

Indoor Navigation

Diplomarbeit im Fach Informatik
vorgelegt
von

Thomas Rauber
Egerkingen, SO
Matrikelnummer 02-707-735

Angefertigt am
Institut für Informatik
der Universität Zürich
Prof. Dr. G. Schwabe

Betreuer:
Christoph Göth

Abgabe der Arbeit: 07. Mai 2007



University of Zurich
Department of Informatics

Zusammenfassung

Navigationssysteme kennt man hauptsächlich aus dem Bereich der Kraftfahrzeuge. Begründet durch die immer leistungsfähigeren mobilen Geräte wie PDAs und Handys sind aber auch Navigationssysteme für die Fussgänger seit einiger Zeit zum Objekt der aktuellen Forschung geworden.

Im ersten Teil dieser Diplomarbeit werden die Grundlagen eines Navigationssystems erläutert. Dabei wird unter anderem auf Wegberechnungsalgorithmen und die Aufbereitung von Wegleitinformationen eingegangen, wie auch einige bereits realisierte Navigationssysteme für Fussgänger betrachtet.

Im zweiten Teil wird auf das im Rahmen dieser Diplomarbeit entwickelte Navigationssystem für den mExplorer eingegangen. Behandelt werden dabei die Architektur, die Benutzerinteraktionen, wie auch die Evaluation des entwickelten Systems.

Abstract

Navigation systems are generally known from motor vehicle. Founded on the more and more powerful devices such as PDAs and mobile phones are pedestrian navigation systems since some time part of the current research.

In the first part of this thesis the basics of navigation systems are illustrated. Among other things it is going into way calculating algorithms and the conditioning of way guidance information.

Further we look at some realised pedestrian navigation systems.

In the second part of this thesis it is going into the navigation system that has been deployed for the mExplorer. In doing so the architecture, the user interaction and the evaluation of the system are discussed.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenstellung	1
2	Literaturanalyse	2
2.1	Navigation	2
2.1.1	Fahrzeugnavigation	2
2.1.2	Fussgängernavigation	3
2.1.2.1	Indoor- und Outdoor-Anwendung	3
2.1.2.2	Uneingeschränkte Bewegungsfreiheit von Fussgängern	4
2.1.2.3	Technische Ausstattung	5
2.2	Wegberechnung	5
2.2.1	Der Algorithmus von Dijkstra	6
2.2.2	A*	7
2.2.3	Routenberechnung im Strassenverkehr	8
2.3	Präsentation der Weginformation	10
2.3.1	Visuelle Präsentation	10
2.3.2	Akustisch	11
2.4	Verwandte Arbeiten	12
2.4.1	REAL: Ein ressourcenadaptiertes mobiles Navigationssystem	12
2.4.1.1	IRREAL	13
2.4.1.2	ARREAL	14
2.4.1.3	Übergang zwischen den beiden Komponenten	15
2.4.2	Deep Map	16
2.4.2.1	Systemüberblick	16
2.4.2.2	Hardwarekomponenten	17
2.4.2.3	Deep Map Tourenplaner	17
2.4.3	INSTAR	19
2.4.3.1	Systemüberblick	19
2.4.3.2	INSTAR in der Praxis	20
2.4.4	Mobile Collaborative Augment Reality	22
2.4.4.1	Mobile Augmented Reality Setup	22
2.4.4.2	Outdoor Collaborative Augmented Reality	24
2.4.4.3	SingPost II	25
2.4.5	GUIDE Project	28
2.4.5.1	Anwendungsanforderungen	28
2.4.5.2	Benutzerspezifische Routen	29
2.4.6	Navigationssysteme für das Handy	31
3	Realisierung des Navigationssystems für den mExplorer	33
3.1	Anforderungsspezifikation	33
3.2	Grundvoraussetzungen zur Navigation	34
3.2.1	Positionsbestimmung	34
3.2.2	Peilung	34
3.2.3	Kenntnis der Umgebung	34
3.3	Details zur Wegberechnung und Wegleitung	37
3.3.1	Wahl des Wegberechnungsalgorithmus	37
3.3.2	Harmonische Wegführung	38
3.4	Architektur	40
3.4.1	Der Navigationseditor	40
3.4.1.1	Die einzelnen Klassen des Navigationseditors im Überblick	41
3.4.1.2	Zusammenspiel der einzelnen Klassen des Navigationseditors	42
3.4.2	Das Navigationsmodul	43
3.4.2.1	Die einzelnen Klassen des Navigationsmoduls im Überblick	44
3.4.2.2	Zusammenspiel der einzelnen Klassen des Navigationsmoduls	46

3.5	Benutzerinteraktion	48
3.5.1	Benutzerinteraktion des Navigationseditors	48
3.5.1.1	Harmonische Integration in das bestehende System	48
3.5.1.2	Funktionalität der Menüleiste	49
3.5.1.3	Weitere Funktionalität	50
3.5.1.4	Zustände und Zustandsübergänge des Navigationseditors	50
3.5.1.5	Visualisierung des Graphen	52
3.5.2	Benutzerinteraktion des Navigationsmoduls	53
3.5.2.1	Funktionalität des Navigationsmoduls	53
3.5.2.2	Zustände und Zustandsübergänge des Navigationsmoduls.....	54
3.5.3	Wegleitung	55
3.5.3.1	Visualisierung des Weges.....	56
3.5.3.2	Richtungsangaben	57
4	Benutzertests	59
4.1	Testsetup.....	59
4.1.1	Spielgebiet	59
4.1.2	Testpersonen.....	60
4.1.3	Technische Ausstattung	61
4.2	Ablauf der Benutzertests.....	62
4.2.1	Navigationstest	62
4.2.2	Test im eigentlichen Spiel.....	62
4.3	Ziel der Tests	63
4.3.1	Ziele des Navigationstests.....	63
4.3.2	Ziel des Tests im eigentlichen Spiel	64
4.4	Auswertung der Benutzertests.....	65
4.4.1	Zeitmessung	65
4.4.2	Navigationsmenü	66
4.4.3	Angezeigter Weg	67
4.4.4	Graphische Wegleitung	68
4.4.5	Akustische Wegleitung	68
4.4.6	Gesamtbeurteilung	69
5	Fazit.....	72
5.1	Bewertung des Erreichten.....	72
5.2	Ausblick	73
5.3	Persönliches Fazit.....	73
5.4	Danksagung	74
	Anhang	75
A	Abkürzungsverzeichnis	75
B	Literaturverzeichnis.....	76
C	Abbildungsverzeichnis	79
D	Tabellenverzeichnis.....	81
E	Inhalt der CD-Rom	82
F	Anwendungsfälle	83
F.1	Anwendungsfälle des Navigationseditors	83
F.2	Anwendungsfälle des Navigationsmoduls	84
G	Config File	86

H	Fragebögen	88
H.1	Fragebogen des Navigationstests	88
H.2	Auszug aus dem Fragebogen zum Test im eigentlichen Spiel.....	94
H.3	Durchschnittswerte des Fragebogens zum Navigationstest.....	95
H.3.1	Durchschnittswerte zum Auszug des Fragebogens zum eigentlichen Spiel	97

1 Einleitung

Der mExplorer ist eine von Christoph Göth entwickelte, mobile Spielumgebung. Die erste Fassung ist in [Göt04] beschrieben. Seit dem wurde der mExplorer vom Entwickler, wie auch von zahlreichen Diplomanden der Universität Zürich, weiterentwickelt.

Ziel des mExplorers ist es, neuen Studenten das noch unbekanntes Universitätsgelände auf spielerische Art und Weise näher zu bringen. Mit einem PDA ausgestattet machen sich die Studenten auf den Weg, um verschiedene Aufgaben zu lösen. Der PDA hilft den Studenten bei der Orientierung, denn auf ihm wird die Karte des Geländes dargestellt. Auf der Karte werden zudem die aktuelle Position des Spielers, Tasks¹ und Point of Interests (POI)² angezeigt. Die Tasks werden vorab definiert und bestimmten Orten des Universitätsgeländes zugewiesen. Um einen Task lösen zu können, muss man sich innerhalb eines gewissen Radius um den jeweiligen Ort befinden. Eine mögliche Aufgabe ist, herauszufinden, in welchem Raum sich die Bibliothek des Instituts für Informatik befindet.

Um den Spielern die Navigation zu erleichtern, soll nun der mExplorer um die Funktionalität eines Navigationssystems erweitert werden. Die Spieler sollen Orte auf der Karte anwählen können, um dann bestmöglich zum Ziel geleitet zu werden.

Die Positionsbestimmung wird anhand von den unterschiedlichen Signalstärken einzelner WLAN Access Points durchgeführt. Nebst der Positionsinformation verfügt der mExplorer auch schon über einen elektronischen Kompass, der die Peilung des PDA's anzeigt und zur Realisierung des Navigationssystems verwendet werden kann. Die Karte des Terrains ist bis jetzt rein bildbasiert und enthält noch keinerlei Information über gehbare Wege oder ähnliches.

1.1 Aufgabenstellung

Im Rahmen dieser Arbeit wird ein Navigationssystem für den mExplorer entwickelt und in das bestehende System integriert.

In einem ersten Teil wird auf die Thematik Navigation und ihre Probleme in den Bereichen der Fussgängernavigation, wie auch der Fahrzeugnavigation, eingegangen. Ebenfalls werden ähnliche, bereits existierende Systeme betrachtet und auf ihre Stärken und Schwächen hin analysiert. Die daraus gewonnenen Erkenntnisse werden dann direkt in die Entwicklung des eigenen Navigationssystems einfließen.

Der Hauptteil dieser Arbeit besteht in der Entwicklung und der Realisierung des Navigationssystems für den mExplorer. Dabei wird ein Editor kreiert, um die bildbasierten Karten mit Informationen über die gehbaren Wege zu versehen. Des Weiteren wird ein adäquater Wegfindungsalgorithmus implementiert, der aufgrund der Information der gehbaren Wege und der aktuellen Position die zu gehende Route berechnet.

Der mExplorer wird dahingehend erweitert, dass durch anklicken ein Zielpunkt festgelegt wird und der Benutzer zu diesem Punkt geleitet wird. Die Führung auf dem Weg erfolgt sowohl akustisch als auch visuell.

¹ Ort an dem eine Aufgabe erledigt werden muss.

² Für den Spieler interessante Orte.

2 Literaturanalyse

Bei der Entwicklung eines Navigationssystems muss man glücklicherweise nicht bei Null beginnen. In den verschiedenen betroffenen Bereichen wurde schon Forschung betrieben, auf deren Resultate man zurückgreifen kann. In diesem Kapitel werden nun die Grundsteine für die spätere Entwicklung des Navigationssystems für den mExplorer gelegt. Es wird auf die verschiedenen Facetten der Navigation eingegangen und gängige Wegberechnungsalgorithmen wie auch Formen der Vermittlung von Navigationsinformationen betrachtet. Abschliessend folgen noch die Betrachtungen einiger Systeme, die sich ebenfalls mit Fussgängernavigation befassen.

2.1 Navigation

Navigation spielt für uns Menschen schon seit jeher eine wichtige Rolle. Sobald wir unsere gewohnte Umgebung verlassen, sind wir darauf angewiesen, unsere aktuelle Position zu kennen und über Methoden zu verfügen, mit denen wir den Weg an unser Ziel finden.

Anfänglich war die Navigation auf See ein grosses Problem. Die Navigation mittels Kompass und der Gestirne war nur sehr ungenau und konnte auch nicht immer angewendet werden.

Im Laufe der Zeit wurden jedoch zahlreiche Hilfsmittel entwickelt. Das wohl momentan am weitesten verbreitete System zur Bestimmung der aktuellen Position, ist das Global Positioning System GPS³. Die heute zur Verfügung stehenden technischen Hilfsmittel beschränken sich aber längst nicht mehr auf die Bestimmung der Position. Navigationssysteme lotsen uns bis ans Ziel. Am geläufigsten sind uns Navigationssysteme im Zusammenhang mit Kraftfahrzeugen, da diese auch am weitesten verbreitet sind. Genauso interessant und hilfreich können aber auch Navigationssysteme für Fussgänger sein. Anwendungsbereiche dafür können beispielsweise fremde Städte oder aber auch grosse Gebäude wie Flughäfen, Universitäten oder Sportstadien sein. Generell dort, wo wir uns auf unbekanntem Terrain bewegen.

2.1.1 Fahrzeugnavigation

Navigationssysteme für Personenwagen sind heutzutage weit verbreitet. Sie werden sowohl in das Fahrzeug integrierte als auch als kleine mobile Systeme angeboten. Es sind sogar Systeme für PDAs und Handys erhältlich. Die Systeme sind mittlerweile sehr zuverlässig und auch erschwinglich geworden. Die günstigsten Systeme sind bereits ab 300 Schweizer Franken erhältlich.

Die Positionsbestimmung wird mit Hilfe eines weltumspannenden Netzes von Satelliten vorgenommen. Zurzeit wird fast ausschliesslich das von den Amerikanern betriebene System GPS verwendet. Mit dem europäischen System Galileo wird aber bald ein weiteres zuverlässiges System zur Verfügung stehen.

Zusätzlich zu den GPS-Daten werden Informationen über die Fahrtrichtung und die zurückgelegte Strecke in die Berechnungen miteinbezogen. Die Fahrtrichtung wird mit Hilfe eines Kompasses bestimmt. Die zurückgelegte Strecke kann über die Anzahl

³ Detaillierte Informationen über die Funktionsweise von GPS können dem Buch „Satellitenortung und Navigation“ [Man04] entnommen werden.

Radumdrehungen berechnet werden. Da gerade in Städten und dicht bebauten Gebieten GPS durch Abschattung des Signals nicht flächendeckend verfügbar ist, helfen diese Informationen, die aktuelle Position auch in diesen Gebieten bestimmen zu können.

Die Forschung im Bereich der Fahrzeugnavigation wird bereits seit geraumer Zeit betrieben und ist daher schon weit fortgeschritten. Kinderkrankheiten, wie Probleme in der Positionsbestimmung oder ähnlichem, sind längst ausgemerzt. So befasst sich die aktuelle Forschung damit, wie die Systeme noch weiter verbessert und auf die speziellen Bedürfnisse zugeschnitten werden können. Prof. Dr. Thomas M. Cerbe befasst sich beispielsweise mit der Fragestellung, wie man durch ein Navigationssystem zeitgenau an sein Ziel geführt werden kann [Cer06].

Ein weiterer interessanter Forschungsbereich sind dynamische Navigationssysteme, die aktuelle Verkehrsinformationen berücksichtigen und so Staus geschickt umfahren. An der Graduate School of Information, Science and Technology der Universität von Osaka befasst man sich damit, das Ziel nicht mühsam eingeben zu müssen, sondern es vom System selbständig erahnen zu lassen [Ter06].

2.1.2 Fussgängernavigation

Individuelle, interessante Stadtbesichtigungen für Touristen, zielstrebiges Auffinden von Hörsälen und Büros oder auch Unterstützung von blinden Personen - die denkbaren Anwendungsszenarien für Fussgängernavigationssysteme sind zahllos.

Kleine, tragbare Geräte wie PDAs und Handys werden immer leistungsfähiger und erlauben nun auch brauchbare Umsetzungen. Die Forschung in diesem Bereich steckt aber noch in den Kinderschuhen und ist keinesfalls schon soweit vorangeschritten wie bei der Fahrzeugnavigation.

In den Grundzügen sind sich Personen- und Fahrzeugnavigationssysteme ähnlich und so können gewisse Erkenntnisse aus der Entwicklung der Fahrzeugnavigationssysteme übernommen werden. Im Wesentlichen gestaltet sich die Entwicklung aber um einiges schwieriger, da verschiedene Problemstellungen komplexer sind. Fussgänger verfügen über eine nahezu uneingeschränkte Bewegungsfreiheit und bewegen sich sowohl im Freien wie auch innerhalb von Gebäuden. Zudem ist ihre maximale Traglast nicht sehr gross. Schweres und sperriges technisches Equipment schränkt ihre Bewegungsfreiheit sehr schnell ein.

Die nun folgenden Unterkapitel 2.1.2.1 bis 2.1.2.3 widmen sich ausführlicher den einzelnen Problemstellungen der Fussgängernavigation.

2.1.2.1 Indoor- und Outdoor-Anwendung

Fussgängernavigation muss den Anforderungen von zwei unterschiedlichen Anwendungsbereichen gerecht werden, denn Fussgänger bewegen sich sowohl im Freien (Outdoor) als auch innerhalb von Gebäuden (Indoor). Massgeblich davon betroffen sind die Positionierungskomponenten der Navigationssysteme. Je nach dem, ob man sich im Indoor- oder Outdoor-Bereich bewegt, sind manche Verfahren zur Positionsbestimmung besser geeignet als andere und manche gar nicht anwendbar.

Im Outdoor-Bereich setzen die meisten Systeme für die Positionierung auf das bewährte System GPS. Es gibt auch einige, die die Positionierung mittels GSM⁴ vornehmen. Im Grossen und Ganzen ist die verwendete Technologie aber sehr einheitlich.

⁴ Global System for Mobile Communications

Bei der Navigation innerhalb von Gebäuden sieht dies etwas anders aus. Durch die fehlende Sichtverbindung zu den einzelnen Satelliten kann GPS nicht verwendet werden.

Im Weiteren werden nun einige Verfahren zur Indoor-Positionsbestimmung erläutert. Dies ist aber keine abschliessende Aufzählung sondern soll lediglich einen kleinen Überblick über die Vielzahl der möglichen Systeme geben. Eine Möglichkeit, die auch beim mExplorer [Göt04] verwendet wird, ist die Positionierung auf Grund von WLAN-Signalstärken. Auf der Grundlage der bekannten Position der unterschiedlichen Access Points wird dabei die Position anhand der unterschiedlichen Stärken der empfangbaren Signale berechnet. Die so ermittelte Position ist immer relativ zu den bekannten Bezugspunkten. Der Vorteil dieser Methode ist es, dass grosse Gebäude, wie Universitäten, oft bereits über die nötige WLAN-Abdeckung verfügen. Alternativ dazu kann die Positionsbestimmung auch durch Infrarot durchgeführt werden. Dazu müssen die Gebäude mit Infrarot-Sensoren ausgestattet sein. Zur Positionierung wird ein Sender mitgetragen, der in bestimmten zeitlichen Abständen Infrarot-Signale mit der Benutzerkennung abgibt. Die von den Sensoren wahrgenommenen Signale werden an einen Server weitergeleitet, der anhand der Daten die Positionsbestimmung vornimmt. Ein System, das diese Technologie verwendet, ist das Active Badge Location System. Es ist in [Wan92] ausführlich beschrieben. Ein weiterer Ansatz befasst sich mit der optischen Erkennung der aktuellen Position. Dabei werden spezifische Merkmale anhand einer Kamera erkannt und dadurch ihre Position ermittelt. Eine solche Umsetzung ist in [Kal02] beschrieben. Des Weiteren werden auch Bluetooth- und Ultraschallsysteme zur Positionsbestimmung eingesetzt.

Bislang hat sich noch kein Verfahren durchsetzen können und wird es wohl in absehbarer Zeit auch nicht. Daher ist vielleicht der beste Ansatz, um möglichst flexibel und grossflächig einsetzbar zu sein, verschiedene Verfahren anzuwenden und nicht auf ein einzelnes Pferd zu setzen. Das im Kapitel 2.4.1 beschriebene System REAL verfolgt diesen ressourcenadaptierten Ansatz.

2.1.2.2 Uneingeschränkte Bewegungsfreiheit von Fussgängern

Fussgänger sind in ihrer Bewegungsfreiheit viel weniger eingeschränkt als Teilnehmer des Strassenverkehrs. Der Bewegungsraum der Automobilisten beschränkt sich auf die Strassen mit den jeweiligen Fahrtrichtungen, also mehr oder weniger auf eine Dimension. Fussgänger jedoch unterliegen diesen Beschränkungen nicht. Ihnen steht gewissermassen eine Dimension mehr zur Verfügung.

Diese weitere Dimension, kombiniert mit der Ungenauigkeit⁵ der Positionsbestimmung, bringt zusätzliche Komplexität mit sich. Der tatsächliche Aufenthaltsort befindet sich innerhalb eines bestimmten Radius ausgehend von der berechneten Position. Schneidet man diesen Kreis mit den grundsätzlich möglichen Positionen, erhält man die tatsächlich möglichen Aufenthaltsorte. Bei Fahrzeugen ist die Reduktion massiv, da ihre möglichen Aufenthaltsorte durch den Strassenverlauf schon sehr eingeschränkt sind, bei Personen ist dies allerdings nicht der Fall.

Je präziser die Positionsbestimmung erfolgt, desto effizienter kann die Wegführung gestaltet werden. Der zu gehende Weg kann beispielsweise stark variieren, je nachdem auf welcher Seite man sich von einer Schnellstrasse oder einem Gebäude befindet.

⁵ Die Genauigkeit kann je nach System und Umgebung von einigen Metern bis zu einigen hundert Metern betragen.

2.1.2.3 Technische Ausstattung

Natürlich spielt die technische Ausstattung nicht nur bei der Fussgängernavigation sondern auch bei der Fahrzeugnavigation eine entscheidende Rolle, nur braucht man bei Fahrzeugen nicht so auf die Grösse und das Gewicht des Equipments zu achten. Ein Fussgängernavigationssystem macht nur dann Sinn, wenn es den Benutzer nicht in der Bewegung einschränkt. Je kleiner und leichter die Systeme sind, desto grösser ist der Tragekomfort für den Benutzer. Für die visuelle Wegführung ist es jedoch von Vorteil, ein möglichst grosses Display zur Verfügung zu haben. Es gilt also ein adäquates Mittelmass zu finden.

Ausfälle des Positionierungssystems können immer wieder vorkommen und stellen daher ein echtes Problem dar, denn ohne Kenntnis der aktuellen Position können auch keine Navigationsanweisungen gegeben werden. Fahrzeugnavigationssystemen sind relativ gut in der Lage solche Ausfälle zu kompensieren. Auf der Grundlage der Fahrtrichtung und der zurückgelegten Strecke kann auf der Grundlage der letzten bekannten Position die aktuelle Position nachgeführt werden. Dies kann auch bei der Fussgängernavigation geschehen, nur sind die dazu benötigten Daten nicht so präzise zu ermitteln.

Die Gehrichtung kann durch einen elektronischen Kompass ermittelt werden. Da das Gerät aber von einer Person getragen wird, wird es unweigerlich stetig in allen drei Dimensionen bewegt, was die Messwerte verfälscht. Ebenso ungenau sind die Daten über die zurückgelegten Distanzen. Die einzige Möglichkeit der Ermittlung stellt ein Schrittzähler dar. Da aber nicht jeder Schritt gleich lang ist und die Schritte nicht präzise gezählt werden können, weisst das Resultat nicht die gewünschte Präzision auf. Dadurch, dass beide benötigten Werte bei Fussgängern nur sehr ungenau bestimmt werden können, ist die Positionsnachführung schon nach kurzer Zeit so verfälscht, dass sie vom System nicht mehr verwendet werden kann.

Wie wir gesehen haben, kommen bei der Fussgängernavigation, verglichen mit der Fahrzeugnavigation, zahlreiche Faktoren erschwerend hinzu. Insbesondere das zentrale Element der Positionsbestimmung ist betroffen. Indoor- und Outdoor-Anwendung stellen hohe Ansprüche an das verwendete System. Die Bewegungsfreiheit erlaubt viel grössere Spielräume für die aktuelle Position. Nur ungenaue Messwerte für Richtung und Distanz lassen es kaum zu, Ausfälle der Positionierung zu überbrücken und zudem muss das System möglichst handlich realisiert werden, um den Benutzer nicht zu sehr einzuschränken.

2.2 Wegberechnung

Bevor die Wegführung stattfinden kann, muss der optimale Weg mittels eines adäquaten Verfahrens berechnet werden. Optimal kann beispielsweise heissen, auf dem kürzesten oder dem schnellsten Weg.

Zu diesem Zweck wurden bereits zahlreiche Algorithmen entwickelt. Die einzelnen Verfahren unterscheiden sich jedoch sehr stark in ihrer Zielstrebigkeit und somit auch in der Dauer der Ausführung.

Für die Berechnung benötigen die Algorithmen eine geeignete Repräsentation der Umgebung. Die später diskutierten Algorithmen werden allesamt auf gewichtete Graphen angewendet. Die Kanten des Graphen stellen dabei die Wege und die Knoten die Kreuzungen dar. Die Gewichtung der Kante widerspiegelt die Kosten, um vom einen Knoten zum andern zu gelangen. Dies kann beispielsweise die benötigte Zeit oder die Entfernung der beiden Knoten sein.

Je nach Anwendungsbereich ist ein gerichteter oder ein ungerichteter Graph zu wählen. Im Falle der Fussgängernavigation ist ein ungerichteter Graph besser geeignet, um die Situation realistisch zu beschreiben. Bislang sind ja noch keine Einbahnstrassen im Fussgängerverkehr eingeführt worden.

Im Weiteren werden nun der Algorithmus von Dijkstra und der A*- Algorithmus betrachtet. A* ist prädestiniert zur Berechnung des kürzesten Weges zwischen zwei Knoten in einem Graphen und baut auf der Grundstruktur von Dijkstra auf. Ebenfalls werden einige Verfahren betrachtet, um die extrem grossen Graphen bei der Wegberechnung im Strassennetz zu vereinfachen, ohne Einbussen bei der Güte der Lösung in Kauf nehmen zu müssen.

2.2.1 Der Algorithmus von Dijkstra

Der Algorithmus von Dijkstra ist einer der bekanntesten zur Berechnung von kürzesten Wegen in Graphen. Erstmals beschrieben wurde er in [Dij59].

Ausgehend vom Startpunkt werden die kürzesten Wege zu allen Knoten im Graph berechnet. Schrittweise arbeitet er die einzelnen Knoten des Graphen ab, bis letztendlich alle berücksichtigt wurden. Die Reihenfolge wird durch die Distanz zum Startknoten bestimmt, wobei jeweils der nächstgelegene, noch nicht bearbeitete Knoten berücksichtigt wird.

Zyklen im Graphen oder ähnliches sind für den Algorithmus kein Problem. Die einzige Einschränkung des Graphen, ist die, dass der Graph keine negativ gewichteten Kanten enthalten darf. Eine Erklärung, wieso negative Kanten mit dem Algorithmus von Dijkstra nicht verträglich sind, abgesehen davon, dass negative Wegstrecken ohnehin nicht intuitiv sind, wird später in diesem Kapitel abgegeben.

Der Algorithmus von Dijkstra vernachlässigt sämtliche redundanten Wege und erstellt einen minimal spannenden Baum⁶ für den Graphen. Während der Ausführung des Algorithmus wächst der Baum kontinuierlich, denn Dijkstra betrachtet nicht alle Knoten auf einmal. Er betrachtet jeweils nur die Nachbarn der bereits fest dem Baum zugeordneten Knoten. Somit werden zur Laufzeit drei Arten von Knoten unterschieden. Knoten, die bereits fest dem spannenden Baum zugewiesen wurden, Peripherieknoten, also Nachbarknoten der Knoten des Baumes sowie Knoten, die dem Algorithmus noch nicht bekannt sind. Am Ende des Algorithmus sind alle Knoten des ursprünglichen Graphen dem spannenden Baum hinzugefügt und somit die kürzesten Wege vom Startknoten zu allen Knoten bekannt. Unter [DijAp] ist ein Java Applet zu finden, in dem die Funktionsweise des Algorithmus von Dijkstra schön zu sehen ist.

Grundsätzlich berechnet Dijkstra die kürzesten Wege zu allen anderen Knoten. Er kann aber auch angewendet werden, wenn der kürzeste Weg zu einem spezifischen Ziel gesucht wird. Dazu wird dem Algorithmus einfach eine Abbruchbedingung hinzugefügt, die erfüllt ist, wenn das explizite Ziel gefunden wurde.

Bei jedem Schritt wird der kürzeste Weg zu einem Knoten des Graphen gefunden und der Knoten somit abschliessend betrachtet. Dabei werden die Distanz zum Startpunkt, sowie der Knoten, über den er erreicht wurde, festgehalten. Dadurch kann der kürzeste Weg zu diesem Knoten rekursiv ermittelt werden. Zudem werden sämtliche Nachbarknoten zur Peripherie zugefügt, sofern sie nicht schon enthalten sind.

Die Knoten der Peripherie werden nach ihrer Entfernung zum Startpunkt sortiert und jeweils der nächstgelegene Knoten als nächstes berücksichtigt. Dadurch wird sichergestellt, dass jeweils der kürzeste Weg zu einem Knoten als erstes entdeckt wird und der Knoten definitiv

⁶ Teilgraph, der alle Knoten des Graphes enthält, keine Zyklen aufweist und die Summe seiner Kanten minimal ist im Vergleich zu allen andern spannenden Bäumen dieses Graphen.

in den Baum übernommen werden kann. Dies ist auch die Begründung, wieso keine negativen Gewichtungen der Kanten zugelassen sind. Denn durch negativ gewichtete Kanten könnten Umwege trotzdem kürzer sein.

Nun folgt der Algorithmus von Dijkstra als Pseudo Code Implementierung. Der Funktion werden der Graph (G) und der Startknoten (v) übergeben. Als Rückgabe liefert sie einen Array mit den Längen der kürzesten Wege zu allen Knoten im Graphen (G), ausgehend vom Knoten (v).

```
ShortestPath(G,v)

  init D array entries to infinity
  D[v]=0

  add all vertices to priority queue Q

  while Q not empty do
    u = Q.removeMin()
    for each neighbor, z, of u in Q do

      ifd[u] + w(u,z) < D[z] then
        D[z] = D[u] + w(u,z)
        Change key of z in Q to D[z]

  return D as shortest path lengths
```

Abbildung 1: Dijkstra Pseudo Code [Coh04]

2.2.2 A*

Ein weiterer interessanter Algorithmus mit Hinblick auf das Auffinden von kürzesten Wegen, ist der A* (A-Star)-Algorithmus. Die Beschreibung dazu folgt den Erläuterungen aus [Rus03]. In den Grundzügen stimmt er mit dem Algorithmus von Dijkstra überein, sie unterscheiden sich aber in der Auswahl des als nächstes zu betrachtenden Knoten. A* ist auf das Auffinden des kürzesten Weges zu einem bestimmten Ziel ausgerichtet und versucht dabei zielstrebig ans Ziel zu kommen.

A* ist ein so genannter informierter Suchalgorithmus. Das heisst, dass zur Suche eine Heuristik beigezogen wird. Heuristik deswegen, weil nicht mit hundertprozentiger Sicherheit garantiert werden kann, dass durch die zusätzlichen Informationen das Problem auch optimal gelöst wird. Die Verwendung der Heuristik zielt auf das schnellere Ablaufen der Suche ab und nimmt dafür Optimalitätsverluste der Lösung in Kauf. Durch eine geschickte Wahl der Heuristik können diese aber minimiert werden.

Wie auch Dijkstra unterscheidet A* zwischen drei Arten von Knoten. Den bearbeiteten Knoten, den Knoten der Peripherie und den noch nicht aufgefundenen Knoten. Der Unterschied liegt in der Abarbeitung der Knoten der Peripherie. A* macht die Auswahl nicht wie Dijkstra anhand der Entfernung zum Startpunkt, sondern anhand einer Evaluationsfunktion. Es wird derjenige Knoten als nächstes abgearbeitet, der den niedrigsten Wert der Evaluationsfunktion hat. Diese Evaluationsfunktion setzt sich wie folgt zusammen:

$$f(n) = g(n) + h(n)$$

Wobei $g(n)$ die tatsächlichen Kosten bis zum jeweiligen Knoten sind. Da der Weg bis zu diesem Knoten ja bereits ermittelt wurde, handelt es sich dabei um einen effektiven Wert. $h(n)$ ist die Heuristikfunktion. Sie bezieht sich auf die noch anstehenden Kosten von diesem Knoten bis zum Ziel. Da der restliche Weg aber noch nicht bekannt ist, ist dies nur eine Annahme und kein effektiver Wert. Je besser die Heuristik auf die jeweilige Problemstellung abgestimmt ist, desto schneller und effizienter kommt der Algorithmus zum Ziel.

Eine sinnvolle Heuristik, im Hinblick auf das Berechnen von kürzesten Wegen zwischen zwei Punkten, ist beispielsweise die horizontale Entfernung des Knotens zum Zielpunkt. Es wird also nicht kreisförmig vom Startpunkt aus gesucht. Der Suchbereich weitet sich also vom Startpunkt ausgehend immer mehr in Richtung Zielpunkt aus. Dies ist in der Abbildung 2 graphisch veranschaulicht. Dadurch erreicht man eine deutliche Reduktion der bearbeiteten Knoten und somit eine Verkürzung der Suche. Die Knoten Z und O in der Abbildung 2 wären bei der Suche nach dem Algorithmus von Dijkstra noch vor S behandelt worden, bei A^* werden sie überhaupt nicht betrachtet.

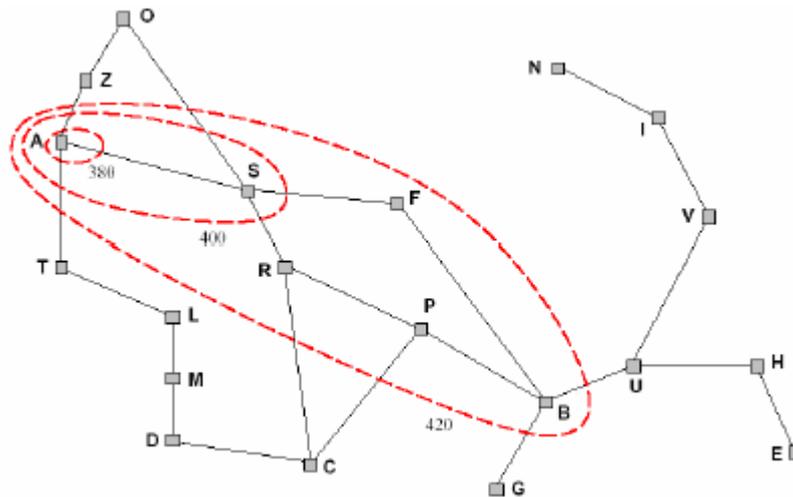


Abbildung 2: Suchverlauf bei A^* [Rus03]

2.2.3 Routenberechnung im Strassenverkehr

Auch wenn Algorithmen wie der Algorithmus von Dijkstra und A^* die Dauer der Suche massiv verkürzen, sind sie dennoch zu langsam für eine effiziente Berechnung von Wegen im Strassennetz. Wenn man sich beispielsweise einen Weg von Zürich nach Barcelona vorstellt, so ist die Anzahl von Strassen und Kreuzungen, die berücksichtigt werden müssen, gewaltig. Diese riesige Menge von Knoten und Kanten ist so nicht in einer vernünftigen Zeit zu bearbeiten. Es bedarf also weiterer Vereinfachungen, um die Suche zu optimieren. Die hier beschriebenen Optimierungen folgen den Erläuterungen von Prof. Sanders in [San03].

