

Entwicklung auf Windows Phone 7

Am Beispiel einer OPENSTREETMAP Anwendung

Martin Rauscher

21. März 2011

1 Einführung

Laber laber laber

1.1 Vorstellung der Beispielanwendung

Im Rahmen dieser Arbeit wurde eine Anwendung erstellt, die das Betrachten von Karten von OPENSTREETMAP (und anderen Anbietern) ermöglicht. Sie ermöglicht die Steuerung über die üblichen Multi-Touch-Gesten, das suchen nach Orten und das erstellen von Favoritenlisten. Desweiteren können Routen geplant werden.

Alle Beispiele in dieser Arbeit sind aus dem Code dieser Anwendung. Für nähere Informationen zur "OpenStreetApp" wird auf XXXXX verwiesen.

1.2 Rahmen dieser Arbeit

Um Anwendungen für WP7 zu schreiben gibt es grundsätzlich zwei Möglichkeiten¹:

1. Entwicklung auf Basis von Silverlight
2. Nutzung des XNA Framework - eine gekapselte DirectX-Schnittstelle - welches hauptsächlich für die Entwicklung von Spielen verwendet wird.

Diese Arbeit beschäftigt sich ausschließlich mit der Entwicklung mit Silverlight, da das Entwicklungsmodell mit XNA gänzlich anders ist und es nur in seltenen Fällen Sinn macht mit XNA nicht-Spiele zu entwickeln.

¹Für spezielle Microsoft Partner ist es auch noch möglich halb-native Anwendungen zu schreiben.

2 Windows Phone 7

Windows Phone 7 ist hinsichtlich der Benutzeroberfläche als auch der Anwendungsplattform eine vollständige Neuentwicklung. Der Kern des Betriebssystems hingegen ist eine veränderte Version von Windows EC 7.0, dem Nachfolger von Windows CE 6, worauf Windows Mobile 6.5 beruht.

Gründe nicht auf das bestehende Windows Mobile aufzusetzen waren [XXXX]

- UI-Konzept zu stark Desktoporientiert
- Fehlende Multi-Touch Unterstützung
- Entwicklung moderner Anwendungen schwierig

WP7 begegnet diesen Problemen mit einer starken Unterstützung für Multi-Touch in Silverlight und den integrierten Kontrollelementen, die es leicht machen sollen, moderne Anwendungen zu schreiben, die sich in das UI Konzept einfügen.

3 SILVERLIGHT und die WINDOWS PRESENTATION FOUNDATION

Das Framework auf WP7 Handys basiert auf dem - vom Desktop bekannten - SILVERLIGHT, welches wiederum auf der WINDOWS PRESENTATION FOUNDATION basiert. In den folgenden Abschnitten soll kurz aufgezeigt werden, wie die beiden Frameworks entstanden sind und was die groben Unterschiede sind.

3.1 WINDOWS PRESENTATION FOUNDATION (WPF)

Ende 2006 stellte Microsoft die finale Version 3.0 des .NET Framework vor. Mit diesem wurde zum ersten Mal die, bis dahin unter dem Codenamen Avalon entwickelte, WINDOWS PRESENTATION FOUNDATION (WPF) der Öffentlichkeit zugänglich gemacht. Microsoft hoffte, dass Entwickler, insbesondere für das nur zwei Monate später erscheinende Windows Vista, nun hauptsächlich mit .NET und WPF entwickeln würden. Bis heute ist allerdings die Akzeptanz von WPF im Consumerbereich eher gering.

Mit WPF führte Microsoft erstmals ein deklaratives Modell zur Gestaltung von Benutzeroberflächen ein. Dabei wird die GUI und die Datenbindungen mit einem auf XML basierenden Format beschrieben. Dies erhöht die Wartbarkeit und Wiederverwendbarkeit verglichen mit den bisher verwendeten, zum Teil riesigen, automatisch generierten Funktionen enorm.

