

# Programmiererfahrungen mit Android

mit besonderem Augenmerk auf die positiven und negativen Erfahrungen

von Arthur Thiessen

eMail: [7thiesse@informatik.uni-hamburg.de](mailto:7thiesse@informatik.uni-hamburg.de)

Seminar Android: Plattform für mobile Geräte

Arbeitsbereich Wissenschaftliches Rechnen

Department Informatik

Uni Hamburg

Hamburg, den 15.08.2010

## Inhaltsverzeichnis:

|  |    |
|--|----|
| 1. Einleitung .....  | 3  |
| 1. Motivation .....  | 3  |
| 2. Projekte .....  | 5  |
| 1. Tetroid .....   | 5  |
| 2. HH-Plan .....   | 5  |
| 3. Positive Erfahrungen mit Android .....                    | 7  |
| 1. Erste Schritte, Einführung und Hilfe .....                | 7  |
| 2. Einarbeitungszeit als Java Programmierer .....            | 8  |
| 3. Bibliotheken .....  | 8  |
| 4. Negative Erfahrungen mit Android .....                    | 10 |
| 1. Einrichtung auf Netbooks .....                            | 10 |
| 2. Performance und Komfort der Testumgebung (Emulator) ..... | 10 |
| 3. Bluetooth .....   | 10 |
| 4. Versionen .....   | 11 |
| 5. Process Manager .....                                     | 11 |
| 6. Activity und Datenübergabe .....                          | 11 |
| 5. Fazit .....   | 13 |
| 1. Persönliche Einschätzung .....                            | 13 |
| 6. Quellen .....   | 14 |

## 1 Einleitung

Im Wintersemester 2009/2010 wurde im Department Informatik der Universität Hamburg das Seminar „Android: Plattform für mobile Geräte“ am Arbeitsbereich Wissenschaftliches Rechnen angeboten.

Im Zuge des Seminars haben sich alle Teilnehmer tiefgründig mit dem Betriebssystem Android beschäftigt. Das Seminar bestand aus einem Vortrag und einer Schriftlichen Ausarbeitung. Das Interesse, an diesem Projekt teilzunehmen, war sowohl seitens der Studenten als auch seitens der wissenschaftlichen Mitarbeiter des Arbeitsbereichs Wissenschaftliches Rechnen sehr groß.

Bei diesem Dokument handelt es sich um die schriftliche Ausarbeitung von Arthur Thiessen zu dem Thema Programmiererfahrungen mit Android. Dabei wird besonders auf die guten Seiten von Android als auch auf die Probleme wertend eingegangen. Es handelt sich hierbei um meine subjektive Einschätzung und Analyse.

### **1.1 Motivation**

Es waren zwei Gründe dafür verantwortlich, dass ich mich für dieses Thema interessiert habe.

Zum einen bin ich auf Grund meines Informatikstudiums mit Java vertraut und sah in dem Seminar eine Möglichkeit meine jahrelang angeeignete Theorie in Praxis umzusetzen. Zudem hat es sich so ergeben, dass das Praktikum „Mobile Computing“ vom Arbeitsbereich VSIS zeitgleich mit dem Seminar „Android: Plattform für mobile Geräte“ statt fand. Dadurch war es eine einmalige Gelegenheit die erlangten Erkenntnisse aus dem Praktikum zu vertiefen.

Zum anderen habe ich die Entwicklung und Verbreitung des Open Source Betriebssystems Android seit längerem beobachtet und mit dem Gedanken gespielt mir ein Smart Phone auf Android Basis zu

zulegen. Daher würden mir die erlangten Kenntnisse in der Android-Architektur nicht nur im Rahmen meines Studiums nützen, sondern auch privat Früchte tragen.

## **2 Projekte**

Im folgenden werden die beiden Projekte vorgestellt, auf die ich mich in diesem Dokument beziehen werde. An beiden Projekten haben Senad Licina und Arthur Thiessen gearbeitet.

