CSC 120: Alarm System

Using APIs

1 Introduction

In this lab, you will be creating an alarm system, which will detect sound above a certain amplitude. If sound is detected, an alarm will be activated. In addition to listening for sound, the alarm can also be sounded at any point for testing by the user. In order to accomplish this on Android, you must utilize its onboard microphone and analyze the amplitude of the incoming audio data. This will be done by using classes and methods from APIs.

2 Objective

The purpose of this lab is to introduce and learn the concept of APIs. API stands for Application Programming Interface and is defined as a set of routines, protocols, and tools for building software applications. For this particular lab, you will be using the Android SDK API.

3 Activity

This section will give you information on the actual work necessary to complete the lab. The Background Information section will provide suplementary information and the Implementation section will provide guidelines for completing the lab.

3.1 Background Information

For this lab, you will be using the `MediaPlayer` and `MediaRecorder` classes. These classes contain methods that utilize the Android's microphone and speakers to play and record media. Before you begin the activity, visit *http://developer.android.com/reference/android/media/MediaPlayer.html* and *http://developer.android.com/reference/android/media/MediaRecorder.html*. Familiarize yourself with both of these classes and their methods.

The alarm system app should be able to listen for sound and, if detected, sound an alarm. There will be three different choices for the alarm sound. The following picture shows the GUI (graphical user interface) of the app:

Both the speaker picture and the sound alarm buttons will be programmed to sound the alarm when clicked. The activate button will be programmed to begin listening for sound when clicked.

3.2 Implementation

This section will guide you through the implementation of the solution.

3.2.1 Research

Research the `MediaPlayer` and `MediaRecorder` classes and methods. List which methods you believe will be crucial in programming the alarm system.

3.2.2 Explore

Open the `AlarmSystemActivity.java` file. Explore the file. Read the comments. Note the parts of the code that you will be required to fill in. They will be labeled.

3.2.3 Import

In order to use the classes and methods from APIs, you have to import the required libraries. Import the `MediaPlayer` and `MediaRecorder` libraries.

3.2.4 Create

Now we will create the media class instances, set them to null, and then instantiate them in the `onCreate()` method. The `MediaPlayer` instance needs to be instantiated with the sound loaded to play. To do this, use the create method with `this` and `R.raw.caralarm1` as the two parameters.

3.2.5 Sound Alarm Programming

Now, let's program the buttons to sound the alarm. When the speaker button or sound alarm button is clicked, we want to play or stop the sound depending on if the alarm is already sounding.

Research how to start the `MediaPlayer` and determine if it is already playing.

3.2.6 Activate Programming

Next, we will program the activate button. This will be done in the listen method of the code. There are 4 functions in the `MediaRecorder` instance that must run before it can start recording. Research how `MediaRecorder` works and determine what these 4 functions are. One of the functions requires an output file. The parameter for this function is `audiofile.getAbsolutePath()`. The `audiofile` variable was declared earlier in the code.

3.2.7 Record

With the `MediaRecorder` (mr) ready to record, determine the functions needed to start recording and code it. Why does the prepare method have to be called before the recorder is started?

3.2.8 Analyze

Next, we will analyze the recorded audio to determine if the amplitude has reached a certain point. Which method in the `MediaRecorder` class do you think would be helpful in achieving this?

In order to accomplish this, we will set up a loop that continually checks the maximum amplitude reached, and then check it against a threshold we set up.

3.2.9 Summary List

The alarm system should now be working properly. Create a list of all the methods used from the `MediaPlayer` and `MediaRecorder` libraries with a description of what the method does and what it accomplished in the program.

4 Conclusion

In this lab, you have learned the concept of using APIs. This was done through extensively researching the `MediaPlayer` and `MediaRecorder` libraries, and then using their methods to implement an alarm system. In the future, you should be able to research other libraries in an API to accomplish other programming projects.

5 Deliverables

The deliverables for this project are as follows:

- The Eclipse project export with your version of `AlarmSystemActivity.java`
- Lab Report