

# Entwicklung der Open-Street-Applikation für Windows Phone 7

Florian Rapp

Universität Ulm, Abt. DBIS  
florian.rapp@uni-ulm.de

**Zusammenfassung** Diese Arbeit beschäftigt sich mit der Entwicklung einer Routing- und Navigationsapplikation, namentlich der *Open Street App*, für das mobile Betriebssystem Windows Phone 7 (WP7). Es wird aufgezeigt, welche Schritte von der Idee dieser Applikation bis zur Fertigstellung umgesetzt, und welche Probleme und Herausforderungen überwunden werden mussten. Die Arbeit soll einen grundlegenden Eindruck der praktischen Entwicklung für WP7 vermitteln. Es soll ebenfalls gezeigt werden, zu welchen Schlüssen für die Weiterentwicklung der Applikation die bisherige Entwicklung geführt hat.

## 1 Einleitung

### 1.1 Rahmen der Arbeit

Im Herbst 2010 veröffentlichte die Firma Microsoft ihr eigenes Betriebssystem für Smartphones, Windows Phone 7 (WP7). Basierend auf WinCE 7, allerdings mit einem komplett neuen und modernen User Interface, sollte ein flexibles, innovatives System auf den Markt gebracht werden. Um mit den Branchenführern *Apple iOS* und *Android* konkurrieren zu können, wurde insbesondere die einfache und schnelle Entwicklung von Applikationen für das neue System von Microsoft angepriesen. Im Rahmen des Seminars *Entwicklung für Windows Phone 7* der Abteilung *Datenbank- und Informationssysteme* der Universität Ulm, war es unser Ziel zu prüfen, wie effektiv sich Programme für WP7 zum aktuellen Stand entwickeln lassen.

### 1.2 Idee der Open Street App

Im Blickpunkt unserer Entwicklung stand nicht nur das reine Erstellen einer Applikation, sondern gerade auch die Verwendung der von Microsoft bereitgestellten Controls und Bibliotheken. Wir wollten herausfinden, ob diese sich effektiv und schnell integrieren lassen. Zum Zeitpunkt des Seminarbeginns war für WP7 keine zufriedenstellendes Third-Party-Routing verfügbar. WP7 liefert von sich aus eine Karte/Routing-Applikation. Unsere Idee war es nun eine eigene Software zu schreiben, welche den Anforderungen der mitgelieferten *Karten-App* genügt und falls möglich übertrifft - unter Verwendung der erwähnten Controls.

### 1.3 Aufbau dieser Arbeit

In Kapitel Zwei wird erklärt mit welchen Quellen und Umsetzungsmitteln gearbeitet wurde, um Kartendaten und zugehörige Informationen zu erlangen und darzustellen. Das folgende Kapitel Drei beschäftigt sich mit der Gestaltung der Applikation im WP7 typischen UI-Stil. Hierbei wurde besonderes Augenmerk auf ein konsistentes Look-and-Feel gelegt im Vergleich zu den native Applikationen für WP7. Kapitel Vier berichtet exemplarisch von den Herausforderungen und Problemen, die bei der Entwicklung aufgekommen sind. Kapitel Fünf erklärt wie einmal entwickelte Applikationen ihren Weg auf den Markt finden. Hier wird auf den Marketplace und die Zukunft der Open Street App eingegangen. Im letzten Kapitel wollen wir einen Ausblick über die möglichen Entwicklungen von WP7 geben, sowie ein Fazit zu der Entwicklung in diesem frühen Stadium ziehen.

## 2 Open Street Map und Konsorten

### 2.1 Umsetzung des Map-Controls

Der Kernaspekt der Open Street App ist die Kartenansicht. Die Karten bilden sowohl die Grundlage für alle Operationen im Routing- und Navigationsbereich, als auch den Dateninput. Es war also unsere erste Aufgabe, passende Karten auf dem Gerät darstellen zu können. Dies wird von Microsoft durch das *MapControl* und durch Verwendung einer *MultiTileSource* ermöglicht. Die *MultiTileSource* ist eine Komponente die eine Quelle für Tiles erhält. Ein Tile bezeichnet einen bestimmten Kartenausschnitt, definiert durch Zoomstufe sowie horizontale und vertikale Start- bzw. Endkoordinaten. Die *MultiTileSource* setzt bei uns dann immer aus vier Tiles unsere aktuelle Karte zusammen die dann im *MapControl* angezeigt wird. Beim Scrollen oder Zoomen der Karte werden die neuen Tiles vom Quell-Server angefragt und das *MapControl* entsprechend aktualisiert.

