



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Sören Voskuhl

Bewegungsbasierte Computerinteraktion zur Navigation in
Informationsbeständen

Sören Voskuhl

Bewegungsbasierte Computerinteraktion zur Navigation in
Informationsbeständen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter: Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 7. März 2009

Sören Voskuhl

Thema der Bachelorarbeit

Bewegungsbasierte Computerinteraktion zur Navigation in Informationsbeständen

Stichworte

Seamless Interaction, Intelligent Home, Bewegungsbasierte Steuerung, Dreidimensionale Objektdarstellung, persönliche Informationsbestände

Kurzzusammenfassung

In der wachsenden Allgegenwärtigkeit von Computern spielt die Benutzerfreundlichkeit eines Systems eine tragende Rolle. Folglich ist die intuitive Bedienung ein wesentliches Kriterium in der Mensch Computer Interaktion (HCI).

Zudem wächst mit der Anzahl der Computer die Größe der persönlichen Informationbestände einer Person. Im Rahmen dieser Bachelorarbeit ist ein System entwickelt worden, welches den Zugriff auf einen Teil dieser Daten koordiniert und vereinfacht. Hierzu stellt es eine Filterfunktion für Bilderbestände bereit und präsentiert relevante Daten in einem dreidimensionalen Raum. Weiterhin ist mit einer bewegungsbasierten Steuerung eine neuartige Interaktionsform zur Navigation durch diese Bestände umgesetzt worden.

Sören Voskuhl

Title of the paper

Motion based computer interaction to navigate in information sources

Keywords

Seamless Interaction, Intelligent Home, Motion based Control, Three-dimensional object representation, Personal information resources

Abstract

In the growing ubiquity of computers, the usability of a system plays a significant role. Consequently, intuitive handling is essential in human-computer interaction (HCI).

Furthermore, a person's personal resource of information increases with the number of computers. Within this bachelor thesis, a system has been developed to coordinate and simplify access to certain parts of the data. This includes providing a filter function for images and presenting relevant data in a three-dimensional space. Additionally, by integrating a motion-based control, a novel form of interaction has been implemented for the navigation through these inventories.

Inhaltsverzeichnis

1	Einführung	7
2	Grundlagen	10
2.1	Human Computer Interaction	10
2.2	Mentale Modelle	10
2.3	Seamless Interaction	11
2.4	Bewegungsbasierte Interaktion	12
2.5	Objektdarstellung in dreidimensionalen Räumen	12
2.6	Ubiquitous Computing	13
2.7	Context Awareness	14
2.8	Intelligent Home	15
3	Analyse	16
3.1	Szenarien	17
3.1.1	Interaktives Museum	17
3.1.2	Digitale Postkarte	17
3.2	Anforderungsanalyse	19
3.2.1	Funktionale Anforderungen	19
3.2.2	Nichtfunktionale Anforderungen	23
3.2.3	Grundlegende Funktionalität des Systems	25
3.3	Objektdarstellung und Navigation in dreidimensionalen Räumen	26
3.3.1	Orientierung in dreidimensionalen Räumen	26
3.3.2	Interaktion mit Objekten in dreidimensionalen Räumen	27
3.3.3	Bewegungstechniken	28
3.3.4	Perspektiven	28
3.3.5	Mentale Modelle in dreidimensionalen Räumen	29
3.4	Fazit	29
4	Bewegungen	30
4.1	Begriffserklärung	30
4.2	Technologien	30
4.2.1	EyeToy	31
4.2.2	Segway	31

4.2.3	Neural Impulse Actuator	32
4.2.4	Nintendo Wii	33
4.3	Interpretation von Bewegungen	35
4.3.1	Beginn und Ende einer Bewegung	35
4.4	Fazit	37
5	Design	38
5.1	Überblick des Systems	38
5.1.1	Zusammenspiel der Komponenten	39
5.2	Architektur	42
5.2.1	Model View Controller	43
5.2.2	MVC im System	44
5.3	Pattern	46
5.3.1	Observer Pattern	47
5.3.2	Singleton	48
5.4	Gestaltung der dreidimensionalen Galerie	49
5.4.1	Generierung der Bilderlandschaft	49
5.4.2	Anordnung der Bilder	51
5.4.3	Orientierung und Perspektive innerhalb der Galerie	52
5.5	Agile Softwareentwicklung	53
5.5.1	Extreme Programming	53
5.5.2	Test-Driven Development	53
5.5.3	Prototyping	54
5.6	Machbarkeitsstudie	55
5.6.1	Bewegungserkennung	57
5.6.2	Cooliris	57
5.6.3	GoogleEarth	59
5.6.4	Ergebnis der Machbarkeitsstudie	60
6	Realisierung	62
6.1	Filter	62
6.1.1	Bilder mit Metadaten versehen	62
6.1.2	Auslesen der Metadaten	63
6.1.3	Verwendete Metadaten	63
6.1.4	Arbeitsweise des Filters	64
6.2	Darstellung	65
6.2.1	Croquet	66
6.2.2	Umsetzung der Galerien	67
6.3	Steuerung	70
6.4	Evaluation	72
6.4.1	Bewertung der Umsetzung	72

6.4.2 Besuchergruppen	73
6.4.3 Usability-Test	73
6.5 Auswertung	77
7 Schluss	79
7.1 Zusammenfassung	79
7.2 Ausblick	80
7.2.1 Erweiterung des Systems	80
7.2.2 Fazit	81
Literaturverzeichnis	83

1 Einführung

Die Entwicklung in der Informatik zeichnet sich durch immer kleinere und leistungsfähigere Computer aus. Aufgrund dieser Eigenschaften halten sie immer mehr Einzug in das alltägliche Leben des Menschen. Mark Weiser beschreibt die zunehmende Allgegenwärtigkeit von Informationstechnologien folgendermaßen ([Weiser, 1993](#)):

„Ubiquitous Computing enhances computer use by making many computers available throughout the physical environment, while making them effectively invisible to the user.“

Diese Definition von allgegenwärtiger Rechenleistung beschreibt die unbewusste Nutzung von Computern, um digitale Informationen aus der gesamten Umgebung aufzunehmen. Um dieses Ziel zu erreichen werden heute verschiedene Technologien entwickelt, welche dazu dienen, die Benutzerschnittstelle zur Interaktion mit Computern für den Menschen als natürlich zu gestalten. Dazu ist es erforderlich, die Informationstechnologien sowie deren Bedienung an die Erwartungen des Menschen anzupassen und nicht, wie es mit klassischen Eingabegeräten der Fall ist, eine Anpassung des Menschen an die Handhabung der Maschine vorauszusetzen.

