Project 1: Getting Started with Android Programming

Tian Guo, 2016/11/01

1 Project Overview

The aim of this project is to get you familiar with Android apps that use the Android Camera, Images and Storage. 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 code zip files, a project description pdf, and a textbook chapter. CriminalIntent.zip is the starter code for the main project while, photo-demo.zip[1] is the starter code for face detection (extra credits). The textbook chapter can be used as a reference to understand the starter code from CriminalIntent.zip. (Please do not distribute the pdf outside class.)
- This project is due by 10:00pm on **Nov. 11** and accounts for 10% of course scores. Please do start early on this project!
- 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.

2 Project Preparations

- For this project, please use an Android (Virtual) Device that runs with API level in the range of $[21, 23]$. If you are completing extra credit part, you will also need a device with API level 25.

3 Project Requirements

3.1 Fix the gradle compile error (10 points)

After importing the starter code, you might encounter the build error when hitting the `run` button. Make changes so that you can start the project and reach the main activity as shown in Figure 1(a). Hint: check out the gradle configuration file `CriminalIntent/app/build.gradle`.

3.2 Display more images (30 points)

Clicking the `NEW CRIME` button on the top will launch a new activity as shown in Figure 1(b). Currently, this activity can only store one image in the top left corner. Taking a second image replaces the first image with the second. Modify the code, so that it can store more images below the ”SEND CRIME REPORT” button as shown in Figure 1(c). Taking images 2,3 and 4 should store those images in the positions below the ”SEND CRIME REPORT” button as shown. Refer to Figure 1(c).

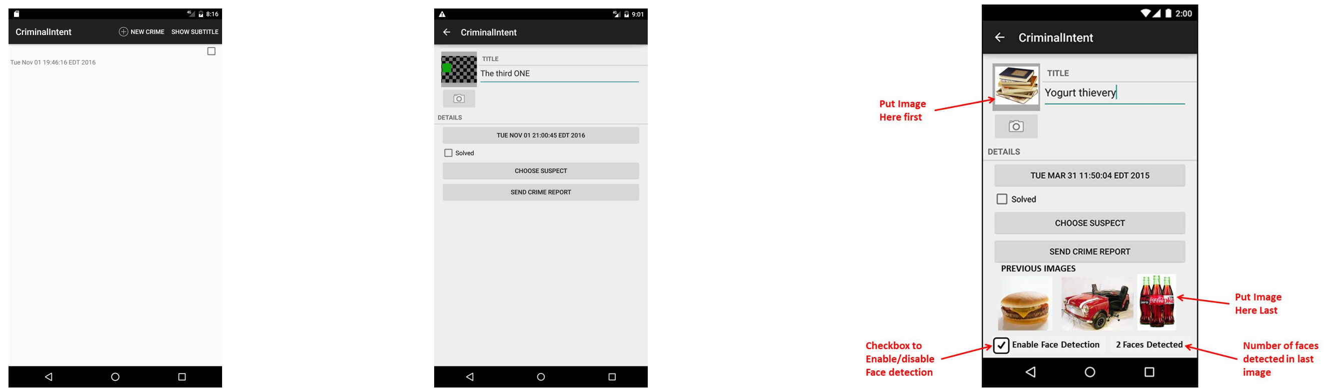
3.3 Recycle views (30 points)

Modify the code so that, if 4 images are already displayed, taking a 5th image should replace image 1 (in top left corner). Taking a 6th image replaces image 2 (leftmost image below the ”SEND CRIME REPORT” button) and so on.

3.4 Save captured images into gallery (30 points)

Currently, when pressing the `ImageButton` to take a picture, it is not saved to gallery. Modify the code so that images will show up in the Android’s Gallery.

[1] Face detection concepts are explained in https://developers.google.com/vision/face-detection-concepts and You can also take a look at other sample codes at https://github.com/googlesamples/android-vision



Put Image Here first

Put Image Here Last

Checkbox to Enable/disable Face detection

Number of faces detected in last image

(a) The app launching activity UI.

(b) The original app UI.

(c) The new app UI.

Figure 1:

3.5 Extra credit: Face detection (15 points)

Add face detection for your project by integrating the code sample photo-demo.zip. Add a checkbox in the bottom left corner. When checked, face detection is enabled and when unchecked, face detection is off. When Face detection is enabled, the rectangles for face detection are overlayed around each face in the preview of the picture. When the picture is taken, the number of faces found in the LAST PICTURE taken is reported in the bottom right corner of the screen as shown. So, for example, if the user tries to take a picture with 2 faces, two rectangles would appear around those faces during the preview, before the picture is taken. After the picture is taken, if face detection is enabled, the text “2 Faces detected” will be displayed in the bottom right corner of the screen. If face detection is not enabled, then the bottom right corner of the screen should be blank (nothing displayed).

3.6 Extra credit: Target at API level 25 (10 points)

As the latest API level is 25, you would also want to support those devices running Android N. Currently, the app will not run on Android N. Figure out the reasons and make changes accordingly so that the app work as expect (as in devices with API level 23).

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. Make sure you have documented your implementation adequately with in-line comments. Also, include a README file that briefly describes the high-level steps you take in implementing each requirement.

- Record session of running this app with all the implemented features on an Android device. If you are using a physical device, you can simply download a free screen capture app, such as `AZ Screen Recorder`. If you are using an AVD, you would have to first install Google Play on the emulator and then download the app. Also check out this link for alternative ways to perform screen capture.

- Build and generate an APK for your app. Check out this post if you don’t know how yet.

- Create a zip file, named `cs4518Project2.zip`, from a directory that contains your modified Android project code, the README file, your APK, and captured screen video file. Submit `cs4518Project1.zip` through Instruct Assist[2].

[2] `https://ia.wpi.edu/cs4518/files.php`