### **2.1 Tetroid**

Tetroid ist zu einem großen Teil in den letzten zwei Wochen des 3-wöchigen Blockpraktikums Mobile Computing (angeboten im Wintersemester 2009 vom Arbeitsbereich VSIS) entstanden. Es handelt sich dabei um einen Tetris-Klon. Allerdings ist es möglich im Multiplayer-Modus gegen andere Spieler zu spielen. Die Verbindung sollte über das Internet und direkt über Bluetooth funktionieren.

### **2.2 HH-Plan**

HH-Plan wurde für das Seminar Android: Plattform für mobile Geräte entwickelt. Es handelt sich dabei um eine Applikation, die -ähnlich wie geofox-Verbindungen mit den öffentlichen Verkehrsmitteln zwischen zwei Punkten zurückgeben kann.

Geplant war ein komplett selbstständiges Programm, welches seine Informationen von einem geofox-Server bezieht und diese anzeigt. Aus diesem Grund haben wir (Senad Licina & Arthur Thiessen) beim Hamburger Berater Team (welches für geofox zuständig ist) um einen Zugang zu den geofox-Servern gebeten. Diesen hätten wir leider nur unter sehr strengen Bedingungen bekommen. Die uns angebotene Schnittstelle basiert auf SOAP.

Da die Einarbeitung und Umsetzung mit der SOAP-Schnittstelle unseren zeitlichen Rahmen gesprengt hätte und dieser Zugriff uns nur zeitlich begrenzt

gestattet werden würde, haben wir uns letztendlich gegen diese Form des Programms entschieden. HH-Plan füllt das Online-Formular aus, welches von geofox zur Verfügung gestellt wird.

HH-Plan kann dabei die GPS-Position ermitteln und selbstständig eine Straße in der unmittelbaren Umgebung als Suchbegriff eintragen. Des weiteren ist ein Zugriff auf die Kontakt-Datenbank geplant.

## **3 Positive Erfahrungen mit Android**

### **3.1 Erste Schritte, Einführung und Hilfe**

Als Anfänger mit Java Kenntnissen ist der Einstieg in Android sehr angenehm und unkompliziert. Die speziell für Android Programmierer erstellte Webseite <http://www.developer.android.com/> bildet eine gute Anlaufstelle für Anfänger und Fortgeschrittene.

Das Einrichten der Entwicklungsumgebung ist gut dokumentiert und alle benötigten Tools sind kostenfrei im Internet erhältlich. Nach wenigen Minuten ist das System startbereit und man kann mit dem ersten „Hello World“ Programm beginnen.

Auf der Developer Seite findet man eine Vielzahl von Tutorials für viele Funktionalitäten. Codebeispiele und verständliche Erklärungen erleichtern den Einstieg enorm. Zudem werden auch grundlegende Hardware-technische Zusammenhänge angesprochen, damit der Programmierer eine Möglichkeit erhält, nicht nur den Code zu lernen, sondern auch die dadurch ausgelösten Vorgänge auf dem Smart Phone selbst.

Falls die Webseite selbst mal keinen Rat hat, wird man in unzähligen Foren/Chat-Rooms schnell fündig. Weil Android ein Open Source Betriebssystem ist, lockt es viele Entwickler an. Darunter sind viele Leihen, aber auch viele Erfahrene und bewanderte Programmierer. Eine Liste der möglichen Anlaufstellen würde an dieser Stelle den Rahmen sprengen, daher der Tipp: Einfach googlen. Man sucht nicht lang.

Mit einem geringen Initialaufwand ist man schnell dabei, sein eigenes App zu schreiben. Das überzeugt.

### **3.2 Einarbeitungszeit als Java Programmierer**

Wenn man eine Programmiersprache lernt, hat man schon alle Hände voll zu tun. Ändert man nun auch noch seine Zielplattform, so möchte man nicht zusätzliche Sprachen lernen müssen. Zumal man sowieso nie mit nur einer Sprache auskommt. Android ist hier sehr angenehm. Als Java-Programmierer hat man absolut keine Probleme die Syntax oder die Semantik zu verstehen. Alles Gelernte lässt sich auch auf den Smart Phones anwenden. Lediglich einige Bibliotheken wird man vergebens suchen. Diese wurden z.B. aus Speicherplatzgründen aus dem Standardpaket herausgenommen.

