

# Location Based Services – Waveguide

Fuhrmann Thomas, #0540440

Greifeneder Felix, #0626383

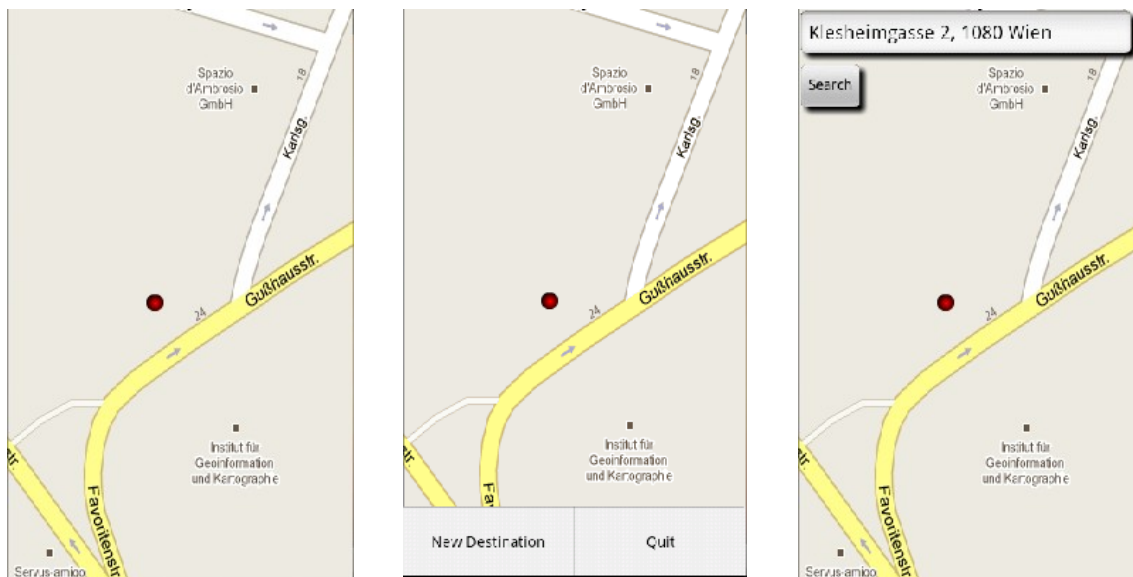
Mayr Markus, #0542042

## Einleitung

Ziel der Aufgabe ist die Erstellung eines Konzepts eines kartenfreien Fußgängernavigationssystem. Zusätzlich wurde ein MockUp und eine Frühfassung eines Programms für das Android Betriebssystem implementiert. Die kartenfreie Navigation soll dem Nutzer ermöglichen seine Aufmerksamkeit auf die Umgebung zu richten es soll möglich sein zu navigieren ohne das Display ständig im Blick zu haben. Wir haben besonderen Wert auf Sprachunabhängigkeit gelegt, Richtungsanweisungen werden durch verschiedene Ton- oder Frequenzhöhen angegeben.

## Navigationablauf

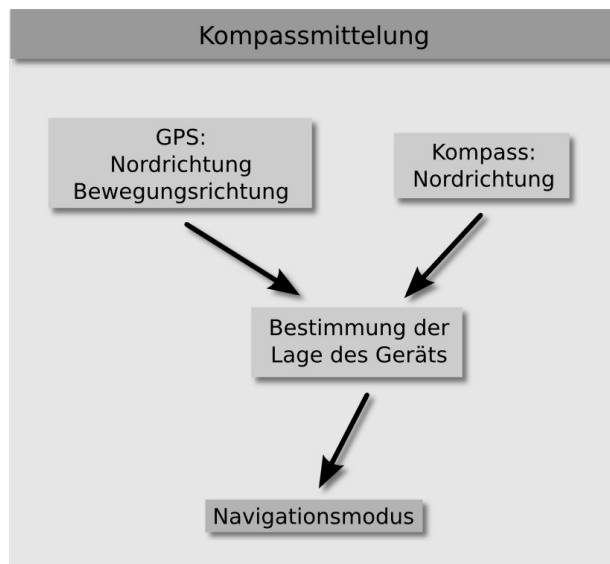
Nach dem Start der Software, wird die momentane Position auf einer Karte dargestellt. Durch drücken der Menü-Taste am Gerät öffnet sich ein Options-Menü. Der Benutzer kann wählen: Quit - beendet das Programm; New Destination - öffnet ein Eingabefenster, zur Bestimmung des nächsten Zieles



