

Lab 7: Android services, Broadcastintents, custom UI components and the accelerometer



Design and Development of Mobile Applications
E. De Coninck, S. Leroux, P. Simoens
2015-2016

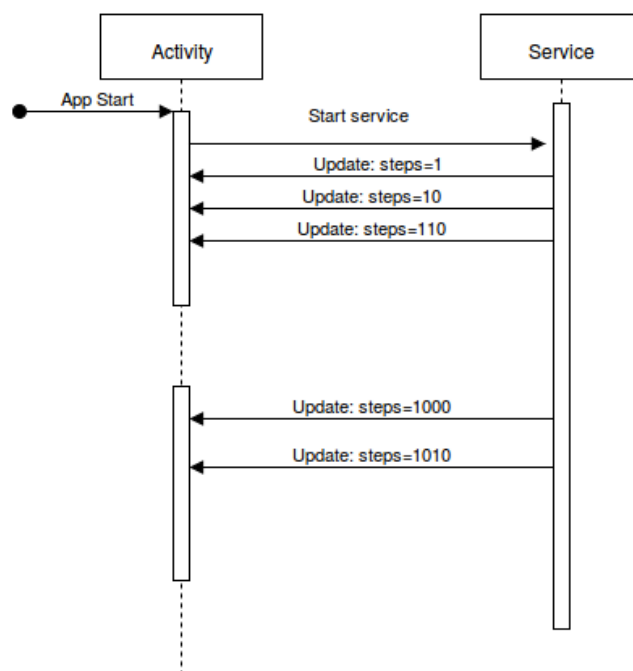
In this lab session we will create a simple step counting application. We will implement an Android service to keep recording steps even when the user is not actively using the smartphone.

1 Overview

This application consists of three parts:

- An Android activity (*StepCounterActivity*) to show the number of steps taken to the user.
- An Android service (*StepCounterService*) to keep processing accelerometer events even when the user is not actively using the smartphone. This service will analyse the data from the accelerometer and will keep the activity up-to-date with the number of steps detected so far.
- A custom UI component (*ProgressmeterView*) to show a graphical illustration of the number of steps taken and the number of daily recommended steps.

BroadcastIntents are used to send notifications from the service to the activity.



2 Activity

Start by implementing a new activity. For now this activity should only display a Textview with the number of steps detected so far.

3 Service

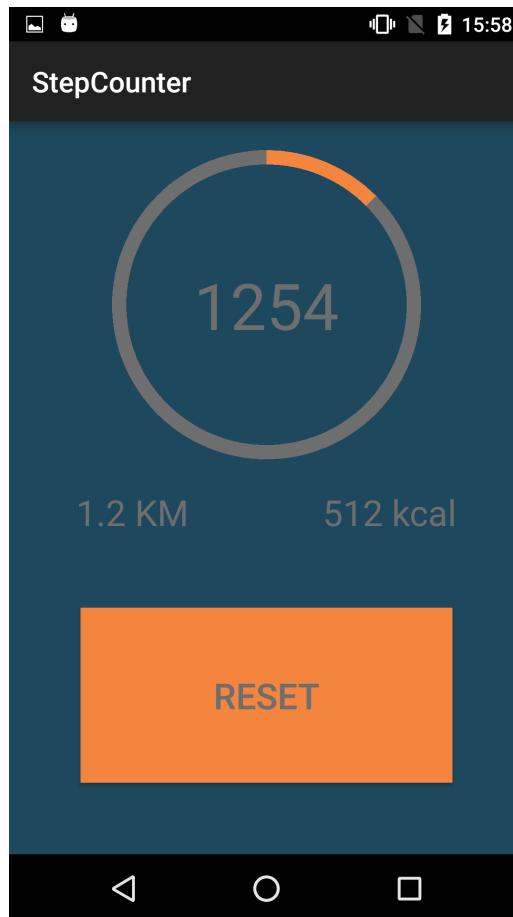
- Create a new Service class (*StepCounterService*).
- Override the lifecycle methods and display log messages when they are called.
- There are two types of services, which one should you use here ?
- This service should implement *SensorEventListener*
- Obtain a reference to the accelerometer and register this class as an observer. Use the linear acceleration sensor which measures the acceleration in x,y and z direction excluding the force of gravity.
- A python version of the basic algorithm to detect steps can be found at <http://nbviewer.ipython.org/url/users.ugent.be/~sleroux/asd3.ipynb>. Implement the corresponding Java code in your Service class. You may need to tune the parameters to obtain more accurate results.

4 Broadcast intents

- We will use broadcast intents to notify interested components when steps are recognized.
- To reduced the overhead, the messages will only be sent after a fixed number of steps are detected
- Register the activity as a *Broadcastreceiver* in the *onCreate()* method. Remove this activity as a receiver in the corresponding lifecycle method.
- The service should send broadcastintents every time at least ten new steps were detected.
- Update the TextField with the new number of steps.

5 Custom user interface

You received the following design from the UI-designer responsible for the look-and-feel of the application. Your task is to implement this UI in code.



- The color codes are : #1E485D, #F4853F and #6E6E6E. These should be configured in the colors.xml file.
- The circular progressmeter is a custom component (the class is provided on Minerva). To create a new custom UI component you should extend the *View* class and override the *onDraw()* method. In this method you can build the component by drawing basic components such as circle, lines and text to the *Canvas*. You can add your custom component to the layout xml file just like every other component.
- The new component should display the progress of the user in obtaining a daily goal of 10000 steps.
- Estimate the distance walked and the calories consumed based on the steps taken and show these values to the user.