

Project 1: Getting Started with Android Programming

Tian Guo, 2016/10/26

1 Project Overview

In this project, you will be writing your first Android app¹ by modifying an existing basketball score app. I will provide you with the starter code (see Section 2.3) and requirements (see Section 3) about new features that you should implement. The goal of this project is to get you started with using Android Studio, designing Android UIs, managing app resources, understanding app life cycles, and intents.

A few important things to know before starting this project.

- 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).
- Inside the same zip file, there should be two pdf files and one code zip file.
- This project is due by 10:00pm on Oct 31 and accounts for 10% of total points. Please do start early on this project. You should be able to follow the instructions to setup Android Studio and complete tasks until Section 3.4 after our first class.
- 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.
- Project instructions are provided based on Android Studio. You may, however, choose to use any other IDEs of your choices. But keep in mind that teaching staffs will less likely be able to provide helps if needed.

2 Project Preparations

Before you can start modifying the code using Android Studio, you need to prepare yourself by following the steps below.

2.1 Prepare Android Studio

Android Studio is the official IDE for developing Android apps. It is based on IntelliJ IDEA from jetbrains² That means, if you use any jetbrains IDEs, Android Studio should feel like familiar to you.

- *Install Android Studio on your computer.* I strongly encourage everyone to setup Android development environment on your personal machine. Please follow the guide on [office Android Site] about how to install Android Studio on different OSes. Alternatively, you can also find YouTube tutorials, for example, [thenew-boston], for installation instructions. However, Android EcoSystem is going through very rapid changes and those unofficial online resources are very likely outdated, and thus might not work straightforwardly. Please only use those online resources as references and at your own cautious.

Optionally, you could also use Android Studio 2.1 that is already installed on ZooLab machines. If you don't have card access to ZooLab yet, please let me know so I can request it for you.

- *Configure Android Studio.* There is really no much you need to configure. I often take the "lazy" approach and only configure when needed. But if you insist, go check out a number of things you can configure [here]. and install some recommended packages through SDK Tools guide [here].
- *Explore Android Studio.* If you want to get comfortable with Android Studio, you can read about [project structure] and its [user interface].

¹If you have any prior experiences with Android programming or mobile development, you should use it as an opportunity to warm up for the term.

²I switched from plugins based Vim environment to jetbrains a few years ago. And I love it. JetBrains provides professional IDEs for educational licenses. If you write a lot of python codes like I do, you should check out Pycharm Professional.

2.2 Prepare your Android Devices

You have two options when it comes to testing your Android apps, running on an Android Virtual Device (AVD) or on a physical Android device.

- To create an AVD, please follow the steps from Android Developer Site [\[here\]](#). I strongly suggest you create an AVD by first *duplicating* and then modifying the cloned AVD.³
- To use your own Android device, you will have to at least enable USB debugging from the Developer Options. For Android 4.2 and higher, please follow the instructions at [\[how to enable on-device developer options\]](#). For Android that runs older versions, you can go to **Settings > Developer options** and check the box for USB debugging.
- Create an Emulator, or enable USB develop on your phone.

Android Studio comes with Android Emulator that allows you to simulate Android Virtual Device (AVD) in your own computer. In a high level, you can equate an AVD to a physical Android device. This is very helpful because you can start developing and testing Android apps without a hardware device. Android Emulator is great as it supports a lot of physical device features, and allows you to test against a wide selection of virtual devices. However, Android Emulator can be slow sometimes. Be patient! Even with a very powerful Macbook Pro, I still experience significant slow down when testing apps using Android Emulator.⁴

2.3 Prepare Starter Code

- To quickly test if your Android development environment is setup properly, please follow the steps [\[here\]](#) by first creating the Hello World Project, and then running the app in the Android Emulator [\[here\]](#). You will not see any error messages if the setup is proper.
- Starter code is downloaded as a zip together with this project 1 description. Import the starter code into Android Studio and Make sure the starter code can compile and run on your Android device or AVD. I have tested the code in a Galaxy Nexus running Android 4.3 (API level 18) and an AVD running Android 5.0 (API level 21). If you have encountered any problems at this step, please try to resolve them by first checking your Android Studio Setup in Section 2.1, and then following the error messages on Android Studio. If you can successfully run a new project with empty activity, then your Android development is correctly setup. Most of the time, the errors come from missing packages that you just need to click to install.

3 Project Requirements

After you successfully run the starter code, you are then asked to make the following modifications to earn corresponding points. Evaluation of your work will be based on a combination of the implemented features, code quality, and compliance with submission guidelines. For each required modification, please take the resulting screenshots and follow the naming conventions: starting with corresponding section number, e.g. the screenshot from Section 3.1 is saved as `31newicon.png`. This helps teaching stuff to provide rapid feedback.

3.1 (10 points) Change Application Icon

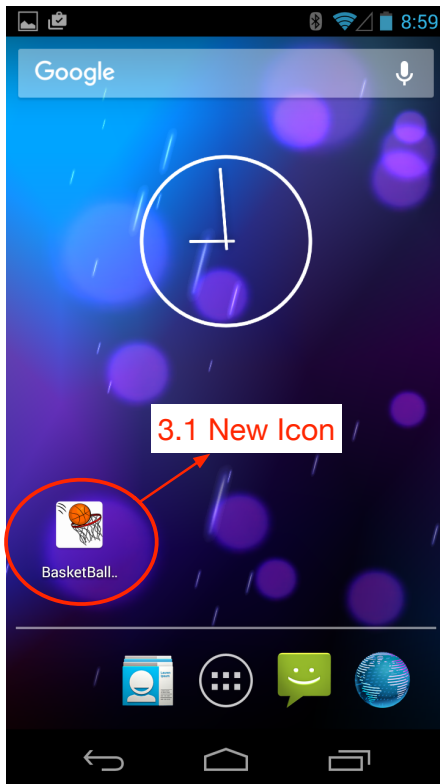
The starter code uses the default android icon. Choose any photos you want, and configure it to be the app icon. Take a screenshot of the app with the new icon, and save the screenshot using `31newicon.png`. An example screenshot is shown in Figure 1(a).