Die Integration von Computern in den menschlichen Alltag hat zur Folge, dass die Anzahl an Informationsträgern mit denen eine Person interagiert, wächst. Um den Zugriff auf die Informationen, welche sich auf den verschiedenen Geräten befinden, zu koordinieren, beschreibt [Pierce \(2008\)](#) mit Personal Information Environments (PIE) ein Modell, das sich aus mehreren heterogenen Geräten eines Anwenders zusammensetzt. Dabei stellen die verschiedenen Computer jeweils unterschiedliche Informationen zur Verfügung und kennen alle weiteren Einheiten innerhalb der Umgebung, um Aufgaben ggf. in Kombination mit anderen Geräten zu bearbeiten. Eine wesentliche Aufgabe eines Computers bei der Informationsabfrage durch den Menschen ist, ausschließlich dem Kontext entsprechende Daten zu liefern und ihn nicht mit irrelevanten Angaben abzulenken. Denn je übersichtlicher und präziser die Daten vorliegen, desto größer wird der Nutzen des Anwenders.

Um eine natürliche Benutzerschnittstelle zu realisieren, wurden bereits Technologien entwickelt, welche dazu dienen sollen, die Computerinteraktion für den Menschen zu vereinfachen. Ein Ansatz liegt hierbei in der multimodalen Interaktion. Zu dieser Interaktionsform gehören z.B. Sprachsteuerungen, bewegungsbasierte Steuerungen sowie Multitouch

Systeme. Um das von Weiser beschriebene Ziel zu erreichen und somit dem Anwender eine nahtlose Interaktion mit dem System zu ermöglichen, werden in diesem Forschungsbereich häufig Kommunikationstechniken zwischen Mensch und der realen Welt auf die Bedienung eines Computers abgebildet.

Der fulminante Erfolg der Suchmaschine Google zeigt, wie unentbehrlich es ist, Daten zu filtern, um aus den Datenbeständen wie sie das Internet liefert, die gewünschten Informationen zu gewinnen. Dabei wächst durch die steigende Anzahl der Computer, die eine Person besitzt, in der Regel auch die Größe seines persönlichen Informationsbestandes. Aus diesem Grund wird in dieser Arbeit betrachtet, in welcher Form diese Datenbestände gefiltert und in eine übersichtliche Darstellung gebracht werden können.

Eine weitere zentrale Fragestellung, die in dieser Arbeit aufgegriffen wird, lautet: Wie lässt sich eine intuitive Interaktion mit einem Computersystem gestalten, mit dem Ziel durch Informationsbestände zu navigieren?

Hierzu wird mit der bewegungsbasierten Computersteuerung eine multimodale Interaktionsform untersucht und das Ziel verfolgt, eine natürlich zu bedienende Computersteuerung zu realisieren. Dabei dienen mentale Modelle als Grundlage, um eine einfache Interaktion mit einem Computersystem zu ermöglichen.

Gliederung der Arbeit

In Kapitel 2 werden die für das Verständnis der Thematik notwendigen Grundlagen vorgestellt. Dazu gehört Human Computer Interaction und einige ihrer Ausprägungen, die Objektdarstellung in dreidimensionalen Räumen sowie den Einzug von Computern in den menschlichen Alltag.

Das Analyse-Kapitel (3) dient dazu, anhand von Szenarien auf die Anforderungen an das System zu schließen. Dabei werden aus den funktionalen und nicht-funktionalen Anforderungen grundlegende Funktionen der zu realisierenden Anwendung abgeleitet. Des Weiteren werden wesentliche Aspekte und Problemstellungen, die sich bei der Objektdarstellung und der Navigation innerhalb dreidimensionaler Räume ergeben, aufgezeigt.

Das Kapitel 4 dient zum tieferen Verständnis menschlicher Bewegungen. Neben einer Begriffserklärung werden an dieser Stelle verschiedene Technologien vorgestellt, die sich mit bewegungsbasierter Computerinteraktion und ihrer semantischen Interpretation befassen.

Im Abschnitt 5 wird das Design des Systems vorgestellt. Nach einer Übersicht der Komponenten wird hier unter Verwendung geeigneter Entwurfsmuster die Architektur der Anwendung erläutert. Außerdem wird die Generierung dreidimensionaler Räume unter Berücksich-

tigung der aus der Analyse bekannten Problemfelder diskutiert. Anschließend werden zwei implementierte Prototypen in einer Machbarkeitsstudie als exemplarische Aufzeigung einer bewegungsbasierten Steuerung vorgestellt und mögliche Eingabegeräte evaluiert.

In der Realisierung (6) folgt eine Beschreibung der technischen Umsetzung der Anwendung, welche sich an dem vorgestellten Design orientiert. Daraufhin wird eine Evaluation der implementierten Steuerung durchgeführt, beruhend auf Reaktionen mehrerer Testpersonen sowie eines systematischen Usability-Tests.

Das letzte Kapitel (7) enthält eine Zusammenfassung und stellt mögliche Erweiterungen und Verbesserungen des Systems vor.

2 Grundlagen

In diesem Kapitel werden kurz die grundlegenden Begriffe vorgestellt, die zum Verständnis dieser Arbeit notwendig sind.

2.1 Human Computer Interaction

Für den Begriff Human Computer Interaction (HCI) gibt es keine allgemein gültige Definition. Laut ACM SIGCHI (ACM's Special Interest Group on Computer-Human Interaction) ist HCI ein Wissenszweig, der sich mit der Gestaltung, Evaluation und Implementation von interaktiven Computersystemen für den menschlichen Gebrauch und der Untersuchung bedeutender verwandter Themen befasst. ([SIGCHI, 1996](#))

Aus Sicht der Informatik liegt der Fokus besonders auf der Interaktion zwischen einem oder mehreren Menschen mit einem oder mehreren Computern. Dabei können letztere in Form von eingebetteten Rechnern in alltäglichen Gebrauchsgegenständen bis hin zu sehr leistungsstarken Großrechnern auftreten. Ein entscheidender Ansatz bei der Human Computer Interaction ist es, dem Akteur, der mit dem Computer interagiert, den Gebrauch des Gerätes so einfach wie möglich zu gestalten. Der Bereich Seamless Interaction befasst sich mit Interaktionen dieser Art ([2.3](#)).

An den Forschungsergebnissen der ACM SIGCHI sind u.a. die Stanford HCI Group ([Stanford, 2008](#)) und Xerox Parc ([Parc, 2008](#)) maßgeblich beteiligt.

2.2 Mentale Modelle

Unter einem Modell versteht man eine auf das Wesentliche reduzierte Darstellung der Realität. Ein mentales Modell ist die Repräsentation eines Gegenstandes oder Prozesses im Bewusstsein. Die Behauptung, dass Menschen solche Modelle aufbauen, wurde erstmals von Kenneth Craik aufgestellt ([Craik, 1943](#)). Diese Abbilder der Wirklichkeit werden aufgrund von äußeren Eindrücken aufgebaut und helfen dem Menschen mit neuen Situationen zurecht zu kommen, indem das zu erklärende Phänomen mit dem bisherigen Wissen in Verbindung

gesetzt wird. Diese Denkmuster sind dynamisch, dies bedeutet, dass sich durch Lernerfahrungen die Erwartungen und Vorstellungen an die Umwelt verändern können.

