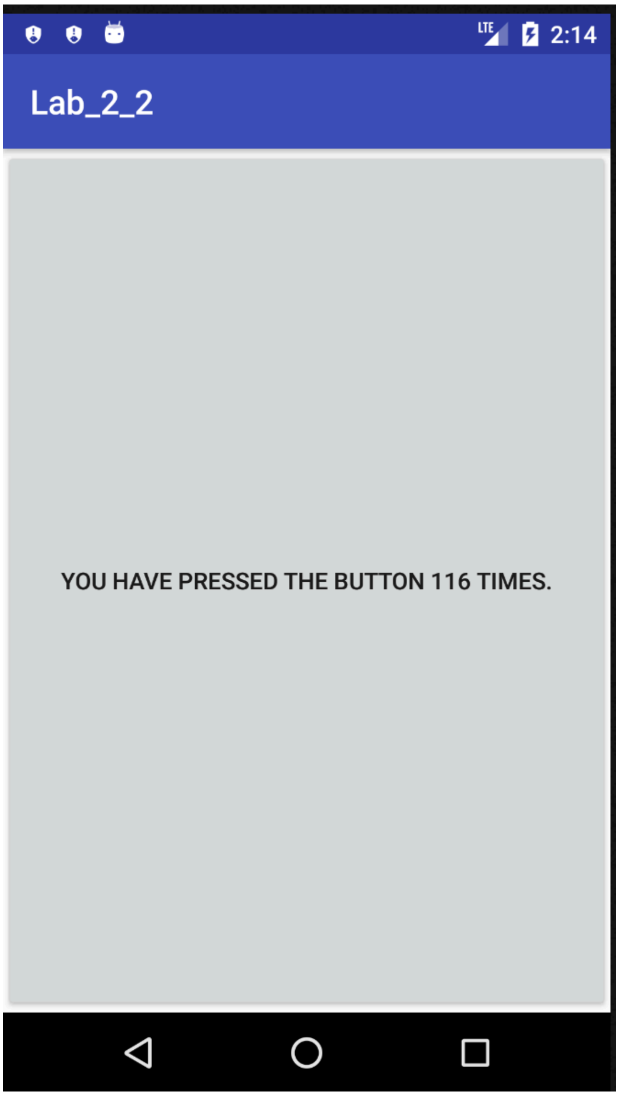
IMPORTANT. You must save all your work in a way that you are able to present them to teacher during the evaluation discussion. You could use for instance GIT.
You also need to have working developer environment where you can run the assignments.

1. UI Programmatically

It's also possible to create UIs without xml files. Let's practice this next.

Create new Android app project. But untick ***Generate Layout File*** option on the Configure Activity Dialog. Now we have a project with one Activity but it doesn't have ui xml file.

Let's make a really simple make. On the UI we have only one **Button**. When user presses this button he / she will get one point.



YOU HAVE PRESSED THE BUTTON 116 TIMES.

You can create UI Views in the code. For example like this:

```
Button gameButton = new Button(this);
gameButton.setText("Hello, I'm the button");
```

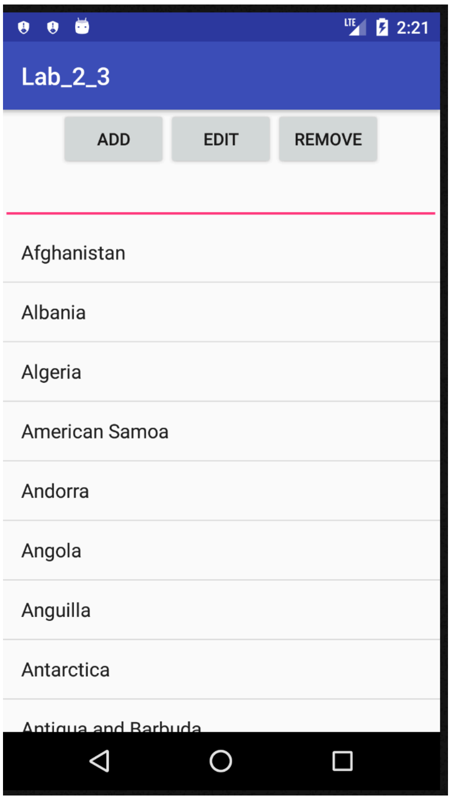
And you can then set this object to be the main view of the Activity in the **OnCreate** method.

```
setContentView(gameButton);
```

Now you just need to implement ways to count presses and update button text when it's pressed.

2. More complex UI programmatically

Create a new project in Android Studio and do the user interface and functionality what we did on the assignment 1 (UI Hierarchies) without xml. Program everything.



You need to create similar structure of the layouts and views as you have on the xml.

LinearLayout (vertical)

- **LinearLayout (Horizontal)**
    - **Add Button**
    - **Edit Button**
    - **Remove Button**
- **EditText**
- **ListView**

Some code to help you a bit

```
LinearLayout btnLayout = new LinearLayout(this);

Button addBtn = new Button(this);
addBtn.setText("Add");
btnLayout.addView(addBtn);
```

3. Implement more functionality

Modify previous assignment so that you implement functionality to **Add** and **Remove** buttons.