Ein erster Ansatz ist die bidirektionale Suche. Dabei wird gleichzeitig vom Start, wie auch vom Ziel aus gesucht. Beendet wird die Suche nicht erst, wenn der Ziel- respektive der Startpunkt erreicht wurden, sondern wenn die beiden aufgespannten Bäume ein gemeinsames Blatt haben. Dadurch kann die Fläche des abgesuchten Terrains erheblich verkleinert werden. Die unidirektionale Suche weitet sich mit zunehmender Entfernung vom Startpunkt auch immer weiter in die Breite aus. In der Abbildung 3 ist dieser Unterschied zwischen der unidirektionalen und der bidirektionalen Suche nach dem Algorithmus von Dijkstra

aufgezeigt. Dieser Sachverhalt tritt natürlich auch bei A* auf, jedoch fällt er dort nicht ganz so extrem aus, da A* den Suchraum seinerseits schon verbessert.

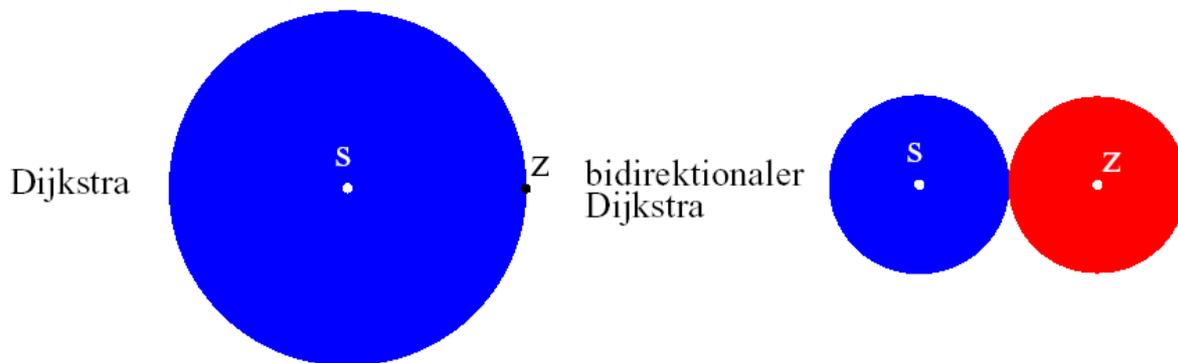


Abbildung 3: Verbesserung des Suchraumes durch bidirektionale Suche [San05]

Ein weiterer Ansatz besteht in der Aufteilung des Suchraumes in unterschiedliche Hierarchiestufen. Je nach dem zu welcher Hierarchiestufe ein Bereich gehört, werden in ihm nur bestimmte Kategorien von Strassen berücksichtigt. Haupt- und Bundesstrassen bilden die höchste Hierarchiestufe, Quartiers- und Nebenstrassen die tiefste. Mit zunehmender Distanz vom Start- bzw. Zielpunkt werden nur noch Strassen höherer Stufe mit in die Suche einbezogen. Je höher die verwendete Stufe, desto geringer die Anzahl zu betrachtender Knoten und Kanten und somit auch desto kürzer die Zeit zur Berechnung.

Im lokalen Bereich, um den Start- und den Zielpunkt ist jedoch der Einbezug aller Hierarchiestufen unabdingbar für die punktgenaue Navigation. Meist liegen der Start und das Ziel an Strassen einer tiefen Stufe. Da die tiefste Stufe aber nur in kleinen Bereichen zum Einsatz kommt, wird die absolute Zahl zu berücksichtigender Kanten und Knoten nicht erheblich vergrössert.

In den grossen Bereichen zwischen Start und Ziel kann man sich auf Haupt- und Bundesstrassen beschränken. Dadurch wird die Suche wesentlich effizienter, ohne aber das Ergebnis zu beeinträchtigen.

Mittels bidirektionaler Suche wird der Suchraum an sich verkleinert und durch die Verwendung von Hierarchiestufen die Anzahl Knoten und Kanten in diesem reduziert.

Bevor A* oder ein anderer Suchalgorithmus überhaupt zur Anwendung kommt, wird das Objekt der Suche, der Graph, auf ein Minimum reduziert. Dadurch erreicht man, dass trotz der immensen Anzahl von möglichen Wegen der Algorithmus innerhalb nützlicher Zeit den optimalen oder einen nahezu optimalen Weg ermittelt.

Abschliessend zur Wegberechnung kann man sagen, dass einige sehr effektive, optimale Algorithmen zur Berechnung von kürzesten Wegen bestehen. In sehr grossen Suchräumen dauert die Suche jedoch zu lange. Massnahmen zur Einschränkung der Suchräume sind daher unumgänglich. Solche Einschränkungen bringen aber zwangsläufig Verschlechterungen der Resultate mit sich. Es gilt also das Optimum zwischen Suchdauer und dem optimalen Ergebnis zu finden.

2.3 Präsentation der Weginformation

Für die Navigation ist eine gute Präsentation der Weginformation von zentraler Bedeutung. Jedes noch so intelligente System zur Wegberechnung wird hinfällig, wenn dem Benutzer der berechnete Weg nicht in verständlicher Art und Weise mitgeteilt werden kann.

Grundsätzlich könnten alle unsere fünf Sinne verwendet werden, um die Informationen zu übermitteln. Normalerweise beschränkt man sich aber auf die beiden am besten ausgeprägten Sinne: Das Hören und das Sehen. Bei handicaperten Personen, wie Blinden, wird der Tastsinn alternativ zum Sehen benutzt.

Egal auf welche Art die Information an den Benutzer weitergegeben wird, sie muss für ihn verständlich sein und darf ihn nicht überfordern. Der Grundsatz dabei lautet: „So viel wie nötig und so wenig wie möglich.“ Zuviel Information lenkt vom Wesentlichen ab, da wir nur beschränkt aufnahmefähig sind. George A. Miller schreibt in [Mill55], dass der Mensch nur 5 ± 2 so genannte Chunks⁷ im Kurzzeitgedächtnis speichern kann.

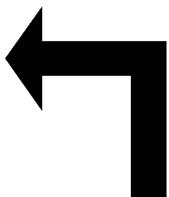
Primär bedarf der Benutzer der Richtungsangaben, anhand denen er das Ziel erreichen kann. Sekundär können noch weitere Informationen präsentiert werden, die das Erreichen erleichtern. Unabdingbar ist nur die einzuschlagende Richtung. Alle weiteren Informationen sind nur sekundär. Solche sekundären Informationen sind beispielsweise die generelle Richtung des Ziels, die Distanz, bis wieder eine Richtungsänderung zu erfolgen hat oder die Distanz bis zum Ziel.

2.3.1 Visuelle Präsentation

Die visuelle Präsentation ist sehr effektiv, da Bilder auf sehr prägnante Art und Weise Information vermitteln können. Die Möglichkeiten sind natürlich stark abhängig von den zur Verfügung stehenden Ausgabegeräten. Die in Kapitel 2.4 betrachteten Systeme verwenden dazu beispielsweise PDAs oder Durchsichtsbildschirmen.

Nun folgen einige Beispiele zur visuellen Präsentation von Weginformationen:

Richtungsangaben (primär):



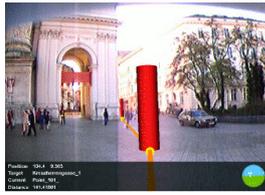
Richtungsangaben können in der Form von einfachen Pfeilgrafiken gemacht werden. Sie sind leicht verständlich und dennoch sehr informativ. Kompliziertere Abzweigungen an Kreuzungen vieler nicht rechtwinklig verlaufender Strassen stellen auch kein Hindernis dar.



[Dag06]

Hier ist der gesamte Weg auf einer Karte visualisiert. Sie enthält somit alle Informationen, die für den Benutzer nötig sind, um ans Ziel zu kommen. Hier muss sich der Benutzer jedoch zuerst orientieren, die Informationen können also nicht besonders schnell abgerufen werden. Informationen über die aktuelle Position und Ausrichtung erleichtern jedoch die Orientierung.

⁷ Ein Chunk entspricht einer Information, die wir in unserem Gedächtnis festhalten. Ein Chunk kann beispielsweise eine Telefonnummer oder ein Name sein.



[Rei03]

Hier zeigen Zylinder dem Benutzer den Weg an. Mit Hilfe einer Durchsichtsbrylle werden sie scheinbar in der realen Welt positioniert. Diese Methode ist sehr leicht verständlich. Der Benutzer muss lediglich den orangen Linien von Zylinder zu Zylinder folgen. Das in Kapitel 2.4.4 betrachtete System Mobile Collaborative Augment Reality verwendet diese Methode.

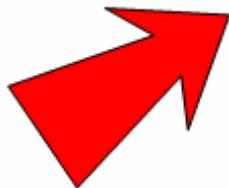
Weitere Informationen (sekundär):



Gerade für Fussgänger ist es sehr schwer abzuschätzen, wie weit nun wirklich 300m sind. Daher ist ein Distanzbalken eine gute Variante, die Distanz zu vermitteln. Anfangs ist der Balken leer und füllt sich langsam auf. Hat man die volle Distanz zurückgelegt, ist der Balken voll.

540 m bis ans Ziel

Die gesamte Distanz bis zum Ziel wird meist textuell dargestellt. Diese Form der Anzeige vermittelt nicht die gleiche Dichte an Information wie der Distanzbalken, ist aber für die Relevanz der Distanz zum Ziel vollkommen ausreichend.



Die allgemeine Richtung des Ziels kann ähnlich wie ein Kompass realisiert werden. Anstelle der Nordrichtung wird einfach die Richtung ans Ziel angezeigt. Für das Erreichen des Ziels ist sie nicht von Relevanz.

2.3.2 Akustisch

Die akustische Wegführung ist für den Benutzer um einiges schwieriger als die optische. Das liegt daran, dass die natürliche Sprache nicht so präzise ist wie graphische Darstellungen. Was bedeutet beispielsweise: „Jetzt bitte links abbiegen.“?

An einer Kreuzung, bei der sich zwei Strassen rechtwinklig schneiden, ist klar, was gemeint ist. Was aber, wenn linkerhand mehr als eine Strasse weg geht. Graphisch wäre es leicht zu zeigen, welche Strasse gemeint ist, akustisch aber nicht.

Auch wenn es grundsätzlich möglich wäre, sollte die akustische Wegführung nicht autonom verwendet werden, sie ist einfach nicht genug mächtig. In Kombination mit graphischer Wegführung kann sie jedoch sehr effektiv und gewinnbringend eingesetzt werden.

Gerade im Strassenverkehr ist es wichtig, dass der Benutzer seinen Blick auf der Strasse hat und nicht auf dem Anzeigegerät. In einfachen Situationen kann sich der Benutzer akustisch führen lassen. Dies ist beispielsweise der Fall, wenn er einfach für eine Weile der Strasse folgen muss. Die Ansage würde dann in etwa wie folgt lauten: „Dem Strassenverlauf weiter folgen.“

Nur bei komplizierteren Abzweigmanövern ist es für den Benutzer nötig, sich von der Umgebung abzuwenden und das Anzeigegerät zu betrachten.

Für die Vermittlung von Information ist die graphische Anzeige wesentlich besser geeignet als die akustische. Durch die hohe Informationsdichte neigt sie aber auch dazu, die volle Aufmerksamkeit des Benutzers auf sich zu ziehen. Optimal ist daher die Kombination von graphischen und akustischen Elementen, um den Benutzer nicht zu binden aber dennoch auf direktem Weg ans Ziel zu führen.

2.4 Verwandte Arbeiten

Auch wenn es sich bei der Personennavigation um einen eher jungen Forschungsbereich handelt, existieren bereits die verschiedensten Ansätze. Die markantesten Unterschiede der einzelnen Realisierungen sind im Bereich der Positionsbestimmung und der Benutzerschnittstellen zu finden. Bei der Positionsbestimmung geht das Spektrum von GPS und GSM für die Positionsbestimmung im Freien bis zu WLAN und Infrarot zur Positionsbestimmung innerhalb von Gebäuden. Bei den Benutzerschnittstellen werden Ansätze wie PDAs als Aus- wie auch Eingabemedium oder aber auch Bedienplatten und Durchsichtsbildschirme verwendet. In den Kapiteln 2.4.1 bis 2.4.6 werden verschiedene Systeme betrachtet, die einen groben Überblick über die in diesem Bereich verfolgten Ansätze geben sollen.

2.4.1 REAL: Ein ressourcenadaptiertes mobiles Navigationssystem

Bei mobilen Systemen kann die Verfügbarkeit von Ressourcen stark variieren. Je nach Aufenthaltsort können die Positions- und Umgebungsinformationen sehr detailliert oder aber auch nur sehr spärlich sein. Ebenso sind die Ressourcen des Benutzers nicht konstant. Ein Tourist, der gemütlich durch die Stadt flaniert, ist viel aufnahmefähiger als ein Geschäftsmann, der unter Zeitdruck von einem Ort zum andern eilt und dabei noch telefoniert. REAL versucht diesen Faktoren gerecht zu werden und dementsprechend den Output für den Benutzer zu generieren. Beispielsweise nimmt REAL bei eingeschränkter Aufmerksamkeit des Benutzers kleinere Umwege in Kauf, wenn dadurch komplexe Weggabelungen umgangen werden können.

Die Beschreibung des Systems REAL stützt sich auf die beiden Publikationen [Wah01] und [Bau02] ab. Entwickelt wurde das System an der Universität Saarbrücken und umfasst die beiden Subkomponenten ARREAL und IRREAL. ARREAL übernimmt die Positionierung und die Navigation im Freien und IRREAL innerhalb von Gebäuden. REAL vereint aktive wie auch passive Lokationssensitivität⁸ so, dass es für den Benutzer verborgen bleibt.

Durch das Zusammenspiel der Subkomponenten soll REAL eine möglichst flächendeckende Navigation ermöglichen.

⁸ Aktiv bedeutet, dass das System seine Position selbst bestimmen kann. Bei passiven Systemen bestimmt eine externe Instanz die Position und übermittelt sie dem System.

2.4.1.1 IRREAL

IRREAL deckt den Bereich der Navigation innerhalb von Gebäuden ab. Als Schnittstelle für den Benutzer dient ein Handheld (PalmOS-Gerät). Dieses hat ein angemessenes Verhältnis zwischen Handlichkeit und Bildschirmgröße. Zudem verfügt es bereits über die für die Positionierung benötigte Infrarot-Schnittstelle. Diese empfängt die Signale der an der Decke montierten Sender. Durch die Reichweite der Deckensender von bis zu 20 Metern ist eine angemessene Raumabdeckung gewährleistet. Die Kommunikation kann jedoch nur unidirektional, ähnlich dem Videotext, geschehen. Informationen an das System, wie beispielsweise das gesuchte Ziel, müssen dem System vor dem Beginn der Navigation separat übermittelt werden.

Die Deckensender werden kontinuierlich vom Server mit den nötigen Informationen versorgt, somit können auch zeitabhängige Informationen übermittelt werden. Übertragen werden interaktive Inhalte in der Form eines Präsentationsgraphen. Dabei kann ein Knoten Text wie auch Grafiken enthalten. Ebenso können in einem Knoten interaktive Bereiche vorhanden sein, die wieder auf einen Knoten zeigen. Nicht alle Knoten werden gleich häufig übertragen. Je höher die Stufe des Knotens ist, desto häufiger wird er übertragen. Durch diese Hierarchie wird die Präsentation abhängig von der Geschwindigkeit des Benutzers. Abbildung 4 zeigt diesen Sachverhalt auf.

Der Pfeil als simple Richtungsangabe ist in der Hierarchie am höchsten und wird dem Benutzer somit als erster dargestellt (Abbildung 4a). Ist die Geschwindigkeit des Benutzers tief, hält er sich länger in einem Bereich auf, wodurch er nach und nach zusätzliche Informationen erhält. Zunächst ist dies der Hilfefknopf wodurch die Anzeige erweitert werden kann (Abbildung 4b). Zuletzt erhält der Benutzer noch ortsspezifische Informationen (Abbildung 4c).

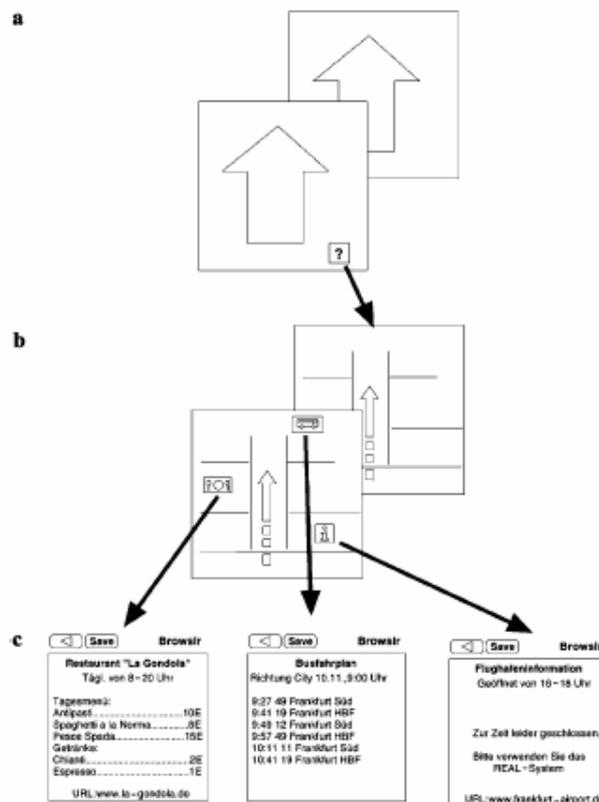


Abbildung 4: Präsentationsbaum in IRREAL [Wah01]

2.4.1.2 ARREAL

ARREAL übernimmt die Navigation im Freien. Dabei wird aktive Lokationssensitivität verwendet, die Intelligenz zur mobilen Navigation ist also im System selbst enthalten. Mitunter deswegen sind die Hardwarekomponenten umfangreicher als bei IRREAL. Das Herz des Systems bildet Notebook (Sony Vaio C1XN), das die nötigen Berechnungen vornimmt. Die Positionierung wird durch ein handelsübliches GPS-Handgerät und einem magnetischen Tracker realisiert. Der magnetische Kompass ist über ein Kabel mit dem System verbunden. Durch zwei zusätzlich angebrachte Knöpfe fungiert er auch als Maus zur Interaktion des Benutzers mit dem System. Für die visuelle Ausgabe steht ein Brillendisplay zur Verfügung, das 640x320 Pixel aufweist.

Trotz alledem ist der Benutzer nicht zu stark in seiner Bewegungsfreiheit eingeschränkt. Das Brillendisplay wird analog zu einer normalen Brille, der magnetische Kompass/Maus in der Hand oder am Gürtel getragen, das Notebook und der GPS-Empfänger sind in einem kleinen Rucksack verstaut. In Abbildung 5 ist ein Benutzer mit den einzelnen Komponenten abgebildet.

In ARREAL stehen zwei Perspektiven zur Verfügung: Vogel- und Egoperspektive. Die Vogelperspektive ermöglicht dem Benutzer eine gute Orientierung, wohingegen ihm die Egoperspektive detailliertere Ansichten auf Gebäude oder ähnlichem erlaubt. Graphische und textuelle Annotationen, wie Strassennamen oder kleine Abbildungen von Landmarken, sind ein- und ausblendbar.

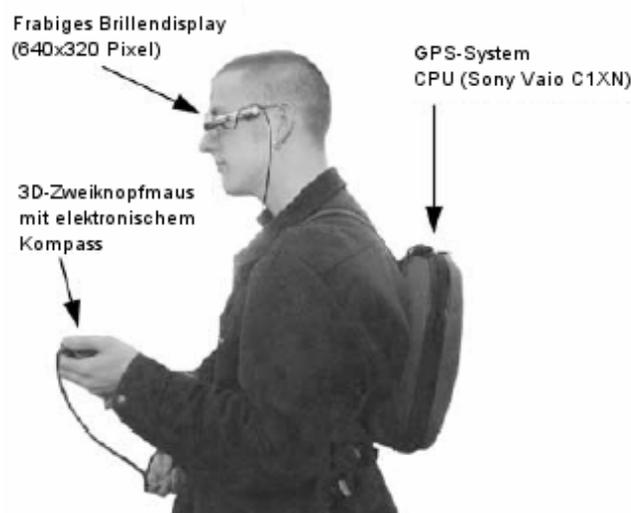


Abbildung 5: Die ARREAL-Systemkomponenten [Wah01]

Abhängig von den verfügbaren Ressourcen wählt das System die Perspektive aus. Die Egoperspektive ist nur sinnvoll, wenn die dazu benötigten Informationen über die Umgebung und die Peilung des Benutzers in ausreichender Form vorhanden sind. Wenn beide Perspektiven vorhanden sind, kann die Perspektive vom Benutzer manuell gewechselt werden. Die Anzeige der Vogelperspektive wird vom System in Abhängigkeit von der Genauigkeit der vorhandenen Informationen und der Geschwindigkeit des Benutzers erstellt. Abbildung 6 zeigt die unterschiedlichen Anzeigen. Je genauer die aktuelle Position bekannt ist, desto kleiner ist der anzeigende Kreis. Schön zu sehen ist der Unterschied in der Abbildung 6a und Abbildung 6c. Nimmt die Geschwindigkeit zu, so wird automatisch der angezeigte Kartenausschnitt vergrößert und die Gebäudebeschreibungen am Rande werden nicht mehr angezeigt. Dies ist in Abbildung 6a und Abbildung 6b zu sehen.

Die so auf die Ressourcen angepasste Anzeige erlaubt es dem Benutzer, sich bestmöglich zu orientieren. Als Navigationshilfe wird dem Benutzer die zu gehende Richtung, wie auch der gesamte Weg, angezeigt. Gut zu sehen ist dies in der Abbildung 6c. Der Pfeil in der linken, unteren Ecke signalisiert dem Benutzer die zu gehende Richtung. Der von der aktuellen Position ausgehende Pfeil bezeichnet den Weg ans Ziel.

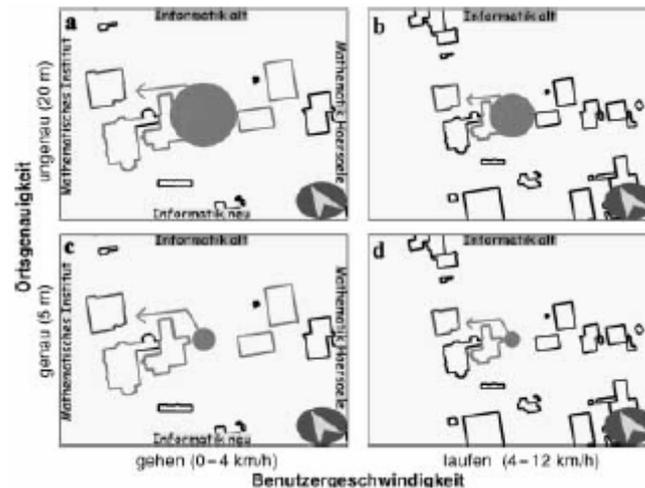


Abbildung 6: Varianten in der Anzeige bei ARREAL [Wah01]

2.4.1.3 Übergang zwischen den beiden Komponenten

Normalerweise liegt der Weg an ein Ziel nicht ausschliesslich in einem Gebäude oder im Freien. Meistens werden beide Bereiche tangiert. Deshalb ist es nötig, dass der Übergang von ARREAL zu IRREAL oder umgekehrt reibungslos und für den Benutzer möglichst verborgen vonstatten geht.

Dazu verwenden ARREAL und IRREAL eine gemeinsame Wissensbasis von Präsentationstechniken. Zudem überlappen sich ihre Modelle der Umgebungen. Die Überlappungen bestehen im Bereich der Gebäude. ARREAL hält Gebäude als einzelne räumliche Entitäten auf einer Karte fest, IRREAL hingegen als 3D-Modelle der Innenräume. Der Übergang von einem zum anderen System läuft automatisch ab. Befindet sich der Benutzer im Sendebereich eines Infrarotsenders, so werden die Daten auf dem Handheld oder dem Brillendisplay angezeigt. Durch die Verwendung eines Präsentationsserver werden diese unterschiedlichen Anzeigemedien unterstützt. Der Server stellt verschiedene, für die einzelnen Anzeigegeräte optimierte, Präsentationen bereit. Wird das Gebäude wieder verlassen, wird das GPS-Signal empfangen und das System übernimmt die Positionierung wieder selbst. Anzeigen sind nur noch auf dem Brillendisplay möglich. Kurzzeitig besteht zwischen ARREAL und dem Präsentationsserver eine Funkverbindung. Dabei werden die Informationen über das Anzeigegerät und das Navigationsziel übermittelt.

REAL ist mit seinen beiden Subkomponenten ARREAL und IRREAL ein sehr interessantes System. Der für den Benutzer mehr oder weniger verborgene Übergang zwischen Indoor- und Outdoorsystem macht das System sehr praktikabel und umfangreich einsetzbar. Zudem verfolgt REAL mit der benutzergerechten Informationsanzeige einen sehr interessanten Ansatz, der vom Prinzip her relativ simpel ist, dadurch aber umso wirkungsvoller.

2.4.2 Deep Map

Deep Map ist ein Touristeninformationssystem der Stadt Heidelberg. Es soll dem Benutzer die Navigation in der fremden Stadt erleichtern und ihm historisches wie auch aktuelles Wissen über Heidelberg vermitteln.

Das Deep Map Projekt ist eine Kooperation der European Media Laboratory und des Geographischen Instituts der Universität Heidelberg.[Zip01]

2.4.2.1 Systemüberblick

Das gesamte System von Deep Map setzt sich aus zahlreichen einzelnen Komponenten zusammen, wie beispielsweise der Komponente zur Routenplanung oder zur Darstellung der Karte. Realisiert wurden die einzelnen Komponenten als Agenten und sind somit weitgehend unabhängig voneinander und dementsprechend leicht austauschbar.

Jeder Agent ist innerhalb des Systems einer der drei logischen Schichten von Deep Map zugeordnet. Die drei logischen Schichten sind: der Interface Layer, der Cognitive Layer und der Knowledge Layer. Abbildung 7 zeigt eine vereinfachte Sicht der Systemarchitektur mit den 3 Schichten.

Der **Interface Layer** enthält die Komponenten zur Interaktion mit dem Benutzer. Er ermöglicht es dem Benutzer, Eingaben an das System zu machen, wie auch dem System, Informationen an den Benutzer weiterzugeben. Die Eingaben des Benutzers können über eine graphische Schnittstelle wie auch per Sprachsteuerung geschehen.

Der **Cognitive Layer** wandelt die Eingaben des Benutzers in für die Basiskomponenten verständliche Anfragen um. Zudem werden die Ausgaben an den Benutzer, wie beispielsweise die Übersichtskarte, aufbereitet.

Der **Knowledge Layer** beinhaltet die grundlegende Funktionalität von Deep Map. Er umfasst unter anderem die Wissensdatenbank, den Tourenplaner und externe Dienste. [Zip99a]

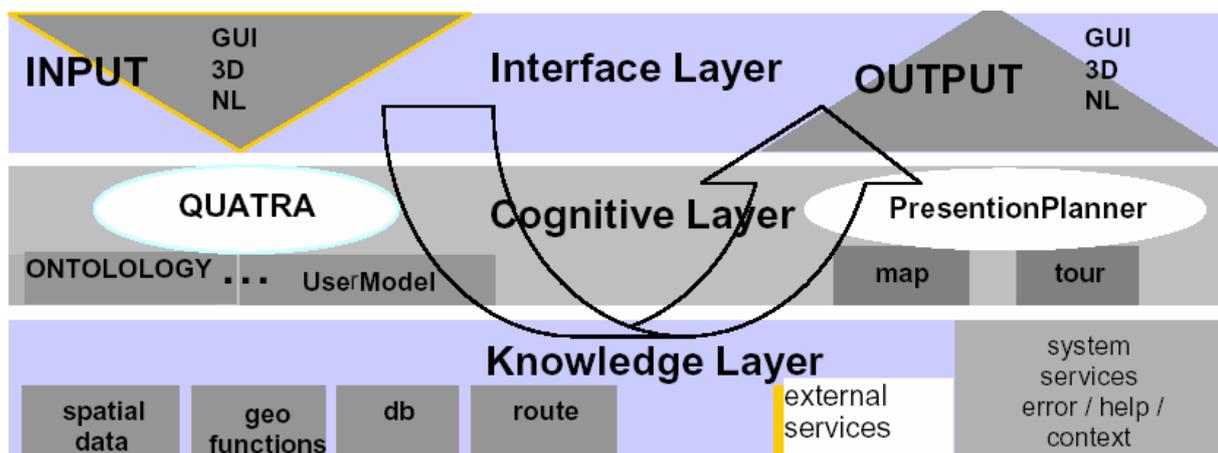


Abbildung 7: Ebenensichtweise der Gesamtarchitektur [Zip99]

2.4.2.2 Hardwarekomponenten

Das Herzstück von Deep Map bildet ein wearable Computer. Dieser stellt die nötige Rechenleistung zur Verfügung, ist aber dennoch handlich genug um am Gürtel getragen werden zu können. Abbildung 8 zeigt, wie die Hardwarekomponenten von Deep Map verwendet werden.

Zur Interaktion mit dem System sind zwei Komponenten vorhanden. Zum einen ein LCD-Display und zum andern ein Headset bestehend aus Kopfhörern und Mikrophon. Auf dem Display werden dem Benutzer diverse Informationen präsentiert, zudem bietet es ihm die Möglichkeit, Eingaben an das System über die graphische Benutzeroberfläche zu tätigen.

Das Headset ist die akustische Schnittstelle zwischen dem System und dem Benutzer. Durch dies können Sprachbefehle an das System und akustische Wegbeschreibungen an den Benutzer übermittelt werden.

Da Deep Map dem Benutzer nicht nur Informationen bereitstellt, sondern ihn auch auf seinem Rundgang führt, ist die Kenntnis der aktuellen Position zwingend nötig. Deep Map verwendet dazu einen GPS Empfänger. [Zip99a]

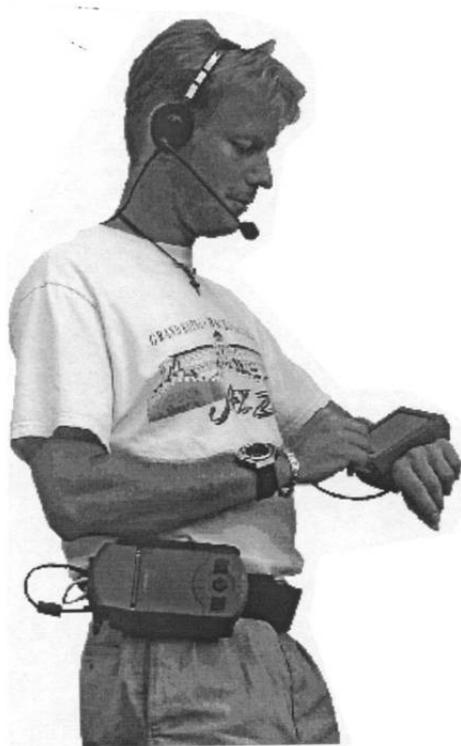


Abbildung 8: Hardwarekomponenten von Deep Map [Zip99a]

2.4.2.3 Deep Map Tourenplaner

Deep Map soll den Benutzer nicht nur auf einer von ihm festgelegten Tour durch die Stadt führen, sondern soll ihn auch bei der Findung der für ihn optimalen Route unterstützen. Diese Aufgabe übernimmt der Tourenplanungs-Agent.

Von diesem Agenten gibt es zwei Versionen. Eine Web basierte und eine mobile Version.

Die Web basierte Version soll den Touristen insbesondere bei ihren Reisevorbereitungen helfen. Sie erlaubt es ihnen sich vorab einen ersten Überblick zu verschaffen und sich schon vor ihrem Besuch zu informieren. Ausführlich beschrieben ist die Web basierte Version in [Zip99b].

Die mobile Version soll dem Benutzer dann vor Ort beiseite stehen, ihm helfen, die geeignete Route zu finden und ihn sicher auf dem Weg führen.

Anhand der vom Benutzer eingegebenen Präferenzen berechnet der Agent die geeignete Route. Beim Erstellen der Route werden sowohl harte wie auch weiche Faktoren berücksichtigt. Harte Faktoren, wie beispielsweise die maximale Dauer der Tour, werden strikte eingehalten, weiche so gut wie möglich. Explizit vom Benutzer gewünschte Ziele werden natürlich fest in die Tour aufgenommen. Der Agent berücksichtigt auch, dass manche Sehenswürdigkeiten mehr Zeit in Anspruch nehmen als andere. Schlussendlich werden dem Benutzer verschiedene Touren vorgeschlagen von denen er eine auswählen kann. [Zip99b]

Neben dem Generieren von massgeschneiderten Routen stellt der Agent noch weitere Funktionen zur Verfügung:

Routenfindung: Sucht den Weg vom aktuellen Standort zu einem angegebenen Ziel.

