JOGGA
Jogging Companion

Thomas Höfer
Falkensteiner Weg 3
83064 Raubling
thoefer@rz.fh-augsburg.de

Sarah-Joy Berthold
Seilerstrasse 7
86153 Augsburg
sarah.joy@web.de

Abstract

Wir beschreiben hier die Rahmenbedingungen für die Umsetzung einer Anwendung für das Android Betriebssystem. Fachlich handelt es sich bei der Applikation um einen Jogging Companion, also eine Anwendung die bei der Auswertung und Bewertung von Lauf-Einheiten hilft.

Categories and Subject Descriptors

D.3.3 Concurrent Programming Structures
D.2.2 User Interfaces
D.3.3 Patterns
H.5.1 Multimedia Information System
I1.2. Sensors

General Terms
Measurement, Design

Keywords

Ubiquitous computing, User Interface, Loose Kopplung, Entwurfsmuster, Multithreading, Sensoren, Beschleunigungsmessung

1. MOTIVATION

Geht man regelmäßig joggen ist es aufgrund der körperlichen Belastung schwierig die eigene Leistung einzuschätzen. Vergleiche mit den letzten Laufeinheiten werden schwierig bis unmöglich. Genau hier setzt JOGGA an - man erhält detaillierte Statistiken über die eigene Leistung.

Unsere Anwendung ist für alle Laufbegeisterten interessant, da damit die Einschätzung der eigenen Lauf-Leistung möglich wird. Mobile Computing & Smartphones boomen seit dem iPhone und nicht zuletzt wegen Android, daher besitzt das Thema höchste Aktualität.

Mit unserer Anwendung haben wir eine sinnvolle Verwendung der Möglichkeiten im Bereich mobile Computing gezeigt und auf Implementierungsebene Best-Practices im Software-Engineering verwendet: Synchronisation in Multithreading-Umgebungen, Komponenten-Entwicklung, Observer Pattern sowie Entkopplung durch die Verwendung von Interfaces.

2. VERWANDTE ARBEITEN

Cardio-Trainer
Diese Anwendung besitzt eine Distanz- und eine Dauer-Anzeige sowie die Darstellung der zurückgelegten Strecke auf einer Karte - diese Daten haben wir als Basisdaten aufgefasst und daher in unsere Applikation ebenfalls eingebaut. Unsere Anwendung loggt die zurückgelegten GPS-Punkte mit, es ist aber noch keine Kartendarstellung implementiert. Jogga zeigt einen Leistungsindex an (min/km), womit man seine Laufleistung auf einen Blick bewerten kann.

Buddy Runner
Diese Anwendung besitzt die o.g. Basisdaten sowie umfangreiche grafische und tabellarische Darstellungen. Buddy Runner besitzt jedoch weder mp3-Player noch die Darstellung eines Leistungsindex.

My Tracks
Damit lassen sich allgemein Routen festhalten. Routen können an beliebigen Stellen kommentiert werden sowie auf eine Map darstellen. Die Anzeige der Basisdaten ist ebenfalls enthalten.

Alles in allem haben wir versucht das für uns relevante aus diesen Anwendung zu integrieren sowie um Daten zu ergänzen, die wir als besonders wertvoll erachten z.B. Anzeige eines Leistungsindex.

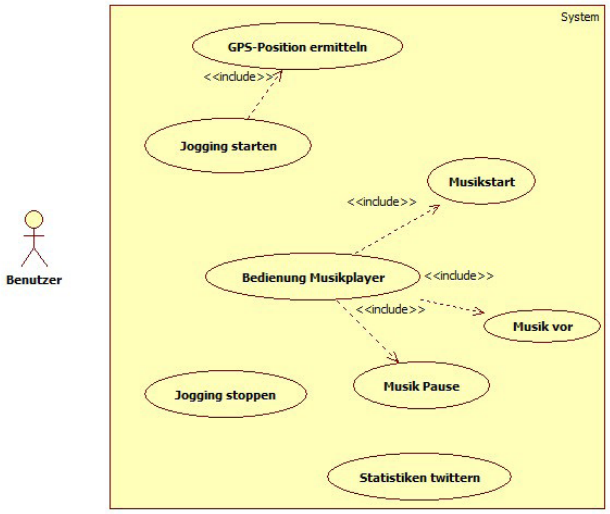
Technisch hatten wir keinen großen Spielraum da auf Android momentan nur Java offiziell unterstützt wird. Vermutlich wäre es auch möglich gewesen neue VM-Sprachen wie Scala oder Clojure zu verwenden.