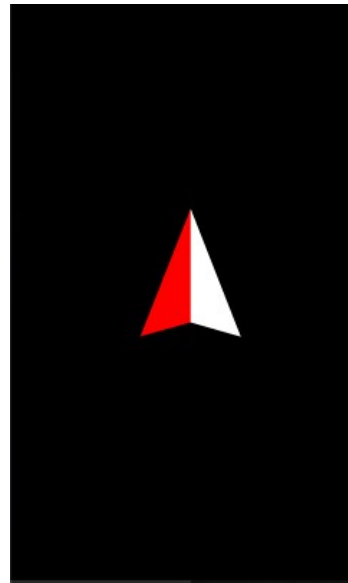
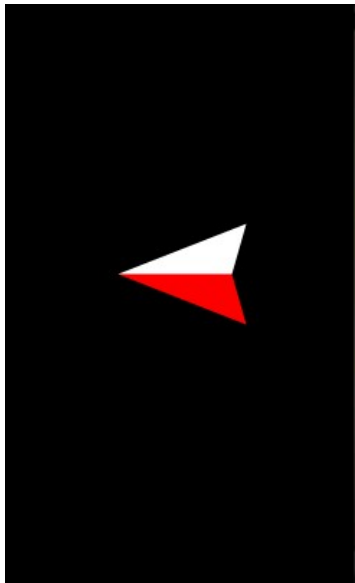
Am Bildschirm wird eine Karte mit der Route dargestellt. Durch drücken der Menü-Taste wird ein Options-Menü geöffnet. Falls es mehrere Möglichkeiten für eine Route gibt, kann der Benutzer wählen: GO! - um die vorgeschlagene Route zu akzeptieren; Next - um die nächste Möglichkeit auf der Karte anzuzeigen. Wenn nur eine Möglichkeit verfügbar ist steht nur GO! zu Auswahl.



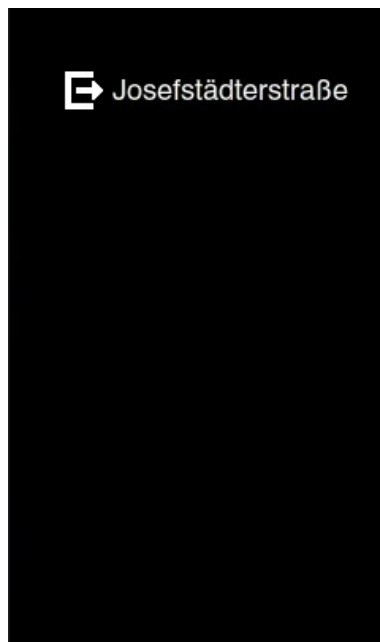
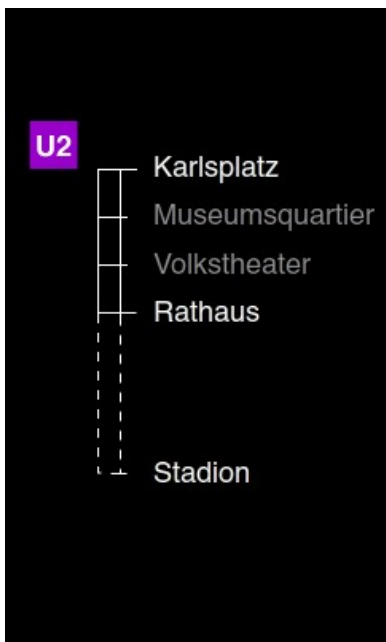
Nach der Bestätigung der Route wechselt die Software in den Navigationsmodus. Der Benutzer erhält nun turn-by-turn Anweisungen, sowohl visuell über den Bildschirm als auch durch ein akustisches Signal. Das Prinzip der akustischen Navigation wird an anderer Stelle in diesem Bericht genauer behandelt. Um dem System zu ermöglichen Richtungsangaben, auch bei nicht horizontaler Lage des Geräts, zu geben muss am Anfang des Navigationsprozesses die Bewegungsrichtung festgestellt werden (mit Hilfe von GPS). Nach dem Ausrichtungsvorgang ist es möglich mit dem akustischen Signal zu navigieren, ohne auf die Lage des Geräts zu achten.