Standortsuche: Das nächstgelegene Objekt einer bestimmten Kategorie finden, wie beispielsweise ein Restaurant.[Zip01]

Das Resultat der Berechnung übergibt der Tourenplanungs-Agent dem Map-Agenten, der die entsprechende Karte mit den erhaltenen Informationen anreichert und sie dem Präsentations-Planer zur Anzeige übergibt. Abbildung 9 zeigt eine mögliche Visualisierung der Karte des Heidelberger Schloss mit hervorgehobenem Tourabschnitt. [Zip99]

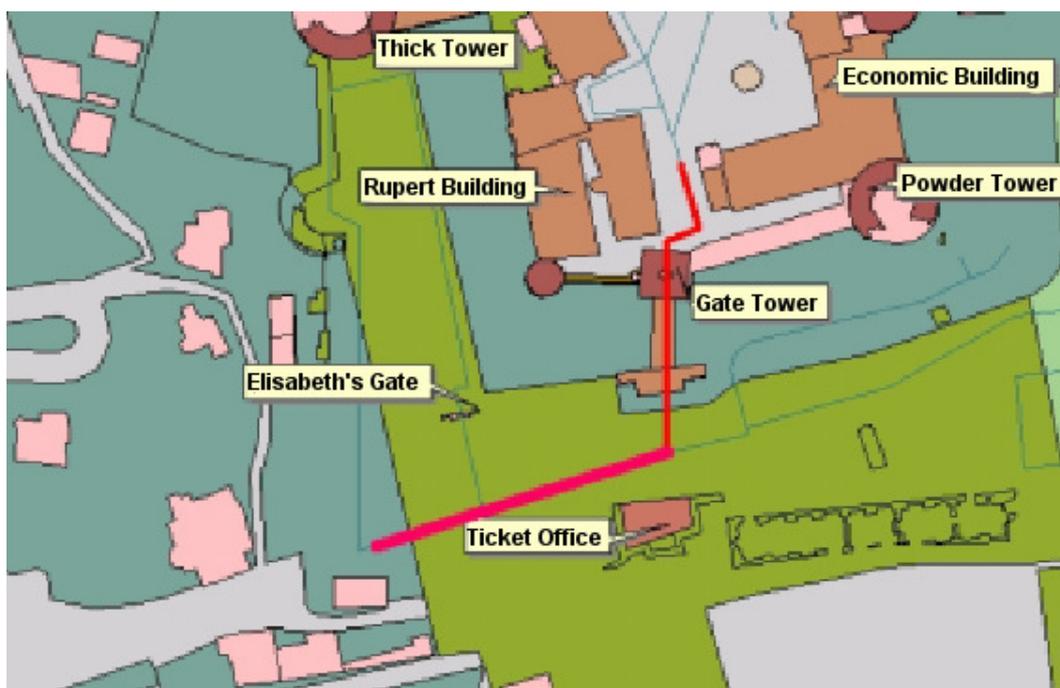


Abbildung 9: Tourenabschnitt vom Deep Map [Zip99]

Dadurch, dass der Benutzer bei der Routenplanung nicht alle Ziele explizit angeben muss, sondern sie seinen Präferenzen entsprechend ausgewählt werden, ist es dem Benutzer möglich, interessante Orte zu besichtigen, die ihm ohne Deep Map verborgen geblieben wären.

Da das System nichts über die Ausrichtung des Benutzers weiss, muss der Benutzer immer wieder für sich die Umgebung mit der angezeigten Karte abgleichen, um auf dem Weg zu bleiben. Dies ist aber nicht zwangsläufig schlecht, denn dadurch fixiert er sich nicht zu sehr auf das Gerät und verpasst so nicht den Grossteil seiner Tour.

2.4.3 INSTAR

INSTAR steht für Information and Navigation System Through Augmented Reality. Es ist aber mehr als nur ein Navigationssystem, das Augmented Reality verwendet. INSTAR versucht sich auf zwei Arten von anderen Navigationssystemen zu unterscheiden. Zum einen macht es den Brückenschlag zwischen Fahrzeug- und Fussgängernavigationssystemen. Es leitet den Benutzer sowohl auf der Fahrt bis zum ungefähren Ziel, wie auch auf dem Fussweg zum exakten Ziel. Zum anderen wird der Benutzer nicht auf die herkömmliche Art und Weise geführt. INSTAR gibt keine Anweisungen wie „Jetzt links abbiegen.“, sondern hebt den Weg in einem Live Video Stream farblich hervor.[Nar03]

2.4.3.1 Systemüberblick

Im Grundsatz ist die Funktionsweise von INSTAR relativ einfach zu verstehen. INSTAR bezieht über Schnittstellen zu externen Geräten die aktuelle Position, wie auch die Routinginformationen, produziert daraus eine Ausgabe und gibt diese über eine Schnittstelle an ein Anzeigegerät weiter. Diese sehr abstrakte Sichtweise von INSTAR als Black Box ist in Abbildung 10 zu sehen.

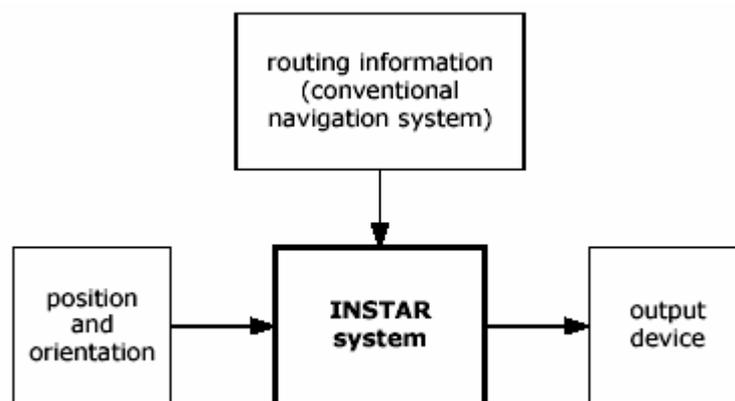


Abbildung 10: INSTAR als Black Box [Nar03]

Da INSTAR in den verschiedensten Umgebungen eingesetzt werden soll, kann es mit den verschiedensten Positionierungsmethoden wie GPS, GSM, WLAN und anderen umgehen. Aktive, wie auch passive Positionierungssysteme, können verarbeitet werden. Zwischen INSTAR und die passiven Systeme wird noch ein zusätzlicher Prozess geschaltet. Dieser Prozess wandelt die relativen Positionsangaben in absolute GPS Koordinaten um. Durch diesen vorgeschalteten Prozess kann INSTAR intern unabhängig vom Positionierungssystem arbeiten.

Die Wegführungsinformationen bezieht INSTAR von einem externen Navigationssystem, da dieses bereits die nötige Qualität aufweist. Die Ausgabe ist jedoch noch nicht optimal, weshalb sie von INSTAR eigenständig generiert wird. Die Ausgabe erfolgt als mit den Weginformationen angereicherter live Video Stream auf dem von INSTAR verwendeten PDA. Dieser angereicherte Video Stream wird vom Benutzer intuitiv verstanden und bedarf keiner zusätzlichen Interpretation, da er keine Abstraktion der realen Welt darstellt. Ebenso ist der Fahrer stets auf die Strasse fokussiert, da sie auch auf dem PDA ersichtlich ist.

Wie die Ausgabe zustande kommt, ist in Abbildung 11 ersichtlich. Auf der Grundlage der aktuellen Position und Orientierung wird anhand der Routinginformation und den topologischen Daten ein virtuelles 3D-Modell des Weges erstellt. Intern wird das 3D-Modell

als „Scenegrph⁹“ gespeichert. Für die fertige Ausgabe wird nun der Scenegrph über den Video Stream der Kamera gelegt. Die fertige Ausgabe ist im Bild rechts unten in der Abbildung 11 und in der Abbildung 12 zu sehen. [Nar03]

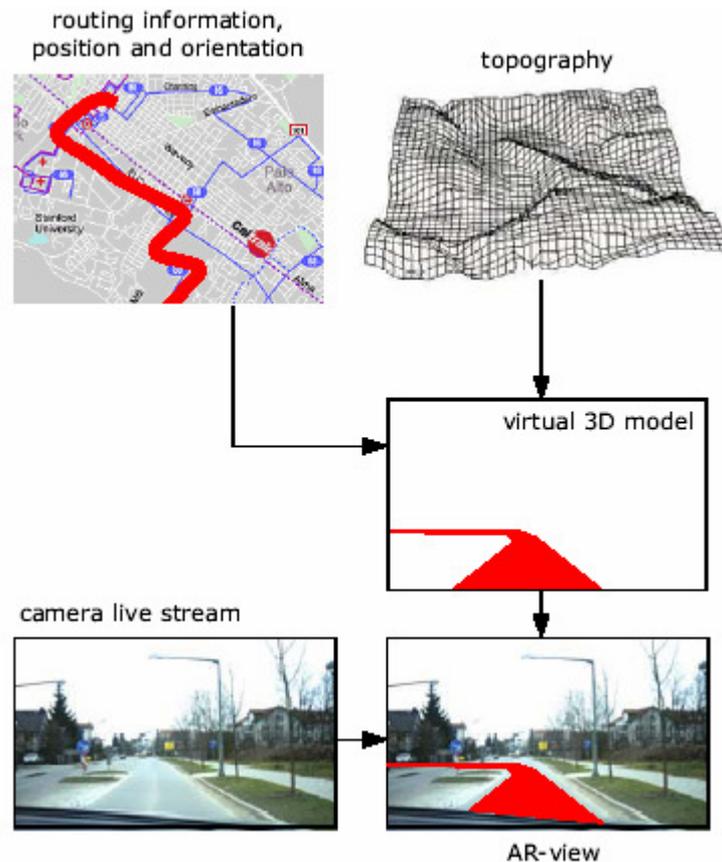


Abbildung 11: Erstellung der AR Ansicht [Nar03]

2.4.3.2 INSTAR in der Praxis

Je nach dem, ob INSTAR in einem Fahrzeug oder zu Fuss verwendet wird, unterscheiden sich die einzelnen Komponenten etwas. Die Unterschiede betreffen aber die externen Geräte und nicht die Hauptkomponente. Die zentrale Einheit wird einheitlich durch einen PDA, in der aktuellen Version einen Compaq iPAQ 3850, gebildet.

Beim Einsatz in einem Fahrzeug muss der PDA nur mit dem Navigationssystem des Fahrzeuges und einer im Fahrzeug installierten Kamera verbunden werden. Die Kamera wird idealerweise hinter dem Rückspiegel montiert. So liefert sie einen Stream, der der Sicht des Fahrers relativ ähnlich ist, ohne diese aber zu beeinträchtigen. Das Navigationssystem des Fahrzeuges liefert sämtliche für den Scenegrphen nötigen Informationen. Abgesehen von der Installation der Kamera ist INSTAR relativ leicht und effektiv in Fahrzeugen einsetzbar. Abbildung 12 zeigt, wie der Fahrer auf seinem Weg geleitet wird. Der Weg ist intuitiv erkennbar und leicht verfolgbar.

⁹ Scenegrph werden in den meisten gängigen Renderer zum Speichern von 3D-Graphiken verwendet [Mül03]



Abbildung 12: INSTAR Wegweisung [Nar03]

Der Einsatz von INSTAR durch Fussgänger erweist sich als etwas aufwändiger als der in Fahrzeugen. Zum einen müssen sämtliche Komponenten mitgetragen werden, zum anderen zeigt die Kamera nicht mehr stets die Situation vor dem Fahrzeug sondern kann frei bewegt werden. Abbildung 13 zeigt die Komponenten von INSTAR im mobilen Einsatz.

Zur Positionsbestimmung wurde ein GPS-Empfänger über den seriellen Port mit dem PDA verbunden. Die Routinginformationen werden von einem lokalen Navigationssystem per WLAN übermittelt. Für das Videosignal wird eine Pocket PC Kamera verwendet. Zur Bestimmung der Peilung des Benutzers ist ebenfalls noch eine Komponente von Nöten. Unglücklicherweise sind durch den GPS-Empfänger, die Kamera und die WLAN Karte bereits alle Anschlussmöglichkeiten des PDAs belegt. Deshalb wurde die Peilungsskomponente an einen zweiten PDA angeschlossen, der die Informationen über WLAN an den ersten PDA übermittelt.[Nar03]

Durch die ganzen externen Komponenten ist INSTAR gerade im Bereich der Personennavigation etwas umständlich in der Handhabung. Es bietet aber dennoch eine gute Möglichkeit, den Benutzer nicht nur bis zum Parkplatz zu leiten, sondern bis ganz an sein Ziel.



Abbildung 13: INSTAR als mobiles System [Nar03]

2.4.4 Mobile Collaborative Augment Reality

Mobile Collaborative Augment Reality (MCAR) ist ein Unterprojekt des Forschungsprojektes Studierstube¹⁰ der technischen Universität Graz. MCAR erlaubt es, die vom Benutzer wahrgenommene, reale Welt durch 3D-Objekte und Audio-Informationen zu erweitern und mit diesen in Interaktion zu treten. Um dem Benutzer sinnvolle Feedbacks geben zu können und das Gefühl einer natürlichen Interaktion vermitteln zu können, findet die Interaktion real time statt. Besonders interessant und vielseitig einsetzbar wird MCAR durch „location based information“¹¹. So werden die verfügbaren Informationen über die aktuell betrachteten Objekte direkt in die reale Welt integriert.

Das Mobile Augmented Reality Setup umfasst die technische Ausstattung und bildet somit die Basis von MCAR. Aufbauend auf diesem wurden einige Applicationen entwickelt: ARLibrary, eine ortsbezogene Application für Bibliotheken, SignPost2 zur Indoor Navigation und OCAR zum Information Browsing und zur Outdoor Navigation. [Rei03]

2.4.4.1 Mobile Augmented Reality Setup

Ein mobiler Computer bildet die zentrale Instanz zur Datenverarbeitung des Mobile Augmented Reality Setup. Dieser wird laufend den aktuell verfügbaren Technologien angepasst. Zuletzt war dies ein Notebook mit einem 2GHz Prozessor und NVidia Quadro4Go Graphikkarte laufend unter dem Betriebssystem Windows XP. Zur Kommunikation mit einer zweiten mobilen Einheit ist er zusätzlich mit einem WLAN Adapter ausgestattet.

Damit die Bewegungsfreiheit nicht zu sehr beeinträchtigt wird, wird die Ausstattung in einem Rucksack mitgetragen. Abbildung 14 zeigt einen Benutzer, der das Mobile Augmented Reality Setup trägt.

¹⁰ Das Projekt Studierstube befasst sich mit den verschiedenen Anwendungsbereichen von Augment Reality. Ausführliche Informationen sind unter <http://studierstube.icg.tu-graz.ac.at/> zu finden.

¹¹ Die angebotene Information ist abhängig vom jeweiligen Aufenthaltsort des Benutzers.



Abbildung 14: Benutzer trägt das Mobile Augmented Reality Setup [Rei01]

Zur Ausgabe wird ein „i-glasses see-through stereoscopic color HMD“ verwendet, das am Helm des Benutzers angebracht ist. Dabei werden die virtuellen Objekte auf die durchsichtigen Scheiben der Brille projiziert und über die Ansicht der realen Welt gelegt. Zur Bestimmung der aktuellen Position ausserhalb von Gebäuden wird ein differential GPS-Empfänger verwendet. Weitere Informationen werden durch einen Sensor und eine Kamera ermittelt, die auch auf den Helm montiert sind.

Der Sensor dient zur Bestimmung der Blickrichtung, also der Himmelsrichtung wie auch des Winkels zur Horizontalebene. Durch die Kamera wird die Umgebung wahrgenommen und die exakte Position relevanter Objekte im Bezug auf das Blickfeld des Benutzers ermittelt. So werden beispielsweise Informationen über einen Kirchturm auch an diesem angezeigt und nicht irgendwo im Himmel.

Zur Steuerung des Systems steht dem Benutzer ein zweidimensionales Menü zur Verfügung. Die Interaktion damit findet über eine Bedienplatte, das PIP (personale interaction panel) statt. Diese werden bei Gebrauch in der Hand getragen und ansonsten am Gurt verstaut.

Abbildung 15 zeigt sowohl einen schematischen Überblick als auch die realen Komponenten des Mobile Augmented Reality Setups. [Rei01]

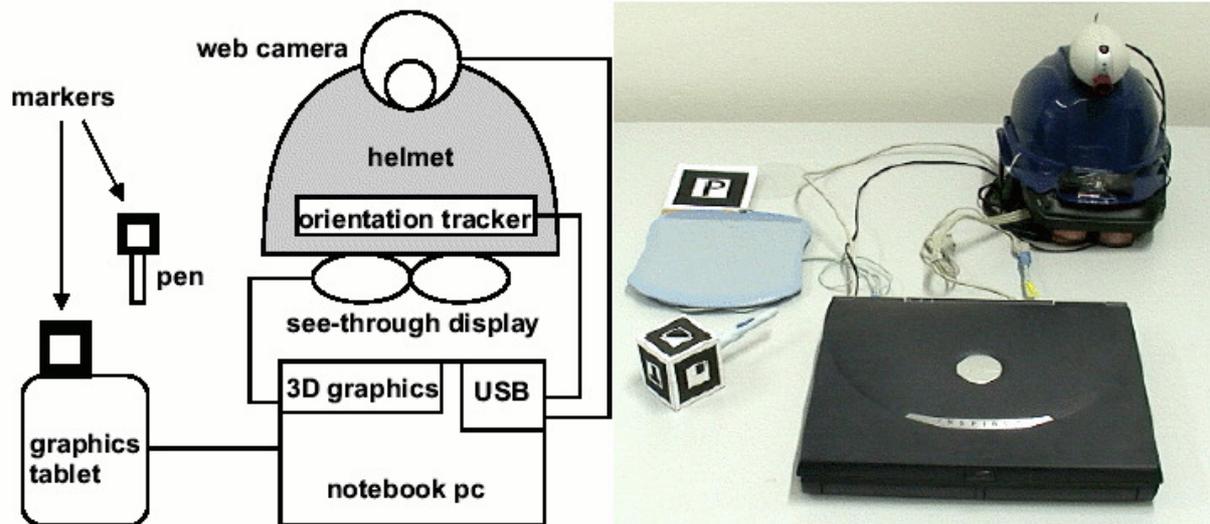


Abbildung 15: Darstellung der Komponenten des Mobile Augmented Reality Setups [Rei01]

2.4.4.2 Outdoor Collaborative Augmented Reality

Diese Applikation bietet dem Benutzer die Möglichkeit zur Navigation und zum Information Browsing im Freien mit Hilfe von Augmented Reality. Auf das Information Browsing von OCAR wird im Rahmen dieser Arbeit nicht eingegangen, Interessierte finden in [Rei03] eine ausführliche Beschreibung.

Grundlage der Navigation sind Kenntnis über die aktuelle Position und die Umgebung, in der man sich befindet. Durch das im vorhergegangenen Kapitel beschriebene Mobile Augmented Reality Setup ist die aktuelle Position wie auch die Blickrichtung verfügbar. Die Informationen über die Umgebung werden einem 3D-Modell der Gebäude und einem Netzwerk der gehbaren Wege entnommen. Im Netzwerk der gehbaren Wege sind Wegpunkte, Strassen, wie auch Adressen für sämtliche Gebäude enthalten.

Zur Navigation kann der Benutzer sein Ziel exakt in der Form einer Adresse oder aber auch nur vage in der Form einer Kategorie angeben. Beispiele für solche Kategorien einer vagen Angabe sind Supermarkt oder Apotheke. Dabei sucht das System selbstständig das nächstgelegene Objekt dieser Kategorie. Danach berechnet das System den kürzesten Weg im Netzwerk der gehbaren Wege und führt den Benutzer an sein Ziel. Das System ist interaktiv, weicht der Benutzer von der berechneten Route ab, so berechnet es die Route neu.

Die Wegführung erfolgt visuell. Die einzelnen Wegpunkte werden durch rote Zylinder markiert, die ihrerseits durch gelbe Pfeile in Marschrichtung verbunden sind. Abbildung 16 zeigt die reine Visualisierung des zu gehenden Weges. Abbildung 17 zeigt den visualisierten Weg eingebettet in die reale Welt. Durch das 3D-Modell der Gebäude wird auch die Sichtbarkeit der einzelnen Wegpunkte und deren Verbindungen berücksichtigt. Durch Gebäude verdeckte Wegpunkte werden nicht angezeigt. Blickt der Benutzer in eine andere Richtung und ist dadurch kein einziger Wegpunkt sichtbar, so wird dem Benutzer eine einfache Richtungsanzeige präsentiert.[Rei03]

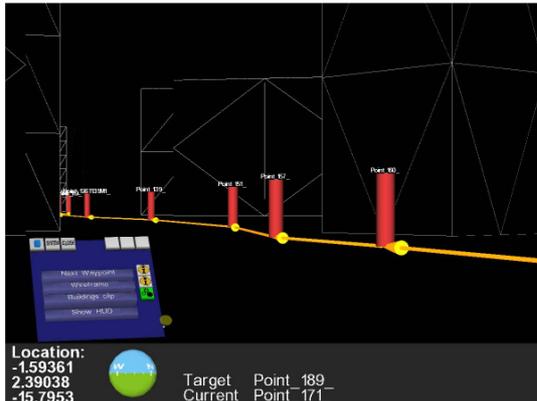


Abbildung 16: Visualisierter Weg [Int06]

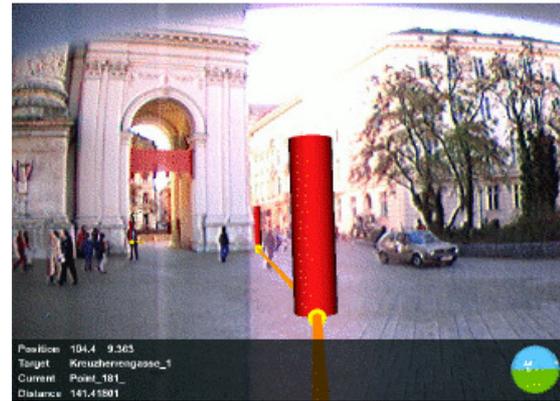


Abbildung 17: Eingebetteter Weg [Rei03]

Sind mehrere Benutzer gleichzeitig unterwegs, so stehen Multi User Optionen zur Verfügung.

Follow: Der Benutzer bestimmt einen anderen, dem er folgen möchte. Das zur Navigation verwendete Ziel ist dabei immer der dem zu verfolgenden Benutzer nächstgelegene Wegpunkt.

Guide: Ein Benutzer kann einen anderen führen und dessen Ziel angeben.

Meet: Unterstützt zwei Benutzer dabei sich zu treffen. Das Ziel wird vom System so gewählt, dass es in der Mitte zwischen den beiden Benutzern liegt. Den Benutzern steht es jedoch offen, den Treffpunkt von dem vom System bestimmten Ort, an einen passenderen zu verlagern. Beispielsweise von einer simplen Strassenkreuzung in ein Café.[Rei03]

2.4.4.3 SingPost II

SingPost II ist der verbesserte Nachfolger der ursprünglichen SingPost Anwendung.

Sing Post II ist ein Indoor Navigationssystem, das den Benutzer durch unbekannte Gebäude an sein Ziel führt.

SingPost II verwendet ebenfalls das Mobile Augmented Reality Setup. Der Benutzer sieht die reale Umgebung und bekommt gleichzeitig von Singpost II Anweisungen über den zu gehenden Weg. Dazu ist die stetige Kenntnis der aktuellen Position nötig. Innerhalb von Gebäuden kann dies aber nicht durch GPS bewerkstelligt werden. In SignPost II wird die Positionsverfolgung durch ARToolkit Markierungen und der auf dem Helm befestigten Kamera realisiert. Zusätzlich wird ein interner Sensor verwendet, der Aufschluss über die Bewegungen des Benutzers, insbesondere schnelle Änderungen der Ausrichtung, gibt. [Kna07]

SignPost II kann in 4 Hauptbestandteile aufgliedert werden:

BAU: Building AUGmentation. Die ist die zentrale Komponente. Sie erstellt die Geometrie des Gebäudes und übernimmt die Positionsverfolgung. [Kna07]

AUG: AUGmentation. Diese Komponente reichert die realen Gegebenheiten mit einem Drahtgittermodell des Gebäudeinneren an. Dies kann sowohl für die gesamte Umgebung geschehen oder aber auch nur für den aktuell sichtbaren Bereich, wobei unsichtbare Linien nicht angezeigt werden. [Kna07]

WIM: World In Miniature. Diese Komponente stellt eine Miniaturansicht des Gebäudes zur Verfügung. Dargestellt wird jeweils die lokale Umgebung. Die Position des Benutzers wird durch ein Logomännchen gekennzeichnet, das stets dieselbe Ausrichtung wie der Benutzer hat. Die WIM kann sowohl auf dem PIP wie auch vor dem Benutzer dargestellt werden. [Kna07]

NAV: NAVigation. Diese Komponente ist eine Navigationsunterstützung. Wenn der Benutzer ein Ziel wählt, wird auf der Grundlage der Informationen von BAU der kürzeste Weg berechnet und der Benutzer dorthin geleitet. Die Weginformationen werden über die Informationen von AUG und WIM gelegt. Ein Pfeil zeigt immer in Richtung des als nächstes zu passierenden Portales. [Kna07]

In der Abbildung 18 ist die oben beschriebene Anreicherung der realen Umgebung durch AUG und WIM dargestellt.

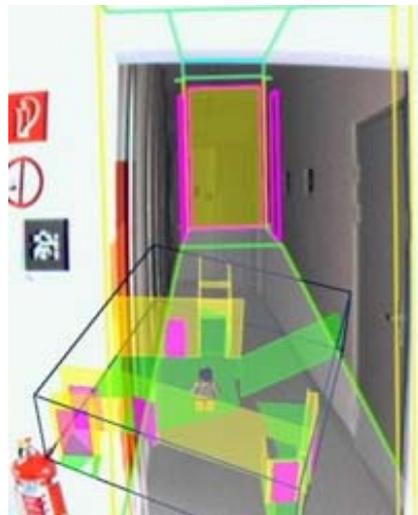


Abbildung 18: Anreicherung der Ansicht durch AUG und WIM [Kna07]

2.4.4.3.1 Positionsverfolgung

Wie bereits erwähnt, wird die Positionsverfolgung mittels Kamera und ARToolkit-Markierungen vorgenommen. Diese Markierungen sind in Abständen von 2 bis 3 Metern an den Wänden angebracht und sind so gestaltet, dass sie möglichst leicht von der Kamera erkannt werden können. Abbildung 19 zeigt einen solchen Marker.

Das System kennt die Positionen der einzelnen Markierungen. Wird eine Markierung erkannt, so kann das System die aktuelle Position anhand der geometrischen Daten der Ansicht auf die Markierung ziemlich genau errechnen.

Je grösser die Anzahl verschiedener Markierungen, desto grösser der Aufwand zum Vergleichen und die Wahrscheinlichkeit von falschen Erkennungen. Daher werden die Markierungen wiederverwendet und sind nur lokal eindeutig. Die Umgebung ist in Raster aufgeteilt und eine Markierung muss nur in einem 3x3-Raster eindeutig sein. Durch das Einbeziehen der letzten bekannten Position, können so Markierungen dennoch eindeutig erkannt werden. Bislang ist es jedoch so, dass der Benutzer am Anfang dem System seine momentane Position mitteilen muss.[Kna07]



Abbildung 19: ARToolkit Marker [Kna07]

2.4.4.3.2 Navigation

Zur Navigation wählt der Benutzer sein gewünschtes Ziel aus der Liste aller Räume aus. Das System berechnet dann den kürzesten Weg von der aktuellen Position zum gewählten Ziel. Der Benutzer erhält vom System verschiedene visuelle Inputs über den zu gehenden Weg. Ein Pfeil zeigt ihm stets die Richtung an, in der sich das nächste Portal, wie beispielsweise eine Tür, befindet. Das nächste Portal wird zudem optisch durch ein gelbes Rechteck hervorgehoben. Zusätzlich wird dem Benutzer in textueller Form sein aktueller Aufenthaltsort, wie auch das gewählte Ziel, angegeben. Die Visualisierung dieser Informationen ist in der Abbildung 20 veranschaulicht: Links oben die textuellen Informationen über die aktuelle Position und das Ziel, in der rechten oberen Ecke die Richtung zum nächsten Portal und in der Mitte das gelb hervorgehobene Portal.

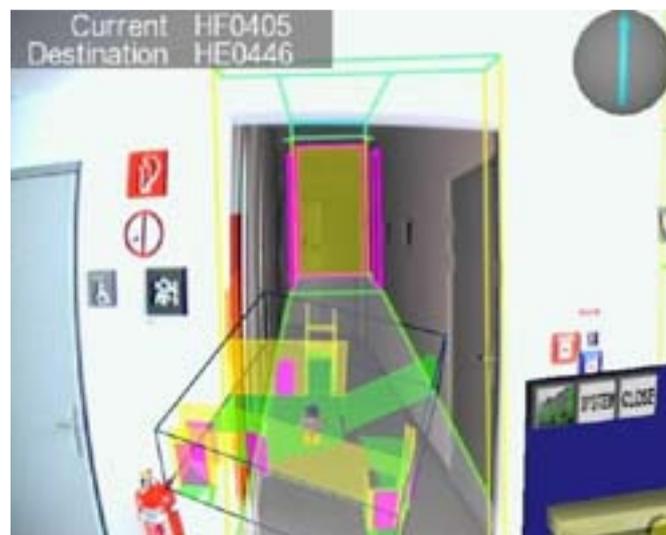


Abbildung 20: Visualisierte Navigationsinformationen [Kna07]

Die Berechnung des Weges wird mittels des Algorithmus von Dijkstra vollzogen und ist in [Kal02] beschrieben. Die Berechnungen erfolgen auf gerichteten Graphen. Die Knoten stellen dabei die einzelnen Räume dar und die Kanten die Verbindungen zwischen den Räumen. Gerichtet deshalb, damit auch Einwegverbindungen wie Notausgänge oder ähnliches abgebildet werden können.

Der Graph wird erzeugt, wenn ein neuer Raumplan geladen wird. Grundlage dazu sind die Informationen über die Räume, die Portale und deren Positionen. Für jeden Raum werden der Mittelpunkt und die Entfernung zu den Portalen des Raumes berechnet und in den Graphen eingetragen.

Das Resultat der Wegberechnung auf dem Graphen ist eine Liste von Räumen und Portalen, die auf dem Weg passiert werden. Jedes Mal, wenn ein Raumwechsel stattgefunden hat, wird der Weg neu berechnet. Dadurch wird das System robust gegen Irr- und Umwege des Benutzers. Hält sich der Benutzer nicht an die Navigationshinweise, so ist das für das System nicht von Bedeutung, es berechnet ohnehin den Weg für jeden Raum neu. [Kna07]

MCAR ist insgesamt ein sehr komplexes System. Die Möglichkeiten der verwendeten technischen Ausstattung werden aber voll ausgeschöpft. Die Indoor-Komponente SingPost II versorgt den Benutzer mit sehr detaillierten Informationen zur Navigation. Die Navigation im Freien ist sehr intuitiv und es ist für den Benutzer ein Leichtes, den Weg zu finden. Ebenso interessant sind die Möglichkeiten zur Interaktion mit anderen Benutzern. Die Navigationsfunktionen Follow, Guide und Meet bieten zusätzliche Erleichterungen für den Benutzer.

2.4.5 GUIDE Project

GUIDE ist ein kontextbezogenes Informationssystem der Universität Lancaster. Es dient zur Unterstützung von Touristen auf ihren Entdeckungsreisen in der Stadt. GUIDE versorgt sie dabei mit positionsspezifischen Informationen über Sehenswürdigkeiten, führt sie auf speziell für sie zugeschnittenen Routen und vieles mehr. Die folgenden Beschreibungen des Systems basieren auf den beiden Publikationen [Che00a] und [Che00b].

2.4.5.1 Anwendungsanforderungen

Damit der Benutzer auf seiner Tour durch die Stadt GUIDE benutzen kann, muss er lediglich mit einem Tablet-PC ausgestattet werden, der anhand eines Stiftes bedient werden kann. Bei der Umsetzung von GUIDE wurde dabei auf Fujitsu TeamPad 7600 zurückgegriffen. Es verfügt über ein ausgewogenes Verhältnis zwischen der gesamten Grösse des Gerätes und der Grösse des Displays. Das Display bietet durch seine Auflösung von 800x600 Pixeln auch die Möglichkeit detaillierte Grafiken anzuzeigen. Zur Positionierung und zur Kommunikation wird die integrierte WLAN-Karte verwendet. Mit einem Gesamtgewicht vom 850g ist das System auch relativ leicht und beeinträchtigt den Benutzer nicht zu sehr. Einziger Nachteil des Systems ist, dass die Batterie bei laufender WLAN-Karte nur gerade Strom für 2 Stunden liefert. Dies beeinträchtigt natürlich die geführten Entdeckungsreisen durch die Stadt massiv. Natürlich hätte zur Positionierung auch GPS verwendet werden können, der Entschied fiel aber aus zwei Gründen auf WLAN. Zum einen bedarf es dazu keiner weiteren Hardware, zum anderen ist in dicht bebauten Gebieten oft nicht die nötige Anzahl von Satelliten sichtbar.

Zur Positionsbestimmung des Benutzers senden gezielt positionierte Stationen Positionsinformationen aus, auf deren Basis das Gerät des Benutzers seine Position bestimmen kann. Dadurch kann aber nur die Position, nicht aber die Peilung des Benutzers bestimmt werden, was bei der Wegführung des Benutzers auf seinem Stadtrundgang berücksichtigt werden muss. Dazu aber mehr im Kapitel 2.4.5.2.

Die Benutzeroberfläche präsentiert sich dem Benutzer ähnlich wie ein Webbrowser. Zu sehen ist sie in der Abbildung 21. Die Anlehnung an einen Webbrowser erleichtert dem Benutzer die Bedienung, da er bereits damit vertraut ist.

Über die Benutzeroberfläche erhält der Benutzer verschiedene Informationen über das System selbst und kann auf die verschiedenen Funktionen zugreifen. Angezeigte Informationen über

das System sind beispielsweise der aktuelle Status des Systems und die WLAN-Signalstärke. Ebenso ist die aktuelle Position in textueller Form ersichtlich.

Die Funktionen, die das System dem Benutzer zur Verfügung stellt, sind zahlreich. Über den „Info Button“ kann er sich Informationen über Sehenswertes bei der aktuellen Position ansehen, wie beispielsweise das Lancaster Castle in der Abbildung 21. Zur besseren Orientierung kann eine Übersichtskarte eingeblendet werden, wie es in der Abbildung 23 zu sehen ist.

Man kann sich eine individuelle Tour zusammenstellen und sich führen lassen.

Über die Kommunikationskomponente kann man sich mit anderen Touristen unterhalten. Da das System auch mit dem Internet verbunden ist, kann man auch direkt Karten für Besichtigungen oder Rundfahrten bestellen.