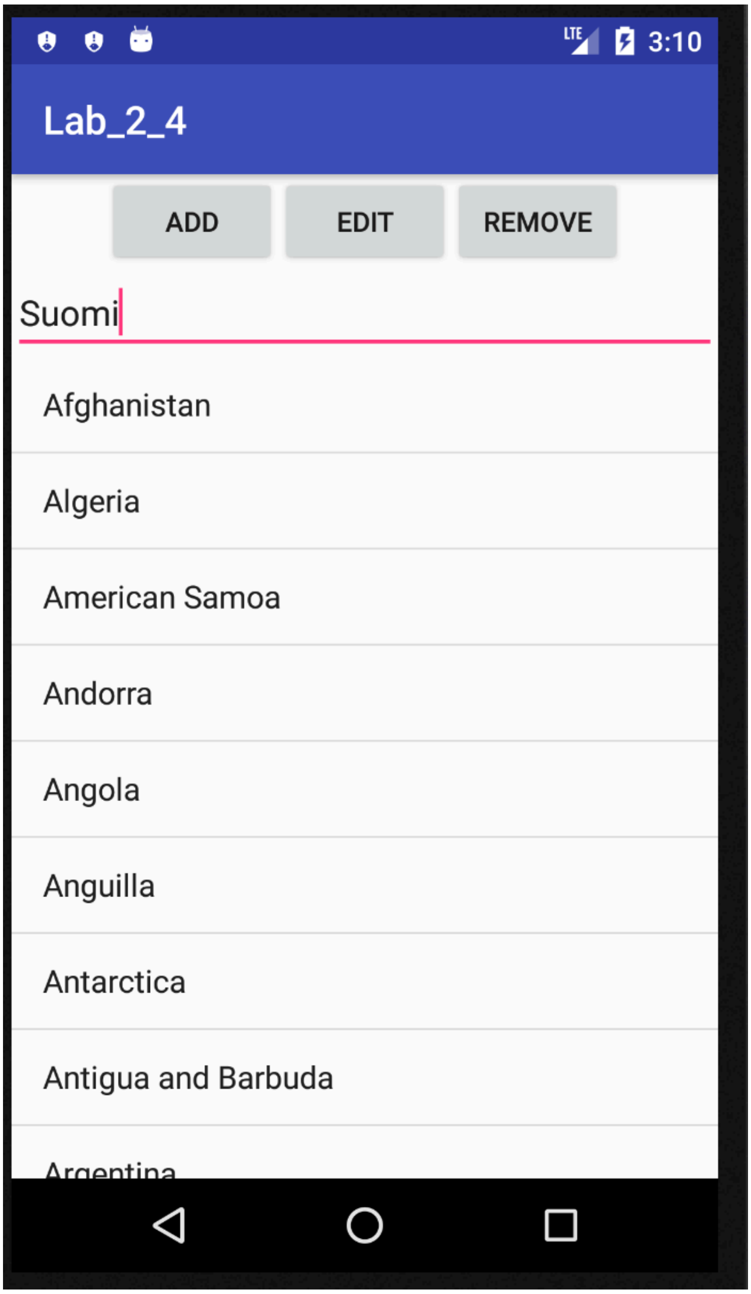
- With add button you can add written country to the list
- With Remove button you can remove written countries from the list

There are several ways to solve this problem, here are few tips from my solution.

You need to use more advanced data structure for the countries than array. I used **ArrayList**. With ArrayList you can easily remove and add objects to it. This country list should me added as member data to the activity.

You could also move the ListView and EditText views to be member datas so you can access them easily when you need to.

You can also create the array adapter object again every time you make changes to the country list. There are more elegant ways to update the list but we will study them later on.



Suomi

Afghanistan
Algeria
American Samoa
Andorra
Angola
Anguilla
Antarctica
Antigua and Barbuda
Argentina

4. Let’s practice a bit more event handling.

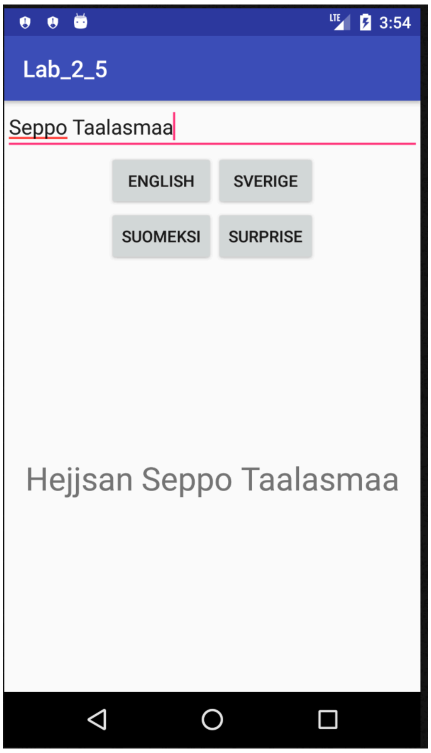
Create an app which greets person with a localised greeting.

Implement the UI like this:

When user enters a name to the EditText (text editor) and clicks the button, software
greets the user using a language displayed on the button

- English: “Hi <name>”
- Sverige “Hejjsan <name>”
- Suomeksi “Terve <name>”
- Suprise “Hola <name>”

Bonus task (not mandatory). Can you make the textfield to update when ever user writes text to the EditText.



Seppo Taalasmaa

Hejjsan Seppo Taalasmaa