Kurz vor jeder Abzweigung schaltet das System auf die Richtung des nächsten Checkpoints. Dies ermöglicht eine flüssige Navigation.



Falls die vorgeschriebene Route Strecken enthält, welche mit öffentlichen Verkehrsmitteln zurück gelegt werden, wird der Benutzer, bei Erreichen der Einstiegsstelle, mit Hilfe eines akustischen Signals, auf Zusatzinformation am Bildschirm hingewiesen. Am Bildschirm werden Informationen über Linie, Richtung und Ausstiegsstelle angezeigt. Beim Erreichen der Ausstiegsstelle wird der Benutzer, falls es sich um eine U-Bahn Station mit mehr als einem Ausgang handelt, auf den richtigen Ausgang hingewiesen.



Beim Erreichen des Ziels wird der Benutzer durch ein visuelles und ein akustisches Signal verständigt.

### **Android – Programmierung**

In diesem Abschnitt möchten wir kurz die Entwicklung einer Applikation für das Android Betriebssystem beschreiben. Der Leser soll einen Überblick über die Voraussetzungen für die Umsetzung einer solchen Applikation gewinnen.

## **Das Android Betriebssystem**

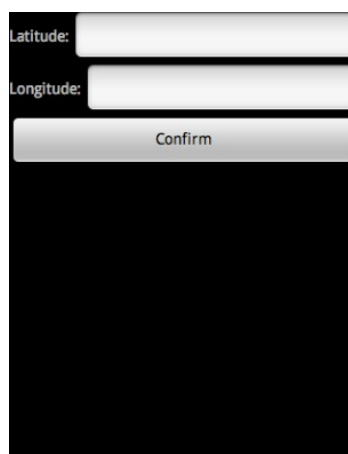
Android ist ein freies und quelloffenes Betriebssystem für mobile Endgeräte (engl. Devices), das von der Open Handset Alliance ([www.openhandsetalliance.com](http://www.openhandsetalliance.com)) entwickelt wird. Von Beginn an wurde bei der Entwicklung des Betriebssystems auf standortbezogene Dienste besonderes Augenmerk gelegt. Das HTC–Dream, das als erstes mobiles Endgerät mit dem Android Betriebssystem ausgestattet wurde, verfügte bereits über GPS–Funktionalität und Bewegungssensoren. Die Architektur des Betriebssystems basiert auf dem Linux–Kernel 2.6, als Laufzeitumgebung arbeitet die Dalvik VM (Virtual Machine). Für die Entwicklung eigener Applikationen benötigt man ein Java–SDK und ein Android–SDK (Software Development Kit). Der in Java geschriebene Quelltext wird mit einem normalen Java–Compiler übersetzt und anschließend von einem Cross–Assembler an die Dalvik VM angepasst. Als Entwicklungsumgebung verwenden wir ein spezielles Plugin für die Eclipse IDE (Integrated Development Environment). Neben Android existieren viele weitere Betriebssysteme für mobile Endgeräte, wie zum Beispiel das Apple iOS oder Symbian OS.

## **Konzeptionelle Überlegungen**

Wir wollen in diesem Abschnitt zeigen, wie sich das Android OS für standortbezogene Dienste nutzen lässt. Als Beispiel wollen wir einen Kompass zur auditiven Unterstützung von Navigationsaufgaben entwickeln. Der Nutzer der Applikation soll mit Hilfe einer Variation der Frequenz oder Tonhöhe eines Audiosignals geleitet werden. Die Sollrichtung wird durch die aktuelle Position und eine Zielposition bestimmt. Die Zielposition könnte zum Beispiel einer von mehreren Zwischenknoten eines Weges (im mathematischen Sinne) sein. Zur Vereinfachung der Aufgabe navigieren wir entlang einer Geraden zwischen Start– und Zielposition. Weicht der Nutzer von der Sollrichtung ab, so wird in diesem Beispiel die Tonhöhe verändert. Je weiter man sich vom Sollkurs entfernt desto höher wird der Ton, bis zu einem Maximalwert bei einer Abweichung von  $\pm 180^\circ$ .