In der Informatik werden Mentale Modelle dazu verwendet, die Erwartungen eines Benutzers an ein System zu erfassen. Dabei kann der Mensch diese während der Interaktion mit einem System modifizieren, damit er ein besseres Resultat des Computers erhält. Nach [Norman \(1983\)](#) hängt der Aufbau dieses Modells dabei vom technischen Verständnis, bereits gesammelten Erfahrungen mit ähnlichen Systemen sowie der Aufbereitung von Informationen des Anwenders ab. Des Weiteren beschreibt er, dass mentale Modelle unvollständig sowie instabil sind. Darunter ist zu verstehen, dass ein Anwender Details über das System vergisst, falls er es längere Zeit nicht verwendet.

Für den Entwickler eines Systems ist es elementar ein System zu gestalten, dass möglichst exakt dem mentalen Modell des Anwenders entspricht, um eine intuitive Bedienung zu ermöglichen ([Jacko und Sears, 2003](#)). Umso größer das Ungleichgewicht zwischen der Intention des Entwicklers und den Erwartungen des Benutzers ist, desto unterschiedlicher sind die Annahmen hinsichtlich der Funktionalität des Systems und umso schwieriger ist es für den Anwender mit dem System zu interagieren.

2.3 Seamless Interaction

Bei einer Seamless Interaction (deutsch nahtlose Interaktion) geht es darum, dem Anwender eines Computers die Benutzung so einfach und intuitiv wie möglich zu gestalten. Dieses wird dadurch erreicht, dass zur Verwendung eines Gerätes keine aufwändige Einarbeitungsphase benötigt wird, wie es beim Durcharbeiten eines Handbuches der Fall ist, sondern direkt in Betrieb genommen werden kann, da sofort erkennbar ist, wie es zu bedienen ist. Ein wichtiger Ansatz bei der Realisierung einer Interaktion dieser Art, ist die Verwendung von natürlichen menschlichen Gesten und Bewegungen aus der zwischenmenschlichen Kommunikation.

In [Ishii u. a. \(1994\)](#) werden Ziele beschrieben, die dazu beitragen, eine nahtlose Interaktion zu ermöglichen. Dabei handelt es sich zum einen um „Continuity“, welche sich mit der Kontinuität von Arbeitspraktiken befasst. Personen entwickeln häufig eigene Praktiken, indem sie eine Auswahl verschiedener Tools treffen und mit einer großen Anzahl von Menschen interagieren. Deshalb ist es essentiell eine gleichbleibende Bedienung zu wahren.

Zum anderen handelt es sich um „Smooth Transition“. Dieses Ziel befasst sich mit dem Übergang zwischen Funktionsräumen und -modi. Hier soll die kognitive Belastung des Anwenders vermindert werden, indem ein dynamischer Wechsel zwischen verschiedenen Funktionen ermöglicht wird.

2.4 Bewegungsbasierte Interaktion

Die Daten der Erfassung und Analyse menschlicher Bewegungen können zur Realisierung von Interaktionen mit Computern verwendet werden. Dabei kann es sich um Körperbewegungen mit Kopf, Armen und Beinen sowie des gesamten Rumpfes handeln. Bevor die erhobenen Daten ausgewertet und weiterverarbeitet werden können, müssen sie vom System erfasst werden. Dieser Erhebungsprozess wird als „Human Motion Capture“ bezeichnet. An dieser Stelle werden zwei bedeutende Einsatzgebiete menschlicher Bewegungsdaten erläutert ([Moeslund und Granum, 2001](#)):

1. **Steuerung:** Dieses Gebiet bezieht sich auf Applikationen, in denen erfasste Bewegungsdaten verwendet werden, eine Steuerungskomponente bereitzustellen. Dabei kann sie als Schnittstelle zu Computerspielen, virtuellen Umgebungen oder Animationen dienen.
2. **Analyse:** Die erhobenen Daten können auch zur Analyse von Bewegungen verwendet werden. Sie dienen dazu klinische Studien durchzuführen oder Athleten bei der Verbesserung ihrer Leistungsfähigkeit zu helfen.

2.5 Objektdarstellung in dreidimensionalen Räumen

Zur Darstellung von Objekten und Informationen in dreidimensionalen Räumen gibt es verschiedenen Ansätze. An dieser Stelle wird mit dem Cone Tree eine Form der Visualisierung vorgestellt, welche einige vorteilhafte Ansätze einer 3D-Darstellung beschreibt.

Unter einem Cone Tree (deutsch Kegelbaum) versteht man eine Visualisierungsmöglichkeit zur Präsentation hierarchischer Informationsstrukturen. Bei dieser Technik, die vom Xerox PARC (Palo Alto Research Center) entwickelt wurde ([Robertson u. a., 1991](#)), werden die Daten in 3D dargestellt, mit der Absicht den zur Verfügung stehenden Bildraum möglichst umfassend auszunutzen und eine Visualisierung der gesamten Struktur zu ermöglichen. Die kognitive Belastung, welche bei der Orientierung in großen Informationsbeständen auftritt, wird durch die Verwendung von interaktiven Animationen vermindert. Dabei besteht die Möglichkeit einzelne Teilbäume oder Knoten zu drehen, damit Informationen in den Hintergrund gerückt werden. Zur Darstellung des Baumes werden Kegel eingesetzt, wobei die Spitze den Vaterknoten repräsentiert und auf der unteren Mantelfläche die Kinder liegen. Bei jedem Kindknoten, der weitere Nachfolger besitzt, beginnt ein neuer Kegel (2.1).