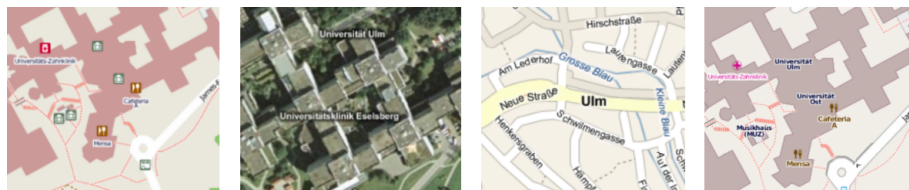


Abbildung 1. Tilesources: Cloudmade [1], VE Areal [5], VE Maps [5], OSM [2]

## 2.2 Verwendung von Services

Im Vergleich zu Applikationen, die bei Start bereits vollständig auf dem Gerät vorhanden sind, zeichnet sich eine Routing-App vor allem durch Informationen, die jederzeit abgefragt werden können, aus. Dem Benutzer müssen zur jeder Zeit und an jedem Ort an dem eine Mobilfunkverbindung aufgebaut werden kann, alle Dienste die man zum Navigieren auf Grundlage einer Karte braucht, bereitgestellt werden. Fast alle diese Daten werden über bestimmte Services angefragt und dann entsprechend auf dem Handy präsentiert. Open Street App verwendet mehrere dieser Services. Grundlegend lassen sich die Anfragen in verschiedene Gruppen einteilen:

**Geocoding** Der Begriff Geocoding oder auch Geotagging bezeichnet die Anreicherung eines (Address)Datensatzes um raumbezogene Information. In unserem Fall verstehen wir darunter die Anreicherung einer Adresse oder eines bestimmten Ortes um eine Koordinate. Dem Benutzer wird über eine Suchmaske die Möglichkeit geboten, einen Suchbegriff einzugeben. Dieser Suchbegriff wird URL-Encoded und dann Teil einer Anfrage an einen REST orientierten Webservice, wie er von Firmen wie Yahoo, Google und Microsoft bereitgestellt wird. Als Antwort werden im Erfolgsfall die Koordinaten des gesuchten Ortes zurückgesendet. Open Street App arbeitet in seiner ersten Version mit mehreren verschiedenen Webservices, um die Suchergebnisse zu optimieren.

**Reverse Geocoding** Unter Reverse Geocoding versteht man die Umwandlung einer Koordinate in eine lesbare Adresse. Dies ist eine Grundfunktionalität die von allen Applikationen, die zur Orientierung dienen, verwendet wird. Über ein GPS Signal liegen der App die aktuellen Koordianten des Benutzers vor. Abbildung 2 zeigt einen beispielhaften Aufruf des Yahoo Placefinder Service [3]. Es werden die Koordinaten des Gebäudes O28 der Universität Ulm übergeben.

---

```

1 Uri adress = new Uri("http://where.yahooapis.com/geocode?q=" +
    48.421997 + "," + 9.956117 + "&gflags=R&locale=" + System.
    Globalization.CultureInfo.CurrentCulture.Name + "&appid
    =12345");
2
3 System.Net.WebClient wc = new System.Net.WebClient();
4 wc.DownloadStringAsync(adress);

```

---

**Abbildung 2.** Reverse Geodocing Anfrage für die Universität Ulm

Das Ergebnis eines solchen Aufrufes kann in verschiedenen Formaten angefordert werden. Üblich sind das JSON (JavaScript Object Notation) und das

XML-Format. Abbildung 3 zeigt einen Ausschnitt des „responses“ zu der Anfrage aus Abbildung 2 im XML-Format. Die Koordinaten wurden in eine Adresse transformiert. Je nach Anforderung können diese „responses“, über entsprechende Optionen der Anfrage, mehr oder weniger Details enthalten.