## **Technische Überlegungen**

Die aktuelle Position kann über GPS bestimmt werden, sofern man sich in keinem Abschattungsbereich befindet in dem nur schlechte oder gar keine Satellitensignale empfangen werden können. Die Koordinaten der Zielposition werden durch deren geographische Länge und Breite angegeben. Hierfür dient eine einfache Benutzerschnittstelle (siehe Abbildung).



The image shows a mobile application interface with a black background. At the top, there are two white input fields. The first field is labeled 'Latitude:' and the second is labeled 'Longitude:'. Below these fields is a white button with the text 'Confirm' centered on it.

Sind Start– und Zielposition bekannt, kann die Sollrichtung als Azimut zwischen der Geraden und magnetisch Nord berechnet werden. Die aktuelle Orientierung wird über digitale Orientierungssensoren berechnet und setzt sich aus drei Komponenten zusammen: roll, pitch und azimuth. Für unsere Navigationsaufgabe benötigen wir nur die azimuth–Komponente. Die aktuelle Abweichung kann nun über die Differenz zwischen azimuth–Komponente und der Sollrichtung

berechnet werden. Die aktuelle Orientierung wird im Sekundentakt abgefragt und daraus die Abweichung berechnet. Ebenso wird die GPS-Position regelmäßig upgedatet und der Sollkurs Neuberechnet. Zu Überlegen ist noch wie oft ein Update der GPS-Position überhaupt notwendig ist, um Ressourcen verbrauchende, redundante Updates zu vermeiden. Wir haben uns für eine möglichst geringe Rate entschieden, da im Bereich der Fußgängernavigation keine schnellen Positionsänderungen passieren. Eine weitere Überlegung betrifft die Orientierung, da die aktuelle azimuth-Komponente nur sinnvoll bestimmt werden kann, wenn sich das Gerät in einer dafür geeigneten Orientierung befindet. Legt man die Achsen des lokalen Gerätekoordinatensystems so fest, dass die z-Komponente orthogonal zum Display steht, dann sollte das Gerät möglichst horizontal gehalten werden um eine sinnvolle Orientierungsbestimmung zu ermöglichen.

Dem könnte Abhilfe geleistet werden, indem man zum Beispiel die aktuelle pitch-Komponente zur Definition des lokalen Gerätekoordinatensystems heranzieht. Weicht das Gerät etwa mehr als  $45^\circ$  von der Horizontalfläche ab, so könnte die z-Achse in Richtung der Längsachse des Geräts gelegt werden und somit die Sensibilität gegenüber der azimuth-Komponente verbessert werden (siehe Abbildung).



## **Realisierung des Mock-Ups**

### **Graphik**

Die visuelle Darstellung des Mock-Ups wurde mit AdobeFlash realisiert. Dies ermöglichte das leichte Integrieren von Animationen.

Als Vorlage für das simulierte Mobiltelefon diente das „HTC Desire“ und eine Karte von „maps.google.com“.

### **Verwendete Töne**

Für das MockUp wurden, abgesehen von den Indikatoren für öffentliche Verkehrsmittel, verschiedene Arten von Tönen und Geräuschen verwendet.

Der Erster ist ein einfacher "Peep" in Form einer 0,25 sek. andauernden Sinuskurve mit einer Frequenz von 1.400 bzw. von 2.000 hz.

Der Zweite umfasst eine in ihrer Frequenz zu- oder abnehmende Sinuskurve, welche auch in ihrer Lautstärke variiert.