Mit keiner dieser Anwendungen ist es möglich eigene Laufergebnisse an Twitter zu schicken um Freunde darüber zu informieren. Um das Rad nicht neu zu erfinden haben wir eine Drittkomponente von Android verwendet, um die Twitterposts zu realisieren.

Da Android in vielen Bereichen neu für uns war haben wir in erster Linie versucht ein bestimmtes Feature funktionsfähig zu bekommen. Anschließend haben wir den Code hinsichtlich Wiederverwendbarkeit refaktorisiert.

Fazit
Unsere Anwendung macht das, was wir als regelmäßige Läufer von einer Jogging-Anwendung erwarten würden. Nicht mehr, aber auch nicht weniger.

3. ANWENDUNGSSZENARIO & Use Cases



3.1.1 Jogging starten

Der Benutzer startet die Applikation um eine Jogging-Einheit zu starten. Dieser Use-Case muss die Vorbedingung erfüllen dass die aktuelle GPS-Position bekannt ist. Vorher ist das Joggen nicht möglich.

3.1.2 Bedienung Musikplayer

Mit dem Musikplayer können alle MP3-Songs innerhalb eines Ordners im Dateisystem abgespielt werden. Es sind die untergeordneten Use-Cases „Musik start", „Musik vor", „Musik Pause" enthalten.

3.1.3 Jogging stoppen

Die Jogging-Einheit wird hiermit beendet.

3.1.4 Statistiken twittern

Die Jogging-Statistiken können getwittert werden um Freunde über die eigenen Laufergebnisse zu informieren.

3.2 Definition der Use Cases

Jogging starten
Unser Haupt-Use-Case ist sicherlich „Jogging starten". Das Programm zeigt anfangs eine Fortschrittsanzeige an, während im Hintergrund versucht wird die Position zu lokalisieren. In der Update-Schleife wird das UI von einem separaten Thread aus mit aktuellen Werten versorgt. Vorbedingung für ein funktionierendes Programm ist der Empfang einer GPS-Position.

Bedienung Musikplayer
Nachdem die GPS-Position empfangen wurde erscheint das Standard-UI, das auch während der Laufphase sichtbar ist. Ab diesem Zeitpunkt kann man den integrierten MP3-Player verwenden.

Jogging stoppen
Wird das Joggen beendet, werden angeforderte Resourcen wie GPS oder der Update-Thread ebenfalls freigegeben.

Statistiken twittern
Statistiken können über eine Drittkomponente von Android („Twidroid") an Twitter gesendet werden.

4. ENTWURF EINES PROTOTYPS

4.1 Android

Technologisch ist Java vom Android OS vorgegeben. Die auf Android eingesetzte Davik-VM ist jedoch eine auf mobile Endgeräte optimierte virtuelle Maschine und daher nicht mit der SUN VM gleich zu setzen.

Wir beschreiben nun detaillierter unsere eingesetzten Techniken & Lösungswege:

Threads
Die Aufgabe des Main-Threads ist es andere Worker-Threads zu starten sowie Touch-Events zu verarbeiten. Es laufen also permanent drei Threads: Main-Thread, Thread zum Zeichnen des Graphen, Worker-Thread der sekündlich die Werte des UI´s berechnet. Bereiche in denen ein Thread exklusiven Zugriff auf eine Datenstruktur benötigt haben wir

als „synchronized" gekennzeichnet.

Observer-Pattern
Für die Implementierung des Schrittzähler haben wir das Observer-Pattern verwendet. Beim Schrittzähler können sich Listener registrieren, die bei jedem neuen Schritt über diesen informiert werden. Die Implementierung als Observer führt zu einer Entkopplung von Anwendung und Schrittzähler. Wir haben die eingebauten „Observer-Vorlagen" aus dem JDK verwendet.

Enkopplung konkreter Datenstrukturen
Interfaces sind reine Polymorphie - dies haben wir uns bei der Umsetzung der Graphen zunutze gemacht und einen Zeichen-Algorithmus implementiert, der nur eine Implementierung des Interfaces IPlottable erwartet um einen Graphen darstellen zu können. Wollte man einen dritten Graphen einzeichnen lässt sich das über IPlottable einfach realisieren.

