CSC 120: Flashlight

Classes and Methods

1 Introduction

In this lab you will create a flashlight application that activates when an object is in close proximity. In creating the application, you will learn how to construct proper classes and methods. This lab will reinforce what you have learned about classes and give you a basis for how to create intuitive and efficient classes and methods.

2 Objective

This lab reinforces proper class and method construction. You will take what you have learned about classes and methods and apply it to a real world application. The hands on approach will help reinforce the theory learned in class. This lab will give you the basic knowledge of Android development to enable you to complete future labs.

3 Activity

This section will provide directions that will help you finish the lab.

3.1 Research

Read through the Hello World tutorial at http://developer.android.com/resources/tutorials/hello-world.html and any other tutorial you might be interested in. The Android framework fundamentals is also good information on the basic components of an Android application - http://developer.android.com/guide/topics/fundamentals.html.

We will be working with the `Camera` class because the LED flashlight is part of the camera. Look at the `Camera` class, specifically `Camera.open()`, `Camera.getParameters()`, `Camera.setParameters()`, `Camera.startPreview()`, and `Camera.stopPreview()`, at the following website - http://developer.android.com/reference/android/hardware/Camera.html We will also be using the Camera.Parameters class using the `Camera.Parameters.setFlashMode` method. http://developer.android.com/reference/android/hardware/Camera.Parameters.html - #setFlashMode(java.lang.String).

3.2 Explore

Open the `FlashlightActivity.java` file and familiarize yourself with the code provided. Read all the comments and note the parts of the code you will be writing. The parts you will be writing will be clearly marked.

3.3 Flashlight Off Method

First, locate the `flashlightClicked()` method of the `FlashlightActivity` class. This method will run whenever you touch the screen. It gets called by the `onTouch()` method. The `onTouch()` method will be covered later. Back in the `flashlightClicked` method, notice the `IF` statement. If the flashlight is on, it runs the `turnOffFlashlight()` method. This method is empty and will be created by you. Likewise, if the flashlight is off, it runs `the turnFlashlightOn()` method. You will create this as well.

Inside of the `turnOffFlashlight()` method, first you will need to set the flash mode to off. Do this by calling the `setFlashMode()` method of `camParams`. You will pass it a string of value "off". Next, we will pass the `camParams` object to the `setParameters()` method of the `Camera` class variable. Finally, you will close the camera preview that was started in `turnOnFlashlight()`. To do this, call the `stopPreview()` method of the `Camera` class variable. Make sure to set `isFlashlightOn` boolean to false, so the application will know the flashlight is off.

3.4 Flashlight On Method

Inside of the `turnOnFlashlight()` method, first you will need to set the flash mode to off. Do this by calling the `setFlashMode()` method of `camParams`. You will pass it a string of value "torch". Next, we will pass the `camParams` object to the `setParameters()` method of the camera class variable. Finally, you will close the camera preview that was started in `turnOnFlashlight()`. To do this, call the `startPreview()` method of the `Camera` class variable. Make sure to set `isFlashlightOn` boolean to false, so out application will know the flashlight is off.

3.5 OnTouch Method

Locate the `onTouch()` method inside the `FlashlightActivity` class. You will edit this method to check if the `MotionEvent` parameter is a "touch down" action. To accomplish this task, call the `getAction()` method of the `MotionEvent (event)` parameter and check if this is equal to the `MotionEvent.ACTION_DOWN` integer. If these are equal, you will call the `flashlightClicked()` method and pass it the `View` (v) variable as the parameter.

4 Conclusion

You should have now successfully programmed your first application for this class. You have learned how to make proper methods and classes in the Android framework by making a flashlight application. Now you are able to take this knowledge and build on what you have learned to complete the future labs.

5 Deliverables

To submit your application, export your Eclipse project as a file system, zip all of the files into an archive and submit them online with the filename `<first_name_initial><last_name>-lab<lab#>.zip`. For example, if your name is John Doe and this is lab #1 (which it is), you would name your file `jdoe-lab1.zip`.