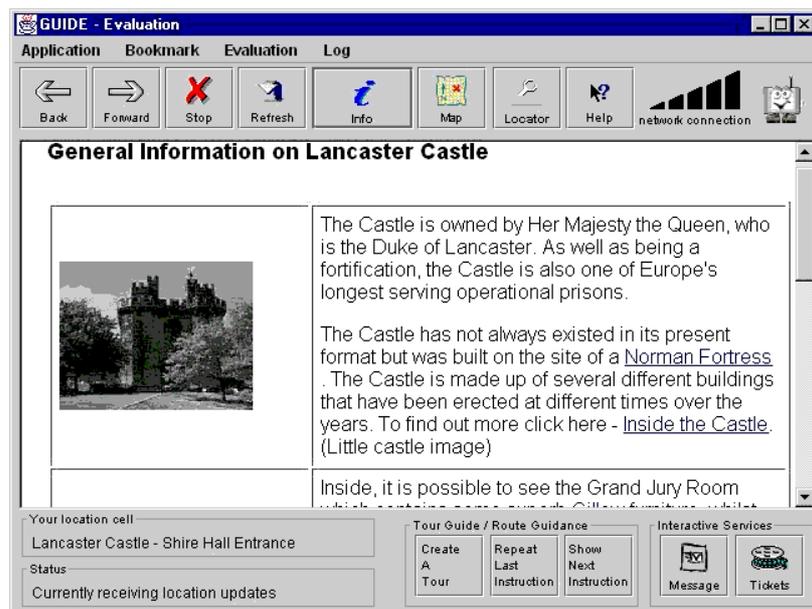


Abbildung 21: GUIDE Benutzeroberfläche [Dist06]

2.4.5.2 Benutzerspezifische Routen

Durch die Informationen zu den aktuellen Gegebenheiten und die Tourführungsfunktion wird GUIDE zu einem vollwertigen Reiseführer. GUIDE erlaubt es Touren ganz nach den individuellen Bedürfnissen zu gestalten und sich dann durch die Stadt führen zu lassen.

Durch den „Create a Tour Button“ gelangt man in den Dialog zum Erstellen einer Tour. Dabei muss man die zu besichtigenden Sehenswürdigkeiten angeben. Diese sind nach Kategorien wie „Historic“, „Recreation“ oder „Popular Attraction“ gegliedert. Nachdem man sämtliche Sehenswürdigkeiten, die man besichtigen will, angegeben hat, erstellt GUIDE die ganz persönliche Besichtigungsrouten. Dabei werden verschiedene Faktoren berücksichtigt. Natürlich spielt die Distanz zwischen den einzelnen Punkten eine wichtige Rolle. Öffnungszeiten der Attraktionen und Zeiten mit besonders grossem Besucherandrang werden aber genauso berücksichtigt. GUIDE versucht den gesamten Weg möglichst kurz zu halten, dabei sollen aber alle Sehenswürdigkeiten auch besucht werden können und das, wenn möglich, ohne lange Zeit in Warteschlangen verweilen zu müssen. Alles in allem versucht GUIDE die Dauer der Tour zu optimieren. Ist die Tour einmal erstellt, so wird der Benutzer von einer Sehenswürdigkeit zur nächsten geleitet. Die Reihenfolge der Attraktionen ist aber nicht statisch sondern dynamisch und kann im Laufe der Stadtbesichtigung ändern. Gründe dafür können sein, wenn man an einem Ort länger verblieben ist als vorgesehen war oder wenn eine Sehenswürdigkeit früher schliesst.

Wie bereits erwähnt, kann GUIDE nur die Position, nicht aber die Peilung des Benutzers bestimmen. Um diesem Faktum gerecht zu werden, wird dem Benutzer die Route als Text mit genauen Richtungsangaben präsentiert. Zudem helfen Fotos des nächsten Zieles es zu finden. Abbildung 22 zeigt die Wegbeschreibung zum John of Gaunt Gateway des Lancaster Castles.

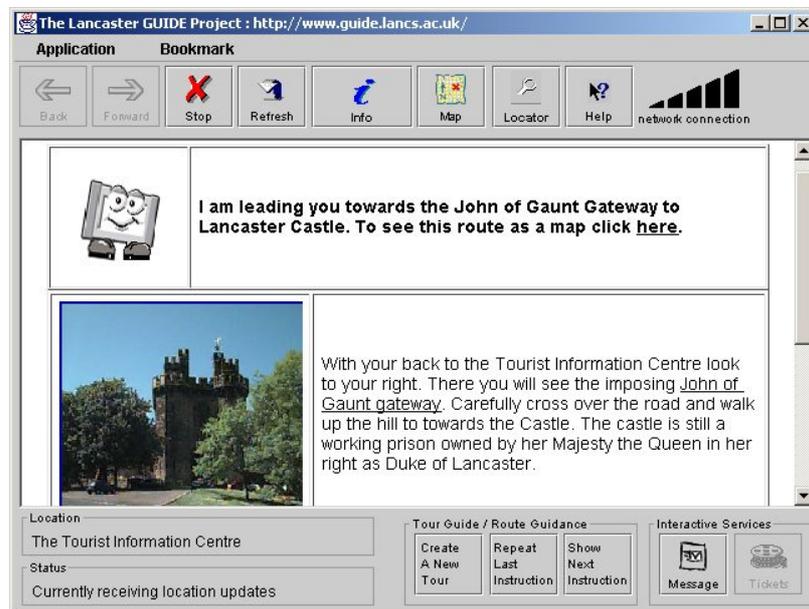


Abbildung 22: GUIDE Wegführung [Dis06]

Da die Ausrichtung des Benutzers nicht bekannt ist, wird ihm gesagt, wie er sich ausrichten muss. So heisst es: „Sehe nach rechts, mit dem Rücken zum Touristen Informations Center. Dort siehst du das imposante John of Gaunt Gateway. Überquere die Strasse vorsichtig und gehe den Hügel hoch in Richtung der Burg...“

An einem Zwischenziel angekommen, klickt man auf „Show Next Instruction“, um weitere Anweisungen zu erhalten und die nächste Etappe in Angriff nehmen zu können. Ist man sich über die Anweisungen nicht mehr ganz im Klaren, so kann man sie auch erneut abrufen.

Zur besseren Orientierung kann man sich auch eine Karte mit der aktuellen Position anzeigen lassen. Abbildung 23 zeigt die Übersichtskarte der lokalen Umgebung und die eigene Position vor der Burg.

Für die Kartenansicht gibt es zwei Varianten: Zum einen eine Übersichtskarte der lokalen Umgebung oder aber einer Karte von ganz Lancaster.

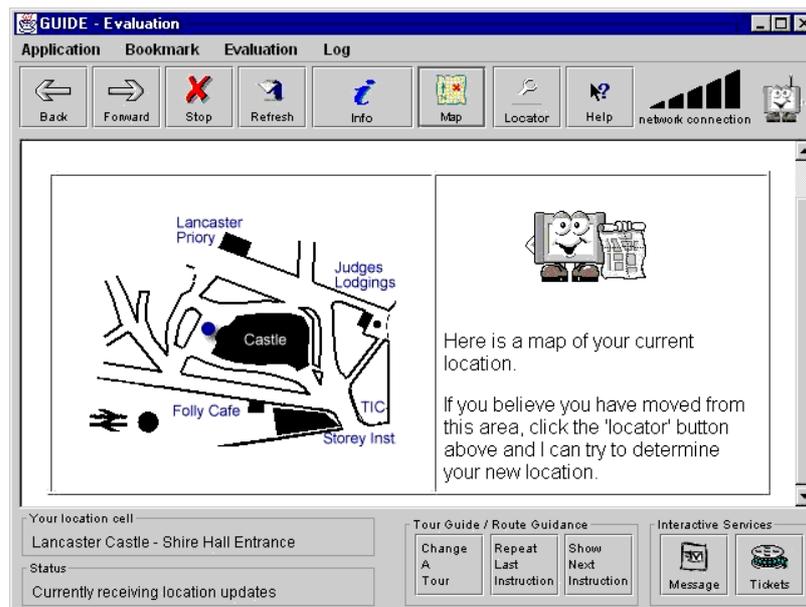


Abbildung 23: GUIDE Übersichtskarte [Dist06]

GUIDE ist ein sehr nützlicher Begleiter für Touristen auf ihren ganz individuellen Touren durch die Stadt. Nach der Eingabe seiner Präferenzen übernimmt er die Routenplanung und führt diese dynamisch aus. Durch seine Wegführung, die Informationen zu den lokalen Gegebenheiten und die Zusatzfunktionen, wie Ticketreservierung, wird er zur guten Alternative zum menschlichen Reiseführer.

2.4.6 Navigationssysteme für das Handy

Es sind bereits zahlreiche kommerzielle Navigationssysteme für das Handy verfügbar. Das Problem dieser Systeme ist jedoch, dass sich die verfügbaren Informationen meist auf Verkaufsargumente und Testberichte beschränken. Dokumentationen darüber, wie sie etwas machen, sucht man vergebens. Daher werden diese Systeme in dieser Arbeit auch nur sehr kurz betrachtet. Grundlage dieser Betrachtung ist der Artikel „Auf schnellstem Wege – Software für die Handy-Navigation“ aus der Zeitschrift CT.

Grundvoraussetzung für alle diese Systeme ist ein GPS-Empfänger zur Bestimmung der Position. Je nach Art des Systems wird zusätzlich noch eine GPRS- oder besser noch eine UMTS-Verbindung benötigt.

Unterschieden wird zwischen Onboard- und Offboard-Systemen. Onboard-Systeme übernehmen die Berechnung der Route selbst, auf der Grundlage von auf einer Speicherkarte vorhandenen Strassenkartendaten. Offboard-Systeme hingegen übermitteln Start und Ziel an einen zentralen Server, der die Route berechnet und sie an das Handy zurückschickt. Dabei wird auf das Übermitteln vollständiger Karten verzichtet. Es wird lediglich ein Korridor übertragen. Wird dieser Korridor verlassen, so wird eine neue Anfrage an den Server gesendet.

Die Entscheidung zwischen Onboard- und Offboard-System sollte man auf Grund der Nutzungshäufigkeit treffen. Bei häufiger Nutzung sind Onboard-Systeme durch die einmaligen Anschaffungskosten günstiger. Bei Offboard-Systemen sind bei jeder Routenanfrage Gebühren für die Datenübertragung fällig.

Grundsätzlich könnten diese Systeme auch für die Personennavigation verwendet werden, da sie auf Handys ausgeführt werden. Gedacht sind sie aber dennoch für den Einsatz im Strassenverkehr. Die Berechnung der Routen erfolgt auf der Grundlage von

Strassenkartendaten und auch das Erscheinungsbild erinnert doch sehr an herkömmliche Systeme aus der Fahrzeugnavigation. [Röb07]

Die Abbildung 24 zeigt anhand von zwei Beispielen, wie sich diese Systeme in etwa präsentieren.



Navigon MN 6 [Nav07]



ALK Copilot live [ALK07]

Abbildung 24: Beispiele für Handy-Navigationssysteme

Navigationssysteme für das Handy sind im kommen, aber sind sie noch lange nicht ausgereift. Problematisch ist, dass bislang der GPS-Empfänger noch kein Standard für das Handy ist. Ebenso ist der Display doch sehr begrenzt.

Obwohl sie durch die Ausführung auf einem Handy gut für die Fussgängernavigation geeignet wären, ist diese Anwendung bislang nicht vorgesehen, da sie den speziellen Anforderungen, die durch die Benutzung von Fussgängern an die Navigation gestellt werden, nicht gerecht werden.

3 Realisierung des Navigationssystems für den mExplorer

In diesem Kapitel wird nun das für den mExplorer entwickelte Navigationssystem vorgestellt. Eingegangen wird dabei auf die Anforderungen an das System, die Grundvoraussetzungen, die erfüllt sein müssen, auf die Architektur des Navigationssystems, die Benutzerinteraktion wie auch einige Besonderheiten der Wegberechnung.

Ein Teil der Inspiration zur Entwicklung stammt aus den in Kapitel 2.4 betrachteten verwandten Systemen. Manche gute Idee wurde aufgegriffen. Sie mussten aber allesamt auf den mExplorer adaptiert werden, denn der mExplorer ist, wie auch alle betrachteten Systeme, ein Unikat.

3.1 Anforderungsspezifikation

Die Anforderungsspezifikation ist von zentraler Bedeutung für ein zu entwickelndes System. Sie legt fest, welchen Anforderungen das System gerecht werden muss. In diesem Fall gehen die grundlegenden Anforderungen an das System aus der Aufgabenstellung hervor und lauten wie folgt:

Gehbare Wege: Die im mExplorer verwendeten Karten sollen mit den Informationen über die gehbaren Wege versehen werden können. Diese Funktionalität soll in den bestehenden Editor integriert werden.

Integration: Das Navigationssystem soll in die bestehende Fassung des mExplorers integriert werden.

Freie Zielwahl: Dem Benutzer soll es möglich sein das Ziel durch Anklicken der entsprechenden Stelle auf der Karte zu wählen.

Wegberechnung: Auf der Grundlage der aktuellen Position und der gehbaren Wege soll mittels eines adäquaten Wegfindungsalgorithmus der kürzeste Weg ans Ziel berechnet werden.

Wegleitung: Der Benutzer soll auf dem berechneten Weg ans Ziel geführt werden. Dies soll sowohl optisch wie auch akustisch erfolgen.

Dies sind die primären Anforderungen, die es unbedingt zu erfüllen gilt. Weitere Anforderungen wie beispielsweise die Wahl des gejagten Teams als Ziel oder freie Farbwahl des angezeigten Weges sind kein Muss, helfen aber die Attraktivität des Systems zu steigern.

3.2 Grundvoraussetzungen zur Navigation

Es gibt drei Grundvoraussetzungen, die erfüllt sein müssen, um navigieren zu können, respektive dass ein Navigationssystem überhaupt einsetzbar ist. Diese Voraussetzungen sind die aktuelle Position, die Bestimmung der Peilung und die Kenntnis der Umgebung. Anhand der aktuellen Position und der Peilung ist es bereits möglich ein Ziel anzusteuern, jedoch noch nicht optimal. Erst durch die Kenntnis der Umgebung ist effiziente Navigation möglich. Sie erlaubt es den Weg zu planen und zu optimieren. Hindernisse wie Flüsse, Gebäude oder ähnliches können elegant umgangen werden.

Die Positionsbestimmung und die Peilung sind keine Bestandteile dieser Arbeit. Da sie aber dennoch für das Verständnis von Bedeutung sind, wird kurz auf sie eingegangen.

3.2.1 Positionsbestimmung

Die Positionsbestimmung ist seit Anfang an ein Bestandteil des mExplorers. Die Bestimmung der Position erfolgt dabei auf der Grundlage von unterschiedlichen Signalstärken der verfügbaren WLAN Access Points. Dazu wird die Ekahau Positioning Engine verwendet. Der Ekahau Client misst die Signalstärken und übermittelt sie an den Ekahau Server, der daraus die Position berechnet. Dem mExplorer-Server wird nun die berechnete Position mitgeteilt und dieser leitet sie an den Client weiter. Die Genauigkeit der Positionsbestimmung ist abhängig von der Anzahl verfügbarer WLAN Access Points. Bei vier empfangbaren Access Points wird eine Genauigkeit von 1-3 Metern erreicht.

Unglücklicherweise muss das Ekahau System vorgängig auf die jeweilige Umgebung geeicht werden, was sehr zeitaufwändig ist. [Göt04]

3.2.2 Peilung

Im Rahmen der Diplomarbeit von Daniel Suter wurde der mExplorer um die Funktionalität der magnetischen Peilung erweitert. Das Herzstück dabei ist ein extern an den PDA angeschlossenes Sensorboard. Dieses Sensorboard misst die verschiedensten Werte, unter anderem auch das Magnetfeld der Erde in drei Dimensionen. Aus dem X- und dem Y-Wert wird dann die zweidimensionale Peilung ermittelt und für den Benutzer graphisch aufbereitet. Im Gegensatz zu einem herkömmlichen Kompass wird dem Benutzer jedoch nicht die Nordrichtung angezeigt, sondern ihm wird auf der Karte angezeigt, in welche Richtung er blickt, also seine Peilung.

Ausführlich erläutert ist das Ganze in [Sut06], der Diplomarbeit von Daniel Suter.

3.2.3 Kenntnis der Umgebung

Bislang wurden im mExplorer rein bildbasierte Karten der Umgebung verwendet. Sie ermöglichen es dem Benutzer, sich in der fremden Umgebung zu orientieren, bieten aber nicht die Möglichkeit Wegberechnungen vorzunehmen. Für Menschen ist es ein Leichtes die Abstraktion der Karte zu lesen und zu verstehen, für eine Maschine jedoch nicht. Was für uns Wände, Türen oder Treppen sind, sind für den Computer alles nur Pixel.

Es war also nötig, die Karten mit den für die Wegberechnung nötigen Informationen zu versehen. Zu diesem Zweck wurde ein Editor erstellt und in den mExplorer integriert. Die Architektur des Editors wird im Kapitel 3.4.1 und die Funktionalität im Kapitel 3.5.1 beschrieben. Der Editor erlaubt es, die Karte mit dem Graphen der gebaren Wege zu versehen.

Die Abbildung 25 zeigt einerseits die Karte und den Graph einzeln, andererseits auch die Vereinigung zwischen der Karte und dem Graph. Der Graph ist für den Benutzer nicht von Bedeutung und wird daher während dem eigentlichen Spiel nicht angezeigt.

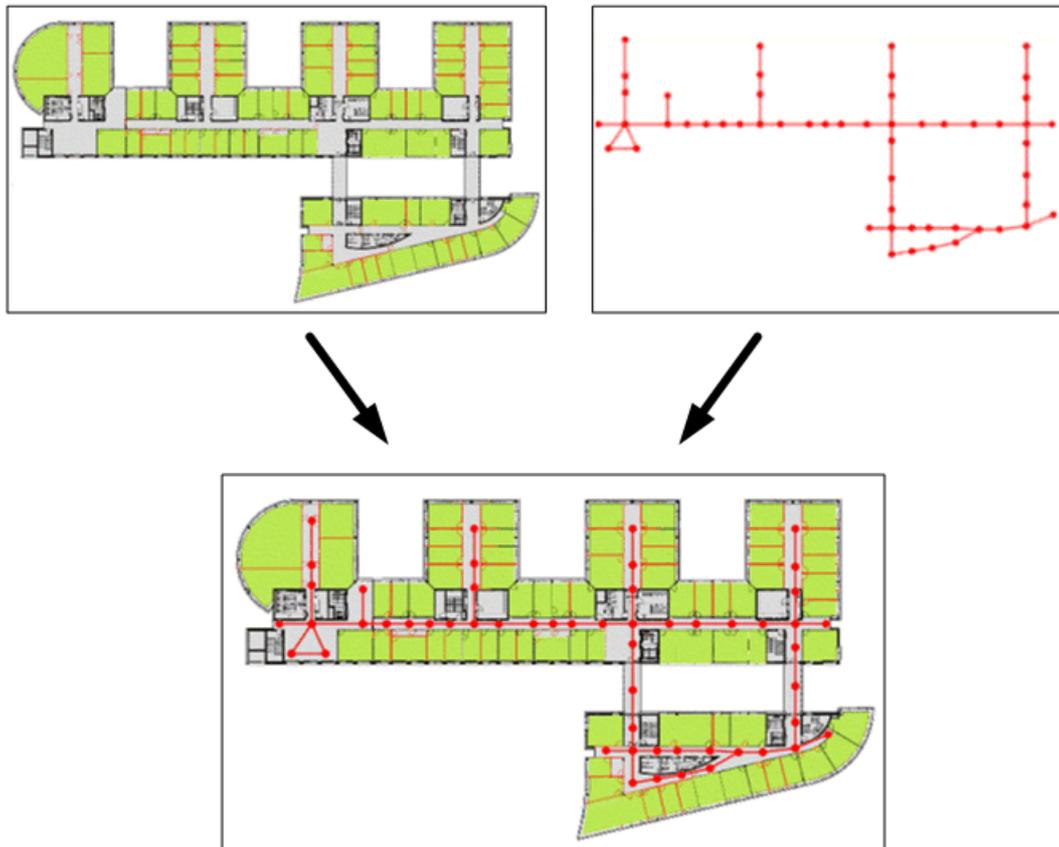


Abbildung 25: Erweiterung der Karte um den Graphen der gehbaren Wege

Der Graph ist ein abstraktes Abbild der gehbaren Wege, also der Wege auf denen sich ein Benutzer bewegen kann. Der Graph setzt sich aus Knoten und Kanten zusammen. Knoten bezeichnen End- oder Kreuzpunkte von Wegen. Kanten sind Verbindungen zwischen zwei Knoten und stellen die eigentlichen Wege dar. Grundsätzlich befinden sich Knoten, wie auch Kanten, auf einer Karte. Bei den Kanten gibt es aber einen Spezialfall, nämlich Kanten, die von einem Knoten einer Karte zu einem Knoten einer anderen Karte führen. Dies rührt daher, dass sich das gesamte Spielgelände über mehrere Karten erstreckt. So gibt es beispielsweise für jedes Stockwerk eines Gebäudes eine eigene Karte. In der Realität kann man sich über Lifte oder Treppen von einem Stockwerk zum anderen bewegen und genau dieser Sachverhalt wird durch die Kantenverbindungen zwischen Knoten verschiedener Karten abgebildet.

Der komplette Graph erstreckt sich folglich über sämtliche Karten des Spielgeländes.

Die Abbildung 26 zeigt wie sich der komplette Graph aus den Teilgraphen der einzelnen Karten und den Verbindungen zwischen den einzelnen Karten zusammensetzt. Rot ausgezogen sind die Teilgraphen der beiden Karten, die zwei Stockwerke eines Gebäudes repräsentieren. Die orangen, gestrichelten Linien stellen die Verbindungen der beiden Stockwerke, wie Treppen oder Lifte, dar. In der dargestellten Situation gibt es also drei

verschiedene Möglichkeiten, um vom einen Stockwerk zum andern zu gelangen. Die beiden beschrifteten Punkte A1 und A2 sind in der Realität ein und derselbe Punkt, denn die Treppe, auf der sie sich befinden, ist auf beiden Karten dargestellt.

Die anderen beiden Möglichkeiten das Stockwerk zu wechseln, sind zwei Lifte. Die entsprechenden Punkte sind daher nicht identisch, befinden sich aber jeweils direkt über, beziehungsweise unter den anderen.

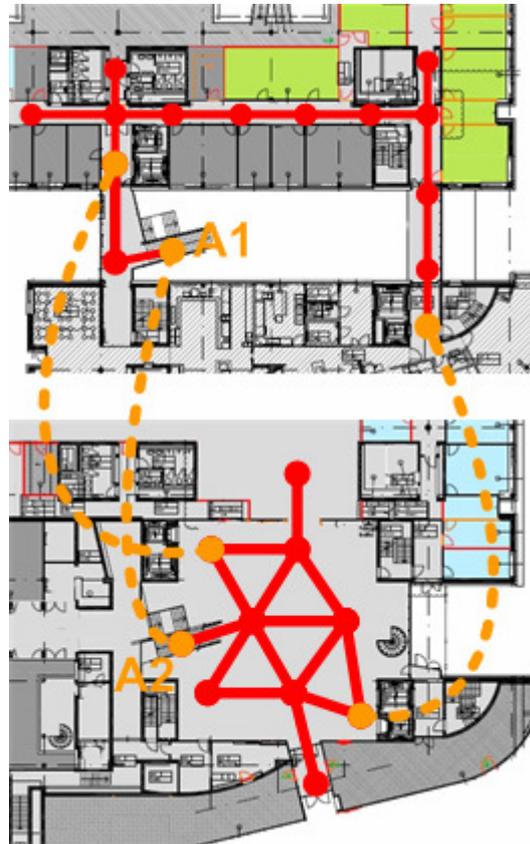


Abbildung 26: Kompletter Graph bestehend aus den Teilgraphen zweier Karten

Auf der Grundlage des Graphen können nun also die für die Navigation nötigen Wegberechnungen durchgeführt werden. Anhand der Distanzen zwischen den einzelnen Knoten, unter Berücksichtigung der unterschiedlichen Kartenmassstäbe, wird der kürzeste Weg ans Ziel bestimmt.

3.3 Details zur Wegberechnung und Wegleitung

Bei der Wegberechnung musste den speziellen Gegebenheiten des mExplorers Rechnung getragen werden. Ebenso bei der Führung des Benutzers an sein Ziel gibt es gewisse Besonderheiten, die die Wegleitung vereinfachen. In den nun folgenden Kapiteln 3.3.1 und 3.3.2 werden diese Besonderheiten erläutert.

3.3.1 Wahl des Wegberechnungsalgorithmus

Wie in Kapitel 2.2 ausführlich beschrieben ist, ist A^* grundsätzlich effizienter als der Algorithmus von Dijkstra bei der Suche, denn A^* ist speziell darauf ausgelegt, den kürzesten Weg zwischen zwei Punkten zu finden. Durch den Einbezug einer Heuristik in die Gewichtung der einzelnen Knoten wird die Suche optimiert.

Bei der Berechnung des kürzesten Weges innerhalb des Navigationssystems des mExplorers wird dennoch der Algorithmus von Dijkstra verwendet, denn wie es sich herausgestellt hat, ist er unter den durch den mExplorer gegebenen Voraussetzungen effizienter einsetzbar.

Der Grund liegt bei den verschiedenen im mExplorer verwendeten Karten. Jeder Knoten hat seine Position auf der jeweiligen Karte. Diese Position hat aber keinen globalen Bezug zum gesamten Spielfeld sondern nur einen lokal auf der entsprechenden Karte.

Bei existierenden globalen Bezügen der einzelnen Positionen zueinander liesse sich die horizontale Distanz eines Knotens zum Ziel leicht berechnen und die Vorzüge von A^* voll ausschöpfen. Ohne die globalen Bezüge ist es sehr aufwändig, sinnvolle Werte für die Heuristik zu berechnen. In einem ersten Schritt könnte man die einzelnen Karten bestimmen, über die der Weg führen wird. Ausgehend davon könnte man die Horizontalabstände zwischen den einzelnen Kartenverbindungen addieren, um brauchbare Werte zu erhalten.

Zum besseren Verständnis wurde die Sachlage in Abbildung 27 graphisch veranschaulicht.

Für die Heuristik von A^* wird ein sinnvoller Wert für die Horizontalabstände zwischen den beiden Punkten A und B benötigt. Anhand der Koordinaten von A und B ist dies jedoch nicht möglich, dass sie sich auf verschiedenen Karten befinden. Bei der bereits erwähnten Möglichkeit werden die drei violett eingefärbten Distanzen zwischen den Verbindungspunkten der Karten addiert. Dieser Wert entspricht zwar nicht der Horizontalabstände zwischen A und B, er ist aber sicherlich nicht länger als der tatsächliche Weg und daher immer noch für A^* geeignet.

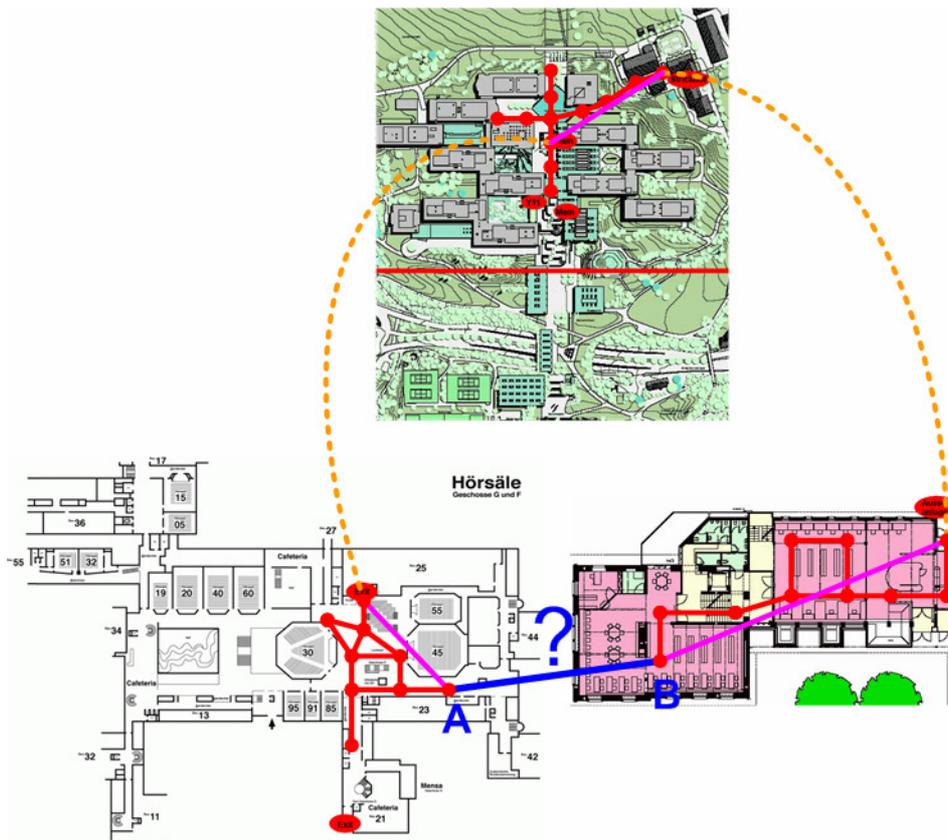


Abbildung 27: Fehlen der globalen Bezüge

Diese Berechnung ist allerdings schon bei einfachen Verhältnissen sehr aufwändig und wird mit steigender Anzahl von Verbindungspunkten stets noch aufwändiger. Daher übersteigen die Kosten der Gewinnung dieses Wertes den aus der Verwendung von A* resultierenden Nutzen. Somit ist es effizienter den allgemeineren Dijkstra zu verwenden als A*.

3.3.2 Harmonische Wegführung

Aus der Berechnung nach Dijkstra resultiert der Pfad, also der kürzeste Weg ans Ziel. Dieser Pfad setzt sich aus den einzelnen Knoten des Graphen zusammen, die der Benutzer auf seinem Weg ans Ziel passiert. Schritt für Schritt wird der Benutzer von einem Knoten zum nächsten geleitet, bis er das Ziel erreicht. Sollte er einmal vom Weg abkommen, so wird die Wegberechnung neu vorgenommen und der Benutzer entsprechend der neuen Berechnung geführt. Dabei kann es jedoch vorkommen, dass der Benutzer für ihn nicht nachvollziehbare Anweisungen erhält. Das System kann den Benutzer anweisen, in eine Richtung zu gehen, die für den Benutzer offensichtlich nicht mit dem berechneten Weg übereinstimmt. Je häufiger solches nicht nachvollziehbares Verhalten auftritt, desto grösser wird die Skepsis des Benutzers gegenüber dem System.

Dieses Verhalten ist durch die Abbildung der gehbaren Wege auf einen Graphen begründet. Der Graph ist zwar sehr gut als Grundlage für die Wegberechnung geeignet, doch bei der Abbildung der gehbaren Wege treten gewisse Schwierigkeiten auf. Korridore oder Wege lassen sich gut abbilden. Sie sind in ihrer räumlichen Ausdehnung auch schon sehr nahe an einer Geraden. Problematisch sind jedoch grosse Flächen wie Plätze oder Innenhöfe. Sie lassen sich nur annäherungsweise durch ein Netz von Knoten und Kanten abbilden.

Es sind eben gerade diese komplexen Annäherungen an die Realität, die die für den Benutzer falsch erscheinenden Wegweisungen verursachen.

Der gewählte Lösungsansatz sieht vor, dass der Benutzer nicht stur von Knoten zu Knoten geleitet wird, sondern die Kurven geschnitten werden. Befindet sich der Benutzer innerhalb eines bestimmten Radius zum nächsten Knoten, so wird er direkt zum übernächsten geleitet. Bei geraden Strecken ändert sich dadurch nichts an der Richtung, bei Abzweigungen jedoch schon. Der eckige, berechnete Weg wird dadurch abgerundet und harmonisiert. Die Abbildung 28 zeigt die Auswirkungen dieser Anpassungen. Blau gestrichelt dargestellt ist jeweils der auf der Grundlage des roten Graphen berechnete Weg vom Punkt 1 zum Punkt 2. Grün dargestellt ist der Weg, auf dem das System den Benutzer tatsächlich führt. Im linken Teil der Abbildung 28 wird gezeigt, wie das System den Benutzer anweist die Kurve zu schneiden anstatt bis in die Mitte des Korridors zu gehen. Dies stimmt auch viel eher mit dem Weg überein, den der Benutzer von sich aus gehen würde. Im rechten Teil der Abbildung 28 werden die Auswirkungen auf das Passieren eines Platzes gezeigt. Auch in diesem Fall zeigt sich die Harmonisierung als sinnvolle Modifikation. Aus dem berechneten Zickzackweg wird eine direkte Verbindung zwischen den beiden Punkten 1 und 2.

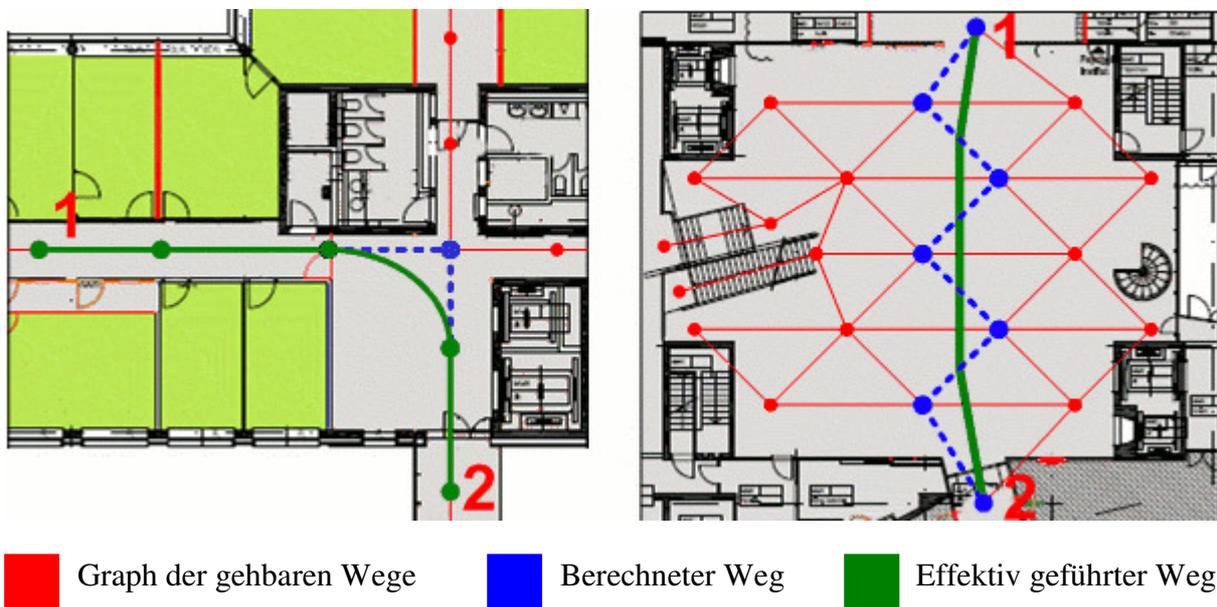


Abbildung 28: Harmonische Wegführung

Durch die Harmonisierung der Wegführung konnte der Brückenschlag zwischen den Vorzügen der Verwendung eines Graphen und der Verständlichkeit der Wegführung für den Benutzer gemacht werden.