---

```
1 <Result>
2   <latitude>48.421997</latitude>
3   <longitude>9.956117</longitude>
4   <line1>James-Franck-Ring</line1>
5   <line2>89081 Ulm</line2>
6   <line4>Germany</line4>
7   <statecode>BW</statecode>
8 </Result>
```

---

**Abbildung 3.** Response für die Geolocation Anfrage(Ausschnitt)

**Point of Interest** Ein weiterer Service den die Open Street App verwendet, ist der „Point of Interest“ Service. Hierbei handelt es sich um das gleiche Prinzip wie beim Reverse Geocoding. Dem Server werden Koordinaten übergeben. Jedoch statt diese Koordinaten in eine Adresse umzuwandeln, werden nach speziellen Punkten in der Umgebung des angegebenen Punktes gesucht. Spezielle „query-strings“ ermöglichen es, einen Radius sowie bestimmte Suchkriterien anzugeben. In ihrer ersten Version beschränkt sich die Open Street App auf das Anzeigen der naheliegendsten Points of Interest, auf eine Anzahl von maximal Zehn begrenzt. Angezeigt werden priorisiert Tankstellen und Gastronomie. Die Grundlage für einen solchen Webservice kann nur gegeben sein, wenn die entsprechenden Adressen nicht nur mit Geoinformationen, sondern auch mit einer entsprechenden Kennzeichnung als Ort des öffentlichen Interesses getaggt und auf dem Server des Servicebetreiber bereitgestellt werden.

**Routing** Eine weitere Hauptfunktionalität ist das Routing. Der Benutzer kann über eine Maske Start- und Zielort eingeben. Diese Eingaben werden per Webservice-Aufruf in Koordinaten umgewandelt. Ein weiterer Aufruf an den Routing Service von Bing [5], liefert eine Liste von Wegpunkten zurück, welche die Route charakterisieren. Das Microsoft MapControl bietet für Routen einen eigenen Layer, welcher mit der erhaltenen Liste von Wegpunkten initialisiert werden kann. Die Karte zeichnet dann eine Route vom Startpunkt zum Ziel.

### 3 Umsetzen der WP7 Designkonzepte

#### 3.1 WP7 Look and Feel

Unter dem Banner der Innovation hat sich Microsoft mit seinem mobilen Betriebssystem Windows Phone 7 von allen Altlasten, in Form von Windows Mobile, getrennt. Was dem Benutzer jedoch vordergründig ins Auge springt, ist die neue „Metro-UI“ (Abbildung 4). Diese Oberfläche zeichnet sich durch ein einfaches und vor allem in sich konsistentes Bedienverhalten aus. Das Phänomen Smartphone gründet sich hauptsächlich auf einem Sachverhalt: Alles ist durch Berührung auf dem Bildschirm steuerbar. Keine Knöpfe müssen mehr gedrückt werden. Dieses Konzept wollten wir ebenfalls für unser App umsetzen. Wichtig war uns hierbei dass die Grundkomponente, die Karte, völlig frei und direkt gesteuert werden kann. Oftmals sind in Applikationen für Smartphones Softwareknöpfe umgesetzt, so dass der Benutzer zwar über Touch steuern kann, jedoch im Prinzip auch nur einen Knopf drückt. Die Kartenansicht der Open Street App verzichtet völlig auf Knöpfe für Zoom/Unzoom oder für das Verschieben der Karte. Jede Bewegung des Fingers wird direkt weitergegeben.



Abbildung 4. „Peoplehub“ im neuen Metro-UI

Zusatzfunktionen wie die Suche, Routing oder die Einstellung sind über die Application Bar verfügbar, wie sie in Abbildung 6 zu sehen sind. Die Kontextmenüs der Open Street App orientieren sich an den nativen Applikationen von

Windows Phone 7. Abbildung 5 zeigt den „Preferences“-Dialog der Applikation. Die verwendeten Controls sowie auch das Layout der Seite selbst orientieren sich an den nativen Einstellungsdialogen von WP7. Der Gedanke hierbei war es, dem Benutzer keine neuen Konzepte und Funktionen aufzuzwängen, sondern über Wohlbekanntes eine schnelle Bedienung zu ermöglichen. So ist es ebenfalls möglich, über den allzeit vorhandenen Zurück-Button, innerhalb der Applikation zu navigieren.