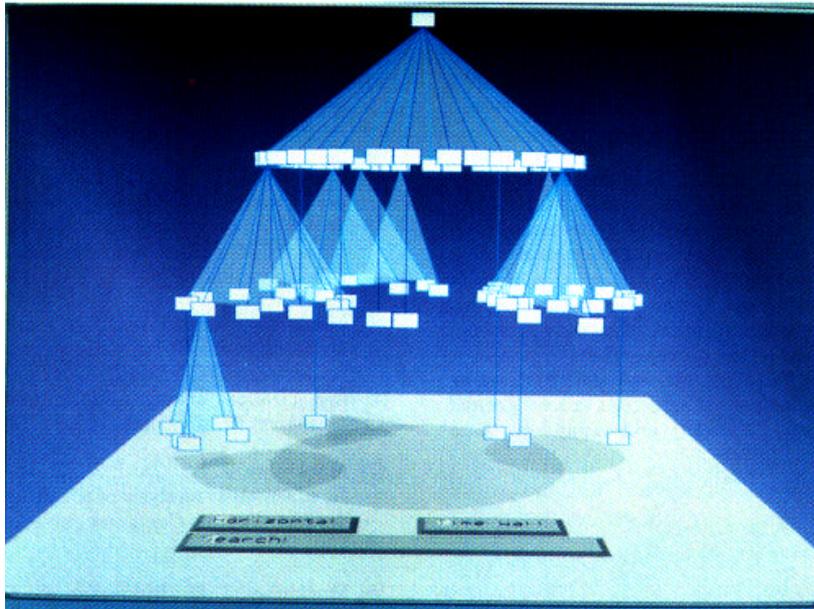


Abbildung 2.1: Beispiel eines Cone Trees ([Robertson u. a., 1991](#))

2.6 Ubiquitous Computing

Der Begriff Ubiquitous Computing, unter dem die Allgegenwärtigkeit von Rechenleistung verstanden wird (siehe [Abbildung 2.2](#)), wurde 1988 von Mark Weiser geprägt. In einem Paper „The Computer for the 21st Century“ ([Weiser, 1991](#)) aus dem Jahre 1991 spricht er davon, dass der Personal Computer (PC) als Gerät menschlicher Aufmerksamkeit im Hintergrund verschwindet und dadurch nur noch unbewusst verwendet wird. Die Idee dabei ist, beliebige Dinge des Alltags miteinander zu vernetzen, um den Menschen beim Erledigen verschiedenster Tätigkeiten zu unterstützen.

Dadurch soll der enge Kanal über Tastatur, Maus und Bildschirm, der zur Interaktion mit einem PC besteht, aufgelöst werden. Eine solche Auflösung beinhaltet, dass eine Schnittstelle zum Unterbewusstsein des Menschen zu Computern geschaffen werden muss, sodass sich die Benutzung als selbstverständlich und nicht als unnatürlich anfühlt. Dabei ist das Ziel, Computer so allgegenwärtig zu machen, wie z.B. Elektrizität, die jederzeit verfügbar ist und ohne weiteres Hintergrundwissen über die Herkunft oder Entstehung verwendet wird.

In [Ishii und Ullmer \(1997\)](#) werden konkrete Ansätze vorgestellt, um das Modell einer GUI auf einem flachen rechteckigen Bildschirm, die von den klassischen Eingabegeräten gesteuert wird, zu erweitern. Mit dem Ziel Computer allgegenwärtig und unsichtbar zu machen, wird hier versucht einen neuen Typen der HCI einzuführen, die "Tangible User Interfaces"(TUIs). Diese Interfaces sollen dabei die reale Welt um digitale Informationen in alltäglichen Ob-

jekten und Umgebungen erweitern, indem der Übergang von der GUI zu TUIs die gesamte Umgebung als Schnittstelle definiert.

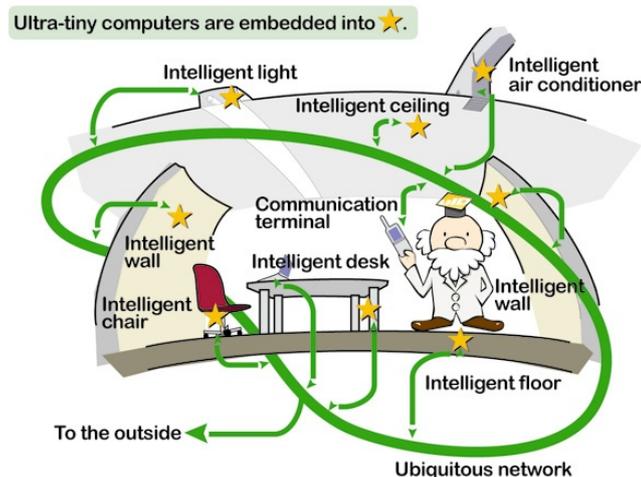


Abbildung 2.2: Beispiel für Ubiquitous Computing (Center, 2008)

2.7 Context Awareness

Man spricht von Context-Awareness, wenn Anwendungen den aktuellen Kontext miteinbeziehen, um Menschen mit Informationen zu versorgen oder Dienste zur Verfügung zu stellen. Bei einer zwischenmenschlichen Kommunikation spielt der Kontext eine große Rolle, um verschiedene Aktionen zu verstehen. Wenn jedoch Menschen und Maschinen kommunizieren, lässt sich der Bezugsrahmen deutlich schwieriger bestimmen. Aus diesem Grund müssen Computer Informationen selbstständig aufnehmen und dynamisch darauf reagieren können. Es gibt vier entscheidende Kategorien, nach denen der Kontext bestimmt werden kann (Dey und Abowd, 1999). Dabei handelt es sich um Ort, Identität, Aktivität und Zeit. Der Ort spezifiziert in welcher Umgebung sich die Anwendung bzw. die Person befindet und welche Daten über den Raum vorliegen. Über die Identität lässt sich feststellen, wer sich innerhalb eines Rahmens befindet und welche Tätigkeit der Mensch oder die Maschine ausführt. Durch die Kategorie Zeit lässt sich über die Uhrzeit oder das Datum der aktuelle Zeitpunkt bestimmen. Für das Sammeln der benötigten Informationen werden verschiedene Sensoren eingesetzt und interpretiert, sodass Schlüsse daraus gezogen werden können und ggf. entsprechend reagiert wird.

2.8 Intelligent Home

Ein Wohnbereich wird als Intelligent Home bezeichnet, wenn innerhalb dieses Bereichs Computer dafür zuständig sind mehr Energieeffizienz, Komfort, Wirtschaftlichkeit, Flexibilität und Sicherheit zu schaffen. Die Rechner sammeln dabei zur Feststellung des Kontextes Informationen über ihr Umfeld und reagieren ggf. entsprechend auf Veränderungen. Hierbei können verschiedene Faktoren, wie Belichtung, Geräuschpegel, Netzwerkkonnektivität, Bandbreite zur Kommunikation sowie die soziale Situation, wie z.B. weitere anwesende Mitmenschen, in die Bestimmung des Kontextes einfließen (Schilit, 1994). Durch diese allgegenwärtige Rechenleistung rückt die Benutzerschnittstelle des Computers immer weiter in den Hintergrund und ermöglicht eine intuitive Interaktion mit dem Gerät.