Eine weitere große Neuerung ist die Abwendung von GDI(+) für das GUI-Rendering. Mit WPF wird erstmals voll auf (hardwarebeschleunigtes) DirectX gesetzt. Das hat den Vorteil, dass komplexe Oberflächen gerendert, transformiert und mit Effekten versehen werden können, ohne dass die CPU zusätzlich belastet wird.

3.2 SILVERLIGHT (WPF/E)

2007 stellte Microsoft SILVERLIGHT (unter dem Codenamen Windows PRESENTATION FOUNDATION/EVERYWHERE (WPF/E) entwickelt) vor. Silverlight ist, plakativ gesprochen, eine stark reduzierte Version des .NET Framework und eine Untermenge² der WPF Funktionalität.

Auch wenn Silverlight oft als Konkurrenz zu Adobes Flash Player gesehen wird, wurde es hauptsächlich mit dem Ziel entwickelt so genannte Rich Internet Applications (RIAs) zu ermöglichen, die hauptsächlich im Geschäftsbereich ihren Einsatzzweck haben. Der einzige Bereich in dem Silverlight momentan tatsächlich in direkter Konkurrenz zu Flash steht ist die Videowiedergabe. Sowohl SMOOTH STREAMING³ als auch hardwarebeschleunigte Videowiedergabe waren zuerst in Silverlight möglich. Deshalb hat Microsoft einige prominente Unterstützter gefunden: U.a. wurden die Olympischen Winterspiele von 2008 und einige amerikanische Sportgroßereignisse exklusiv via Silverlight gestreamt; außerdem setzt das deutsche Videoportal "Maxdome" auf Silverlight.

Silverlight wurde vom Grund auf Plattformunabhängig konzipiert. Microsoft stellt es für die meisten Windows Versionen sowie für Mac OS X zur Verfügung. Desweiteren ist ein Großteil der Spezifikationen und einige Teile des Codes öffentlich zugänglich. Darauf aufbauend entwickelt das Mono Projekt unter Leitung von Novell einen Linux Port. "Moonlight" ist momentan Kompatibel mit Silverlight 3.

3.3 Unterschiede

nochmal

Mit dem Ziel der Plattformunabhängigkeit gingen natürlich auch höhere Ansprüche an einen geringen Ressourcenverbrauch einher. Das hatte zur Folge, dass einige Funktionen - z.B. Hardwarebeschleunigung und einige Controls - erst in späteren Versionen zur Verfügung standen. Bis jetzt sind noch einige Konzepte, wie z.B. Trigger und implizite Styles, noch immer nur in WPF verfügbar.

²Aufgrund von unterschiedlichen Releasezyklen hat die aktuelle WPF einige Klassen nicht, die in Silverlight enthalten sind. Dies wurden aber über ein Silverlight Toolkit nachgereicht.

³Bei Smooth Streaming handelt es sich um eine Technik die Videomaterial dynamisch neu kodiert, um sich der zur Verfügung stehenden Bandbreite anzupassen.

4 Entwicklen mit Silverlight

4.1 Struktur einer Silverlight Anwendung

Eine Silverlight Anwendung wird immer in Form einer in “*.XAP” umbenannten ZIP-Datei verteilt. In dieser Datei sind die vier Bestandteile einer App untergebracht:

- Das Manifest (WMAppManifest.xml), welches die Fähigkeiten der App festlegt.
- Das Anwendungsobjekt (App.cs und App.xaml), welches die zentrale Steuereinheit bildet
- Die Verschiedenen “Seiten” der App
- Die Ressourcen (Bilder, Töne, etc.)

xxxxxxx

navigation

4.2 XAML

XAML, bindings, MVVM etc namespaces

XAML ist eine deklarative Sprache zur Beschreibung von Datenstrukturen, insbesondere Benutzeroberflächen, die in Silverlight und WPF verwendet wird. Sie ist ein XML Dialekt, der um verschiedene Konzepte erweitert wurde:

4.2.1 Komplexe Attributdefinitionen

In XML können Attributen nur Strings, Zahlen und Referenzen zugewiesen werden, während der Inhalt von Element beliebig komplex sein kann. In XAML können Attribute durch eine simple Erweiterung der Semantik von Elementnamen, ebenfalls beschrieben werden. Im Beispiel 1 wird so auf das Attribut `Background` über `Grid.Background` zugegriffen.