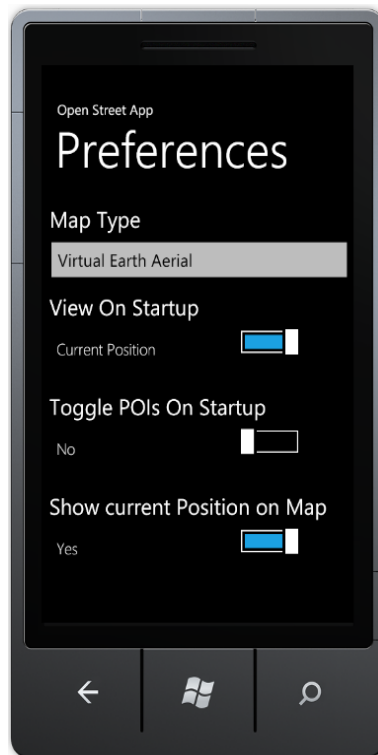


Abbildung 5. WP7 typische UI controls

## 4 Herausforderungen

### 4.1 Probleme der gegebenen Controls

Die von WP7 gegebenen Controls erlauben es sehr schnell zu Ergebnissen zu kommen [6]. Da sie jedoch als Basis für alle Entwickler dienen, sind einige Aspekte natürlich nicht optimal für spezielle Anwendungen. In unserem Fall

war das MapControl, welches für die Kartendarstellung eine hervorragende Grundlage bietet, nicht völlig ohne Anpassung verwendbar. Die Methoden, die die Berührungen des Touchscreens verarbeiten und weitergeben, waren uns nicht direkt genug. Um dieses Problem zu umgehen, haben wir einen eigenen Layer geschrieben, welcher die Touchevents auffängt, nach unseren Vorstellungen skaliert und dann erst an die verarbeitenden Methoden weiterreicht. So war es uns möglich, die Steuerung noch schneller und direkter umzusetzen.

## 4.2 Services

Wie in Abschnitt 2.2 beschrieben, werden viele Services verwendet, um an die benötigten Daten zu kommen. Die meisten dieser Services sind für Entwickler kostenfrei. Jedoch ist es meist erforderlich sogenannte „Credentials“ zu besitzen, normalerweise eine Authentifizierungsnummer, welche durch Onlineregistrierung erworben, und bei Programmstart an die jeweiligen Serviceprovider übermittelt wird. Diese mehrfachen Anmeldungen bei Programmstart sorgen, auch abhängig von Reaktionszeiten der Server, für Verzögerungen im Ladevorgang. Es war deshalb notwendig, mit verschiedenen Anbietern zu experimentieren und die für uns Optimalen auszuwählen. Diese Auswahl richtete sich jedoch nicht alleine nach der Geschwindigkeit. Wie bereits im Voraus vermutet, liefern verschiedene Anbieter auch verschiedene Ergebnisse für die einzelnen Serviceanfragen, sowohl was Qualität als auch Detaillierungsgrad betrifft. Nicht alle Services ließen sich entsprechend lokalisieren und waren daher für uns nicht verwendbar. Die weitere Entwicklung der Open Street App sieht es vor, alle verwendeten Services auf die API von Bing [5] umzustellen, um eine einheitliche Grundlage zu schaffen.

## 4.3 Tombstoning

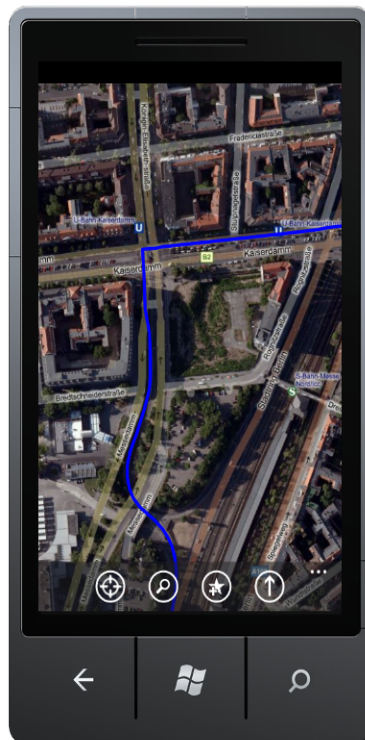
Eine Eigenschaft mit denen moderne mobile Betriebssysteme werben ist das Multitasking. In der vorliegenden Version von WP7 ist dieses weit von einer perfekten Umsetzung entfernt. Jedoch ist die Grundlage des Multitasking die schnelle Wiederaufnahme einer App. Diese Grundlage muss von Entwicklern umgesetzt werden. Tombstoning nennt sich der Prozess, in dem eine App ihre Daten und ihren Kontext sichern kann, bevor sie endgültig unterbrochen wird, durch eine andere App oder das OS. Wichtig war für uns herauszufinden, in wie weit sich ein Neustart einer App von einer Wiederaufnahme unterscheidet. Speziell welche Methoden aufgerufen werden und welche Daten noch vorhanden sind. Es war notwendig, eigene Sicherungsroutinen zu implementieren, welche dann dafür sorgen, dass der Benutzer die Karte mit allen eventuell angezeigten Informationen nach Wiedereinstieg genauso vorfindet wie beim Verlassen der App.