Ein Projekt, das zu einem Intelligent Home beitragen soll, indem es dem Anwender die Bedienung verschiedener Geräte innerhalb eines Wohnbereichs erleichtert, ist LinuxMCE (Linux Media Center Edition)¹. Diese Erweiterung für die Linux-Distribution Kubuntu bietet neben einer Serverfunktion für Musik und Filme, die Möglichkeit als zentrale Steuereinheit für die digitale Hausverwaltung zu fungieren. Mit zusätzlicher Hardware lassen sich diverse Geräte, wie Lampen oder Heizungen, an das System anschließen, welche sich dann über einen sogenannten Orbiter steuern lassen. Die Funktionalität dieser Steuerung kann von verschiedenen Geräten, wie einem Handy oder einem PC, übernommen werden. Somit ist es möglich das gesamte System durch den Einsatz entsprechender Geräte anhand von Bewegungen und Gesten zu steuern.

¹<http://linuxmce.com/>

3 Analyse

Mark Weiser hatte bereits im Jahre 1991 genaue Vorstellungen davon, was ein Intelligent Home in der Zukunft leisten könnte. Innerhalb eines Szenarios, in dem er einen Tag im Leben von Sal beschreibt (Weiser, 1991), wird erklärt, in welcher Form sie bei alltäglichen Tätigkeiten von Computern unterstützt wird.

Unter anderem von dieser Vision inspiriert, wird seit einigen Jahren verstärkt daran gearbeitet, dem Menschen im privaten Wohnbereich mehr Komfort und Sicherheit durch allgegenwärtige Computer zu ermöglichen. Dabei zeigen aktuelle Forschungsprojekte, wie das Philips HomeLab² oder das „Amigo“-Projekt³, dass sich dieser Bereich stetig weiterentwickelt.

Ein Aspekt bei der Gestaltung eines Intelligent Home ist, dass jede Person ein persönliches Informationsfeld besitzt. Sie greift auf diese Informationen zu, indem sie nicht mit einem einzelnen Computer interagiert, sondern einer Ansammlung verschiedener Geräte, wie z.B. Desktops, Laptops, Tablets, PDAs und Handys. Einerseits steigt durch den Einsatz mehrerer Computer die Anzahl der zugreifbaren Informationen, andererseits ist jedes Gerät in seinen Funktionen beschränkt. In Dearman und Pierce (2008) wird das erhebliche Problem des Anwenders beschrieben, die verschiedenen Geräte zu verwalten und den Zugriff auf die Informationen zu koordinieren. Aus diesem Grund wurde das Modell des Personal Information Environment (PIE) entwickelt (Pierce, 2008). Hier werden alle Geräte eines Benutzers sowie deren Aktivitäten zusammengeführt. Auf diese Weise soll dem Anwender auf einfache Art die Benutzung oder die Kombination von Computern ermöglicht werden, um somit die gewünschten Informationen zu erhalten.

Diese Informationsbestände einer Person können sehr umfangreich und unübersichtlich sein. Ein interessanter Ansatz ist deshalb die Informationsfilterung, damit ausschließlich wichtige Daten zugreifbar gemacht werden. Aus diesem Grund muss der Computer entscheiden, welche Daten oder Objekte innerhalb einer Situation relevant sind, indem er den aktuellen Kontext erkennt.

Da die Darstellung den Anwender kognitiv entlasten soll, bietet sich eine dreidimensionale Darstellung an. Dabei sollte es für den Menschen möglichst einfach sein auf diese Informationen zuzugreifen und durch sie zu navigieren. Ein intuitiver Zugriff lässt sich dabei z.B.

²<http://www.research.philips.com/technologies/misc/homelab/>

³<http://www.hitech-projects.com/euprojects/amigo/>

durch eine auf Gesten und Bewegungen basierende Steuerung realisieren. Dies zeigen verschiedene Forschungsprojekte, die sich bereits mit großem Erfolg auf dem Markt etabliert haben. Dazu gehören beispielsweise die Nintendo Wii⁴ und das iPhone⁵, welche u.a. aufgrund ihrer intuitiven Bedienung eine große Anzahl von Benutzern aufweisen. Das Ziel dieser Arbeit ist deshalb, eine prototypische Applikation zu entwickeln, die gefilterte Informationen zur Verfügung stellt, durch welche mittels Gesten und Bewegungen navigiert werden kann.

Im Folgenden wird der Kontext dieser Arbeit durch zwei Szenarien erläutert.

3.1 Szenarien

3.1.1 Interaktives Museum

Dieses Szenario beschreibt den Besuch einer Person in einem interaktiven Museum. Dabei soll die Möglichkeit bestehen, selbst zu entscheiden welche Bilder in der Ausstellung gezeigt werden. Allerdings werden dabei nicht alle Bilder einzeln ausgewählt, sondern es soll über eine grafische Oberfläche eingegrenzt werden können, welche Daten der Besucher für interessant hält und somit zum aktuellen Zeitpunkt relevant sind. Hierzu sind unterschiedliche Auswahlmöglichkeiten notwendig, wie z.B. Datum, Thema oder Fotograf bzw. Maler der Bilder. Außerdem sollten diese Möglichkeiten kombinierbar sein. Nachdem die Kriterien bestimmt sind, nach denen die Bilder ausgewählt werden, ist der Teil der Vorbereitung abgeschlossen und es beginnt der eigentliche Rundgang.

Hier beginnt ein interaktives Surfen durch die Bilderlandschaft. Das bedeutet, dass eine Galerie aufgebaut werden muss, in der die relevanten Bilder gezeigt werden und durch welche nun navigiert werden kann. Vor dem Hintergrund dem Anwender ein räumliches Gefühl zu vermitteln, sollten die Bilder in einem dreidimensionalen Raum dargestellt werden. Dabei soll die Navigation durch diesen Raum möglichst intuitiv und einfach sein, um eine „handbuchfreie“ Benutzbarkeit zu erzielen. Dazu soll eine direkte Umsetzung von Bewegungen auf die Steuerung innerhalb der Galerie stattfinden.

3.1.2 Digitale Postkarte

In diesem Szenario wird beschrieben, in welcher Form eine dreidimensionale Galerie verwendet werden kann, um Informationen für Menschen zugänglich zu machen.

⁴<http://www.nintendo.com/wii>

⁵<http://www.apple.com/iphone>

Viele Menschen verschicken aus ihrem Urlaub eine Postkarte an ihre Bekannten und Freunde. Allerdings sind diese Postkarten in den meisten Fällen vorgefertigte Bilder und nicht individuell gestaltet. Diese statischen Postkarten können durch eine persönliche Galerie abgelöst werden, welche über das Internet als Link verschickt wird. Dabei stellt der Sender eine Anzahl an Fotos zusammen, welche er dem Empfänger zur Verfügung stellen möchte und sendet diese der anderen Person über das Internet zu. Im nächsten Schritt werden diese Bilder in der Galerie des Empfängers dargestellt. Der Rezipient nimmt auf diese Weise interaktiv am Urlaub des Anderen teil. Für die Realisierung dieses Konzeptes bietet sich ein System an, welches Bilder für die aktuelle Komposition auswählt, indem es Daten aus ihnen ausliest.

