

Assignment 01

First Steps

Prepare the Android development environment and create your first test app to check all involved components.

Download the Android development IDE of your choice

Android currently offers two convenient ways to develop Apps for Android. Choose an IDE and set it up correctly. ADT might be more suitable if you are already familiar with eclipse.

- 1) Android Development Tools (ADT) for Eclipse
<http://developer.android.com/sdk/index.html>
- 2) Android Studio
<http://developer.android.com/sdk/installing/studio.html>

Both environments come pre-bundled with the latest Android SDK and Emulator.

Create your first app and run it on your device or in the emulator

Create a first sample App using the default values and get familiar with the structure of the IDE. Identify the IDE modules that let you inspect logs generated on the device (LogCat).

For running the App in the emulator, first create a new emulator configuration. Afterwards start the emulator and run the App in it.

If you have your own Android device, try running the App on it. You will probably have to install a driver on Windows machines; Linux and Mac should display your device as a startup target directly. You might also have to activate the developer mode on your device.

Note: Have a look at the Android tools reference for more information (<http://developer.android.com/tools/workflow/index.html>).

Pirate Slang Conversion App

As your first assignment, you will build an App that translates a given text into pirate slang and display the result to the user. It makes use of standard design elements.

Modify the main layout

The default activity layout should consist of two text elements (one for inserting the text to translate, the other for displaying the translation) and two buttons (one for clearing both text fields, one for submitting a new translation request). Together, all elements should take up the available content area.

Afterwards, wire all elements up to the activity in order to change the text programmatically and react to button events.

Create a translation class

Create a new class for handling the translation.

This class should take the input from the first text element and translate it into pirate speech. This means that every “r” in the input text should be replaced by “rrr”.

When the user presses the translate button the text from the input should be translated and shown in the second text element.

Handle possible errors

If the input is empty when the translate button is pressed the user should be notified that a text must be inputted first. You can use a “Toast” for this (<http://developer.android.com/guide/topics/ui/notifiers/toasts.html>).

If the input doesn’t contain any “r” the user should also be notified that he needs to use more pirate-compatible words. 😊

Master Students: Add an Image of a pirate and customize the output TextView to look like a speech bubble

This applies to master students only!

You can use an ImageView to display the image (<http://developer.android.com/reference/android/widget/ImageView.html>).

For customizing the TextView you should use its `android:background` property in the layout file, linking to a separately created style.xml.

See <http://developer.android.com/guide/topics/resources/drawable-resource.html#Shape>

The final result should look approximately like this:



Submission

Please zip up your complete Android project and hand it in via Uniworx.

Deadline is 16.10.2014 at 14:00.

Projects that do not compile due to errors will not be accepted.