Die dritte Kategorie dient als Signal für das Ein- oder Aussteigen an Stationen für öffentliche Verkehrsmittel und das Erreichen des Zielpunktes. Im Prinzip wird mit diesen Geräuschen auf ein besonderes Ereignis hingewiesen. Diese Samples wurden nicht selber erstellt, sondern es wurde auf fertige Tonbibliotheken zurückgegriffen. Die für diese Kategorie verwendeten Töne umfassen:

- **Erreichen der U-Bahn:** "Schließen der U-Bahn-Türen"
- **Erreichen der Straßenbahn:** "Straßenbahnglocke"
- **Erreichen der Ausstiegsstation der U- oder Straßenbahn:** "Alarmsignal vor der Abfahrt einer NewYork-Metro" (sollte durch einen eindeutigeren Ton ersetzt werden)
- **Erreichen des Endziels:** "Applaus"

### ***Daten der Töne***

Die Navigationstöne liegen in einem Frequenzbereich von 220 bis 660 Hz. Abhängig von der Frequenz wurde, um ein gleichmäßiges Wahrnehmen im Straßenlärm zu ermöglichen, die Lautstärke vom Faktor 0,2 für die höchste Frequenz bis zum Faktor 0,8 für die niedrigste verändert.

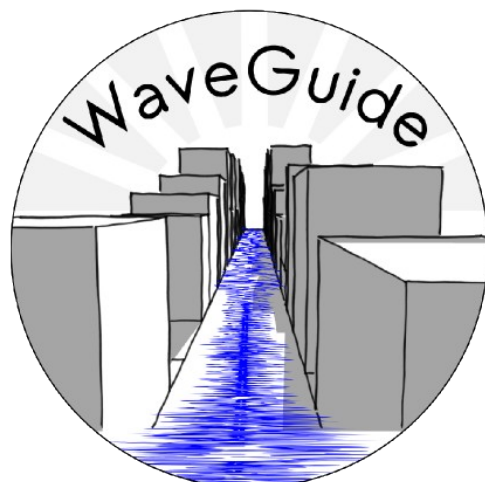
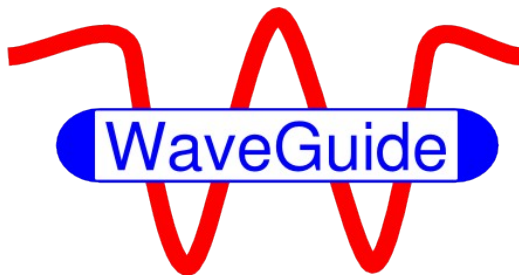
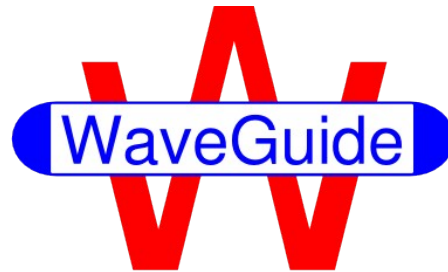
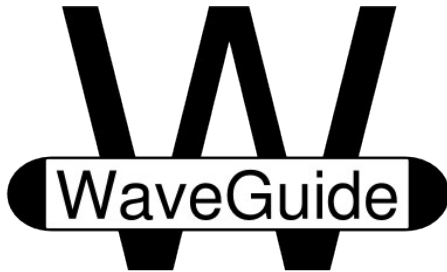
### ***Generierung der Töne***

Zur Erstellung der Töne wurde das OpenSource Programm "Audacity" verwendet, welches auf Windows, MacOS und Linux Betriebssystemen lauffähig ist. Es weist alle notwendigen Funktionen auf und ermöglicht neben dem Mischen mehrerer Spuren auch den Export in mp3 oder wav Dateien.

## Logo

Es wurden verschiedene Logos in verschiedenen Ausformungen gestaltet. Ziel war, den Namen des Navigationssystems "WaveGuide" in einer Welle widerspiegeln zu können.

Eine Übersicht über einige der erstellten Varianten bietet folgende Abbildung.



Im der gewählten, endgültigen Version (siehe Abbildung unterhalb) wird die Wellenform durch einen farbigen Akzent des Buchstabens "W" und seiner fetteren Schreibweise betont. Das im rechten Bereich des Logos positionierte Symbol deutet zusätzlich eine sich ausbreitende Welle an.