Android wendet eine ganze Reihe von Konzepten an, mit denen man sich erst einmal vertraut machen muss, bevor man gute Ergebnisse erzielen kann. So muss man zum Beispiel erst einmal lernen, was eine Activity ist und wie man diese benutzt.

### **3.3 Hardware/Software Schnittstelle**

Als Anfänger wird man großen Respekt und vielleicht sogar Hemmungen davor haben, das volle Potential von Android auszuschöpfen. Grund dafür können sein, dass man sich Benutzung der Funktionalitäten wie GPRS oder Gravity-Sensoren schwerer vorstellt, als sie tatsächlich sind.

Im Handumdrehen findet man eine Anleitung zum Programmieren eines Bewegungssensors auf der Developer Seite von Android und nach einer kleinen Einarbeitungsphase staunt man über die verblüffenden Ergebnisse.

So macht z.B. Tetroid Gebrauch von Sensoren (Gravitation und Beschleunigung), Bluetooth und dem internen Handyspeicher. Der HH-Planer greift auf GPRS zu.

Lediglich bei der Auswahl der Android Version sollte man darauf achten, dass die Funktionalitäten vorhanden sind.

### **3.4 Bibliotheken**

Möchte man für Android programmieren und braucht eine Funktionalität, die nicht Teil des Standardpaketes ist, so hat man die Möglichkeit im Internet zu suchen, ob es die gewünschten

Module nicht bereits gibt. Meist wird man schnell fündig.

Als Beispiel: Für das Projekt Tetroid wurde ein Mechanismus gebraucht, der die Nachrichten zwischen zwei Geräten verarbeitet. Die Nachrichten kamen im XML Format an und mussten daher geparkt werden. Die Möglichkeiten waren: Selbst programmieren, oder suchen, ob es bereits etwas ähnliches gibt. Aus Zeitgründen ist es häufig sinnvoll bereits existierende Software zu verwenden und somit kam kXML zum Einsatz. Die Einbindung verläuft schnell und reibungslos und man hat sofortigen Zugriff auf die Zusatzfunktionalitäten der neuen Bibliothek.

Allerdings können Bibliotheken auch Probleme bereiten. So können diese schlecht dokumentiert sein oder das Verhalten entspricht nicht der Beschreibung. Somit sind Pakete, die nicht von den Android Entwicklern stammen mit Vorsicht zu genießen. Unter Umständen handelt man sich zahlreiche Stunden Debuggens ein.

## **4 Negative Erfahrungen mit Android**

### **4.1 Einrichtung auf Netbooks**

Das Einrichten der Entwicklungsumgebung stellte sich als problematisch heraus, wenn man auf einem etwas schwächeren Netbook programmieren wollte. Die benötigte Software ist groß und speicherplatzintensiv. Die Ressourcen reichten zwar aus, um den Code schreiben zu können, jedoch ergaben sich Probleme bei der Ausführung der Testumgebung. Der Emulator, auf dem man das Programmieren testen kann, startete nicht oder nur nach langer Wartezeit. Gelegentlich hängte sich das gesamte System auf, was Datenverlust zur Folge hatte. Somit ist man gezwungen einen leistungsstarken Rechner zu benutzen und ggf. das Entwickeln von Applikationen für Android Smart Phones auf langen Zugfahrten zu vergessen.

### **4.2 Performance und Komfort der Testumgebung (Emulator)**

Der Emulator zum Testen der Applikationen ohne Einsatz eines echten Smart Phones stellte sich als schlecht heraus. Zum einen ist dieser sehr langsam im Starten und in der Bedienung, zum anderen ist das Verhalten eines Programms auf dem Emulator nicht immer die gleiche wie die auf einem echten Mobiltelefon. Dadurch ergeben sich zusätzliche Entwicklerstunden, die man beim Programmieren fürs iPhone nicht hat. Denn dort verhält sich der Emulator genau so, wie das Endgerät beim Benutzer.