3.4 Architektur

Als Bestandteil des mExplorers verwendet das Navigationssystem natürlich die bereits bestehende Client-Server Architektur. Die Kommunikation zwischen Client und Server findet asynchron auf der Basis von Nachrichten statt. Die Architektur des Servers, wie auch des Clients, ist geschichtet. Der Client weist drei Schichten auf: Die Netzwerk-, die Logik- und die Präsentationsschicht. Der Server besteht aus vier Schichten. Neben den Schichten, die auch der Client enthält, hat er zudem noch eine Datenschicht. Ausführlicher wird die Architektur des mExplorers in [Bru05] geschildert.

Das entwickelte Navigationssystem selbst umfasst zwei unabhängige Module:

Navigationseditor: Der Navigationseditor kommt beim Erstellen des Spiels zum Zuge und erlaubt es, die verwendeten Karten mit den für die Navigation nötigen Informationen über die gehbaren Wege zu versehen. Er ist in den bereits bestehenden Spieleditor des mExplorers integriert.

Navigationsmodul: Das Navigationsmodul wird auf dem Client ausgeführt. Es stellt die zur Navigation nötigen Benutzerschnittstellen bereit, berechnet den kürzesten Weg und übernimmt die Wegführung des Benutzers.

Im Weiteren wird nun die Architektur der beiden einzelnen Module des Navigationssystems erläutert.

3.4.1 Der Navigationseditor

Der Navigationseditor erlaubt es, die Daten der gehbaren Wege zu generieren, verwaltet und präsentiert sie. Daher erstrecken sich die einzelnen Klassen des Navigationseditors über alle drei Schichten.

Die Abbildung 29 zeigt einen Überblick über die Einordnung der einzelnen Klassen des Navigationseditors in die geschichtete Architektur des Editors des mExplorers.

Die Klassen „Node“ (Knoten) und „Edge“ (Kante) dienen zur Repräsentation der gehbaren Wegen und sind somit auf der Datenschicht angesiedelt. Die Klasse „Graph“ verwaltet und bearbeitet die erstellten Knoten und Kanten und ist somit Bestandteil der Logikschicht. Sie wurde der Klasse „Game“ als neues Attribut hinzugefügt. Die Präsentation wurde dem bestehenden Editor angepasst. So wurde die Präsentation des Navigationseditors wie auch die Steuerung der entsprechenden GUI-Komponenten in die Klasse „GuiThread“ integriert.

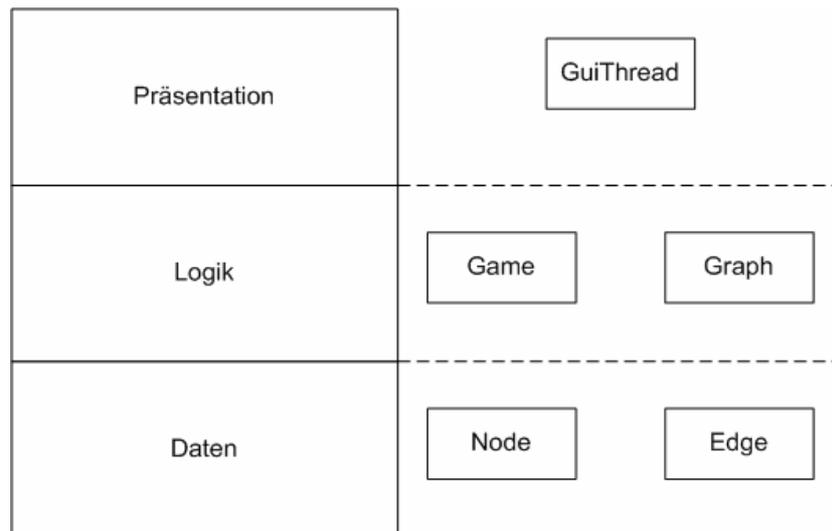


Abbildung 29: Einordnung der Klassen des Navigationseditors in die geschichtete Architektur

3.4.1.1 Die einzelnen Klassen des Navigationseditors im Überblick

Die folgenden Erläuterungen zu den einzelnen Klassen des Navigationseditors geben einen Überblick über den Aufbau und die Funktionalität der einzelnen Klassen. Dabei werden die einzelnen Klassen aber nicht vollumfänglich erläutert, sondern nur auf die für das Verständnis relevanten Attribute und Methoden eingegangen. Methoden, die innerhalb der Klassen für Umwandlungen oder für den Zugriff auf private Attribute verwendet werden, werden daher ausgeblendet. Interessierte können jedoch die vollumfänglichen Klassen im Java-Doc oder dem Source Code selbst einsehen. Das Java-Doc wie auch der Source Code sind auf der beiliegenden CD enthalten.

Node: „Nodes“ oder zu Deutsch Knoten repräsentieren die einzelnen Wegpunkte der gehbaren Wege. Sie befinden sich an speziellen Punkten der gehbaren Wege, so beispielsweise an Kreuzungen oder den Endpunkten der Wege.

Knoten weisen ein Attribut „**position**“ vom Typ „Position“ auf, das dessen Position angibt. Die Position wird durch eine X- und eine Y-Koordinate, sowie einen eindeutigen Kartennamen, gekennzeichnet. Des Weiteren haben Knoten ein Attribut „**distanceToStart**“ vom Typ „int“. Dieses Attribut dient zum Festhalten der Distanz zum Startpunkt beim Berechnen der kürzesten Wege. Das Attribut „**connectedToMaps**“ ist vom Typ Vector und speichert die Namen der Karten zu denen der Knoten Verbindungen aufweist.

Edge: Edges oder zu Deutsch Kanten sind Verbindungen zwischen zwei Knoten. Sie repräsentieren gewissermassen die gehbaren Wege, also die Wege, auf denen sich der Benutzer bewegen kann.

Kanten weisen zwei Attribute „**node1**“ und „**node2**“ vom Typ „Node“ auf. Sie sind der Anfangs- beziehungsweise der Endpunkt der Kante. Da sämtliche Kanten bidirektional sind, spielt die Reihenfolge der Knoten keine Rolle. Jede Kante muss immer einen echten Anfangs- und Endpunkt haben, daher müssen beide Attribute Werte ungleich „null“ aufweisen.

Kanten haben auch eine Länge, die im Attribut „**length**“ vom Typ „int“ festgehalten wird. Der Wert dieses Attributs muss bei jeder Verschiebung eines der beiden Knoten nachgetragen werden. Einzige Ausnahmen sind Kanten, die sich von einer Karte zu einer andern erstrecken. Dies wird durch ein Attribut „**overMaps**“ vom Typ „boolean“ angezeigt.

Graph: Diese Klasse enthält die gesamte Logik der gehbaren Wege. Sie erstellt, verwaltet und löscht sämtliche Knoten und Kanten. Da es in einem Spiel nur eine Instanz dieser Klassen geben darf, ist sie als Singleton implementiert. Für die Darstellung der Knoten und Kanten werden jeweils nur diejenigen der aktuell angezeigten Karte benötigt, daher werden sie in den beiden Hashtables „**nodesOfMap**“ und „**edgesOfMap**“ in Vektoren gespeichert. So können die Knoten und Kanten der jeweiligen Karte rasch und effizient bereitgestellt werden.

Werden Knoten und Kanten erstellt, so werden sie in die entsprechenden Vektoren eingetragen. Werden Kanten gelöscht, so werden sie dementsprechend entfernt. Bei Knoten, die entfernt werden, reicht es nicht einfach den Knoten zu löschen. Zuerst werden alle Kanten, die von diesem Knoten ausgehen gelöscht und erst dann der entsprechende Knoten.

Über die Methode „**shortestPath(Node pos, Node target)**“, die als Parameter den Start- und den Zielknoten hat, kann der kürzeste Weg zwischen diesen beiden Knoten berechnet werden. Dazu müssen aber alle Knoten zusammenhängend sein. Um dies zu überprüfen, verfügt die Klasse „Graph“ über die Methode „**isGraphCoherent()**“. Diese Methode überprüft den Graphen darauf, ob alle Knoten zusammenhängend sind. Als Rückgabewert liefert diese Methode einen String mit den Namen aller Knoten, die nicht Bestandteil des Graphen sind.

GuiThread: Diese Klasse ist die zentrale Instanz der Präsentationsschicht. Sie ist für die Darstellung wie auch die Handhabung der GUI-Komponenten verantwortlich. Sie wurde nicht neu erstellt, sondern lediglich erweitert. Die Erweiterungen betreffen die Darstellung wie auch die Steuerung des neu hinzugefügten Navigationseditors.

3.4.1.2 Zusammenspiel der einzelnen Klassen des Navigationseditors

Die Abbildung 30 zeigt ein vereinfachtes Klassendiagramm der Klassen des Navigationseditors. Dieses Diagramm ist bewusst kein vollständiges Klassendiagramm sondern umfasst nur die für das Verständnis des Navigationseditors nötigen Klassen. Es soll das Zusammenwirken der einzelnen Klassen verdeutlichen, nicht aber die Gegebenheiten in allen Details wiedergeben. Zur besseren Übersicht wurde auf die Angabe der Attribute und Methoden der einzelnen Klassen verzichtet. Interessierte können dies im Java Doc oder direkt im Source Code nachsehen. Das Java Doc und der Source Code sind auf der beiliegenden CD enthalten.

Die Beziehung zwischen der Klasse „Node“ (Knoten) und der Klasse „Position“ ist eine Aggregation. Dies bedeutet, dass die eine Position ein Bestandteil eines Knotens ist. Weitere Beziehungen vom Typ „Aggregation“ sind zwischen den Klassen „Edge“ und „Node“, „Graph“ und „Node“ wie auch „Graph“ und „Edge“ zu finden. Kanten enthalten also Knoten, und Graphen enthalten sowohl Knoten wie auch Kanten.

Die Klasse „Graph“ enthält also Knoten und Kanten. Zwischen der Klasse „Graph“ und der Klasse „Game“ besteht ebenfalls eine Aggregations-Beziehung, speziell daran ist jedoch, dass eine Instanz vom Typ „Game“ genau eine Instanz vom Typ „Graph“ enthält. Zwischen den Klassen „Game“ und „GuiThread“ besteht eine Beziehung vom Typ „Benutzung“. „GuiThread“ benutzt Attribute und Methoden der Klasse „Game“, um seine Funktionalität zu erbringen.

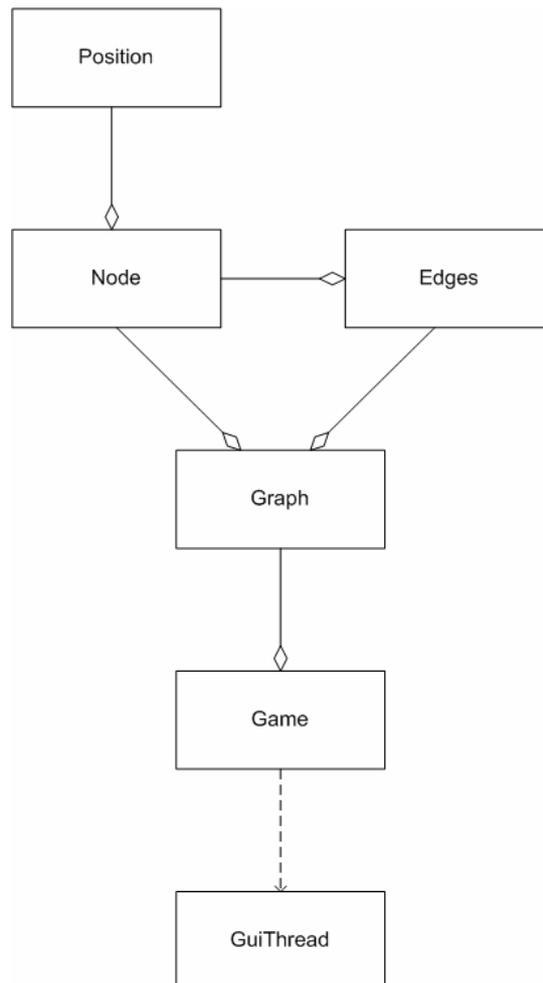


Abbildung 30: Klassendiagramm des Navigationseditors

3.4.2 Das Navigationsmodul

Das Navigationsmodul wird zur Laufzeit auf dem Client ausgeführt. Es stellt die Benutzerschnittstelle für die Navigation zur Verfügung, berechnet den kürzesten Weg und ist für die Wegführung des Benutzers verantwortlich. Das Navigationsmodul selbst erstreckt sich lediglich über die Logik- und die Präsentationsschicht des mExplorer Clients. Es ist aber abhängig von dessen Netzwerk- und dessen Daten-Schicht. Meldet sich der Client beim Server an, so übermittelt ihm dieser sämtliche Daten des Spiels über die Netzwerkschicht und somit auch die Informationen über die gehbaren Wege. Von der Datenschicht ist das Navigationsmodul insofern abhängig, dass es die von den „SensorReader“ gelesenen und vom „RecordStore“ gespeicherten Daten für die Richtungsangaben benötigt.

Die Abbildung 31 zeigt einen Überblick über die einzelnen Klassen des Navigationsmoduls und deren Einordnung in das Schichtenmodell des mExplorer Clients.

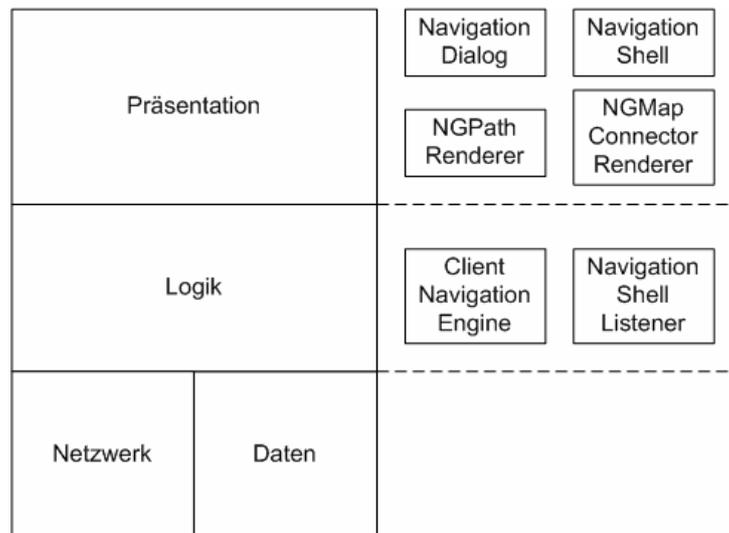


Abbildung 31: Einordnung der Klassen des Navigationsmoduls in die geschichtete Architektur

Die zentrale Klasse des Navigationsmoduls ist die „ClientNavigationEngine“. Sie umfasst die gesamte für die Navigation benötigte Logik. Sie hat Kenntnis darüber, ob die Navigation an ist, wird über Änderungen der Blickrichtung informiert, berechnet die Richtungsangaben und leitet diese an die für die Anzeige verantwortlichen Klassen weiter.

Für die Anzeigen des Navigationsmoduls sind gleich mehrere Klassen verantwortlich. Der berechnete Weg wird, wie auch die Tasks und ähnliches, direkt auf die Karte gezeichnet. Daher wurde die Klasse „NGMapRenderer“, die die Karte auf dem „NGMapCanvas“ darstellt, dementsprechend erweitert und die dazu notwendigen Renderer, „NGPathRenderer“ und „NGMapConnectorRenderer“, erstellt. Die „NavigationShell“ ist eine zusätzliche Shell, die über das Display gelegt wird und die Anzeige der Richtungsangaben übernimmt. Die letzte Klasse des Navigationsmoduls ist der „NGNavigationDialog“. Sie stellt die graphische Benutzeroberfläche zur Steuerung der Navigation zur Verfügung.

3.4.2.1 Die einzelnen Klassen des Navigationsmoduls im Überblick

Die folgenden Erläuterungen zu den einzelnen Klassen des Navigationsmoduls geben einen Überblick über den Aufbau und die Funktionalität der einzelnen Klassen. Dabei werden die einzelnen Klassen aber nicht vollumfänglich erläutert, sondern nur auf die für das Verständnis relevanten Attribute und Methoden eingegangen. Methoden, die innerhalb der Klassen für Umwandlungen oder für den Zugriff auf private Attribute verwendet werden, werden daher ausgeblendet. Interessierte können jedoch die vollumfänglichen Klassen im Java-Doc oder dem Source Code selbst entnehmen. Das Java-Doc, wie auch der Source Code, ist auf der beiliegenden CD enthalten.

ClientNavigationEngine: Wie alle anderen Klassen der Logikschicht wurde die „ClientNavigationEngine“ von der Klasse „LogicModule“ abgeleitet. Als zentrale Klasse des Navigationsmoduls darf jeweils nur eine Instanz existieren und daher wurde sie als Singleton implementiert. Im Attribut „usedSensor“ vom Typ „int“ wird festgehalten, ob ein Sensor verwendet wird und um welchen es sich handelt, den anstelle des Sensorboards kann auch ein mit einem elektronischem Kompass ausgestatteter GPS-Empfänger verwendet werden. Der Wert wird bei der Initialisierung des Clients entsprechend dem Config-File gesetzt. Das Attribut „audio“ ist vom Typ „boolean“ und hält fest, ob der Benutzer die akustischen Richtungsangaben hören möchte oder nicht. Das Ziel der Navigation wird im Attribut „target“ gespeichert. Darin wird der dem gewünschten Ziel nächstgelegene Knoten des Graphen gespeichert und daher ist es auch vom Typ „Node“. Um der Sachlage gerecht zu

werden, dass auch dynamische Ziele in der Form des zu jagenden Teams gewählt werden können, gibt es zusätzlich ein Attribut „**teamTargetMovable**“ vom Typ „boolean“. Es zeigt an, ob das Ziel beweglich ist oder nicht. Je nach Wert dieses Attributs wird das Ziel bei Positionsänderungen des zu jagenden Teams neu berechnet oder nicht. Da der Graph sehr oft benötigt wird und zur Laufzeit auch nicht verändert wird, wird er bei der Initialisierung aus dem „GameStore“ ausgelesen und intern im Attribut „**graph**“ gehalten. Da die Farbe der Anzeige des berechneten Weges wählbar ist, bedarf es eines Attributs, das jeweils die aktuell gewählte Farbe speichert. Dazu wird das Attribut „**pathColor**“ vom Typ „Color“ verwendet. Die Methode „**calculateNextPossibleTargetToPreferredDestinations(PositionInformation preferredDestination)**“ berechnet den der gewünschten Position nächstgelegenen Knoten des Graphen. Dazu vergleicht sie die Positionen der einzelnen Knoten des Graphen mit der übergebenen „PositionInformation“ und liefert den nächstgelegenen Knoten zurück. Die Methode „**calculatePath()**“ berechnet anhand der aktuellen Position und dem Ziel des Teams den exakten Pfad. Für die grobe Berechnung ruft sie ihrerseits die Methode „**shortestPath(Node pos, Node target)**“ der Klasse „Graph“ auf. Das erhaltene Resultat wird dann aber zusätzlich noch entsprechend der in Kapitel 3.3.2 beschriebenen harmonischen Wegführung angepasst. Je nach dem, wie die aktuelle Position im Bezug auf den ersten, respektive zweiten Knoten liegt, wird die aktuelle Position direkt hinzugefügt oder vorgängig der erste Knoten noch entfernt. Zur Berechnung der Richtungsangaben steht die Methode „**calculateDirection()**“ zur Verfügung. Zur Berechnung werden die aktuelle Position, der als nächstes anzulaufenden Knoten, sowie die Blickrichtung benötigt. Die aktuelle Position und der nächste Knoten sind die ersten beiden Elemente des berechneten Weges und werden daraus bezogen. Mittels des Skalarprodukts wird der Winkel zwischen dem Vektor der Blickrichtung und dem Vektor zwischen der aktuellen Position und dem nächsten Knoten des Weges berechnet. Die Grösse des Winkels gibt an, wie fest sich der Benutzer drehen soll oder ob die beiden Richtungen übereinstimmen. Zusätzlich wird bestimmt, ob sich der Benutzer bei einer nötigen Drehung nach links oder rechts drehen muss. Am Anfang der Methode wird zuerst noch überprüft, ob das Ziel bereits erreicht wurde. Ist dies gegeben, so wird die Richtung gar nicht erst berechnet, sondern dem Benutzer direkt mitgeteilt, dass er sein Ziel erreicht hat. Die Überprüfung wird mittels der Methode „**isTargetReached()**“ vorgenommen. Diese überprüft, ob der als nächstes anzulaufende Knoten mit dem Ziel übereinstimmt und ob die Entfernung kleiner als der bestimmte Schwellenwert ist.

NavigationShell: Die „NavigationShell“ wurde nach dem Vorbild der „SensorShell“ von Daniel Suter erstellt und dient zur Visualisierung der Richtungsangaben. Die Methode „**getArrowImage(String text)**“ wählt entsprechende dem übergebenen String das Bild aus und übergibt dieses der Methode „**imageToRegion(Image image)**“ zur weiteren Aufbereitung. Diese wiederum färbt die entsprechende Region der „NavigationShell“ ein. Die „NavigationShell“ selbst wird über das Display gelegt. Da abgesehen von der Region der Richtungsangabe die „NavigationShell“ transparent ist, bleibt der Rest der Kartenansicht trotz der unabhängigen Anzeige der Richtungsangaben sichtbar.

NGPathRenderer: Der „NGPathRenderer“ übernimmt das Zeichnen des berechneten Weges auf der Karte. Wie alle anderen Renderer, die Objekte auf der Karte darstellen, ist er von der Klasse „NGMapObjectRenderer“ abgeleitet. Er zeichnet den Weg Stück für Stück mittels der Methode „**draw(GC gc, PositionInformation p1, PositionInformation p2, Color color, String map)**“. Die beiden übergebenen Parameter p1 und p2 entsprechen dem Anfangsbeziehungsweise dem Endpunkt eines Wegabschnittes. Der Parameter „**color**“ gibt an, in welcher Farbe der Weg gezeichnet werden soll. Der übergebene String entspricht dem Namen der Karte, auf der der Weg gezeichnet werden soll. Anhand dieses Parameters wird

sichergestellt, dass nur Wegabschnitte gezeichnet werden, die auch wirklich auf dieser Karte liegen.

NGMapConnectorRenderer: Auch diese Klasse ist von der Klasse „NGMapObjectRenderer“ abgeleitet. Sie dient zum Zeichnen der Punkte des Weges von denen aus der Weg auf eine andere Karte führt. Dies wird durch die Methode „**draw(GC gc, PositionInformation p1, String text, Color color, String map)**“ bewerkstelligt. Der Parameter p1 gibt an, an welcher Stelle die Kartenverbindung gezeichnet werden soll. Die Parameter „**map**“ und „**color**“ erfüllen denselben Zweck wie beim „NGPathRenderer“. Der übergebene String enthält den Namen der Karte, auf der der Weg weiter geht. Diese wird aus Gründen der Übersicht aber nur dargestellt, wenn die Karte im Zoom-Modus betrachtet wird.

NGNavigationDialog: Diese Klasse stellt die graphische Benutzeroberfläche zur Steuerung der Navigation bereit. Sie umfasst mehrere Steuerelemente und ordnet sie entsprechend ihrer Anzahl optimal auf dem Display an. Ausführlich besprochen wird diese Benutzeroberfläche im Kapitel 3.5.2.1.

NavigationShellListener: Diese Klasse dient zur Benachrichtigung der NavigationShell über neue Richtungsangaben über das Observer/Observable Patern.

Die Klassen „NGApplicationController“, „NGMapCanvas“ und „NGMapRenderer“ wurden dementsprechend angepasst, dass sie die neuen Renderer beziehungsweise den neuen Dialog zum gegebenen Zeitpunkt aufrufen.

3.4.2.2 Zusammenspiel der einzelnen Klassen des Navigationsmoduls

Das in der Abbildung 32 dargestellte Klassendiagramm veranschaulicht das Zusammenspiel der einzelnen Klassen des Navigationsmoduls. Es enthält die einzelnen Klassen des Navigationsmoduls wie auch einige weitere für das Verständnis notwendige Klassen. Auf die Angabe von Attributen und Methoden der Klassen wurde bewusst verzichtet, da sie das Diagramm nur unnötig vergrößern und nicht weiter zum Verständnis beitragen.

Die zentrale Klasse ist die „ClientNavigationEngine“, sie enthält die für die Navigation benötigte Logik. Als Logikklasse ist sie von der Klasse „LogicModul“ abgeleitet und erbt von dieser das Attribut „GameStore“. Über die Klasse „GameStore“ kann auf sämtliche Spieldaten zugegriffen werden, so auch auf den Graphen der gehbaren Wege. Ebenso wird hier die durch Positionsänderungen der einzelnen Teams ausgelösten Events bearbeitet.



Abbildung 32: Klassendiagramm des Navigationsmoduls

Der Zugriff auf alle Logikklassen erfolgt zentral über die „ClientLogicFacade“. Sie hat eine Aggregations-Beziehung zur „ClientNavigationEngine“, sie enthält also eine „ClientNavigationEngine“. Die Klassen „NGApplicationController“, „NGMapCanvas“ und „NGMapRenderer“ benutzen zur Erbringung ihrer Aufgaben die Funktionalitäten der „ClientNavigationEngine“. Der Zugriff darauf erfolgt, wie bereits erwähnt, indirekt über die „ClientLogicFacade“.

Die Klassen „NGApplicationController“ und „NGMapCanvas“ benutzen gegenseitig die Funktionalität des Anderen zur Leistungserbringung. Der „NGApplicationController“ benutzt ebenfalls die Klasse „NGNavigationDialog“. Diese ist als Dialog von der allgemeinen Klasse „NGDialog“ abgeleitet.

Über den seriellen Port liest die Klasse „SensorReader“ die Daten des Sensorboards ein. Die Weiterleitung der Sensordaten an die „ClientNavigationEngine“ erfolgt über den „ListenerStore“ mittels des Observer/Observable-Patern. Die „ClientNavigationEngine“ registriert sich beim „ListenerStore“, so dass er sie über neue Sensordaten informiert. Treten die Daten auf, so feuert der „SensorReader“ über den „ListenerStore“ einen Event, der an alle registrierten Listener gesendet wird. Genau nach demselben Prinzip wird die

„NavigationShell“ von der „ClientNavigationEngine“ über neue Richtungsangaben informiert. Für das Verwalten der registrierten Listener und das Versenden von ausgelösten Events ist die Klasse „NavigationShellListener“ zuständig.

Die Darstellung der Karte übernimmt die Klasse „NGMapCanvas“. Zum Zeichnen der Karte benutzt sie jedoch die Funktionalität der Klasse „NGMapRenderer“. Diese wiederum greift zum Zeichnen der einzelnen Objekte der Karte auf die verschiedenen, speziell darauf ausgerichteten, Renderer zurück. Die speziellen Renderer des Navigationsmoduls sind der „NGMapConnectorRenderer“ und der „NGPathRenderer“. Sie werden beide von der Klasse „NGMapObjectRenderer“ abgeleitet.

3.5 Benutzerinteraktion

Die Kommunikation zwischen dem Anwender und dem System wird auch als Benutzerinteraktion bezeichnet. Sie kann vom Benutzer oder auch vom System ausgelöst werden. Eine vom Anwender initialisierte Benutzerinteraktion ist beispielsweise das Drücken eines Buttons oder eines anderen Steuerelements. Eine vom System ausgelöste Benutzerinteraktion ist unter anderem die Präsentation eines Ereignisses an den Benutzer.

Die beiden Kapitel 3.5.1 und 3.5.2 erläutern, welche Funktionalität das Navigationssystem dem Benutzer bereitstellt und wie er darauf zugreifen kann. Das Kapitel 3.5.3 erörtert abschliessend, wie der Benutzer vom Navigationssystem auf seinem Weg ans Ziel geführt wird. Im ersten Schritt, der Entwicklung der graphischen Benutzeroberfläche zur Benutzerinteraktion, wurden Anwendungsfälle für die möglichen Aktionen des Benutzers erstellt. Anhand dieser wurde das Konzept der Benutzeroberfläche auf Mängel überprüft. Die Anwendungsfälle sind im Kapitel F des Anhangs zu finden.

3.5.1 Benutzerinteraktion des Navigationseditors

Die Benutzerinteraktionen des Navigationseditors sind darauf ausgerichtet, den Graphen der gehbaren Wege zu erstellen. Dieser wird vorgängig zum Spiel erstellt und ist während dem Spiel fix, folglich sind Benutzerinteraktionen im Navigationseditor auch nur während dem Erstellen des Spiels möglich, nicht aber zur Laufzeit.

3.5.1.1 Harmonische Integration in das bestehende System

Der Navigationseditor wurde so sanft wie möglich in den mExplorer integriert, bereits bestehende Funktionalität und Abläufe wurden so wenig wie möglich verändert.

Wie in der Abbildung 25 zu sehen ist, ist der Graph der gehbaren Wege sehr dominant und kann sich somit störend auf das Erstellen der restlichen für das Spiel benötigten Elemente, wie Tasks oder POI, auswirken. Um dem entgegenzuwirken, lässt sich der Navigationseditor ein- und ausschalten. Der Graph wird nur bei eingeschaltetem Navigationseditor angezeigt. Bei ausgeschaltetem Navigationseditor ist die Handhabung des mExplorers nahezu identisch wie zuvor.

3.5.1.2 Funktionalität der Menüleiste

Die wichtigsten Funktionen des Navigationseditors wurden direkt in die Menüleiste des mExplorers integriert, so dass der Benutzer leicht und schnell auf sie zugreifen kann.

Die für die Buttons verwendeten Icons sind so gewählt, dass sie die dahinter verborgenen Funktionen möglichst gut wiedergeben. Für Benutzer, die das erste Mal damit arbeiten, ist zusätzlich ein Hilfetext hinterlegt, mit dem auch ihnen die Funktionalität klar wird.

Die Abbildung 33 zeigt einen Ausschnitt der Menüleiste des mExplorers. Die Bedienelemente des Navigationseditors sind dabei graphisch durch die rote Umrahmung hervorgehoben.



Abbildung 33: Integration der wichtigsten Funktionen in die bestehende Menüleiste

Die vier in der Menüleiste bereitgestellten Funktionen sind:



Navigation An/Aus: Über diesen Button kann der Navigationsmodus beziehungsweise ausgeschaltet werden. Je nachdem, ob der Editor an oder aus ist, zeigt der Button „Nav On“ oder „Nav Off“ an.

Da jeder Knoten genau einer Position auf einer Karte zugeordnet wird, lässt sich der Editor nur einschalten, wenn im Spiel auch tatsächlich eine Karte vorhanden ist.



Knoten erstellen: Durch Anwählen wird dieser Button aktiviert und bleibt aktiv, bis er abermals gedrückt wird. Bei aktivem Button können Knoten ganz einfach durch einen Klick auf die entsprechende Stelle der Karte erstellt werden. Durch die Arretierfunktion lassen sich auch viele Knoten innerhalb kurzer Zeit erstellen.



Kanten erstellen: Dieser Button wird analog zum „Knoten Button“ durch einmaliges Drücken aktiviert und durch erneutes Drücken deaktiviert. Ist er aktiviert, so lassen sich Kanten erstellen, indem man zuerst den einen Knoten und danach den zweiten Knoten der zu erstellenden Kante durch Klick auf die Karte auswählt.



Graph prüfen: Bei grossen Graphen, die sich über mehrere Karten erstrecken, verliert man leicht den Überblick darüber, ob alle Knoten in den Graphen eingebunden sind oder ob man einzelne Knoten oder Teilgraphen vergessen hat. Daher stellt dieser Button die Möglichkeit zur Verfügung, den Graphen zu prüfen. Als Resultat der Prüfung werden sämtliche nicht eingebundenen Knoten oder die Meldung über die Korrektheit des Graphen ausgegeben.

Dies sind die grundlegenden Funktionen des Navigationseditors. Sie reichen aus, um den für die Navigation benötigten Graph der gebaren Wege zu erstellen und werden mit Abstand am häufigsten benutzt.

3.5.1.3 Weitere Funktionalität

Neben der grundlegenden Funktionen der Menüleiste stellt der Navigationseditor noch weitere zur Verfügung. Manche von ihnen sind aufgrund ihrer seltenen Anwendung, andere wiederum aufgrund der Art ihrer Anwendung, nicht in der Menüleiste platziert.

Werden Knoten oder Kanten fälschlicherweise gesetzt, so können sie natürlich auch wieder gelöscht werden. In das entsprechende Menü gelangt man durch Rechtsklick auf der Karte.

Wird der Rechtsklick direkt über einem Knoten ausgeführt und man wählt „Delete Node“ so wird der Knoten direkt gelöscht. Befindet sich jedoch kein Knoten an der entsprechenden Stelle, so gelangt man in einen Dialog, in dem man den zu löschenden Knoten durch seine ihn identifizierende Nummer angeben kann. Soll eine Kante gelöscht werden, so gelangt man jedes Mal in einen Dialog, dabei müssen die beiden Knoten der zu löschenden Kante angegeben werden. Die Abbildung 34 zeigt die beiden Dialoge zum Löschen von Knoten und Kanten, wie auch das Menü über das sie aufgerufen werden können.