## 4.4 Routing

Ein weiteres Problem bezüglich der Performance stellte sich beim Implementieren des Routings. Eine Route wird, wie oben beschrieben, durch eine Men-

FLORIAN RAPP

ge von Wegpunkten beschrieben und dann von unserem MapControl durch Zeichnen von Punkten und die Verbindung dieser durch Linien dargestellt. Da es sich bei diversen Routen jedoch um einige tausend Punkte handeln kann, waren enorme Verluste in der Performance die Folge. Insbesondere das Neuzeichnen der Route nach dem Zoomen oder Verschieben der Karte war problematisch und verbrauchte enorme Ressourcen. Die Lösung war eine Vereinfachung der Route. Für jede Zoomstufe werden nicht alle Wegpunkte, sondern ein Maximum von 400 Punkten angezeigt. Die ursprüngliche Menge von Punkten wird hierbei für jede Zoomstufe schrittweise um die Punkte verringert, die den geringsten Unterschied in der resultierenden Route bedeuteten. Das Ergebnis war eine fast genau so detaillierte Route wie unter Verwendung aller Wegpunkte, jedoch ein sehr viel flüssigeres Verhalten.



**Abbildung 6.** Vereinfachte Route



## **5 Marketplace**

### **5.1 Struktur**

Die Struktur des Marketplace orientiert sich an den Branchenkonkurrenten. Ein Entwickleraccount ermöglicht es, eigene Applikationen einzureichen. Diese werden durch Microsoft geprüft und dann im Marketplace veröffentlicht. Ein normaler Account kostet 99 Dollar im Jahr, für Studenten ist das erste Jahr kostenlos.

### **5.2 Entwicklung**

Die Entwicklung des Marketplace ist ebenfalls vergleichbar mit denen der Konkurrenz. Die Zuwachsrates entsprach im ersten halben Jahr in etwa der des Appstores bei Einführung des iPhone 1. Mittlerweile gibt es über zehntausend Apps [9]. Davon sind viele eine Portierung bekannter Apps aus anderen Stores. Firmen die sich auf die Entwicklung mobiler Applikationen spezialisiert haben können so ein noch breiteres Publikum ansprechen, bei vergleichbar geringem Mehraufwand, den eine solche Portierung darstellt.

### **5.3 Zukunft der Open Street App**

Die Open Street App wurde im Sommer 2011 fertiggestellt und steht nun vor der Veröffentlichung im Marketplace. Grundsätzlich bestand für uns die Auswahl aus drei Konzepten für unserer Applikation:

- kostenpflichtige App für die im Voraus bezahlt werden muss
- kostenfreie App mit (eventuell Drittanbieter-) Werbung
- werbe- und kostenfreie Gratisapp

Da wir die Ergebnisse unserer Arbeit allen WP7 Nutzern zur Verfügung stellen wollten, war die Variante der kostenpflichtigen Applikation nicht relevant für unsere Veröffentlichung. Die Möglichkeit unser Programm durch Werbung anzureichern, war lange Zeit das Konzept unserer Wahl. Applikationen die vom Kaufsystem auf eine Werbefinanzierung umgestiegen sind, haben in den letzten Monaten einen großen Anteilszuwachs zu verzeichnen. Gerade der Spielebereich hat sich durch den hohen Wiederspielwert als profitabel für Werbeeinblendungen gezeigt [7]. Schlussendlich haben wir uns dafür entschieden, die App frei von Kosten und Werbung in den Store zu bringen. Eine dauerhafte Werbeeinblendung ist bei der Bedienung einer Karte störend und würde die Usability und damit die Zufriedenheit der Benutzer einschränken.