Die Idee aus einem Bilderbestand verschiedene Zusammenstellungen zu generieren, um sie über das Internet weiteren Nutzern zur Verfügung zu stellen, wird ebenfalls von der Software Picasa⁶ sowie der Webanwendung Flickr⁷ aufgegriffen. Dabei lassen sich Online-Fotoalben generieren, in welche digitale Bilder transferiert werden können.

Neben dem Consumer-Bereich lässt sich ebenfalls der Gebrauch im Business-Bereich beschreiben. Hier soll von einem Hersteller eine Werbebotschaft an einen Kundenkreis verschickt werden. Dazu stellt der Werbende ein oder mehrere Galerien zusammen (siehe Abb. 3.1) und lässt sie dem Interessierten zukommen. Der Empfänger sollte dann ohne eine Einarbeitungszeit die Botschaften überblicken können und sich den Informationen widmen, die für ihn bedeutend sind. Bei den vorgestellten Produkten sollte es dem Kunden außerdem ermöglicht werden, weitere Daten abzurufen, welche näher über den Artikel informieren, um bei Bedarf direkt Kontakt zum Hersteller aufnehmen zu können. Dazu sollte die Applikation plattformunabhängig sein und eine externe Schnittstelle zur interaktiven Steuerung besitzen, damit eine möglichst einfache Anbindung an verschiedene Systeme gewährleistet wird.

⁶<http://picasa.google.com/>

⁷<http://www.flickr.com/>

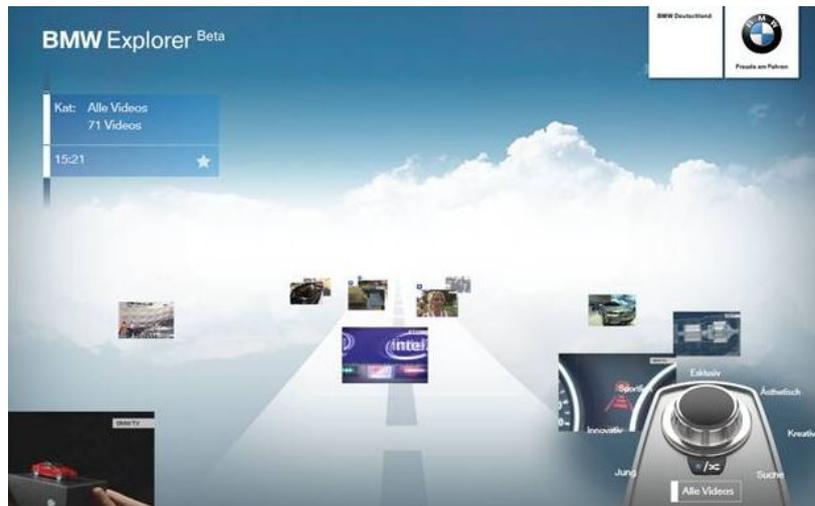


Abbildung 3.1: BMW Explorer Beta (BMW, 2008)

3.2 Anforderungsanalyse

In diesem Abschnitt werden einige essentielle Anforderungen erläutert, die ein System erfüllen muss, damit die beschriebenen Szenarien erfüllt werden können. Darauf folgend werden einige nichtfunktionale Anforderungen untersucht, um zu bestimmen welche Eigenschaften das System haben sollte. Damit diese Bedingungen festgelegt werden können, werden in diesem Abschnitt Anwendungsfälle betrachtet, die bei einer Interaktion mit dem System auftreten können. Daraus ergeben sich konkrete Interaktionsmöglichkeiten zur Steuerung sowie Anforderungen an die Bereitstellung der Informationen.

3.2.1 Funktionale Anforderungen

Die funktionalen Anforderungen beschreiben notwendige Funktionen, die das System offerieren muss. Diese befassen sich mit der Zusammenstellung bzw. mit der Filterung der darzustellenden Informationen und der bewegungs- und gestenbasierten Steuerung innerhalb der Informationsbestände.

Use Cases

Eine Übersicht der möglichen Anwendungsfälle beim Auswählen der relevanten Informationen bietet Abbildung 3.2.

Abbildung 3.3 beschreibt einen Use Case beim Navigieren durch die Datenbestände.

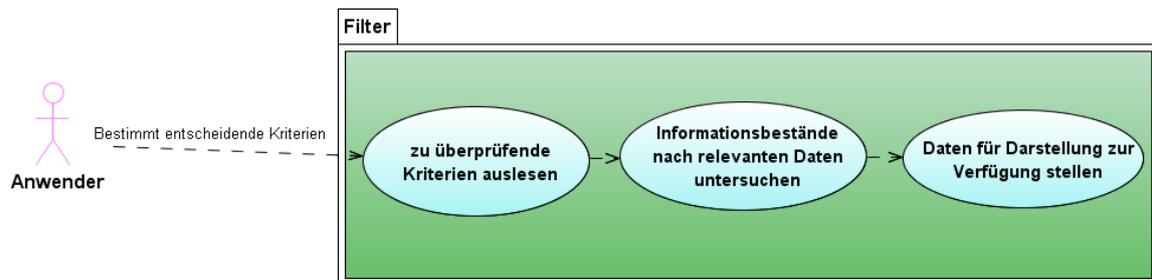


Abbildung 3.2: Anwendungsfall „Auswahl der relevanten Daten“

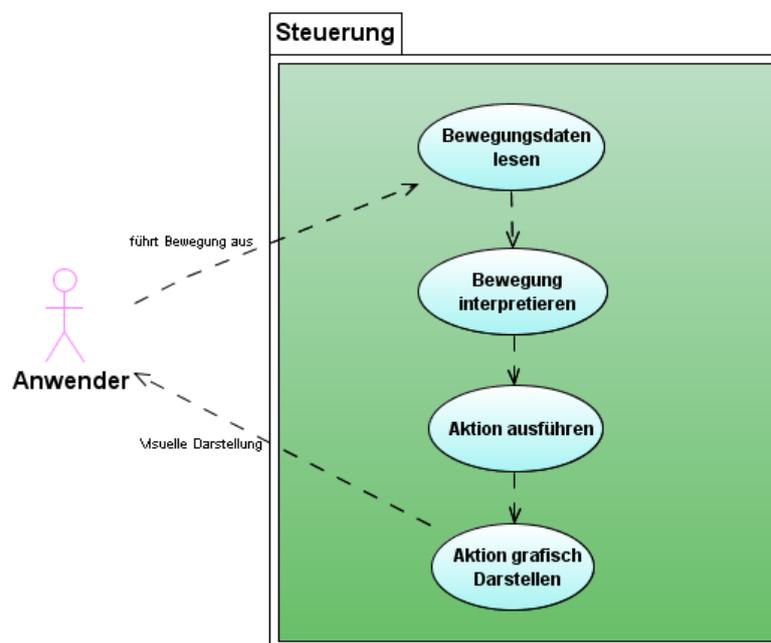


Abbildung 3.3: Anwendungsfall „Navigation durch Informationsbestände“

Daten auswählen

Dem Anwender die Möglichkeit einer schnellen und intuitiven Auswahl der zu verwendenden Daten zu ermöglichen, beinhaltet die Notwendigkeit, ihm ausschließlich analysier- und unterscheidbare Kriterien anzubieten. Deshalb muss die Applikation in der Lage sein, Metadaten oder weitere vorliegende Informationen auszulesen. Diese Daten können einerseits dazu verwendet werden, Auskünfte darüber zu geben, nach welchen Aspekten die Daten eingegrenzt werden können und andererseits zu entscheiden, ob sie dem Benutzer angezeigt werden.