Abbildung 34: Menü und Dialoge zum Löschen von Knoten und Kanten

Knoten können zudem auch auf der Karte verschoben werden. Dazu klickt man auf sie und zieht sie mit gedrückter Maustaste an die neue Position. Nach dem Verschieben werden ihre Position und die Längen aller Kanten, die von ihm ausgehen, aktualisiert.

Durch alle seine nun beschriebenen Funktionalitäten erlaubt es der Navigationseditor dem Benutzer, auf einfache aber dennoch effektive Weise, die gehbaren Wege im Spiel anzugeben und somit die Grundlage zur Navigation zu schaffen.

3.5.1.4 Zustände und Zustandsübergänge des Navigationseditors

Der Navigationseditor verfügt über verschiedene Zustände, wie beispielsweise „Knoten erstellen“ oder „Kanten erstellen“. Die Verfügbarkeit der einzelnen Funktionen ist abhängig vom jeweiligen Zustand des Navigationseditors. Zustandsübergänge werden durch die in den beiden vorherigen Kapiteln 3.5.1.2 und 3.5.1.3 beschriebenen Benutzerinteraktionen ausgelöst.

In der Abbildung 35 sind die einzelnen Zustände und Zustandsübergänge des Navigationseditors als Statechart veranschaulicht. Wird der Navigationseditor aktiviert, so gelangt man in den Ausgangszustand. Durch betätigen des Knoten-Buttons wechselt man in den „Knoten erstellen“-Zustand. Dieser Zustand erlaubt einfaches Erstellen von Knoten durch bloße Mausklicks auf die jeweilige Karte, dafür sind die meisten andern Funktionen inaktiv.

Ähnlich ist es beim „Kanten erstellen“-Zustand. Er erlaubt es, Kanten durch anklicken von gesetzten Knoten zu erstellen. Beim Anwählen des ersten Knotens wechselt der Zustand zu „erster Knoten selektiert“ und beim Anwählen des zweiten Knotens wieder zurück.

Das Verschieben von Knoten, wie auch das Löschen von Knoten und Kanten, ist nur im Ausgangszustand möglich. Wählt man Knoten- oder Kantenlöschen, so wird ein Zustandsübergang zum jeweiligen Zustand ausgelöst und der entsprechende Dialog wird geöffnet. Nach korrekter Angabe des zu löschenden Knotens bzw. der beiden Knoten der zu

löschenden Kante, wird das Löschen ausgeführt und man gelangt zurück in den Ausgangszustand. Das Verschieben eines Knotens bewirkt keinen Zustandsübergang, die Position des Knotens wird aktualisiert und der Navigationseditor bleibt im Ausgangszustand.

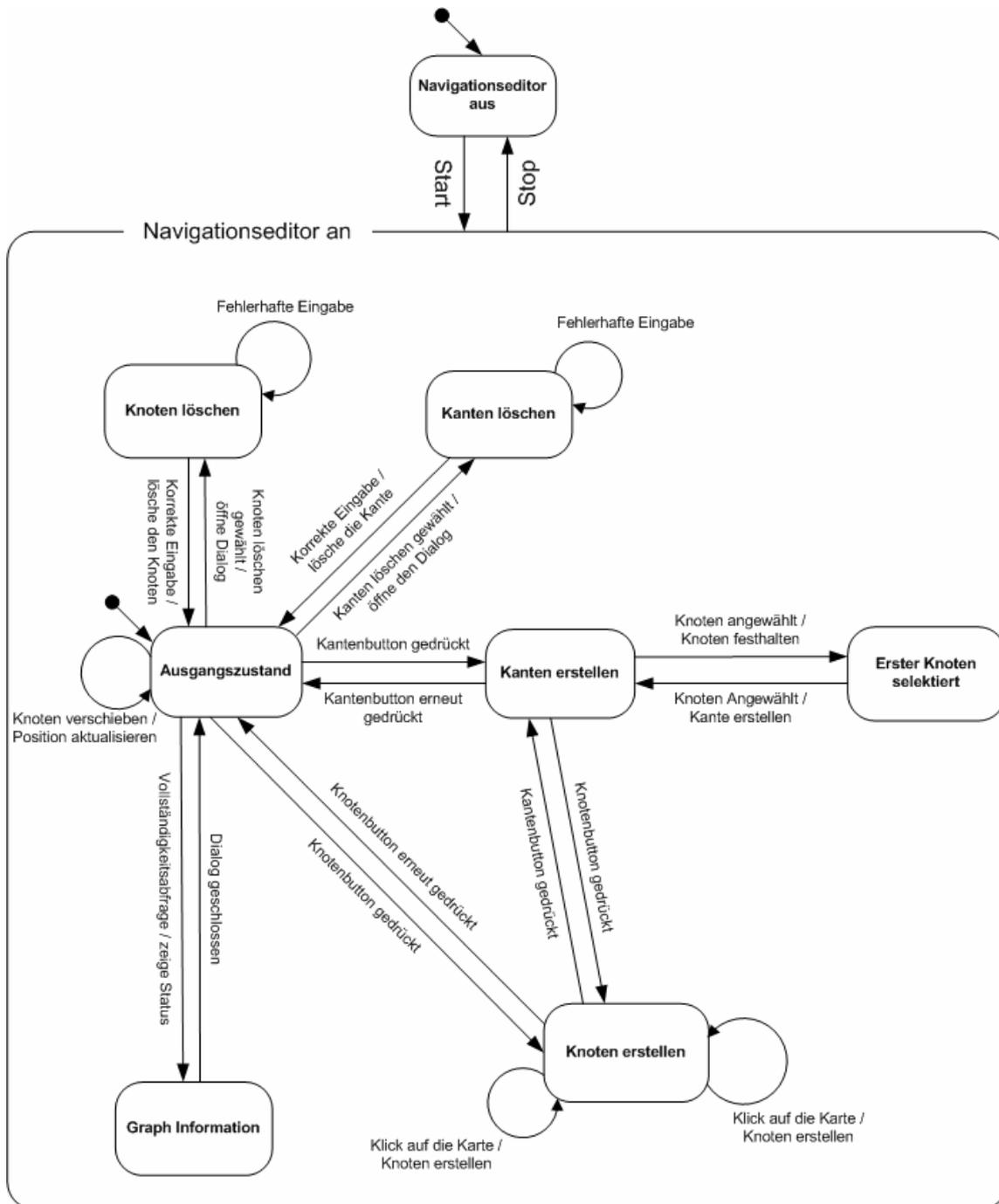


Abbildung 35: State Chart des Navigationseditors

Dieses selektive Aktivieren der einzelnen Funktionen mag auf den ersten Blick als hinderlich und einschränkend erscheinen. In Wahrheit erlaubt es jedoch eine effektivere Bedienung des Navigationseditors. Insbesondere die beiden Hauptfunktionen, das Erstellen von Knoten und Kanten, wird massiv vereinfacht. Durch die eigenständigen Zustände können Knoten und Kanten durch reine Mausklicks erstellt werden, ohne in Konflikt mit anderen Funktionalitäten zu kommen.

3.5.1.5 Visualisierung des Graphen

Als Reaktion auf die Eingaben des Benutzers wird der Graph vom Navigationseditor visualisiert. Dadurch ist es dem Benutzer möglich, zu überprüfen, ob der Graph wunschgemäß erstellt wurde. Kanten werden einheitlich als rote Linien dargestellt. Die Repräsentation der Knoten ist vom Typ des Knotens abhängig. Normale Knoten werden als rote Kreise dargestellt. Knoten, die Verbindungen zu anderen Karten aufweisen, werden als rote Pfeile visualisiert. Unabhängig von seinem Typ wird zu jedem Knoten die Nummer angezeigt, über die er angesprochen werden kann. Die Abbildung 36 zeigt einen Kartenausschnitt mit Kanten und den unterschiedlichen Knotentypen. Beim Knoten mit der Nummer 9 handelt es sich um einen normalen Knoten. Der Knoten mit der Nummer 5 ist ein Verbindungsknoten. Der Text neben ihm zeigt an, zu welchen Karten er verbunden ist. In diesem Fall ist die zu den Karten „ifi-neu-eg“ und „ifi-neu“. Der Knoten hat also zwei Verbindungen zu anderen Karten. Eine zum Stockwerk darunter und eine zum darüberliegenden. Der Knoten mit der Nummer 3 ist kein eigener Typ, aber seine Visualisierung zeigt dem Benutzer an, dass er als erster Knoten einer Verbindung selektiert wurde.

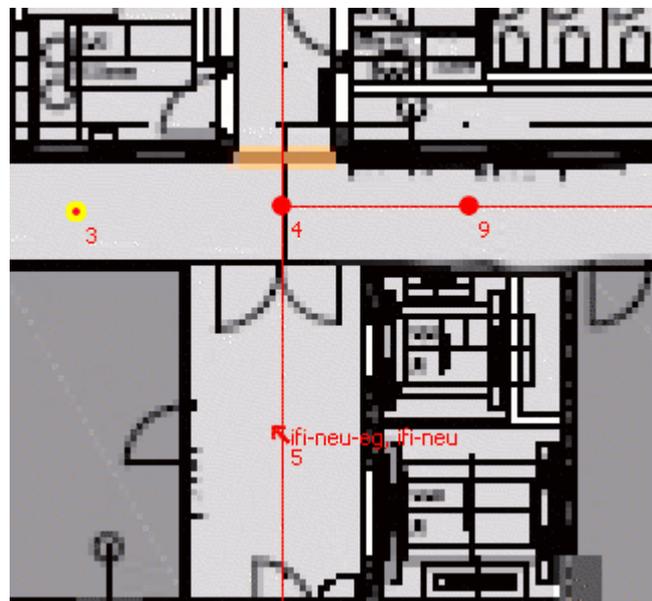


Abbildung 36: Kartenausschnitt mit unterschiedlichen Knoten

3.5.2 Benutzerinteraktion des Navigationsmoduls

Der mExplorer Client ermöglicht dem Benutzer den Zugriff auf die einzelnen Funktionen, wie „Chat“, „Annotation“ und anderem, über eine Steuerleiste, in der die entsprechenden Menüs aufgerufen werden können. Über diese Steuerleiste kann neu auch das Navigationsmenü aufgerufen werden.

3.5.2.1 Funktionalität des Navigationsmoduls

Das Navigationsmenü ist die zentrale Komponente für den Benutzer zur Steuerung der Navigation. Es erlaubt dem Benutzer den Zugriff auf alle Funktionen des Navigationsmoduls; diese sind:

- Start-/Stopp Navigation:** Die Navigation kann nach Belieben ein- oder ausgeschaltet werden.
- Wahl des Ziel:** Der Benutzer kann ein beliebiges Ziel auf der Karte oder falls Huntig auf dem Team-Level betrieben wird auch das gejagte Team als Ziel wählen. Wurde das Team gefangen, so ist diese Funktion inaktiv, bis ein neues zu jagendes Team zugeteilt wurde.
- Wahl der Farbe des Pfades:** Damit der zu gehende Weg auch gut auf der Karte zu erkennen ist und er sich optimal vom Rest der Karte abhebt, kann die Farbe individuell gewählt werden.
- Audio an/aus:** Die akustische Wegführung kann ein- oder ausgeschaltet werden.

In der Abbildung 37 ist das Navigationsmenü zu sehen. Es ist so gestaltet, dass es dem Benutzer leicht fällt, die gewünschte Funktion aufzurufen. Startet man die Navigation oder wählt man „Select new Target“ so gelangt man automatisch in die Kartenansicht des Spiels, wo man die gewünschte Karte auswählen kann und durch einen Klick auf die gewünschte Stelle das Ziel anwählt. Der Button „Hunted Team“ wird nur angezeigt, wenn Hunting auf dem Team-Level betrieben wird und das zu jagende Team auch eingeloggt ist. Wählt man diesen Button an, so gelangt man ebenfalls in die Kartenansicht zurück und der kürzeste Weg zum gejagten Team wird direkt angezeigt. Wählt man das farbige Kästchen neben Path Color an, so gelangt man in einen Dialog, in dem man die Farbe für den Pfad auswählen kann.



Abbildung 37: Navigationsmenü

3.5.2.2 Zustände und Zustandsübergänge des Navigationsmoduls

Genau wie beim Navigationseditor sind auch beim Navigationsmodul nicht immer alle Funktionen verfügbar sondern sind abhängig vom jeweiligen Zustand. Die Abbildung 38 veranschaulicht den Zusammenhang zwischen Zuständen und Zustandsübergängen des Navigationsmoduls. Vom Ausgangszustand gelangt man in den Zustand „Ziel wählen“ oder durch Wahl des zu jagenden Teams als Ziel direkt in den Zustand „Navigieren“. Im Zustand „Ziel wählen“ kann man zwischen den einzelnen Karten wechseln und durch einen Klick auf die entsprechende Karte das gewünschte Navigationsziel wählen. Ist ein Ziel angegeben, so wird der Weg berechnet und es findet ebenfalls der Zustandsübergang zu „Navigieren“ statt. Der Zustand „Navigieren“ ist der zentrale Zustand aus dem Übergänge in beinahe alle anderen Zustände möglich sind. In diesem Zustand können die Audiosignale der Navigation an- bzw. ausgeschaltet und auch die Farbe des berechneten Weges so angepasst werden, dass er sich optimal von der Karte abhebt. Wird die Farbwahl aktiviert, so wechselt der Zustand zu „Farbwahl“ und ein Dialog zur Wahl der Farbe öffnet sich. Nach der Wahl der gewünschten Farbe gelangt man wieder zurück in den Zustand „Navigieren“. Hat man das Ziel erreicht oder will man einfach ein neues Ziel wählen, so wechselt der Zustand wie zu Beginn zu „Ziel wählen“ und nach erfolgter Zielangabe wieder zu „Navigieren“. Der Wechsel von einem fixen zum gejagten Team als Ziel bewirkt keinen Zustandsübergang. Es wird lediglich das neue Ziel eingetragen und der Weg neu berechnet. Des Weiteren kann die Navigation auch deaktiviert werden, dabei wird der übergeordnete Zustand „Navigation an“ verlassen und ein Zustandsübergang zu „Navigation aus“ findet statt.

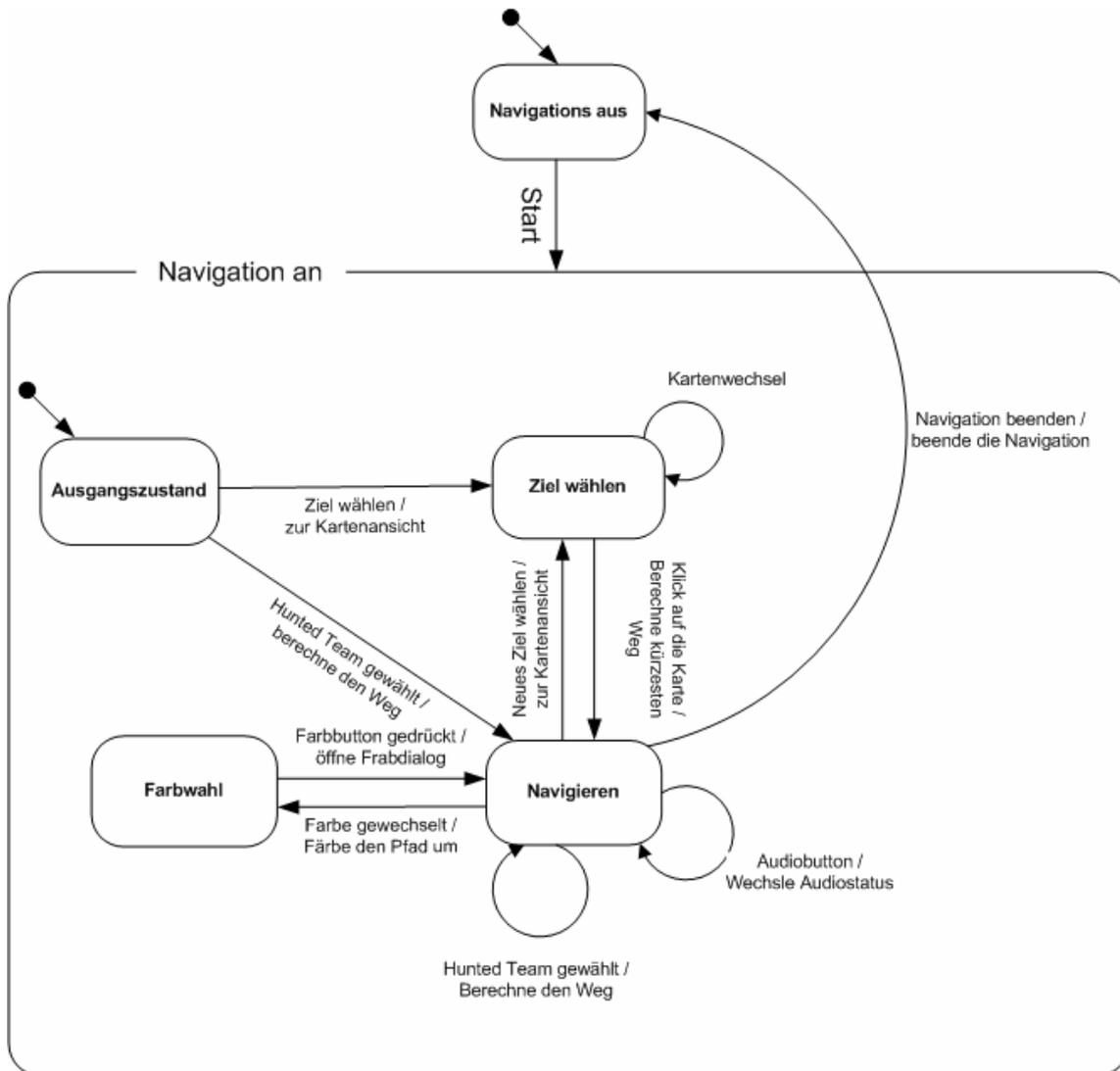


Abbildung 38: Statechart des Navigationsmoduls

3.5.3 Wegleitung

Nach erfolgter Zielangabe berechnet das System den kürzesten Weg ans Ziel und visualisiert ihn für den Benutzer. Neben dem visualisierten Weg erhält der Benutzer zusätzliche Informationen, die ihm helfen sollen, das Ziel schnellstmöglich zu finden.

Da die Navigation kein eigenständiges System ist, sondern Bestandteil des mExplorers, wurden diese weiteren Informationen auf das Nötigste beschränkt. Auf nebensächliche Informationen, wie Distanzangaben oder Nordrichtung, wurde verzichtet. Zum einen können nebensächliche Informationen den Benutzer vom Relevanten ablenken, zum anderen ist das verfügbare Display auch so klein genug. Übermäßige Präsentation von Navigationsinformationen hätte sich daher unweigerlich negativ auf den Rest des mobilen Spiels ausgewirkt.

3.5.3.1 Visualisierung des Weges

Die Visualisierung des berechneten Weges auf der Karte ist der zentrale Punkt der Navigation. Die Anzeige des Weges ist die primäre Informationsquelle für den Benutzer während der Navigation. Dadurch verfügt der Benutzer bereits über alle nötigen Informationen, um von seiner aktuellen Position bis ans Ziel zu gelangen. Durch den Abgleich der angezeigten Karte mit der Realität ist es ein Leichtes, dem angezeigten Weg zu folgen.

Die Abbildung 39 zeigt wie der berechnete Weg visualisiert wird. Links ist die Zoom-Ansicht der einen Karte zu sehen, rechts die Normalansicht der anderen. Die rote Aura repräsentiert die aktuelle Position und der rote Pfeil die aktuelle Blickrichtung. Die blaue Linie ist der visualisierte Weg ans Ziel.

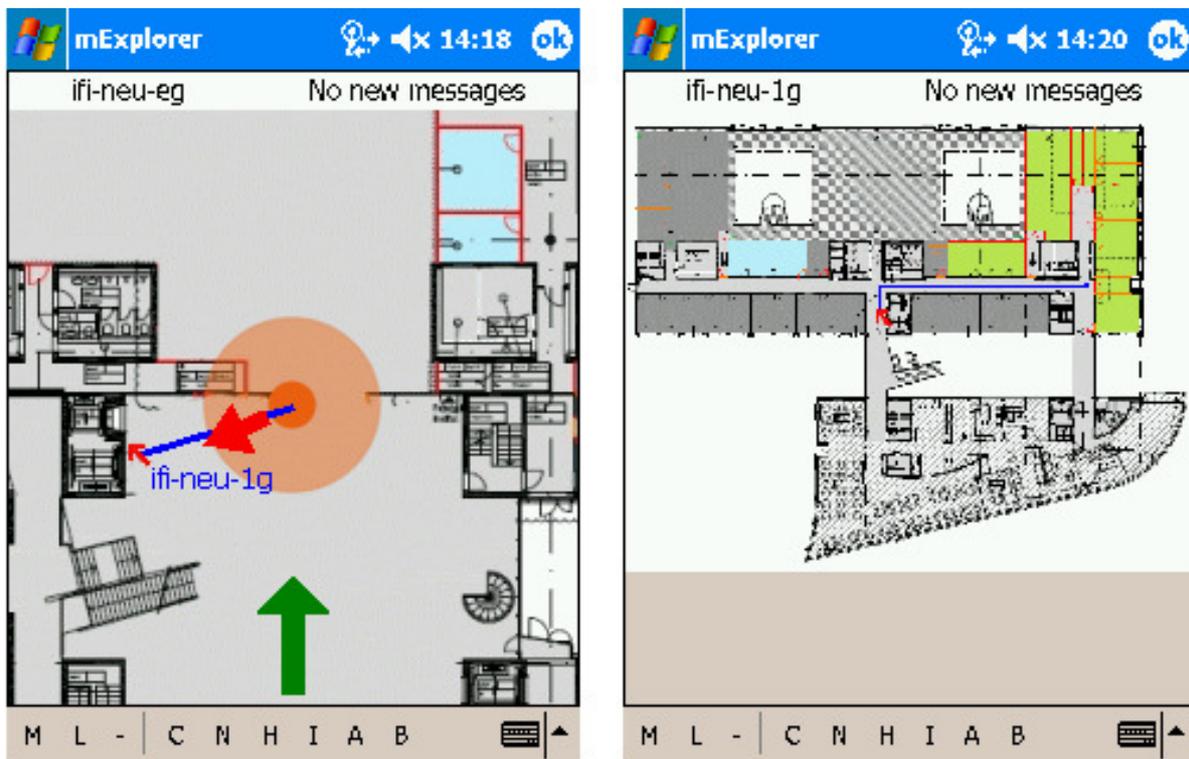


Abbildung 39: Visualisierung des berechneten Weges

Das Verfolgen des Weges auf einer Karte ist für den Benutzer einfach, etwas schwieriger ist es jedoch, wenn der Weg über mehrere Karten führt, denn wie im Kapitel 3.2.3 erläutert wird, kann sich das gesamte Spielgelände über mehrere Karten erstrecken. Muss an einem Punkt des Weges auf eine andere Karte gewechselt werden, so wird dies, analog zum Navigationseditor, durch einen roten Pfeil symbolisiert. Zu sehen ist dies in der Abbildung 39. Von der Karte links in der Abbildung 39 muss der Benutzer auf die Karte rechts in der Abbildung 39 wechseln, wo der Weg weiter geht. Benutzern, denen die Umgebung gänzlich unbekannt ist, ist es natürlich nicht möglich zu wissen, welche Übergänge zu welchen Karten führen. Daher wird jeweils bei einem Übergang von einer zu einer anderen Karte angegeben zu welcher Karte der Benutzer wechseln muss. Dies ist ebenfalls auf der linken Karte der Abbildung 39 zu sehen. Diese Beschriftung der Kartenübergänge wird aber nur vorgenommen, wenn der Benutzer sich im Zoom-Modus befindet. Bei normaler Kartenansicht ist die Schrift entweder zu klein als dass sie gelesen werden kann oder nimmt einen Grossteil der Karte in Anspruch.

3.5.3.2 Richtungsangaben

Zusätzlich zum visualisierten Weg erhält der Benutzer Richtungsangaben in akustischer, wie auch visueller Form. Auch ohne diese Richtungsangaben ist es dem Benutzer möglich, effizient ans Ziel zu gelangen. Sie sind als Ergänzung gedacht, als sekundäre Informationsquelle. Sie sind hilfreich für Anwender, denen es schwer fällt, sich anhand einer Karte zu orientieren. Durch die Richtungsangaben müssen sie sich nicht abmühen, die Karte auszurichten, bis sie mit den realen Gegebenheiten übereinstimmt.

Die Richtungsangaben sollten nicht nur klar verständlich und eindeutig sein, sie sollten auch so gewählt sein, dass der Benutzer sie auch umsetzen kann. Angaben, wie beispielsweise „um 70° nach links drehen“ können vom Anwender nicht umgesetzt werden und überfordern ihn nur. Daher wird nur zwischen 5 einfachen Richtungsangaben unterschieden. Es gibt eine Richtungsangabe für gerade aus und jeweils 2 für links und rechts. Die beiden Varianten beim links beziehungsweise rechts Drehen unterscheiden sich dadurch, wie fest der Benutzer drehen muss. Die Abbildung 40 veranschaulicht die Unterscheidung der einzelnen Richtungsangaben.

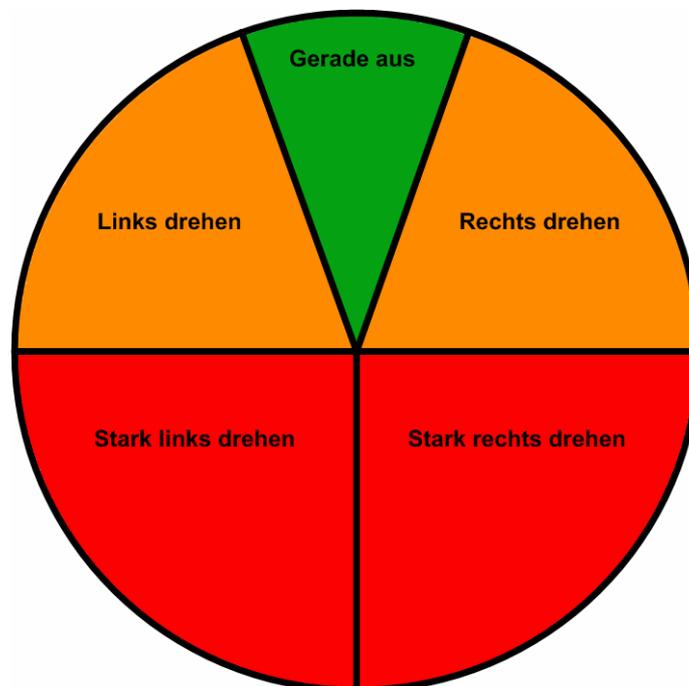


Abbildung 40: Unterscheidung der Richtungsangaben

Weicht die momentane Blickrichtung nicht mehr als 15° von der zu gehenden Richtung ab, so wird der Benutzer angewiesen, geradeaus zu gehen. Liegt der Winkel zwischen der Blickrichtung und der zu gehenden Richtung im Bereich zwischen 15° und 90° , so erhält der Anwender die Information, sich nach links bzw. nach rechts zu drehen. Ist der Winkel größer als 90° , so soll er sich stark links bzw. stark rechts drehen.

Die Richtungsangaben erfolgen sowohl akustisch wie auch visuell. Die visuellen Richtungsangaben werden im unteren Bereich des Displays dargestellt. Zu sehen ist sie im linken Teil der Abbildung 39. Der Benutzer blickt mehr oder weniger genau in die zu gehende Richtung und wird daher angewiesen, geradeaus zu gehen.

Die einzelnen, zur optischen Richtungsangabe verwendeten Symbole sind:



Gerade aus: Der gerade Pfeil weist den Benutzer an, geradeaus zu gehen. Zudem vermittelt ihm die grüne Färbung, dass die Richtung korrekt ist.



Rechts drehen: Der gebogene Pfeil nach rechts weist den Benutzer an, sich nach rechts zu drehen. Die orange Färbung vermittelt ihm zudem, dass seine momentane Richtung von der Richtung ans Ziel abweicht.



Links drehen: Der gebogene Pfeil nach links weist den Benutzer an, sich nach links zu drehen. Die orange Färbung vermittelt ihm zudem, dass seine momentane Richtung von der Richtung ans Ziel abweicht.



Ganz rechts drehen: Der enorm gebogene Pfeil nach rechts weist den Benutzer an, sich beträchtlich nach rechts zu drehen. Die rote Färbung vermittelt ihm zudem, dass die momentane Richtung beträchtlich von der Richtung ans Ziel abweicht.



Ganz links drehen: Der enorm gebogene Pfeil nach links weist den Benutzer an, sich beträchtlich nach links zu drehen. Die rote Färbung vermittelt ihm zudem, dass die momentane Richtung beträchtlich von der Richtung ans Ziel abweicht.

Durch die verwendeten Symbole, wie auch deren Färbung, ist es möglich, rasch zu erkennen, ob die Richtung stimmt oder inwiefern sie von der Richtung ans Ziel abweicht. Anhand der Pfeile lässt sich rechts oder links gut unterscheiden. Bei der Bestimmung der Grösse der Drehung erhält der Benutzer zusätzlich noch Informationen durch die Farbgebung der Pfeile. Die Farbwahl fiel nicht zufälligerweise auf die auch bei einer Ampel verwendeten Farben. Grün wird in unserem Kulturkreis generell mit richtig assoziiert, rot hingegen mit falsch. Orange ist auch nicht richtig, aber nicht ganz so falsch wie rot. Daher auch grün für „richtige Richtung“, orange für „kleine Abweichung“ und rot für „grosse Abweichung“.

Die akustischen Richtungsangaben erfolgen analog zu den visuellen. Zu jeder visuellen Richtungsangabe existiert das akustische Pendant. Die akustischen Richtungsangaben weisen den Anwender ebenfalls an, sich geradeaus zu begeben; links oder rechts fest oder weniger fest zu drehen. Diese akustischen Richtungsangaben sind jedoch viel weniger prägnant als ihre visuellen Gegenstücke. Bilder vermitteln viel mehr Information und zudem neigt natürliche Sprache dazu, nicht eindeutig zu sein.

Die primäre Komponente der Navigation ist ganz klar die Karte mit dem visualisierten Weg ans Ziel. Die Richtungsangaben, egal ob visuell oder akustisch, sind als sekundäre Komponente gedacht. Sie sollen dem Anwender die Navigation zusätzlich erleichtern, ihn aber nicht vollkommen eigenständig ans Ziel führen.

4 Benutzertests

Neben den zahlreichen technischen Tests, die während und nach der Implementierung durchgeführt wurden, fand ein Benutzertest statt. Dabei wurde das entwickelte Navigationssystem von echten Benutzern, die mit dem System nicht vertraut waren, unter realen Bedingungen getestet. Dadurch sollten der Nutzen, wie auch die Usability ermittelt werden.

4.1 Testsetup

In diesem Kapitel wird kurz auf die Gegebenheiten des Benutzertests eingegangen. Das Spielgebiet, die Testpersonen und die verwendete Technik haben direkten Einfluss auf den durchgeführten Benutzertest und sind daher für die in Kapitel 4.4 durchgeführte Testauswertung von Relevanz.

4.1.1 Spielgebiet

Der Benutzertest fand auf dem Gelände der Universität Zürich Irchel statt. Das Spielgebiet erstreckte sich dabei über vier unterschiedliche Karten. Die Abbildung 41 zeigt die einzelnen Karten des Spielgebiets. Die Karte 1. ist die Hauptkarte, sie umfasst den Hörsaalbereich, die Mensa und weitere interessante Orte. Die Karte 2. zeigt den Aussenbereich. Sie ist im Vergleich zu den restlichen Karten nicht sehr detailliert, deckt dafür aber den gesamten Aussenbereich ab. Die Karte 3. zeigt den Bereich der Informatikdienste. Die letzte Karte, die Karte 4. zeigt den Strickhof, in dem sich die Hauptbibliothek befindet.

Die meiste Zeit des Spiels bewegen sich die Spieler auf den detaillierten Karten 1., 3. und 4. Die grobe Übersichtskarte wird hauptsächlich dazu verwendet, um von einem Innenbereich zu einem andern zu gelangen.

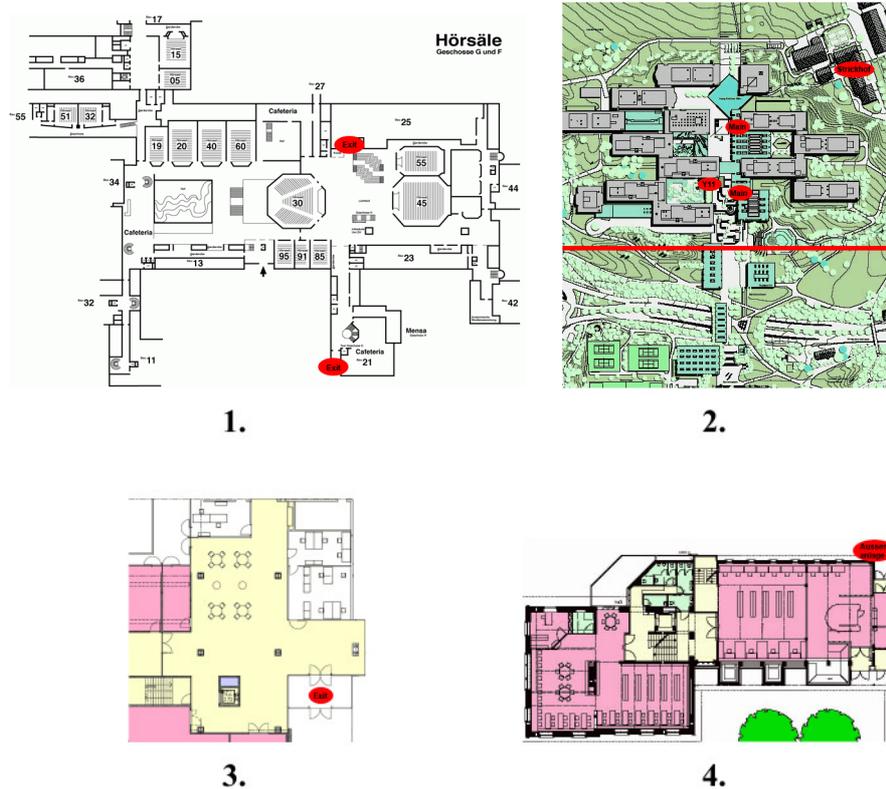


Abbildung 41: Die vier einzelnen Karten des Spielgebiets des Benutzertests

4.1.2 Testpersonen

Am Test nahmen insgesamt vierzehn Personen teil. Davon waren acht Personen männlich und sechs Personen weiblich. Die einzelnen Personen wiesen ein Alter zwischen 18 und 47 Jahren auf. Das Durchschnittsalter betrug 28 Jahre.

