Lappeenranta University of Technology
CT30A9300 Qt Code Camp

Smart Alarm

Report

Date: 31.10.2010

Group 3

Mihai Iusan 0377037
Email: mihai.iusan@gmail.com

Camilo Cifuentes 0376999
Email: cgcamilo@gmail.com

Bhupesh Mudari 0377118
Email: mudaribhupesh@gmail.com

Contents

1. Introduction

In this document we present the project that we created during the Qt codecamp 2010. We will start showing the idea which pushed us to create the Smart Alarm. We will continue by presenting the features that we implemented. After the features, we will present the problems that we faced during development which will be followed by our solution. The technical description will explain the application architecture using the class diagram. We will also describe the technologies that we used to achieve the application goals. In the final part we finalize with our conclusions.

You can find the project at: http://codecamp.fi/doku.php/qt2010/grp3/start

2. Idea

The idea of creating the Smart Alarm comes from the concept that many people use their mobile phones for waking up in the morning. Also people might want different ways to stop the alarm.

Motivation

The reasons that made us to create new methods for turning the alarm off were:

- The phone might be too far for the user
- When trying to stop the alarm we might hit it and fall down from the table
- Maybe we cannot touch it because we have dirty hands, we are driving or we cannot see it
- Or we just want to stop the alarm in a fancy way

Solution

To achieve this goal we identified three methods, stopping the alarm by:

- Making sounds
- Shaking the phone
- Flipping the phone

3. Features

For implementing the idea, we should have the following features:

- Setting up the alarm
- Stop the alarm by:
    - Pressing a button (normal way)
    - Making Sound
    - Shaking the phone
    - Flipping the phone
- Saving the settings
- Playing the alarm sound

4. What is the problem

To stop the alarm by sound is not simple, you need to consider the external environment, because it might be a loud environment. Also the sound that the alarm produces can interfere and stop it.

When we implemented the stop the alarm by shake option, we noticed that it is important to define the sensor sensitivity. Also when we tested it on the device we noticed that the sensitivity is not linear if we compare it with the value.

When stopping the alarm by flipping the phone we have to know the initial position of the device, so the alarm will not stop randomly.

When we start the application for the first time, we noticed that the configuration file might be missing. This problem occurs also with the sound file.

5. Solution

Considering the environment, we need to analyze the sound level before the alarm starts. To avoid the alarm sound interference, we should read the sound only when the alarm sound pauses.

Note: when the application is working in the simulator, the sound analysis will work also when the alarm is ringing. This problem occurs because the function "***isFinished()***" has no effect in the windows environment.

Because we do not know the accelerometer sensitivity, we implemented the sensitivity bar so the user can chose the level. For setting the sensitivity bar range, for the future, we should make more tests on the devices so we will identify the sensitivity curve.

To avoid random stop of the alarm by the phone position, we need to check the initial orientation in order to calculate the opposite position that will trigger the alarm to stop.

To avoid the missing configuration file problem, we check if the file exists and if there is no configuration file, we create a new one with default settings.

For non existing sound file, the solution is created by checking if the file exists, if it is not found, a file browser will appear and you will have the opportunity to select a sound file.

Note: Only *.wav* files can be played.

6. Technical description

For describing the architecture of the application we present the following class diagram:

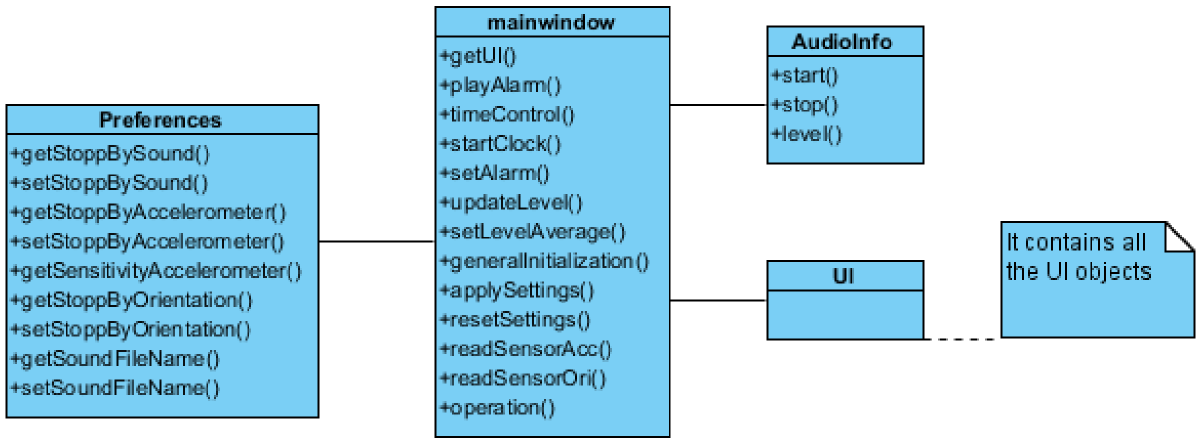

Figure 1: Class Diagram

The *Preference* class is responsible to store and retrieve the information from the configuration file. It also checks if this file exists, otherwise it create a new one.

The *AudioInfo* class handles the microphone. It reads the input audio level.

The UI class contains all the user interface objects, like button, texts and layouts.

The *mainwindows* class implements the logic related to the alarm that includes the three ways to stop it. So, if the alarm should be stopped by sound, this class will make use of the *AudioInfo* class. If the alarm should be stopped by shake or flip the device it uses the internal functions. And for reading or setting the preferences it uses the *Preferences* class. Also this class makes use of all the user interface objects.

7. Involved technologies

To implement the general functionality we used the main QT modules such as *QtCore*, *QtGui* and *QtMultimedia*. For sensors and audio functionality we needed to use QT mobility libraries like *QSensor*, *QAccelerometer* and *QOrientationSensor*.

8. Conclusions

During the developing of the application we found out that in Qt platform is easier to make more complex applications. For example connecting objects events you can use slots and signals which we believe is a very good concept that makes your work easier.

Using the mobility package we noticed that is easy to read the sensors. The only problem that we had at the beginning was that we didn’t know how to declare the mobility API in the project file.

Also the simulation is very useful when developing, like this you do not need to have the device for testing purpose. But also there is a drawback because there are functions that do not work in all the environments. For example the function *isFinished()* of *QSound* class.

Because the software doesn’t run the same in all the environments, we also noticed other problems, for example on *Symbian*, our microphone implementation does not work.

The final conclusion is that Qt is a powerful platform that helps the developer create software in a easier manner. Even though we found some problems, we believe that in time everything will run properly. In the future we believe that the platform will be used in a larger extent.