MVC
Android bietet mit Activities bereits Komponenten, die Controller-Semantik besitzen. Die View-Komponenten sind mit Android-XML-Views umgesetzt. Unser Model ist in mehrere Schichten unterteilt, angefangen von rohen Messwerten bis hin zu Form-Model, das von den Android-Views referenziert wird.

4.2 Konzeptueller Aufbau des Prototypen

Activities
Diese sind von Android vorgegeben und entsprechen einem visuellen Einstiegspunkt in eine Anwendung. Eine Activity besteht im weiteren aus Subkomponenten, welche u.a. für die graphische Ausgabe zuständig sind.

System-Service - Accelerometer
Der Beschleunigungsmesser (Accelerometer) in Android ist als System-Service implementiert. Bei diesem Service kann man Callbacks registrieren welche bei einer Wertänderung informiert werden.

System-Service - LocationManager
Das Prinzip ist das Gleiche wie beim Accelerometer: Man registriert Callbacks um über Wertänderungen informiert zu werden.

Komponenten
Schrittzähler & Jukebox sind als Komponenten programmiert, welche in der Activity verwendet werden.

5. IMPLEMENTIERUNG

5.1 Hard- und Softwareumgebung

Entwickelt wurde die Anwendung für das Android Betriebssystem Version 2.1. Erfolgreich getestet haben wir es ebenfalls auf dem älteren Android 1.5.

Von Android wird momentan nur Java als Programmiersprache zur Verfügung gestellt.

Die View-Komponenten haben wir mit Android-XML erstellt, was ähnlich zu HTML funktioniert.

5.2 Kern der Implementierung

Wir haben mehrere Komponenten umgesetzt und daher auch mehrere interessante Teile. Wir haben uns aber letztendlich dafür entschieden den Schrittzähler näher zu dokumentieren:

public class Pedometer extends Observable
Um das Rad nicht neu zu erfinden verwenden wir die vom SDK mitgelieferte Observable-Klasse.

(float)Math.sqrt(Math.pow(x, 2) + Math.pow(y, 2) + Math.pow(z, 2)) - SensorManager.STANDARD_GRAVITY;
Mit dieser Formel berechnen wir die Länge unseres 3D-Beschleunigungsvektors und ziehen davon die Erdanziehung ab.

int result = (int)(val * 100);
Hiermit bestimmen wir die Sensibilität des Beschleunigungsmessers.

```
// Vereinfachter Codeausschnitt!
if(result >= 50) {
   state = DIR.TOP;
} else if(result<=-50) {
   state = DIR.BOTTOM;
}
```

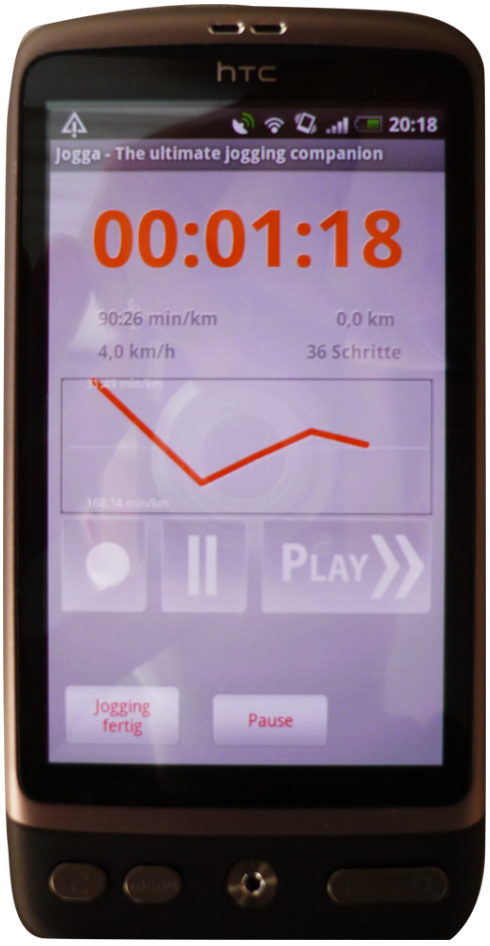
Einen Schritt definieren wir als „Nulldurchlauf", d.h. unsere Variable result oszilliert beim gehen ständig über 50 bzw. unter -50.

notifyObservers(steps);
Diese Methode wurde von Observable geerbt und benachrichtigt alle Listener über einen neuen Schritt.