Total Testpersonen	14
Männliche Testpersonen	8
Weibliche Testpersonen	6
Durchschnittliche Kenntnis der Umgebung	4.07
Durchschnittliche Kenntnis im Umgang mit PDAs	2.57

Tabelle 1: Persönliche Werte der Testpersonen

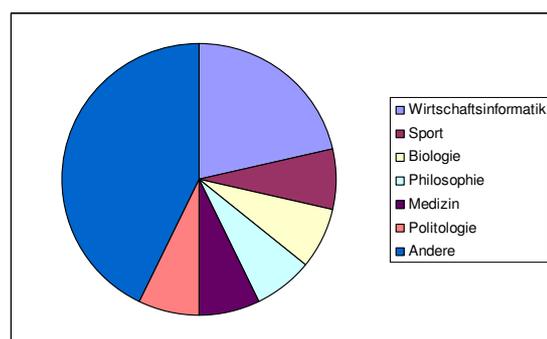


Abbildung 42: Kreisdiagramm der Tätigkeiten der Testpersonen

Der Grossteil der Testpersonen waren Studenten aus den verschiedensten Fachbereichen, wie Wirtschaftsinformatik, Biologie, Medizin, ... Es nahmen aber auch Berufstätige und eine Hausfrau am Test teil.

Wie anhand der unterschiedlichen Studienrichtungen und Tätigkeiten der Testpersonen zu ahnen ist, variierten die Kenntnisse im Umgang mit Computern und PDAs sehr. Insbesondere bei den PDAs gab es zahlreiche Personen, die noch nie damit in Berührung kamen, was sich

auch deutlich in den Zahlen widerspiegelt. Auf einer diskreten Skala von 1 bis 7, wobei 1 „Keine Kenntnis im Umgang mit PDAs“ und 7 „sehr gute Kenntnis im Umgang mit PDAs“ bedeutet, liegt der Durchschnittswert bei 2.57.

Die Kenntnis der Umgebung des Irchels wurde durchschnittlich mit 4.07 bewertet. Die Spannweite der Nennungen erstreckt sich von 1 bis 6. Einige Testpersonen studieren am Irchel und kennen daher das Gelände sehr gut, andere wiederum waren das erste Mal vor Ort.

4.1.3 Technische Ausstattung

Gespielt wird jeweils in Zweier-Teams, daher wurden für die vierzehn Testpersonen insgesamt sieben Geräte eingesetzt. Vier Geräte waren vom Typ Qtek 9000 Pocket PC und drei vom Typ HP iPAQ 4700. Die HP Geräte können mit dem in 3.2.2 beschriebenen Sensorboard ausgestattet werden. Leider waren zum Zeitpunkt des Tests nur drei Sensorboards verfügbar und daher verfügten auch nur drei Geräte über die Funktionalität der Anzeige der Blickrichtung und der Richtungsangaben zur Navigation. Abgesehen von den Richtungsangaben kann auf den vier Geräten ohne Sensorboard die Navigation dennoch benutzt werden. Dabei wird jedoch lediglich der Weg ans Ziel berechnet und angezeigt. Die Abbildung 43 zeigt ein mit dem Sensorboard ausgestattetes HP Gerät. Das Sensorboard an sich ist sehr robust, nur leider ist die Anbringung am Gerät etwas empfindlich. Die Stromversorgung erfolgt über zwei direkt an die Batterie des HP Geräts angeschlossene Kabel, deren Lötstellen am Sensorboard leider leicht brechen können.

Beide Geräte, sowohl HP iPAQ 4700 wie auch Qtek 9000 werden durch einen Stift mittels eines Touch-Screen bedient.



Abbildung 43: Ein mit einem Sensorboard ausgestatteter PDA

4.2 Ablauf der Benutzertests

Der durchgeführte Benutzertest ist in zwei unterschiedliche Subtests unterteilt. Im ersten Test, dem Navigationstest, wurde explizit die Navigationsfunktion getestet. Im zweiten Test, dem Test im eigentlichen Spiel, wurde die Navigation so getestet, wie sie letztendlich auch eingesetzt werden soll, als Hilfsfunktion innerhalb des mobilen Spiels. Neben der Navigation wurden in diesem zweiten Test auch noch Erweiterungen des mExplorers bezüglich des Fokusproblems getestet. Diese Erweiterungen stammen von Simon Frey, der seine Diplomarbeit zu diesem Thema schreibt. Die Ergebnisse seiner Arbeit können unter [Fre07] nachgelesen werden.

4.2.1 Navigationstest

In diesem Test sollten die Benutzer ganz gezielt die Navigation einsetzen, um ans vorgegebene Ziel zu gelangen. Dadurch sollten sonstige Faktoren, die sich auf die Bewertung auswirken können, auf ein Minimum reduziert werden.

Als erstes wurden die Spieler angewiesen, einen bestimmten Punkt ohne Benützung des Navigationssystems anzulaufen und danach einen weiteren mit Hilfe des Navigationssystems. Die Punkte, die angelaufen werden sollten, waren von Team zu Team verschieden, um sicherzustellen, dass die Teams die Punkte eigenständig finden und nicht einfach einem anderen Team nachlaufen.

Bei Beginn der Aufgaben, wie auch beim Erreichen eines jeden Punktes, mussten die Teams die Antwort eines Tasks übermitteln, wodurch die exakte Zeit des Erreichens festgehalten werden konnte.

4.2.2 Test im eigentlichen Spiel

Dieser Test ist darauf ausgelegt, das Navigationssystem als Bestandteil des mExplorers zu testen. Dazu wurde ein vollständiges Spiel erstellt, das die Testpersonen spielen sollten.

Das Spiel umfasste folgende Komponenten:

Points of Interest: Points of Interest machen die Spieler auf für sie interessante Orte, wie beispielsweise einen Kopierer oder eine Cafeteria, aufmerksam. Points of Interest waren acht Stück platziert.

Annotationen: Annotationen haben wie die Points of Interest eine Hinweisfunktion. Sie können jedoch von den Spielern frei gesetzt werden.

Bookmarks: Bookmarks sind analog zu den Annotationen, jedoch sind sie nicht öffentlich sichtbar, sondern privat. Die Spieler können sie setzen, damit sie für sie selbst relevante Orte besser wieder finden.

Chat: Über die Chatfunktion kann man innerhalb des Spiels kommunizieren. Es können Nachrichten an andere Teams, den Spielleiter oder an alle verschickt werden.

Hunting: Hunting ist eine Funktion, die es erlaubt, andere Teams zu fangen. Jedes Team kann ein anderes fangen und wird von einem anderen gefangen. Das zu fangende Team wird als Schaf dargestellt und das jagende Team als Wolf.

Navigation: Die Navigation unterstützt die Spieler beim Auffinden von beliebigen Orten und kann auch benutzt werden, um das gejagte Team zu fangen.

Tasks: Tasks sind Aufgaben, die an spezifischen Orten platziert wurden und die auch nur an diesen Orten gelöst werden können. Ein möglicher Task ist beispielsweise „Was ist das heutige Menü 1 der Mensa?“. Acht dieser Tasks waren über das gesamte Spielgebiet verteilt.

Während ca. einer Stunde wurde das Spiel gespielt, wobei die Spieler frei waren, die genannten Funktionen des Spiels nach Lust und Laune zu benutzen. Sie konnten Bookmarks und Annotationen setzen, diese wie auch Points of Interests anschauen, untereinander chatten, jagen und die Navigation benutzen, um Orte zielstrebig anzulaufen.

4.3 Ziel der Tests

Ziel des Benutzertests mit den beiden oben beschriebenen Tests, dem Navigationstest und dem Test im eigentlichen Spiel, war es, das entwickelte Navigationssystem bezüglich des Nutzens und der Usability zu evaluieren. Es galt Fragen wie „Kommen die Benutzer mit der Bedienung zu Recht?“ oder „Braucht es eigentlich ein Navigationssystem?“ zu klären.

4.3.1 Ziele des Navigationstests

Ziel dieses Tests ist es sowohl objektive wie auch subjektive Daten über das Navigationssystem zu erheben. Wie bereits oben erwähnt, mussten die einzelnen Teams im Rahmen dieses Tests sowohl einen bestimmten Punkt ohne, wie auch einen anderen mit der Hilfe des Navigationssystems anlaufen. Dabei wurde jeweils die Zeit des Starts, wie auch des Erreichens des Punktes durch das Übermitteln eines Tasks, festgehalten. In der Abbildung 44 ist die Sachlage graphisch veranschaulicht.

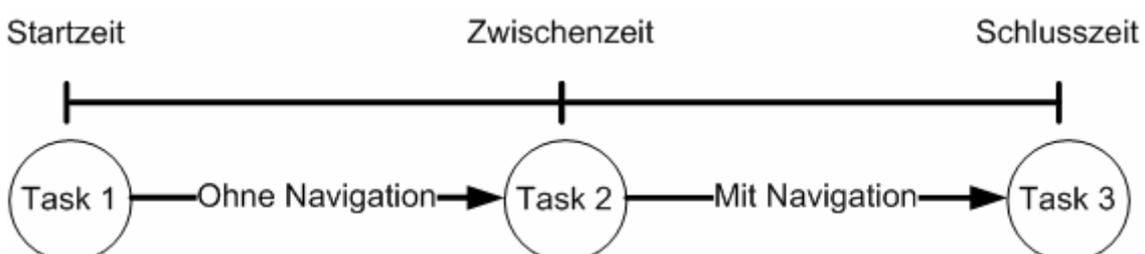


Abbildung 44: Zeitmessung im Navigationstest

So können also für jedes Team zwei Zeiten ermittelt werden. Die erste ist die Zeit, die benötigt wurde, um einen Punkt ohne Hilfe der Navigation zu finden und die andere, die Zeit, die mit Hilfe der Navigation gebraucht wurde. Da die Strecken zwischen den einzelnen Punkten nicht gleich lang sind, müssen die einzelnen Zeiten noch mittels der Länge der Strecke normiert werden. Die normierten Zeiten geben erste Aufschlüsse über den Nutzen des Navigationssystems.

Zusätzlich zu den gemessenen Zeiten wurden nach diesem Test die subjektiven Empfindungen der einzelnen Testpersonen bezüglich des Navigationssystems anhand eines Fragebogens ermittelt. Der Fragebogen umfasste folgende Bereiche:

Angaben zur Person: Hierbei geht es darum Daten über die Testpersonen zu erlangen, um sie und ihre Kenntnisse des Ircchels und der Technik besser einordnen zu können.

Gesamtbeurteilung: Dieser Bereich befasst sich mit der Frage, ob die Benutzer das Navigationssystem als sinnvoll erachten. Ebenso wird die Frage gestellt, ob die Benutzer den Wegangaben vertrauen oder ob sie kürzere Wege nehmen würden.

Navigationsmenü: Ist die Bedienung intuitiv oder sind die Benutzer hoffnungslos überfordert. Finden sie alle Funktionen oder sind einige gar nicht vorhanden?

Anzeige des berechneten Weges: Genügt die Anzeige des Weges den Anforderungen der Benutzers. Ist er Verständlich und gut sichtbar? Hilft die Anzeige den Benutzern ihr gesuchtes Ziel zu finden?

Graphische Wegleitung: Sind die graphischen Richtungsangaben gut verständlich und sind sie hilfreich?

Akustische Wegleitung: Erleichtern die akustischen Richtungsangaben das Auffinden des Ziels? Sind sie verständlich oder werden sie von den Umgebungsgeräuschen völlig übertönt?

Nutzen des Navigationssystems: Bringt das System als Ganzes dem Benutzer einen zusätzlichen Nutzen? Welche Elemente tragen dazu bei und welche nicht?

Weitere Einflüsse: In diesem Bereich werden noch einige Fragen über den mExplorer an sich gestellt. Dabei sollen Elemente ausserhalb des Navigationssystems, die aber dennoch Einfluss darauf haben, ermittelt werden.

Diese kurzen Erläuterungen des Fragebogens sollen einen Überblick darüber geben, welche subjektiven Daten im Rahmen des Navigationstests anhand des Fragebogens ermittelt werden sollten. Ausführlicher wird auf die einzelnen Fragen in der Auswertung im Kapitel 4.4 eingegangen. Zudem können Interessierte den kompletten Fragebogen im Kapitel H.1 des Anhangs einsehen.

4.3.2 Ziel des Tests im eigentlichen Spiel

Bei diesem Test ging es darum zu evaluieren, wie sich das entwickelte Navigationssystem auf das bestehende System auswirkt. Ist die Navigation gut integriert oder wirkt sie sich störend auf andere Elemente des Spiels aus?

Aus dem vorhergegangenen Navigationstest kennen die Benutzer die Navigation bereits und sind nun frei, diese zu benutzen oder auch nicht. Dadurch kann auch ermittelt werden, ob und wofür die Spieler die Navigation im Rahmen eines echten Spiels einsetzen.

Die Erhebung der dazu benötigten Daten erfolgte im Anschluss an das gespielte Spiel anhand eines Fragebogens. Dieser Fragebogen deckte einerseits die genannten Fragestellungen zum Navigationssystem ab, andererseits auch das Fokus-Problem betreffend Fragestellungen. Einen Auszug mit den Fragen, die die Navigation betreffen, ist im Kapitel H.2 des Anhangs enthalten.

4.4 Auswertung der Benutzertests

In erster Linie erfolgt die Auswertung der Daten des Tests in den einzelnen Teilgebieten des Navigationssystems zu denen auch gezielt Daten erhoben wurden. Zwangsläufig wird diese Trennung aber nicht strikt sein, denn die einzelnen Teilgebiete beeinflussen sich auch gegenseitig. Eine vollständige Evaluation bedarf also auch zwangsläufig einer globalen Betrachtung, um bereichsübergreifende Korrelationen zu erkennen.

Bei der Auswertung des Fragebogens wurden den einzelnen Antwortmöglichkeiten die Werte 1 bis 7 zugeordnet. 1 entspricht dabei „trifft nicht zu“ und 7 „trifft zu“.

4.4.1 Zeitmessung

Bei den durch die Zeitmessung erhobenen Daten handelt es sich um objektive Werte. Sie wurden durch reines Messen der Zeit erhoben und beruhen nicht auf subjektiven Empfindungen des Benutzers.

Die Auswertung der Daten ergab folgende gemessene Zeiten für die sieben Teams ergab folgende Werte:

	Ohne Navigation	Mit Navigation
Durchschnittliche Geschwindigkeit:	3.35 km/h	3,37 km/h
Maximale Geschwindigkeit	5.8 km/h	6.41 km/h
Minimale Geschwindigkeit	1.82 km/h	1.93 km/h

Tabelle 2: Überblick über die gemessenen Zeiten

Die Werte sind durch die Ungenauigkeit der Positionierung nicht als absolute Werte zu verstehen, sondern als Richtgrößen, die die effektiven Größen bestmöglich annähern. Da davon aber beide ermittelten Werte gleichermassen betroffen sind, hat dieser Effekt keine grundlegenden Auswirkungen auf die Auswertung der beiden Werte.

Unglücklicherweise war zum Zeitpunkt des Tests der WLAN-Empfang im Bereich des einen Punktes nicht optimal, so dass an dieser Stelle keine Positionierung möglich war.

Dieses Faktum wurde aber anhand der einzelnen Log-Files bestmöglich ausgemerzt. In den Log-Files sind nicht nur erfolgreiche Versuche einen Task zu erfüllen ersichtlich, sondern auch fehlgeschlagene. Da sich die Teams an der richtigen Stelle befanden, die Position auf ihrem PDA aber nicht nachgetragen wurde, versuchten sie erfolglos den Task zu erfüllen. Für die Auswertung wurde jeweils der erste erfolglose Versuch als Zeitpunkt des Erreichens des Punktes und der letzte als Zeitpunkt des Verlassens des Punktes gewertet. Von diesem Problem waren sowohl Zeiten ohne, wie auch mit Navigation betroffen, daher dürfte sich dieses Problem nicht verfälschend auf die Auswertung auswirken.

Die durchschnittliche Geschwindigkeit mit der Hilfe der Navigation ist mit 3,37 km/h nur geringfügig höher als die ohne Navigation mit 3.35 km/h. Zudem ist eine Stichprobe von sieben Werten ohnehin nicht gross genug, um wirklich aussagekräftige Resultate zu liefern. Aber dennoch sind diese beiden Werte positiv für das entwickelte Navigationssystem zu werten. Bereits bei der ersten Anwendung sind die Benutzer so gut mit dem System zu Recht gekommen, das eine minim bessere Geschwindigkeit resultierte als ohne das System.

Auffällig ist, dass die Varianz zwischen den einzelnen Geschwindigkeiten recht gross ist. Die Höchstgeschwindigkeit von 6.41 km/h steht einer minimalen Geschwindigkeit von 1.82 km/h

gegenüber. Dies kann durch die Kenntnisse des Geländes des Irchels wie auch der Kenntnisse im Umgang mit PDAs begründet werden. Das Team, dessen Geschwindigkeiten mit, wie auch ohne Navigation mit Abstand am niedrigsten waren, kannte weder den Irchel noch hatte es schon Kontakt mit PDAs. Beide Teammitglieder antworteten auf die Frage der Kenntnis des Irchels und der Kenntnisse im Umgang mit PDAs mit der Tiefstnote von 1. Dagegen war bei den schnellen Teams immer zumindest ein Mitglied versiert im Umgang mit PDAs und kannte sich auf dem Irchelgelände aus.

Es konnten zwei verschiedene Strategien beim Anlaufen eines Ziels beobachtet werden. Teams, die neu am Irchel waren und PDAs nicht kannten (Anfänger) tasteten sich Schritt für Schritt heran. Sie hielten jeweils nach kurzen Strecken wieder inne und warteten, bis ihre Position nachgetragen wurde. Versierte Teams (Profis) dagegen legten jeweils viel grössere Strecken zurück bis sie erneut den PDA konsultierten und warteten bis ihre Position nachgetragen wurde. Dadurch fällt die Geschwindigkeit der Anfänger zwangsläufig geringer aus als die der Profis.

4.4.2 Navigationsmenü

Bei den Fragen betreffend des Navigationsmenüs wurden durchgehend Durchschnittswerte erreicht, die geringfügig über dem Mittelwert von 4 (Werte von 1 bis 7) liegen. In der Tabelle 3 sind die wichtigsten Werte zum Navigationsmenü zusammengefasst.

Bedienbarkeit des Navigationsmenüs (Frage Nr. 14)	4.14
Bedienbarkeit des mExplorers (Frage Nr. 45)	4.54
Einfaches Wählen des Ziels (Frage Nr. 15)	4.57

Tabelle 3: Werteübersicht zum Navigationsmenü

Die Einfachheit beim Wählen eines Ziels (Frage Nr. 15) wurde mit 4.57 und die Bedienung allgemein (Frage Nr.14.) mit 4.14 bewertet. Dies ist jedoch nicht ganz verwunderlich, da sehr viele Teilnehmer zum ersten Mal einen PDA benutzt haben. Daher sind auch diese durchschnittlichen Werte durchaus positiv zu bewerten. Trotz der geringen Kenntnis im Umgang mit PDAs sind die Testpersonen mit der Bedienung des Navigationsmenüs so zurecht gekommen, dass sie es mit einem arithmetischen Mittel von über 4 bewertet haben. Zusätzlich muss man natürlich auch noch betrachten, dass die Testpersonen die Bedienbarkeit des mExplorers an sich (Frage Nr. 45) mit einem arithmetischen Mittel von 4.54 nicht viel besser bewertet haben. Die vermutete Beziehung zwischen der Bedienbarkeit des Navigationsmenüs und den mExplorer an sich lässt sich anhand der erhobenen Daten auch bestätigen. Zwischen diesen beiden Merkmalen besteht eine 60% Korrelation auf dem 0.05 Signifikanzniveau.

Korrelationen

			Navigation smenü	mExplorer
Spearman-Rho	Navigationsmenü	Korrelationskoeffizient	1.000	.604(*)
		Sig. (2-seitig)	.	.049
		N	14	11
	mExplorer	Korrelationskoeffizient	.604(*)	1.000
		Sig. (2-seitig)	.049	.
		N	11	11

* Die Korrelation ist auf dem 0,05 Niveau signifikant (zweiseitig).

Tabelle 4: Korrelation zwischen der Bedienung des Navigationsmenüs und der des mExplorers

Aus den sonstigen Anmerkungen zum Navigationsmenü bei der Frage Nr. 18 geht hervor, dass einige Testpersonen bei der Bewertung des Navigationsmenüs auch bereits den mExplorer an sich mitbewertet haben. Es wurden Anmerkungen wie „Bessere Beschreibung, C/D/E/F¹² sind zu schwach“ oder „Buchstaben vs. Text“ gemacht. Diese beziehen sich eindeutig auf die Menüleiste des mExplorers und nicht auf das Navigationsmenü an sich. Da diese negativen Punkte bei der Bewertung des Navigationsmenüs mit eingeflossen sind und nicht beim mExplorer selbst, lässt sich das geringfügig schlechtere Abschneiden zumindest teilweise erklären.

Im Grossen und Ganzen kann ein positives Fazit bezüglich des Navigationsmenüs gezogen werden. Die Benutzer sind mehrheitlich damit zu Recht gekommen. Die vorhandenen Schwierigkeiten haben ihren Ursprung nicht im Navigationsmenü selbst sondern bei dem erstmaligen Kontakt mit dem System selbst und den verwendeten PDAs.

4.4.3 Angezeigter Weg

Die Anzeige des berechneten Weges wurde generell als sehr hilfreich für das Auffinden eines unbekanntes Ortes (Frage Nr. 19) bewertet. Das arithmetische Mittel der Bewertung liegt dabei bei 5.83. Die tiefste Bewertung war eine 4 und dies auch nur von einer Person. Vier Personen haben jedoch die Höchstnote von 7 gegeben. Einen Überblick über alle Werte zur Anzeige des Weges, auf die im Text eingegangen wird, verschafft die Tabelle 5.

Hilfreich zum Auffinden eines unbekanntes Ortes (Frage Nr. 19)	5.83
Erkennbarkeit des angezeigten Weges (Frage Nr. 21)	5.81
Leichtigkeit, um dem Weg zu folgen (Frage Nr. 20)	5.25
Korrektheit der Anzeige (Frage Nr. 23)	5.18

Tabelle 5: Werteübersicht zur Anzeige des Weges

Ebenfalls sehr gut bewertet mit einem Durchschnitt von 5.81 wurde die Erkennbarkeit des Weges auf der Karte (Frage Nr. 21). Dies hängt sicherlich auch damit zusammen, dass die Farbe des Weges nach Bedarf variiert werden kann.

Den Testpersonen fiel es leicht dem angezeigten Weg ans Ziel zu folgen. Die entsprechende Frage (Frage Nr. 20) wurde mit einem Durchschnittswert von 5.25 bewertet. Eine Testperson merkte dazu an, dass es einfacher wäre dem Weg zu folgen, wenn sich die Karte entsprechend der Blickrichtung ausrichten würde. Dies ist aber leider wie Daniel Suter in seiner Arbeit [Sut06] ermittelte, mit den verwendeten PDAs nicht realisierbar.

Ebenfalls positiv ist, dass die Benutzer die Korrektheit des angezeigten Weges (Frage Nr. 23) bei einem festen Ziel durchschnittlich mit 5.18 bewerteten. Dieser Wert wird aber durch die Positionsungenauigkeit beeinflusst. Anhand der erhobenen Daten konnte jedoch die Existenz dieser Beziehung nicht erhärtet werden. Es ist auch anzunehmen, dass manche Personen die Positionsungenauigkeit bei der Bewertung der Korrektheit des Weges ausblendeten und andere nicht. Manche werden also einen korrekten Weg als falsch bezeichnet haben, da ihre Position nicht genau mit der angezeigten Position übereinstimmte.

Aus den erhobenen Daten geht also hervor, dass die Benutzer den angezeigten Weg für korrekt halten. Er ist gut ersichtlich und man kann ihm gut ans Ziel folgen.

¹² Diese Buchstaben beziehen sich auf die Menüleiste des mExplorers. Zu sehen ist diese in der Abbildung 39.

4.4.4 Graphische Wegleitung

Wie bereits erwähnt, waren beim Test leider nur drei Sensoren verfügbar. Grundsätzlich können also nur sechs Personen über diesen Bereich Auskunft geben. Ein Team, das im ersten Teil des Tests keinen Sensor hatte, bekam jedoch im zweiten Teil einen PDA mit Sensor und bewertete diesen Bereich noch nach dem Ende des zweiten Teils. Daher sind für diesen Bereich Daten von acht verschiedenen Testpersonen vorhanden.

In der Tabelle 6 sind die Durchschnittswerte, auf die in der Auswertung Bezug genommen wird, ersichtlich.

Nutzen der graphischen Richtungsangaben (Frage Nr. 26)	5.12
Verständlichkeit der Pfeile (Frage Nr. 27)	6
Erkennbarkeit der Pfeile (Frage Nr. 28)	6.37
Unterscheidbarkeit der Pfeile (Frage Nr. 29)	6

Tabelle 6: Werteübersicht zur graphischen Wegleitung

Obschon die richtungsangehenden Pfeile nur als zusätzliche Hilfe gedacht sind, wurden sie mit einem arithmetischen Mittel von 5.12 als hilfreich eingestuft (Frage Nr. 26). Hilfreich sind sie insbesondere dadurch, dass der Benutzer auch durch rasches Hinsehen weiss, wohin er sich bewegen soll, ohne lange die Karte mit der Umgebung abgleichen zu müssen. Wenn die Drehung der Karte realisierbar wäre, wäre diese natürlich noch um einiges hilfreicher, aber die Richtungsangaben sind eine ressourcenfreundlichere Alternative, die in die gleiche Richtung geht.

Insbesondere die graphische Umsetzung der Richtungsangaben anhand der gewählten Pfeile ist den Daten zufolge gut gelungen. Durchschnittlich wurden die Verständlichkeit der Pfeile (Frage Nr. 27) mit 6, die Erkennbarkeit (Frage Nr. 28) mit 6.37 und die Unterscheidbarkeit (Frage 29) mit 6 bewertet.

Auch wenn 8 Bewertungen keine wirklich aussagekräftigen Schlüsse zulassen, können immerhin Tendenzen ermittelt werden, die es erlauben auch bei den graphischen Richtungsangaben eine positive Bilanz zu ziehen.

4.4.5 Akustische Wegleitung

Die Voraussetzung für die akustischen Richtungsangaben ist genau wie bei den graphischen die Ausstattung des PDAs mit einem Sensorboard. Daher konnten auch hier nur Daten von acht Testpersonen erhoben werden. Die Tabelle 7 fasst die später im Text ausführlich behandelten Durchschnittswerte zusammen.

Nutzen der akustischen Richtungsangaben (Frage Nr. 31)	3
Klarheit der Bedeutung der akustischen Richtungsangaben (Frage Nr. 32)	6.14
Verständlichkeit der akustischen Richtungsangaben (Frage Nr. 33)	6.14

Tabelle 7: Werteübersicht der akustischen Wegleitung

Die akustischen Richtungsangaben sind verständlich gesprochen (Frage Nr. 33) und ebenso ist klar, was damit gemeint ist (Frage Nr. 32). Beides wurde von den Testpersonen mit einem Durchschnittswert von 6.14 bewertet. Jedoch erachten sie es nicht als nützlich. Auf die Frage,

wie hilfreich die akustischen Richtungsangaben sind (Frage Nr. 31), antworteten sie durchschnittlich mit einer 3. Nicht hilfreich ist eine Information, wenn sie nicht verstanden wird oder wenn die Information redundant ist. Dass es nicht daran liegt, dass die Benutzer die Angaben nicht verstehen, wurde ja bereits anhand der Fragen Nr. 32 und Nr. 33 ermittelt. Die Redundanz der vermittelten Information ist aber durchaus eine plausible Erklärung. Die akustischen Richtungsangaben vermitteln grundsätzlich dieselbe Information wie die graphischen Richtungsangaben. Jedoch sind Bilder viel aussagekräftiger als Worte und zudem erhält der Benutzer beim Betrachten der graphischen Richtungsangaben noch zusätzliche Informationen. Er sieht dabei auch gerade die Übersichtskarte und weiss daher viel genauer, in welche Richtung der Weg weiter geht. Daher sind die akustischen Richtungsangaben nicht so informativ wie die optischen.

Der einzige Vorteil der akustischen Richtungsangaben wäre der, dass sie der Benutzer auch erhält, wenn er nicht auf den Display schaut. Das Problem dabei ist jedoch, dass die Blickrichtung im Gehen ständig hin und her springt. Um brauchbare Werte zu erhalten, muss der PDA kurze Zeit ganz ruhig und horizontal gehalten werden, bis sich der Kompass einpendelt. Dabei kann der Benutzer auch gerade auf den PDA schauen.

Kommentare wie „Braucht es nicht unbedingt“ bei den weiteren Äusserungen (Frage Nr. 35) sind daher nicht erstaunlich.

Zusätzlich kommt noch hinzu, dass auch sämtliche nicht am Spiel beteiligten Personen die Angaben auch hören. Dies kann einigen Personen auch peinlich sein, weshalb sie die akustischen Richtungsangaben vorzugsweise ausschalten. Dies ging aus mündlichen Diskussionen mit einzelnen Personen im Anschluss an den Test hervor. Eine Äusserung auf einem Fragebogen, die lautet „Störend auf Umgebung“ geht in dieselbe Richtung.

Zu den akustischen Richtungsangaben kann man zusammenfassend sagen, dass sie grundsätzlich verständlich sind, aber nicht unbedingt benötigt werden. Da sie im Gehen keine verlässlichen Resultate liefern, weisen sie keinen Vorteil zu den graphischen Richtungsangaben mit der höheren Informationsdichte auf.

4.4.6 Gesamtbeurteilung

Die Gesamtbeurteilung des Navigationssystems fällt durchaus positiv aus. Die behandelten Durchschnittswerte sind in der Tabelle 8 zusammengefasst.

Sinnvolle Erweiterung (Frage Nr. 8)	5.28
Sinnvolle Erweiterung (Frage Nr. 2.24)	5.9
Integration in den mExplorer (Frage Nr. 2.25)	6.2
Keinen negativen Einfluss auf den Rest des Spiels (Frage Nr. 2.26)	6.3
Benutzung im eigentlichen Spiel (Frage Nr. 2.27)	5.33
Navigation benutzt, um einen bestimmten Punkt anzulaufen (Frage Nr. 2.28)	4.91
Navigation benutzt, um das gejagte Team zu fangen (Frage Nr. 2.29)	3.33
Hätten anderen Weg genommen (Frage Nr. 11)	1.5
Hätten kürzeren Weg genommen (Frage Nr. 12)	2.14
Genauigkeit der Positionsanzeige (Frage Nr.41)	1.8

Tabelle 8: Werteübersicht der Gesamtbeurteilung

Die Frage, ob das Navigationssystem eine sinnvolle Erweiterung ist, wurde nach dem ersten Test (Frage Nr. 8) durchschnittlich mit 5.28 und nach dem zweiten Test (Frage Nr. 2.24) mit 5.9 bewertet.

Die Testpersonen erachten die Navigation als gut in den mExplorer integriert (Frage Nr. 2.25) und bewerteten dies mit 6.2. Ebenso finden sie, dass sich die Navigation nicht negativ auf den Rest des Spiels auswirkt (Frage Nr. 2.26) und bewerteten dies mit 6.3.

Die Testpersonen haben die Navigation auch noch im zweiten Test, wo die Benutzung nicht mehr obligatorisch war, benutzt. Die Frage dazu, die im Fragebogen im Anschluss an das eigentliche Spiel gestellt wurde (Frage Nr. 2.27), wurde durchschnittlich mit einem Wert von 5.33 bewertet. Dieser Wert ist nicht extrem hoch aber dennoch sehr positiv zu werten. Es gilt zu beachten, dass im zweiten Teil nebst der Navigation noch zahlreiche weitere Funktionen wie Chat und ähnliches benutzt werden konnte. Daher ist es sehr positiv, dass die Testpersonen die Navigation immer noch benutzten, auch wenn sie sie im ersten Teil schon benutzen mussten.

Mehrheitlich wurde die Navigation benutzt, um einen bestimmten Punkt auf der Karte anzulaufen. Dabei hätten die Testpersonen auch ohne die Navigation mehrheitlich denselben Weg genommen um ans Ziel zu gelangen. Die Frage, ob sie einen anderen Weg genommen hätten (Frage Nr. 11) wurde durchschnittlich mit 1.5 beantwortet. Die Frage, ob sie einen kürzeren Weg genommen hätten (Frage Nr. 12) wurde mit 2.14 bewertet. Dass diese beiden Werte sehr tief ausgefallen sind, ist sehr positiv. Stimmen der vom System berechnete Weg und der Weg, denn man selbst gewählt hätte, mehrheitlich überein, so erhöht das das Vertrauen in das System und die Bereitschaft es auch zu benutzen.

Um das gejagte Team zu fangen wurde die Navigation eher selten benutzt. Die Frage dazu (Frage Nr. 2.29) erreichte nur einen Wert von 3.33. Der Vollständigkeit halber muss man jedoch erwähnen, dass nicht allzu viele Teams gefangen werden konnten. Es war eine Rotation von 20 Minuten eingestellt. Das bedeutet, dass alle 20 Minuten die zu fangenden Teams wechselten. Fängt man ein Team, so versucht das System ein neues Team zu finden, das gejagt werden kann. Dazu darf das Team aber von niemandem anderen gejagt werden und man darf es zuvor noch nie gefangen haben. Es kann also erst nach dem Ablauf der 20 Minuten garantiert werden, dass alle Teams wieder ein Team jagen können. Daher wurden also generell wenig Teams gefangen und so ist es auch nicht verwunderlich, dass die Navigation nicht sehr oft zum Fangen eines Teams verwendet wurde.