Zusätzlich sollte es dem Anwender erlaubt sein, die Kriterien miteinander zu kombinieren, sodass nur Daten verwendet werden, die alle ausgewählten Eigenschaften erfüllen. Auf der anderen Seite sollte es genauso möglich sein, Kriterien, die in der Suche keine Rolle spielen, zu deaktivieren.

Wenn der Benutzer eine Auswahl von Aspekten getroffen hat, ist es naheliegend, dass er diese Anfrage an das System in einem gewissen Zeitraum erneut stellen wird. Dies könnte der Fall sein, wenn der Anwender die Daten ein weiteres Mal betrachtet oder zu überprüfen beabsichtigt, ob sich diese verändert haben und folglich nicht mehr relevant sind bzw. neue Daten die gewünschten Eigenschaften erfüllen. Deshalb sollten die gestellten Anfragen vom System gespeichert werden, damit sie bei einer erneuten Verwendung direkt zur Verfügung stehen. Damit neue Daten, welche die Kriterien erfüllen, automatisch angezeigt werden und nicht relevante Daten aus der Ergebnisliste entfernt werden, muss das System im Hintergrund eine neue Sortierung der Daten vornehmen, sobald die vordefinierte Suche gestartet wird.

Der Ansatz Suchanfragen zu speichern wird bereits in Spotlight⁸, der integrierten Suchfunktion in Apples Betriebssystem Mac OS X, aufgegriffen. Hier ist es möglich eine durchgeführte Anfrage zu speichern und mit einem Namen zu versehen. Die Ergebnisse dieser Suche werden dann in einem sogenannten „Intelligenten Ordner“ abgespeichert und stehen zur Verfügung, sobald der Ordner geöffnet wird. Dabei werden keine Daten von ihrem Originalspeicherort kopiert oder verschoben, sondern lediglich an dieser Stelle angezeigt, sodass kein Speicherplatz für die enthaltenen Daten belegt wird. Ein großer Vorteil dieser Suchfunktion ist, dass der Inhalt dieser Ordner sofort aktualisiert wird, wenn an einer Stelle im Dateisystem eine Datei gespeichert wird, die den verknüpften Suchkriterien entspricht (Pogue u. a., 2005). Der Einsatz solcher intelligenter Ordner im hier vorgestellten System ist aufgrund seiner eben vorgestellten Funktionalität sehr empfehlenswert.

Damit während der Einrichtungsphase eine intuitive Steuerung gewährleistet wird, sollte dem Benutzer eine grafische Oberfläche angeboten werden, mit der Absicht von dieser den gesamten Vorgang durchführen zu können. Des Weiteren ist es notwendig, dem Anwender zu ermöglichen, die Applikation in ihren Ursprungszustand zurück zu setzen, um eine neue

⁸<http://www.apple.com/macosx/features/300.html#spotlight>

Suche zu starten und vorherige Daten, die ab jetzt eventuell keine Rolle mehr spielen werden, wieder zu löschen.

Durch Informationen navigieren

Für einen Benutzer ist es sehr entscheidend, in welcher Weise ihm Informationen präsentiert werden. Dabei steht er bei großen Mengen an Informationen vor einer Flut, die schwer zu beherrschen ist. Häufig sucht der Anwender dabei nur wenige bestimmte Daten, die für ihn interessant sind. Damit er sich in dieser Menge zurechtfindet, sollte es ihm ermöglicht werden, innerhalb eines Raumes durch die Bestände zu navigieren und sich ausschließlich auf die für ihn wichtigen Objekte zu konzentrieren. In diesem Zusammenhang kann für einen gewissen Zeitraum die Notwendigkeit bestehen, sich nur einem einzelnen Objekt zuzuwenden. Deshalb ist es erforderlich, Elemente bewegen zu können, um sie bei Bedarf näher heranzuholen und somit die weiteren Daten in den Hintergrund zu rücken.

Für die Navigation ist eine Seamless Interaction notwendig, damit der Anwender sich vollständig auf die vorliegenden Datenmengen konzentrieren kann. Eine solche Interaktion lässt sich dadurch realisieren, dass dem Anwender intuitiv bewusst ist, welche Handlung seinerseits zu welcher Reaktion des Systems führt.

Elementare Interaktionen

Um die Anforderungen an eine intuitive Bedienung innerhalb der Galerie zu erfüllen, sind einige elementare Interaktionen notwendig, mit dem Ziel die Navigation innerhalb dieser zu realisieren.

Innerhalb einer räumlich dargestellten Informationslandschaft führt der Anwender Aktionen, zur Bewegung durch die Daten, aus. In diesem Kontext lässt sich die Steuerung innerhalb des Raumes am intuitivsten realisieren, indem man die Bewegungen des Benutzers auf äquivalente Aktionen im Raum abbildet. Hieraus ergeben sich Handlungen des Anwenders, die durch das System abgefangen, interpretiert und als Reaktion auf der grafischen Oberfläche dargestellt werden müssen.

Grundlegende Interaktionen, die sich hierbei mit dem System ergeben sind:

Navigation: Hierunter versteht man das Bewegen und Navigieren innerhalb des Raumes, in welchem die Daten präsentiert werden. Dabei sind die Richtungen, in die man sich bewegen kann z.B. vor, zurück, rechts und links. Dies soll erreicht werden, indem der Benutzer durch einfache Bewegungen angibt, in welche Richtung er gehen möchte.

Objekt bewegen: Der Anwender ist in der Lage, Objekte in der Vorder- bzw. Hintergrund zu rücken oder um auf der anderen Seite nach weiteren Informationen zu schauen. Dazu

muss der Benutzer zur Durchführung weiterer Aktionen einzelne Objekte auswählen können.

Objekt anvisieren: Der Benutzer kann Objekte anvisieren, um den Informationsträger heranzuholen und es in einer größeren Darstellung betrachten zu können. Außerdem muss der Anwender das Objekt wieder ablegen können, sodass er sich in der ursprünglichen Informationslandschaft wiederfindet.