### **4.3 Bluetooth**

Die Möglichkeit zwei Geräte kabellos und kostenfrei zu verbinden bietet sich hervorragend dafür an, eine attraktive Multiplayer Funktion in Spiele einzubauen. Denn dann ist man unabhängig vom Empfang und Internet-Tarif. Wichtig dabei ist lediglich, dass die zwei Geräte im selben Raum sind.

Das Problem ist allerdings, dass die Bluetooth Funktion erst ab Android 2.0 verfügbar ist. Zwar gibt es inoffizielle Bibliotheken, die es angeblich auch unter Version 1.6 ermöglichen sollen Bluetooth zu benutzen, so sind diese nur unzureichend dokumentiert und liefen nicht auf unseren Testgeräten.

#### **4.4 Versionen**

Android wächst und wächst und bringt damit ständig neue Features auf den Markt. Mit jeder Version wird es mächtiger und größer, doch damit ist es auch kaum möglich auf dem neusten Stand zu sein. Außerdem sind die Handys, die mit der aktuellsten Version arbeiten sehr teuer.

#### **4.5 Process Manager**

Als Programmierer schwankt man immer zwischen Komfort und Kontrolle. Je mehr man von dem einen hat, desto weniger hat man von dem anderen. So hat Android die Eigenschaft, dass es sich, falls es Speicherprobleme bekommt, selbstständig Platz verschafft. Dadurch braucht sich der Entwickler nicht mehr um lästiges Zusichern von ausreichendem Speicherplatz zu kümmern. Stattdessen startet man einfach ein Programm und kann davon ausgehen, dass es ausgeführt wird. Die Beschaffung der benötigten Ressourcen wird dadurch realisiert, dass Android Prozesse pausieren und sogar schließen kann. Dafür hat es eine interne Hierarchie von Prozessstypen. Je weiter unten ein Prozess in dieser Rangfolge angesiedelt ist, desto wahrscheinlicher ist es, dass es unter Umständen geschlossen wird, ohne dass es von dem Benutzer beabsichtigt war.

#### **4.6 Activity und Datenübergabe**

Activities bilden Programmabschnitte (mehr Info in der Ausarbeitung von Senad Licina). Jedes Programm besteht aus mindestens einer Activity und es kann zu einem Zeitpunkt immer nur eine Activity aktiv sein. Außerdem sind diese unabhängig von einander. So muss man beim Bau von GUIs beachten, dass man für jedes Fenster, eine eigene Activity schreibt. Dabei muss der Entwickler unter Umständen Daten von einer Activity in eine andere transportieren. Nun gibt es zwar komfortable Methoden für das übergeben von Daten in Form von primitiven Datentypen, jedoch keine für selbst erstellte.

Die einzige Lösung ist es die betroffenen Objekte zu serialisieren und dann als Zeichenkette zu transportieren. Dann braucht man wiederum Methoden, die die Objekte wiederherstellen. Dies bedeutet bei komplexen Strukturen erheblichen Programmieraufwand für den Entwickler und verlorene Zeit, die man besser hätte an anderen Stellen gebrauchen können.

## **5 Fazit**

Hier folgt nun ein kurzes Fazit zum Android Betriebssystem basierend auf den oben beschriebenen Erfahrungen.

### **5.1 Persönliche Einschätzung**

Meiner Meinung nach ist das Betriebssystem Android sehr stark und hat viel Potential. Bisher wurde fleißig an Problemen gearbeitet, sodass einige Dinge, die ich in dieser Ausarbeitung aufgezählt habe, bereits Geschichte sind. Das gibt Hoffnung, dass Android vielleicht das perfekte Betriebssystem (aus programmiertechnischer Sicht) wird.

Es war ein lehrreicher Ausflug in die Welt der Smart Phones und sicherlich nicht das letzte mal, dass ich für dieses System programmiert habe.

## **6 Quellen**

<http://www.developer.android.com>