Location Lab - Part 2

Use Location information within your app.

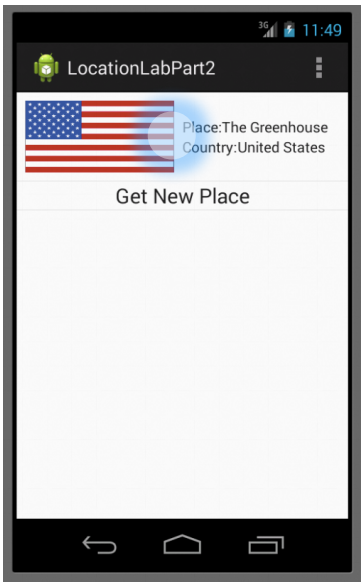
Objectives:

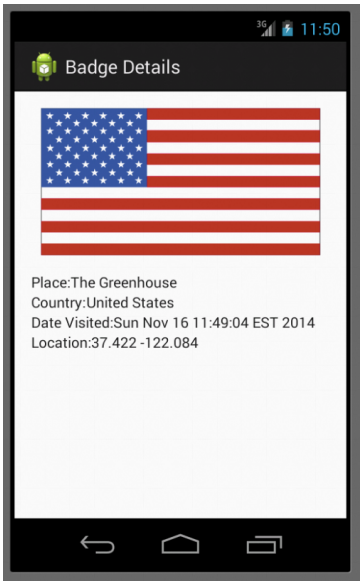
This week's lab will help you learn how to use location information in your Android applications. Upon completion of this lab, you should have a better understanding of how to listen for and respond to Location measurements.

This application displays a ListView containing a set of Place Badges. You implemented most of this application in a previous lab. In that Lab however the app's user manually entered the GPS coordinates for a given location.

This week you will make it so that the application receives locations from the device itself. To implement this functionality, you will need to acquire location readings from Android. Exactly how you implement this is up to you, however, your app will need to listen for location updates from the NETWORK_PROVIDER (which we will control for testing purposes) and the GPS_PROVIDER (so you can enter locations via the emulator). You can also inject locations into the app, by clicking on the menu and selecting one of the menu items, specifically "Place One", "Place Two," or "Place No Country."

In addition, when the user clicks on a Place Badge, the app will open a detail view showing additional information about the place.





Testing:

The test cases for this Lab are in the LocationLabTest project. You can run the test cases either all at once, by right clicking the project folder and then selecting Run As>Android Junit Test, or one at a time, by right clicking on an individual test case class (e.g., TestOneValidLocation.java) and then continuing as before. The test classes are Robotium test cases.

Warnings:

1. These test cases have been tested on a Galaxy Nexus AVD emulator with API level 18. To limit configuration problems, you should test you app against a similar AVD.
2. Our MockLocationProvider relies on a method that was first included in API level 17. Therefore, the TestCases will fail to compile on earlier platforms.
3. You will need to go into your device's Developer Options and make sure that you've enabled "Allow Mock Locations."
4. The application screenshots shown above require a working network connection. You will also need to create an account at http://www.geonames.org/login. Your username will need to be updated in PlaceDownloaderTask.java. If you do not have a working network connection, you can still use the app, but the flag images will use a simple placeholder.

Submission

To submit your work you will need to submit the LocationLabUMDPart2 project files we've asked you to modify. These files should be stored in specific directories as described below and then compressed in a zip file. Then you will submit this zip file to the Coursera system. The automatic grading system will test your submission and give you feedback. This process may take some time, especially if many students are submitting at the same time. To make sure your submission is correctly graded, pay attention to the following aspects:

1. Your project files must be compressed in a zip file named LocationLabSubmit.zip.
2. When decompressed, your submission should contain one top-level directory named LocationLabSubmit. Inside that directory there should be one directory named LocationLab containing the following file: PlaceViewActivity.java, PlaceDownloaderTask.java, PlaceBadgeDetailActivity.java.

The directory structure of the unzipped submission file should be as follows:

```
LocationLabSubmit /
        LocationLab /
                PlaceViewActivity.java
                PlaceDownloaderTask.java
                PlaceBadgeDetailActivity.java
```