## **6 Ausblick und Fazit**

### **6.1 Weiterentwicklung der Plattform**

Mit Windows Phone 7 hat Microsoft einen großen Schritt in die richtige Richtung getan und sich auf eine neues, zukunftsorientiertes Design eingelassen.

WP7 ist jedoch noch nicht „fertig“ und leidet wie die meisten Systeme der ersten Generation noch an Kinderkrankheiten. Das Fehlen von „Copy und Paste“, sowie eine vernünftige Suchfunktion, waren zwei der Hauptkritikpunkte, sind jedoch mittlerweile behoben. Im Laufe des Jahres 2011 sind noch größere Updates geplant, die das System unter anderem um einen HTML5 fähigen Browser, Hardwarebeschleunigung und Multitasking erweitern.

## 6.2 Chancen für die Zukunft

Zum Zeitpunkt des Open Street App Entwicklungsstartes waren neun verschiedene WP7 Geräte verfügbar [4]. Durch die Partnerschaft zwischen Nokia und Microsoft wird diese Palette stetig ausgebaut werden. Der Marketplace wird in Zukunft gerade durch die portierten sowie neuentwickelten Apps kein Negativkriterium bei der Wahl eines Smartphones sein. Die Metro-UI-Oberfläche wird durch die Einbindung im kommenden Betriebssystem Windows 8 von einem breiten Publikum verwendet werden. Somit kann die Lücke zwischen Desktop Computern und mobilen Geräten ein Stück weit geschlossen und ein einheitliches Bedienkonzept etabliert werden.

## 6.3 Fazit

Die Entwicklung für WP7 war, gerade im frühen Stadium, erstaunlich entgegenkommend. Durch die Verwendung von Silverlight besteht für C# erprobte Entwickler kaum ein Unterschied zur Entwicklung auf dem PC. Durch die Bereitstellung der Controls durch Microsoft lassen sich schnell Ergebnisse erzielen, welche effizient im Emulator getestet werden können. Eine Vielzahl von bereitgestellten Hilfen, Tutorials und Beispielprojekten [8] ermöglichen einen einfachen Einstieg in die Entwicklung. Unabhängig von Präferenzen in Bezug auf das persönlich verwendete mobile Betriebssystem, kann WP7 von uns als Einstieg in die mobile Entwicklung nur empfohlen werden.

## Literatur

1. CloudMade. Geocoding and geosearch. <http://developers.cloudmade.com/wiki/geocoding-http-api/Documentation>. Zugriffen am 05.04.2011.
2. OpenStreetApp Community. Osm api v0.6. [http://wiki.openstreetmap.org/wiki/API\\_v0.6](http://wiki.openstreetmap.org/wiki/API_v0.6). Zugriffen am 05.04.2011.
3. Yahoo! Inc. Yahoo! placefinder guide. <http://developer.yahoo.com/geo/placefinder/guide/>. Zugriffen am 05.04.2011.
4. Jan Kluczniok. Wp7: Alle smartphones mit microsofts neuem handy-os im Überblick. <http://www.netzwelt.de/news/84335-wp7-smartphones-microsofts-neuem-handy-os-ueberblick.html>. Zugriffen am 05.04.2011.
5. Microsoft. Bing maps api. <http://msdn.microsoft.com/en-us/library/dd877180.aspx>. Zugriffen am 05.04.2011.
6. Microsoft. *Programming Windows Phone 7*. Microsoft Press, 2010.

## ENTWICKLUNG DER OPEN-STREET-APPLIKATION FÜR WINDOWS PHONE 7

7. Elbert Perez. Ads and wp7 games. <http://www.occasionalgamer.com/2011/02/09/ads-and-wp7-games/>. Zugegriffen am 09.02.2011.
8. Puja Pramudya. *Silverlight for Windows Phone - learn & practice*. Microsoft Innovation Center, 2010.
9. Jacob Schulman. Windows phone 7 marketplace hits 10,000 apps, microsoft wp7 updates still way outnumbered. <http://www.engadget.com/2011/03/11/windows-phone-7-marketplace-hits-10-000-apps-wp7-updates-still/>. Zugegriffen am 11.03.2011.