Algorithmus 1 Beispiel für Komplexe Attributdefinitionen

```
<Grid x:Name="LayoutRoot" Background="Transparent"/>
<!-- oder -->
<Grid x:Name="LayoutRoot">
    <Grid.Background>
        <SolidColorBrush Color="Transparent"/>
    </Grid.Background>
</Grid>
```

4.2.2 Attached Properties

4.2.3 Markup Extensions

4.3 Datenbindung

4.3.1 Dependency Properties

4.3.2 Converters

4.3.3 MVVM?

4.4 Reactive Extensions (Rx)

4.5 Besonderheiten auf dem Handy

4.5.1 Performance

Die Entwicklung auf Mobilgeräten unterliegt zum Teil gänzlich anderen Anforderungen, als die Entwicklung für den Desktopbereich. Während auf "normalen" PCs sowohl Rechenleistung als auch Arbeitsspeicher im Übermaß vorhanden ist, müssen sich Entwickler im mobilen Bereich über diese Aspekte wieder Gedanken machen. Nur dann kann eine ausreichende Performance auf dem Gerät zu erzielt werden.

Beispiele im Vergleich WPF SL/WP7

lazy loading

ui virtualisierung

4.5.2 TOMBSTONING

Eine Konsequenz der geringen Ressourcen ist das so genannte TOMBSTONING. WP7 unterstützt kein Multitasking⁴, wie man es auf dem Desktop gewöhnt ist. Stattdes-

⁴Das OS selbst unterstützt Multitasking und ein zukünftiges Update soll es auch Apps zugänglich machen.

sen wird ein Programm sobald es inaktiv wird aus dem RAM entfernt. Bevor dies geschieht wird der App zuvor die Möglichkeit gegeben ihren Zustand zu sichern.

Dies geschieht an zwei Stellen:

Zum einen kann jede Page lokale Daten halten. Da Pages bei jedem Aufruf neu erstellt werden, müssen diese Daten, für eine eventuell später ausgeführte “zurück” Navigation, bei jedem Verlassen gespeichert werden. Dies geschieht durch setzen eines STATE Dictionaries, auf das der Navigationsservice Zugriff hat.

Zum Anderen kann die APP Klasse globale Daten halten. Diese können entweder auch mit einer Dictionary basierten Methode gespeichert werden, oder aber auch manuell mit dem normalen Dateisystem.

Wenn eine Anwendung oder eine Seite keinen inhärenten Zustand hat, ist die Verwendung dieser Methoden aber nicht verpflichtend. Sie werden dann einfach neu erstellt, wie beim ersten Aufruf.

4.5.3 Design

Ein weiterer Aspekt, auf den hier aber nicht näher eingegangen werden soll, ist die Bauform. Durch sie bedingt steht sehr viel weniger Bildschirmfläche zur Verfügung, was u.U. ein Komplett unterschiedliches Design von Nöten machen kann. Außerdem sind in diesem Bereich Eingabemethoden üblich, die auf dem Desktop oft garnicht zur Verfügung stehen. (Z.B. Multi-Touch-Screens, Beschleunigungs- und Lagesensoren)

5 Das Microsoft-Ökosystem

5.1 Marketplace (aka App-Store^{TM5})

5.2 Entwicklungsumgebung

6 Fazit - Erfolgchancen von WP7

noch-probleme

Literatur

[1] Erste Quelleden

⁵Apple hat “App-Store” als Markenzeichen eintragen lassen, wogegen Microsoft gerade Einspruch erhoben hat.