5.3 Umfang des Prototypen

Implementiert:
Joggingdauer, Joggingstrecke, Leistungsindex, Geschwindigkeit, Schrittzähler, Geschwindigkeit und Leistungsindex als Graph dargestellt, MP3-Player, Twitter-Client, GPS-Lokalisierung

Teilweise implementiert:
Statistiken an Facebook senden, Anwendung beendet sich noch nicht sauber, Pause-Button



6. EVALUATION

Letztendlich soll unsere Anwendung für jeden Jogger mit durchschnittlichen Technikfähigkeiten bedienbar sein. Da-

her haben wir einen kurzes mündliches Interview mit 3 Probanden durchgeführt.

6.1 Ziele der Evaluation und Methodik

1. Sind Sie mit der Bedienung zurecht gekommen?

2. Vermissen Sie bestimmte Funktionen?

3. Sind die Display-Anzeigen sinnvoll?

Diese Fragen stellen wir in einem Interview 5 Probanden nachdem Sie sich kurz mit JOGGA auseinander gesetzt haben.

6.2 Durchführung der Evaluation

Person A) Frage 1:
Hat sich gewundert dass GPS so lange gedauert hat. Hatte keine Schwierigkeiten mit der Bedienung.

Person A) Frage 2:
Fällt dazu nichts ein

Person A) Frage 3:
Ja sicher

Person B) Frage 1:
Hatte keine Probleme mit der Bedienung

Person B) Frage 2:
Eine Kalorienanzeige wäre noch sinnvoll

Person B) Frage 3:
Siehe Antwort zu Frage 2

Person C) Frage 1:
Das Twitterfenster der „Twidroid"-Anwendung erschien öfter als gewünscht

Person C) Frage 2:
Daten abspeichern und auf PC exportierbar wäre interessant.

Person C) Frage 3:
Ja, würde aber gerne noch eine Favoritenliste beim MP3-Player einstellbar haben wollen.

6.3 Interpretation der Evaluationsergebnisse

Die Ergebnisse verwundern uns nicht, da wir die Anwendung selbst gut kennen und über die Schwächen bescheid wissen. Mit ein wenig Entwicklungszeit können wir die evaluierten Probleme jedoch definitiv ausmerzen.

7. ZUSAMMENFASSUNG

7.1 Erzielte Ergebnisse

Die vorliegende Studie beschäftigt sich mit der generellen Umsetzbarkeit einer Jogging-Anwendung für Android-Smartphones. Der Prototyp besteht aus allen zum Ziel gesetzten Komponenten: Schrittzähler, Geschwindigkeitsmesser, Leistungsindex, MP3-Player, Dauer, Graphische Auswertung und dem Versand an Twitter. Da es aufgrund der sehr knappen Entwicklungszeit zu kleineren Problemen in der Anwendung kommt ist nicht verwunderlich. Kein Problem ist jedoch so tiefgreifend, dass es nicht relativ einfach korrigierbar wäre. Durch den komponentenorientierten Aufbau

ist die Anwendung einfach refaktorisierbar. Die Interview-Evaluation mit 3 Probanden hat eigentlich nur die Probleme bestätigt, über die wir ohnehin bescheid wussten. Grundsätzlich war es unser Ziel eine in unseren Augen für Jogger passende Anwendung zu schreiben. Durch die Aussagen der Probanden lässt sich schlussfolgern, dass wir unser Ziel erreicht haben.

7.2 Erweiterungsmöglichkeiten / Ausblick

Der Pause-Button sollte nocht funktionsfähig gemacht werden, die Verwendung der Drittkomponente „Twidroid" sollte ebenfalls nochmals verbessert werden. Die Anregungen „Kalorienanzeige" bzw. „Datenexport" können problemlos noch integriert werden.

Konkurrenzanwendungen sind JOGGA vor allem noch hinsichlich der Kartendarstellung überlegen. Wir loggen die GPS-Daten aber bereits mit, so dass einer Erweiterung der Anwendung in diese Richtung ebenfalls nichts im Wege steht.

8. DANKSAGUNG

Wir danken Prof. Thomas Rist von der HS Augsburg für die positive Rückmeldung zu unserer Anwendung.

9. REFERENZEN

[1] Professional Android 2 Application Development, Autor Reto Meier, Wiley Publishing Inc., ISBN 978-0-470-56552-0