

Optional Project 4: Activity Recognitions

Tian Guo, 2016/11/26

1 Project Overview

In this project, you will be developing an Android App that performs different tasks automatically based on user activities. The aim of this project is to get you familiar with Android sensors, in the context of Google activity recognition API, as well as SQLite database, Maps and Media Player.

- This project is due by 10:00 pm on **Dec. 8** and accounts for 10% of course scores.
- Each student must complete this project individually. However, I do encourage you to share ideas, advice and resources with each other. Please contact me if you have questions about what constitutes appropriate collaboration.
- If you have any questions regarding this project, please post it in the forum with the part you have questions with (using line numbers to the left).

2 Project Preparations

I have included the starter code “ActivityRecognition” along with some resources you might find useful in completing the project.

- A slide deck that compiles basic information related to project requirements.
- An online tutorial about activity recognition: <http://code.tutsplus.com/tutorials/how-to-recognize-user-activity-with-activity-recognition-cms-25851>
- An online tutorial about Android music player: <https://github.com/SueSmith/android-music-player>
- A mp3 file.

3 Project Requirements

3.1 Saving data into SQLite (40 points)

- (20 points) Currently, recognized user activities are simply printed out as console log. Your tasks are to design a SQLite database schema, and save the activity data into the local SQLite database. Write down your SQLite database schema.
- (20 points) Then you should implement a “Summary” button that displays a summary of all user activities in the past 24 hours using data from SQLite database. The summary should at least include the time span, i.e. start time and end time for the 24 hours window, the activity name, and the corresponding activity time. Take a screenshot of the activity summary.

3.2 Plotting user path in the Map (40 points)

- (20 points) Next, when your Android app detects a user switches to walking or running activities, your task is to first display a map of user’s current location. Take a screenshot of the displayed map.
- (20 points) Then you should add a thin red line starting user’s current location along user’s movement path. When the user switches to activities other than walking and running, you should save the coordinates of starting and finishing locations into the local SQLite database. Take a screenshot of the map with the thin red line.
- (10 points) Finally, you should implement a new screen activity that replaces the map. The new screen activity should at least display the basic information of a user’s current activity, e.g., starting time and activity name. Take a screenshot of the new screen activity.

40 3.3 Playing music (10 points)

- 41 • (10 points) Last, let us add some music to the app. Your task is to implement a “music” button that once
42 clicked, plays a mp3 music file locally. A sample mp3 file is provided, but you are welcome to use any mp3
43 files of your choices.

44 4 Project Submission

- 45 • Please make sure you have documented your implementation adequately with in-line comments.
- 46 • Create a zip file, named `cs4518Project4.zip`, from a directory that contains your modified Android
47 project code, the screenshot images, and your writeup. Submit `cs4518Project4.zip` through Instruct
48 Assist¹.

¹<https://ia.wpi.edu/cs4518/files.php>