3.2 (10 points) Greetings with Toast Message

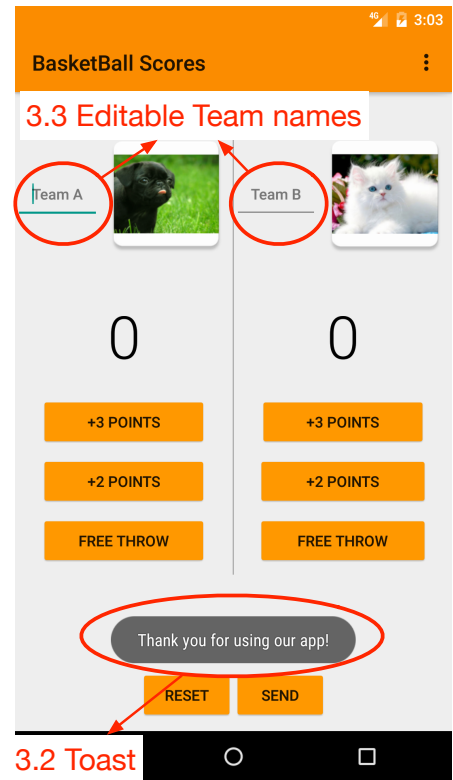
Modify the code so that a Toast message will be displayed after the application is first launched. Save the screenshot containing your customized Toast message as `32toast.png`. An example screenshot is shown in Figure 1(b).

³Just a precaution in case something funny happens.

⁴I usually test apps directly on my Android phone and thus have not looked into ways to fine tune the Emulator to improve performance. If you find a way to do so, please feel free to share with us by posting in the forum.



(a) The app with a new icon.



(b) Toast and editable team names.

Figure 1: Screenshots for Section 3.1, Section 3.2 and Section 3.3.

3.3 (10 points) Allow Manually Input Team Names

Modify the code so that the hard coded Team Names can be changed. That is, users can input any two team names when using the app. Save the screenshot containing two team names of your choice as `33names.png`. An example screenshot is shown in Figure 1(b).

3.4 (10 points) Team Profile Photos

Modify the app layout so that team profile photos are displayed next to the team names. For this step, you need to choose two pictures you want to use and save them as drawable resources. Save the corresponding screenshots as `34profiles.png`.⁵ An example screenshot is shown in Figure 2(a).

3.5 (10 points) Supporting Localization

And a second `string.xml` of a language of your choice. In your phone or emulator, change the language from default English to the second language of your choice. Relaunch your app, and take the screenshot of localization effects. Save the screenshot as `35local.png`.

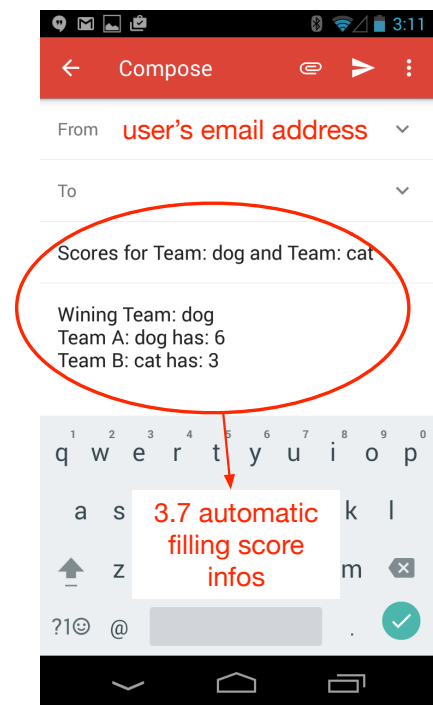
3.6 (25 points) Correct Scores

At this point of time, you might notice that when you rotate your Android device (or AVD), the scores will be reset to 0. Recall why such reset is happening, and modify the code so that, scores will remain unchanged after device rotations. The teaching staff will test this feature by first incrementing scores for both teams, and comparing the scores after rotating the device. Screenshot is not required.

⁵A quick way to convert pictures to predefined sizes is similar to what you did for Section 3.1.



(a) Team profile photos and localization support.



(b) Sending scores using an Mail client.

Figure 2: Screenshots for Section 3.4, Section 3.5, and Section 3.7.

3.7 (25 points) Sending Scores

Add a button `Send` at the bottom next to `Reset`, so that user can send the final results through an existing Mail client *automatically*. The message body should contain at least three lines that reads like the following. Save the screenshot as `37send.png`.

```
Wining Team: team dog
Team A: team dog has: 6
Team B: team cat has: 3
```

Specifically, the first line summarizes the winning team of the game and the following two lines reports their scores respectively. An example screenshot is shown in Figure 2(b).

3.8 (20 points) Bonus Features

Each extra feature is given a 10 bonus points. For each extra feature you implemented, please briefly describe what the feature is, how you implement it, as well as the screenshots if applicable. Example screenshot names: `38bonus1.png` etc.

4 Project Submission

- Before submitting, please take the time to make sure you have a clean build by following this short guide about clean your project build.
- Please include a README file If you complete the bonus features in Section 3.8.
- Then put all the screenshot photos into a folder called `screenshots`. Then put both the screenshots and the modified Android project into a new folder named `cs4518Project1`. Create a zip file from the new folder and submit `cs4518Project1.zip` through Instruct Assist.