Auffallend ist, dass die Testpersonen die Positionsanzeige (Frage Nr. 41) mit einem Wert von 1.8 als sehr ungenau einstufen. Leider wurden in vergangenen Versuchen keine identischen Daten erhoben, die effektive Vergleiche zulassen würden. Aus den Daten von Tests, die Ende 2005 durchgeführt wurden, wurde die Ungenauigkeit der Positionsanzeige als ziemlich störend bewertet. Eine Hypothese ist, dass die Genauigkeit der Positionsanzeige im Zusammenhang mit der Navigation schlechter bewertet wird als sonst, obwohl sie sich nicht verändert hat. Bei der Bewertung der Genauigkeit kommt das Frame of Reference¹³ Problem zum Tragen. Durch die Benutzung der Navigationsfunktion haben wir einen anderen Blickwinkel auf die Positionsgenauigkeit und dies wirkt sich, wenn auch unbewusst, auf unsere Bewertung aus. Bei der Benutzung des Navigationssystems sind wir sehr darauf fixiert, wo wir sind und wo wir hin wollen. Daher erwarten wir eine höhere Genauigkeit der Positionierung als sonst und deshalb bewerten wir eine gegebene Genauigkeit aus diesem Blickwinkel schlechter als sonst.

Abschliessend kann für das Navigationssystem ein durchaus positives Fazit gezogen werden. Die Benutzer erachten es als eine sinnvolle und gut integrierte Erweiterung des mExplorers. Sie vertrauen der Korrektheit des berechneten Weges, was für die Akzeptanz des Systems unabdingbar ist. Die graphischen Richtungsangaben sind klar verständlich und gut unterscheidbar. Im Gegensatz zu den akustischen Richtungsangaben werden sie auch als hilfreich eingestuft.

¹³ Das Frame of Reference Problem besagt, dass Bewertungen immer subjektiv vom Blickwinkel des Betrachters abhängig sind.

Einziges Wermutstropfen ist, dass der Benutzer aufgrund der technischen Gegebenheiten von Zeit zu Zeit innehalten muss, bis die Position und die Blickrichtung nachgetragen wurden.

5 Fazit

In diesem Kapitel wird eine abschliessende Betrachtung der Arbeit gemacht. Dabei wird überprüft, inwiefern die Arbeit den Anforderungen der Aufgabenstellung gerecht wird und einen Ausblick auf Möglichkeiten der Weiterführung der Arbeit gemacht. Zum Schluss folgt noch ein persönliches Fazit.

5.1 Bewertung des Erreichten

Die Arbeit umfasst im Grossen und Ganzen zwei Teile, einen theoretischen Literaturteil und eine praktischen Implementierungsteil.

Im Literaturteil werden die theoretischen Grundlagen der Navigation erarbeitet. Dabei wird auf die Unterschiede zwischen Fussgänger- und Fahrzeugnavigation eingegangen, wie auch Wegberechnungsalgorithmen und Möglichkeiten zur Präsentation von Wegleitinformationen betrachtet. Abgeschlossen wird der Teil durch die Betrachtung bereits existierender Systeme und deren Eigenschaften.

Der praktische Teil beschreibt die Entwicklung und die Evaluation des Navigationssystems für den mExplorer. Das entwickelte System umfasst zwei unabhängige Komponenten:

Navigationseditor: Der Navigationseditor ermöglicht es, den Graphen der gehbaren Wege zu erstellen. Er wurde in den bereits existierenden Spieleditor integriert ohne die bestehende Funktionalität zu beeinträchtigen.

Navigationsmodul: Das Navigationsmodul ist für die eigentliche Wegführung des Benutzers zuständig und wurde in den mExplorer-Client integriert. Es gibt sowohl graphische wie auch akustische Wegleitinformationen an ihn weiter.

Die Dokumentation behandelt die Architektur, die Benutzerinteraktion wie auch Besonderheiten beider Komponenten.

Zur Evaluation des entwickelten Systems wurde ein Benutzertest durchgeführt und die dabei erhobenen Daten gezielt auf den Nutzen und die Usability ausgewertet.

Somit erfüllt die Arbeit sämtliche Anforderungen der Aufgabenstellung.

5.2 Ausblick

Im Laufe der Arbeit hatte es sich gezeigt, dass auf Grund der zeitlich verzögerten Positionierung die Navigation nicht im Gehen, sondern nur wenn der Benutzer für eine gewisse Zeit am Ort still steht, effektiv durchgeführt werden kann. Die technischen Gegebenheiten schränken also die Möglichkeit zur Weiterführung etwas ein. Eine interessante Möglichkeit wäre jedoch die kategorisierte Navigation, wie sie auch in dem in Kapitel 2.4.2 betrachteten System Deep Map angewendet wird.

Bislang berechnet das System den kürzesten Weg an ein vom Benutzer angegebenes Ziel, also von A nach B. Dabei kennt der Benutzer aber bereits den Standort seines Ziels. Bei der kategorisierten Navigation gibt der Benutzer sein Ziel nicht in der Form eines spezifischen Punktes auf der Karte an, sondern anhand einer Kategorie. Darauf hin wird dann der kürzeste Weg zum vom System ermittelten Punkt berechnet. Beispielsweise könnte der Benutzer angeben, dass er Durst hat, worauf ihm das System den Weg zur nächsten Mensa/Cafeteria anzeigen würde. Im Ansatz ist dies bereits möglich, durch die Wahl des gejagten Teams als Ziel. Dies ist aber erst eine einzige Kategorie. Durch die Betrachtung der Bedürfnisse der Benutzer können jedoch leicht noch viele mehr gefunden werden. Mögliche Kategorien wären beispielsweise Arbeitsplätze, Hörsäle, Büros, Kopierer, Toiletten, ...

5.3 Persönliches Fazit

Das Anfertigen dieser Arbeit und insbesondere das Entwickeln des Navigationssystems war sehr interessant. Es erlaubte es mir, im Studium erworbenes Wissen aus diversen Bereichen an einem konkreten Projekt anzuwenden. Es zeigte sich wieder einmal, dass bei der Software-Entwicklung der verbleibende Aufwand nur schwer abzuschätzen ist und daher oftmals massiv unterschätzt wird. Der vor Beginn der Arbeit erstellte Zeitplan bewarte mich aber davor, zu sehr in Verzug zu kommen und so konnte ich die Arbeit ohne zu grossen Zeitdruck zu Ende bringen.

Der Benutzertest war in der Vorbereitung sehr zeitintensiv, aber dennoch erlebte ich ihn als sehr positiv. Da ich das System selbst entwickelt habe, war mein Blickwinkel darauf sehr einseitig. Durch den Test erhielt ich ein Feedback über das System von Personen, die das System aus ganz anderen Blickwinkeln betrachteten.

Abschliessend ziehe ich somit ein durchwegs positives Fazit und kann sagen, dass ich diese Arbeit auch ein zweites Mal wählen würde.

5.4 Danksagung

An diesem Punkt möchte ich mich ganz herzlich bei all den Personen bedanken, die mich bei der Anfertigung dieser Arbeit unterstützt hatten.

Mein besonderer Dank geht an meinen Betreuer Christoph Göth. Seine Tür stand mir stets für Fragen offen und so konnte ich auch sehr von seinen Erfahrungen profitieren.

Ebenso möchte ich Peter Vorbürger danken. Bei allen Problemen, die das Sensorboard betroffen hatten, konnte ich immer auf seine Hilfe zählen, auch wenn es nur eine gebrochene Lötstelle war.

Auch danken möchte ich Sarina Zimmermann für die Korrektur der Arbeit.

Abschliessend möchte ich mich auch noch bei Simon Frey für die gute Zusammenarbeit beim Benutzertest bedanken.

Anhang

A Abkürzungsverzeichnis

AUG	Augmentation
AR	Augmented Reality
BAU	Building Augmentation
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
MARS	Mobile Augmented Reality Setup
MCAR	Mobile Collaborative Augment Reality
NAV	Navigation
OCAR	Outdoor Collaborative Augment Reality
PDA	Personal Digital Assistant
PIP	Personal Interaction Panel
POI	Point of Interests
WIM	World In Miniature
WLAN	Wireless Local Area Network

B Literaturverzeichnis

[ALK07] ALK Technologies: ALK Copilot live, <http://www.alk.com/copilot/smartphone.asp>,
Letzter Besuch: 21. April 2007

[Bau02] Baus, Jörg; Kray, Christian; Krüger, Antonio; Wahlster, Wolfgang: A resource-
adaptive mobile navigation system, University Saarbrücken, 2002

[Bru05] Brunner, Dennis: Architekturentwicklung für mobile Spiele, Diplomarbeit, Institut
für Informatik der Universität Zürich, 2005

[Cer06] Cerbe, Thomas M.; Giszczak, Adam: Just in Time Navigation in Road Transport – A
new Approach in Navigation, Cebit in Motion – Forum für Telematik und Navigation,
Hannover, 2006

[Che00a] Cheverst, Keith; Davies, Nigel; Mitchell, Keith; Friday, Adrian; Efstratiop,
Christos: Developing a Context-aware Electronic Tourist Guide: Some Issues and
Experiences; Universität Lancaster, 2000

[Che00b] Cheverst, Keith; Davies, Nigel; Mitchell, Keith; Friday, Adrian: Experiences of
Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project; Universität
Lancaster, 2000

[Coh04] Cohen Jonathan: Graphs, Department of Computer Science, Johns Hopkins
University, 2004

[Dag06] <http://www.daguangming.nl/route.gif>, Letzter Besuch: 19. Januar 2007

[DijAp] Dijkstra-Algorithmus (Java Applet); <http://www-m9.ma.tum.de/dm/java-applets/routenplanung/Algorithmen/Dijkstra>; letzter Besuch: 16. Februar 2007

[Dij59] Dijkstra, Edgser W.: A note on two problems in connection with graphs, Numerische
Mathematik 1:269-271, 1959

[Dis06] Distributed Multimedia Research Group: The Guide Project; Universität Lancaster,
2006, <http://www.guide.lancs.ac.uk> , Letzter Besuch: 8. Januar 2007

[Fre07] Frey, Simon: Das Fokus-Problem im mExplorer, Diplomarbeit, Institut für Informatik
der Universität Zürich, 2007

[Göt04] Göth, Christoph; Häss, Urs_Peter; Schwabe, Gerhard: Requirements for mobile
learning games shown on a mobile game prototype, Proceedings MLearn2004 conference,
Rome, Italy, 2004

[Int06] Interactive Media Systems Group: Outdoor Collaborative Augmented Reality,
Institute for Software Technology and Interactive Systems, Technische Universität Wien,
2006, <http://www.ims.tuwien.ac.at/research/mobile/ocar/> , Letzter Besuch: 22. Januar 2007

[Kal02] Kalkusch, Michael; Lidy, Thomas; Knapp, Michael; u.A: Structured Visual Markers for Indoor Pathfinding, Proceedings of the First IEEE International Workshop on ARToolKit (ART02). 2002

[Kna07] Knapp, Michael.; Reitmayr, .Gerhard.; Schmalstieg, Dieter.: SignPost 2 - Mobile AR Navigation System; <http://www.studierstube.org/projects/mobile/SignPost2/> , Letzter Besuch 22. Januar 2007

[Kal02] Kalkusch, Michael; Lidy, Thomas; Knapp, Michael; u.A: Structured Visual Markers for Indoor Pathfinding, Proceedings of IEEE International Augmented Reality Toolkit Workshop 2002

[Man04] Mansfeld, Werner: Satellitenortung und Navigation (2. Auflage), Fried. Vieweg & Sohn Verlag/ GWV Fachverlag GmbH, Wiesbaden, 2004

[Mil55] Miller, George A.: The Magical Number Seven, Plus or Minus Two - Some Limits on Our Capacity for Processing Information, Harvard Universität, 1955

[Mül03] Müller, Arnold; Conrad, Stefan; Kruijff, Ernst: Multifaceted Interaction with a Virtual Engineering Environment using a Scenograph-orientated Approach; Proceeding of the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2003; Tschechische Republik, 2003

[Nar03] Narzt, Wolfgang; Pomberger, Gustav; Ferscha, Alois; Kolb, Dieter und Andere: Pervasive Information Acquisition for Mobile AR-Navigation Systems; 5th IEEE Workshop on Mobile Computing Systems & Applications, Monterey, California, USA, ISBN: 0-7695-1995-4/03, pp: 13-20, October 2003

[Nav07] Navigon Mobile Navigator 6,
http://www.navigon.com/site/de/de/mobile_navigation/cellphone/symbianphone/screenshots,
Letzter Besuch: 21. April 2007

[Rei01] Reitmayr, Gerhard; Schmalstieg, Dieter: Collaborative Mobile Collaborative Augmented Reality, Technische Universität Wien, 2001

[Rei03] Reitmayr, Gerhard; Schmalstieg, Dieter: Collaborative Augmented Reality for Outdoor Navigation and Information Browsing, Technische Universität Wien, 2003

[Röb07] Röbbke-Doerr, Peter: Auf schnellstem Wege- Software für Handy-Navigation, CT – Magazin für Computer und Technik, Ausgabe 9, 2007

[Rus03] Russell, Stuart J.; Norving, Peter: Artificial Intelligence: A Modern Approach, Prentice Hall, 2002

[San05] Sanders, Peter; Schultes, Dominik: Genauer Hochleistungs-Routenplaner, Universität Karlsruhe, 2005

[Sut06] Suter, Daniel; Indoornavigation unterstützt durch Magnetfeldsensorik, Diplomarbeit, Institut für Informatik der Universität Zürich, 2006

[Ter06] Terada, Tsutomu; Miyamae, Masakazu; u.A.: Design of a Car Navigation System that Predicts User Destination, Osaka University, 2006

[Wah01] Wahlster, Wolfgang; Baus, Jörg; Kray, Christian; Krüger, Antonio: REAL: Ein ressourcenadaptierendes mobiles Navigationssystem, Universität Saarbrücken, 2001

[Wan92] Want, Roy; Hopper, Andy; Falcão, Veronica; Gibbson Jonathan: The Active Badge Location System , Olivetti Research Ltd. (ORL), Cambridge, 1992

[Zip99a] Zipf, Alexander; Chandrasekhara, Vasu; Häussler Jochen; Malaka, Reiner: GIS hilft Touristen bei der Navigation – ein erster Prototyp der Mobilen Deep Map Systems für das Heidelberger Schloss, Universität Heidelberg, 1999

[Zip99b] Zipf, Alexander; Malaka: Web- basierte Planung und animierte Visualisierung von 3D Besichtigungstouren im Rahmen des Touristeninformationssystems Deep Map, Heidelberg, 1999

[Zip01] Zipf, Alexander; Projektschlussbericht: Deep Map/GIS, Universität Heidelberg, 2001

C Abbildungsverzeichnis

Abbildung 1: Dijkstra Pseudo Code [Coh04]	7
Abbildung 2: Suchverlauf bei A* [Rus03].....	8
Abbildung 3: Verbesserung des Suchraumes durch bidirektionale Suche [San05]	9
Abbildung 4: Präsentationsbaum in IRREAL [Wah01].....	13
Abbildung 5: Die ARREAL-Systemkomponenten [Wah01].....	14
Abbildung 6: Varianten in der Anzeige bei ARREAL [Wah01]	15
Abbildung 7: Ebenensichtweise der Gesamtarchitektur [Zip99]	16
Abbildung 8: Hardwarekomponenten von Deep Map [Zip99a]	17
Abbildung 9: Tourenabschnitt vom Deep Map [Zip99].....	18
Abbildung 10: INSTAR als Black Box [Nar03]	19
Abbildung 11: Erstellung der AR Ansicht [Nar03]	20
Abbildung 12: INSTAR Wegweisung [Nar03].....	21
Abbildung 13: INSTAR als mobiles System [Nar03].....	22
Abbildung 14: Benutzer trägt das Mobile Augmented Reality Setup [Rei01]	23
Abbildung 15: Darstellung der Komponenten des Mobile Augmented Reality Setups [Rei01]	24
Abbildung 16: Visualisierter Weg [Int06]	25
Abbildung 17: Eingebetteter Weg [Rei03].....	25
Abbildung 18: Anreicherung der Ansicht durch AUG und WIM [Kna07]	26
Abbildung 19: ARToolkit Marker [Kna07]	27
Abbildung 20: Visualisierte Navigationsinformationen [Kna07]	27
Abbildung 21: GUIDE Benutzeroberfläche [Dist06].....	29
Abbildung 22: GUIDE Wegführung [Dis06].....	30
Abbildung 23: GUIDE Übersichtskarte [Dist06].....	31
Abbildung 24: Beispiele für Handy-Navigationssysteme	32
Abbildung 25: Erweiterung der Karte um den Graphen der gehbaren Wege	35
Abbildung 26: Kompletter Graph bestehend aus den Teilgraphen zweier Karten.....	36
Abbildung 27: Fehlen der globalen Bezüge	38
Abbildung 28: Harmonische Wegführung	39
Abbildung 29: Einordnung der Klassen des Navigationseditors in die geschichtete Architektur	41
Abbildung 30: Klassendiagramm des Navigationseditors	43
Abbildung 31: Einordnung der Klassen des Navigationsmoduls in die geschichtete Architektur	44
Abbildung 32: Klassendiagramm des Navigationsmoduls	47
Abbildung 33: Integration der wichtigsten Funktionen in die bestehende Menüleiste.....	49
Abbildung 34: Menü und Dialoge zum Löschen von Knoten und Kanten	50
Abbildung 35: State Chart des Navigationseditors	51
Abbildung 36: Kartenausschnitt mit unterschiedlichen Knoten	52
Abbildung 37: Navigationsmenü	54
Abbildung 38: Statechart des Navigationsmoduls	55
Abbildung 39: Visualisierung des berechneten Weges.....	56
Abbildung 40: Unterscheidung der Richtungsangaben.....	57
Abbildung 41: Die vier einzelnen Karten des Spielgebiets des Benutzertests.....	60
Abbildung 42: Kreisdiagramm der Tätigkeiten	60
Abbildung 43: Ein mit einem Sensorboard ausgestatteter PDA	61

Indoor Navigation	80
Diplomarbeit von Thomas Rauber	
Abbildung 44: Zeitmessung im Navigationstest	63
Abbildung 45: Beispiel eines Config-Files	86

D Tabellenverzeichnis

Tabelle 1: Persönliche Werte der Testpersonen	60
Tabelle 2: Überblick über die gemessenen Zeiten	65
Tabelle 3: Werteübersicht zum Navigationsmenü	66
Tabelle 4: Korrelation zwischen der Bedienung des Navigationsmenüs und der des mExplorers	66
Tabelle 5: Werteübersicht zur Anzeige des Weges.....	67
Tabelle 6: Werteübersicht zur graphischen Wegleitung	68
Tabelle 7: Werteübersicht der akustischen Wegleitung.....	68
Tabelle 8: Werteübersicht der Gesamtbeurteilung.....	69

E Inhalt der CD-Rom

Die beiliegende CD-Rom enthält folgende Daten:

Verzeichnis	Inhalt
Diplomarbeit	PDF der Arbeit Word-Dokument der Arbeit
Source Code	Source Code des Servers Source Code des Clients
Javadoc	Javadoc des Servers Javadoc des Clients
Daten des Benutzertests	Daten des Fragebogens zum Navigationstest Daten des Fragebogens zum eigentlichen Spiel Daten der Zeitmessung im Navigationstest Log-Files der Tests
Install-Package	Install-Package für den Client
Zusammenfassung	Zusammenfassung der Arbeit auf deutsch und englisch.

F Anwendungsfälle

Hier sind alle Anwendungsfälle des Navigationseditors und auch des Navigationsmoduls angefügt.

F.1 Anwendungsfälle des Navigationseditors

Anwendungsfall	Navigationseditor an
Akteur	Benutzer/Spieler
System	Editor
Referenzen	-
Beschreibung	Wird der Navigationseditor eingeschaltet, so werden die Buttons zum Erstellen von Knoten und Kanten, wie auch zum Prüfen des Graphen aktiv. Der Graph der gehbaren Wege wird angezeigt. Ist der Navigationsmodus an, so kann der Server nicht gestartet werden. Versucht man dies, so erscheint eine entsprechende Fehlermeldung.

Anwendungsfall	Navigationseditor aus
Akteur	Benutzer/Spieler
System	Editor
Referenzen	-
Beschreibung	Wird der Navigationseditor ausgeschaltet, so werden dessen Buttons inaktiv und der Graph wird nicht mehr angezeigt.

Anwendungsfall	Knoten erstellen
Akteur	Benutzer/Spieler
System	Editor
Referenzen	-
Beschreibung	Wird der Button "Create Nodes" angewählt, so können Knoten durch blosses Klicken auf der Karte erstellt werden. Bei jedem Klick wird ein Knoten an der entsprechenden Stelle der Karte angezeigt und dem Graphen hinzugefügt.

Anwendungsfall	Kanten erstellen
Akteur	Benutzer/Spieler
System	Editor
Referenzen	-
Beschreibung	Wird der Button "Create Edges" angewählt, so können Kanten durch anklicken der entsprechenden Knoten erstellt werden. Wird der erste Knoten angewählt, so wird er entsprechend markiert. Wird der zweite angewählt, so wird die Kante gezeichnet und den Graphen hinzugefügt und der markierte Knoten wieder zurückgesetzt.

Anwendungsfall	Knoten verschieben
Akteur	Benutzer/Spieler
System	Editor
Referenzen	-
Beschreibung	Ist weder Knoten noch Kanten erstellt, so können Knoten mit gedrückter Maustaste verschoben werden. Wird der Knoten wieder losgelassen wird seine Position aktualisiert und die Längen der Kanten, die von ihm ausgehen, neu berechnet.

Anwendungsfall	Kanten löschen
Akteur	Benutzer/Spieler
System	Editor
Referenzen	-
Beschreibung	Eine Kante wird durch Angabe deren beiden Knoten im entsprechenden Dialog gemacht. Bei fehlerhaften Eingaben erscheint eine Fehlermeldung. Bei korrekter Angabe der Knoten wird die Kante aus dem Graph gelöscht und nicht mehr angezeigt.

Anwendungsfall	Knoten löschen (Dialog)
Akteur	Benutzer/Spieler
System	Editor
Referenzen	Kanten löschen
Beschreibung	Wählt ein Benutzer "Delete Node" an einer Stelle, an der kein Knoten ist, so öffnet sich ein Dialog zum Löschen des Knotens. Wird dort ein gültiger Knoten angegeben, so werden zuerst alle dessen Kanten gelöscht (-> Kanten löschen) und danach der Knoten selbst. Anschliessend wird weder der Knoten noch dessen Kanten mehr angezeigt.

Anwendungsfall	Knoten direkt löschen
Akteur	Benutzer/Spieler
System	Editor
Referenzen	Kanten löschen
Beschreibung	Wählt ein Benutzer "Delete Node" bei einem Knoten, so werden zuerst dessen Kanten gelöscht (-Kanten löschen) und danach der Knoten selbst. Anschliessend wird weder der Knoten noch dessen Kanten mehr angezeigt.

F.2 Anwendungsfälle des Navigationsmoduls

Anwendungsfall	Ziel wählen
Akteur	Benutzer/Spieler
System	Client
Referenzen	-
Beschreibung	Beim Start der Navigation kann der Anwender ein beliebiges Ziel durch einen Klick auf die entsprechende Karte wählen. Zur Wahl des Ziels kann zwischen den einzelnen Karten gewechselt werden. Wurde das Ziel gewählt, so wird der Weg berechnet und auf der Karte angezeigt.

Anwendungsfall	Ein-/Ausschalten der Navigation
Akteur	Benutzer/Spieler
System	Client
Referenzen	Ziel wählen
Beschreibung	<p>Die Navigation kann nach belieben ein- bzw. ausgeschaltet werden: Ist die Navigation ausgeschaltet, so kann sie jederzeit durch einen Klick auf "Start Navigation" eingeschaltet werden, worauf ein Ziel angegeben werden muss (-> Ziel wählen).</p> <p>Läuft die Navigation, so kann sie durch einen Klick auf "Stopp Navigation" beendet werden. Daraufhin gelangt der Benutzer automatisch in die Kartenansicht. Auf der Karte wird nun der berechnete Weg nicht mehr angezeigt.</p>

Anwendungsfall	Hunted Team als Ziel wählen
Akteur	Benutzer/Spieler
System	Client
Referenzen	-
Beschreibung	<p>Durch einen Klick auf "Hunted Team" kann der Benutzer das gejagte Team als Ziel wählen. Daraufhin gelangt er wieder in die Kartenansicht, wo der kürzeste Weg zum gejagten Team angezeigt wird.</p>

Anwendungsfall	Farbe wählen
Akteur	Benutzer/Spieler
System	Client
Referenzen	-
Beschreibung	<p>Bei laufender Navigation kann der Spieler die Farbe des angezeigten Weges durch einen Klick auf "Path Color" verändern. Dadurch gelangt er in einen Farbwahl-Dialog, wo er die gewünschte Farbe auswählen kann. Kehrt er in die Kartenansicht zurück, so wird der Weg neu in der gewählten Farbe dargestellt.</p>

Anwendungsfall	Audio an/aus
Akteur	Benutzer/Spieler
System	Client
Referenzen	-
Beschreibung	<p>Bei laufender Navigation kann der Spieler die akustischen Richtungsangaben über eine Check-Box an- bzw. ausschalten. Ist die Check-Box angewählt, so ertönen die akustischen Richtungsangaben, sonst nicht.</p>

G Config File

Hier folgt ein Beispiel der Konfigurationsdatei mit einer Erläuterung zu den einzelnen Parametern.

```
#Client Configuration File
team_name=Team 7
team_pass=7777
server_ip=130.60.156.137
server_port=9999
map_path=java\\maps\\
icon_path=java\\icons\\
audio_path=java\\audio\\
skype_name=unizh_game_1
skype_port=11444
tag_name=Tag-10
sensor=0
xoffset= 1200
yoffset= 1200
```

Abbildung 45: Beispiel eines Config-Files

team_name: Der Name des Teams (muss identisch auch im Spiel vorkommen).

team_pass: Das Passwort des Teams (muss identisch auch im Spiel vorkommen).

server_ip: IPv4-Adresse des Servers

server_port: Port des Servers

map_path: Der Pfad zum Verzeichnis, in dem die Maps abgelegt sind.

icon_path: Der Pfad zum Verzeichnis, in dem die Icons abgelegt sind.

audio_path: Der Pfad zum Verzeichnis, in dem die Audio-Dateien abgelegt sind.

skype_name: Skype-Benutzername

skype_port: Skype-Port

tag_name: Angabe des Ekahau Tag, den das Team mit sich trägt.

sensor: Angabe über den verwendeten Sensor (**0 = kein Sensor, 1 = Sensorboard, 2 = GPS-Empfänger**)

xoffset: Offset zur Kalibrierung der x-Achse des Magnetfeldsensors (variiert von Sensor zu Sensor)

yoffset: Offset zur Kalibrierung der y-Achse des Magnetfeldsensors (variiert von Sensor zu Sensor)

17.a	Wenn nein, was fehlt?
18. Weitere Anmerkungen zum Navigationsmenü:	

Angezeigter Weg

	trifft nicht zu	---	--	-	0	+	++	+++	trifft zu
19. Der angezeigte Weg ist sehr hilfreich, um einen unbekanntem Ort zu finden.	<input type="checkbox"/>								
20. Es ist mir leicht gefallen, dem angezeigten Weg zu folgen.	<input type="checkbox"/>								
21. Der angezeigte Weg war auf der Karte leicht erkennbar.	<input type="checkbox"/>								
22. Es war mir immer klar, wenn der Weg auf einer anderen Karte weiter ging.	<input type="checkbox"/>								
23. Wenn ich ein Ziel auf der Karte auswählte, wurde immer der korrekte Weg angezeigt.	<input type="checkbox"/>								
24. Wenn ich das gejagte Team als Ziel auswählte, wurde immer der korrekte Weg angezeigt.	<input type="checkbox"/>								
25. Weitere Anmerkungen zur Anzeige des Weges:									

Graphische Wegleitinformationen (Pfeile)

	trifft nicht zu	---	--	-	0	+	++	+++	trifft zu
26. Die Pfeile sind sehr hilfreich, um dem Weg ans Ziel zu folgen.	<input type="checkbox"/>								
27. Es ist klar verständlich, was die Pfeile bedeuten.	<input type="checkbox"/>								
28. Die Pfeile sind gut zu erkennen.	<input type="checkbox"/>								
29. Die Pfeile sind gut unterscheidbar.	<input type="checkbox"/>								
30. Weitere Anmerkungen zu der graphischen Wegleitung:									

Akustische Wegleitinformationen

	trifft nicht zu	---	--	-	0	+	++	+++	trifft zu
31. Die Ansagen waren hilfreich, um dem Weg ans Ziel zu folgen.	<input type="checkbox"/>								
32. Es war klar verständlich, was mit den Ansagen gemeint war.	<input type="checkbox"/>								
33. Die Ansagen waren verständlich gesprochen.	<input type="checkbox"/>								
34. Die Geräusche aus der Umgebung wirkten sich störend auf das Verständnis der Ansagen aus.	<input type="checkbox"/>								
35. Weitere Anmerkungen zu der akustischen Wegleitung:									

49. Ich habe die Zoomfunktion der Karte benutzt.	<input type="checkbox"/>						
50. Es traten keine technischen Probleme auf.	<input type="checkbox"/>						
51. Nenne 3 positive Aspekte?							
52. Nenne 3 negative Aspekte?							

H.3 Durchschnittswerte des Fragebogens zum Navigationstest

Benutzertest des Navigationssystems des mExplorers

Fragen zum persönlichen Profil

5. Ich bin sehr vertraut im Umgang mit Computergeräten.	5.35
6. Schon vor dem Spiel war ich sehr vertraut im Umgang mit PDAs.	2.57
7. Schon vor dem Spiel kannte ich das Gelände des Irchels gut.	4.07

Gesamtbeurteilung

8. Ich erachte das Navigationssystem als sinnvolle Erweiterung.	5.28
9. Das Navigationssystem ist hilfreich, um gezielt einen bestimmten Punkt anzulaufen.	5
10. Ohne das Navigationssystem hätte ich länger gebraucht, um ans Ziel zu kommen.	2.42
11. Ohne das Navigationssystem hätte ich einen anderen Weg ans Ziel genommen.	1.5
12. Ohne das Navigationssystem hätte ich einen kürzeren Weg ans Ziel genommen.	2.14
13. Nachdem ich einen Punkt mittels Navigation erreicht hatte, könnte ich diesen Punkt auch leicht wieder ohne Navigation finden.	5.5

Bedienung des Navigationsmenüs

14. Die Bedienung des Navigationsmenüs ist mir leichtgefallen.	4.14
15. Es ist mir leicht gefallen ein beliebiges Ziel auf der Karte zu wählen.	4.57
16. Es ist mir leicht gefallen, das gejagte Team als Ziel zu wählen.	4
17. Das Navigationsmenü erlaubt den Zugriff auf alle für die Navigation nötigen, Funktionen.	4.64

Angezeigter Weg

19. Der angezeigte Weg ist sehr hilfreich, um einen unbekanntem Ort zu finden.	5.83
20. Es ist mir leicht gefallen, dem angezeigten Weg zu folgen.	5.25
21. Der angezeigte Weg war auf der Karte leicht erkennbar.	5.81
22. Es war mir immer klar, wenn der Weg auf einer anderen Karte weiter ging.	4.5
23. Wenn ich ein Ziel auf der Karte auswählte, wurde immer der korrekte Weg angezeigt.	5.18
24. Wenn ich das gejagte Team als Ziel auswählte, wurde immer der korrekte Weg angezeigt.	3.5

Graphische Wegleitinformationen (Pfeile)

26. Die Pfeile sind sehr hilfreich, um dem Weg ans Ziel zu folgen.	5.12
27. Es ist klar verständlich, was die Pfeile bedeuten.	6
28. Die Pfeile sind gut zu erkennen.	6.37
29. Die Pfeile sind gut unterscheidbar.	6

Akustische Wegleitinformationen

31. Die Ansagen waren hilfreich, um dem Weg ans Ziel zu folgen.	3
32. Es war klar verständlich, was mit den Ansagen gemeint war.	6.14
33. Die Ansagen waren verständlich gesprochen.	6.14
34. Die Geräusche aus der Umgebung wirkten sich störend auf das Verständnis der Ansagen aus.	3.28

Nutzen des Navigationssystems

36. Es ist nützlich, dass man ein beliebiges Ziel auf einer Karte wählen kann.	5.77
37. Es ist nützlich, dass man das gejagte Team als Ziel wählen kann.	5.42
38. Die Anzeige des Weges ans Ziel ist nützlich.	5.55
39. Die Anzeige der Pfeile, in welche Richtung man sich bewegen muss, ist nützlich.	5.83

40. Die Ansagen, in welche Richtung man sich bewegen muss, sind nützlich.	5.5
---	-----

Position und Kompass

41. Die angezeigte Position stimmte immer mit meiner wirklichen Position überein.	1.88
42. Meine Position wurde genau angezeigt.	3.5
43. Meine Blickrichtung wurde immer korrekt angezeigt.	2.8

Fragen zum mExplorer an sich

44. Allgemein sind die Anzeigen genug gross.	5.08
45. Das Spiel ist allgemein leicht zu bedienen.	4.54
46. Es ist einfach in die einzelnen Menüs des Spiels zu gelangen.	5.16
47. Es ist klar, wo sich welche Funktion aufrufen lässt.	4.41
48. Es ist einfach, zwischen den einzelnen Karten zu wechseln.	4.88
49. Ich habe die Zoomfunktion der Karte benutzt.	6.72
50. Es traten keine technischen Probleme auf.	2.81

H.3.1 Durchschnittswerte zum Auszug des Fragebogens zum eigentlichen Spiel

2.24 Die Navigation ist eine sinnvolle Erweiterung des mExplorers.	5.9
2.25 Die Navigation ist gut in den mExplorer integriert.	6.2
2.26 Die Navigation behindert die restlichen Funktionen des Spiels nicht.	6.3
2.27 Ich habe die Navigation während dem Spiel benutzt.	5.33
2.28 Ich benutzte die Navigation um bestimmte Orte gezielt anzulaufen.	4.91
2.29 Ich nutzte die Navigation um andere Teams zu fangen.	3.33