Benutzbarkeit

Unter der Anforderung „Benutzbarkeit“ versteht man laut der Norm ISO/IEC 9126 (ISO/IEC, 1991) die Verständlichkeit, Erlernbarkeit, sowie die Bedienbarkeit eines Softwareproduktes. Da die hier vorgestellte Applikation eine Seamless Interaction zwischen dem Menschen und dem System aufweisen soll, spielt diese Bedingung eine große Rolle. Aus diesem Grund ist es von Bedeutung, dass die Anwendung ohne eine Einarbeitungszeit verwendet werden kann und der Benutzer selbsterklärend den Weg durch den Vorbereitungsmodus und durch die Galerie findet. In diesem Kontext besteht die Notwendigkeit, dem Anwender bei jedem Start einen identischen Ablauf zu gewährleisten und möglichen Fehleingaben entgegen zu wirken, indem Eingabevorschläge gemacht werden. Für die Steuerung des Systems ist es entscheidend, dass die benötigten Interaktionsgeräte nach Möglichkeit keine Einrichtungsphase benötigen, sondern direkt über eine externe Schnittstelle mit der Applikation kommunizieren können.

3.2.2 Nichtfunktionale Anforderungen

Neben den funktionalen Anforderungen an das System, welche die Aufgaben des Systems beschreiben, werden in diesem Abschnitt die nichtfunktionalen Anforderungen, d.h. die Eigenschaften des Systems, erläutert.

Zuverlässigkeit

Eine Anforderung an qualitativ hochwertige Software ist die Zuverlässigkeit eines Systems (ISO/IEC, 1991). Dabei zählt u.a. die Fehlertoleranz und Wiederherstellbarkeit zu den Kriterien dieses Merkmals. Die Zuverlässigkeit einer Applikation zeichnet sich dadurch aus, dass das Leistungsniveau unter bestimmten Bedingungen einen bestimmten Zeitraum aufrecht erhalten werden kann. Dies beinhaltet auch bei Fehleingaben des Benutzers den Programmablauf zu sichern, indem aussagekräftige Fehlermeldungen ausgegeben werden oder die Applikation versucht, aus den fehlerhaften Angaben, verwendbare Daten zu generieren.

Unter der Wiederherstellbarkeit versteht man die Zeit und den Aufwand der betrieben werden muss, um das Leistungsniveau wiederherzustellen, wenn das System im Programmfluss unterbrochen wurde. In diesem Fall dürfen keine Informationen verloren gehen, sondern müssen für eine weitere Verwendung gesichert werden.

Sicherheitsanforderungen

Bei persönlichen Informationsfeldern spielt die Vertraulichkeit der Informationen eine entscheidende Rolle. Um die vertraulichen Daten der Anwender zu schützen, sollte das System eine Authentifizierung verlangen. Falls es sich bei den Daten allerdings um öffentlich zugängliche Informationen handelt, wie es z.B. bei einer Werbebotschaft der Fall wäre, sollte die Applikation die Daten direkt zugreifbar machen. Hierzu ist es erforderlich, Zusatzinformationen auszulesen, um anhand dieser zu entscheiden, für wen die vorliegenden Informationen zugänglich sind.

Portierbarkeit

Falls -wie in einem der Szenarien beschrieben- eine Galerie über das Internet verschickt wird, kann der Sender nicht davon ausgehen, dass der Empfänger das gleiche Betriebssystem verwendet. Deshalb ist es bedeutsam, dass die Anwendung betriebssystemunabhängig und auf mehreren Plattformen ausführbar ist.

Performance

Bei einer interaktiven Steuerung ist für den Benutzer die Lieferung einer Reaktion des Systems in Echtzeit essentiell. Damit für den Anwender das Gefühl einer Abbildung seiner Bewegungen auf die Reaktionen des Systems entsteht, ist es entscheidend, dass keine erkennbaren Verzögerungen entstehen. Aus diesem Grund sollte die Applikation direkt reagieren, sobald eindeutig feststeht, welche Bewegung der Anwender ausführt, auch wenn sie noch nicht vollständig beendet ist.

Des Weiteren sollte die Analyse der Daten keine Verzögerungen zwischen der Einrichtungsphase und dem Starten der Informationsgalerie hervorrufen, damit dem Benutzer ein fließender Übergang zwischen diesen Schritten vermittelt wird.

3.2.3 Grundlegende Funktionalität des Systems

Aufgrund der hier vorgestellten Szenarien und Anforderungen soll eine Applikation entstehen, die Daten aus einem persönlichen Informationsfeld filtern kann, mit dem Ziel sie grafisch darzustellen. Dabei soll für die Navigation durch die Datenbestände eine externe Schnittstelle zur Verfügung stehen, die dem Benutzer eine Steuerung durch die vorliegenden Daten offeriert.

Da sich das Informationsfeld eines Menschen auf sehr viele verschiedene Gebiete erstreckt, wird der Fokus des Filters auf Fotobestände des Benutzers gesetzt. In diesem Kontext soll es möglich sein, aus verschiedenen Eigenschaften zu wählen, nach denen die Bilder geordnet werden. Sinnvolle Eingrenzungskriterien sind u.a. der Zeitraum in dem sie geschossen wurden, der Fotograf oder die Kategorie des Bildes. Damit der Anwender ein zufriedenstellendes Ergebnis erhält, also relevante Daten möglichst präzise eingrenzen kann, sollten auch Kombinationen zwischen den Kriterien möglich sein. Diese Suchanfragen an das System sollen gespeichert werden, um sie dem Benutzer für eine erneute Anfrage zur Verfügung zu stellen, ohne dass er die entscheidenden Kriterien erneut zusammenstellen muss.

Vor dem Hintergrund die Bedienung des Filters intuitiv zu gestalten, müssen die Merkmale nach denen die Daten eingegrenzt werden können bereits beim Start der Applikation ausgelesen werden, um dann nur noch analysierbare Attribute anzuzeigen zwischen denen der Anwender wählen kann. Zum Auslesen der Daten müssen die Fotos mit entsprechenden Daten versehen werden. Daraus resultierend müssen sie als Metadaten an das Bild „angehängt“ sein.

Für die Darstellung der Bilder soll eine dreidimensionale Galerie aufgebaut werden, in welcher sich der Anwender zur Betrachtung der einzelnen Bilder bewegen kann. Der Vorteil bei dieser Form der Präsentation liegt darin, dass für den Anwender ein räumliches Gefühl entsteht und sich über die dritte Koordinate weitere Interaktionsmöglichkeiten zur Bewegung innerhalb des Raumes ergeben. Ein weiterer positiver Effekt einer solchen Darstellung liegt in der kognitiven Entlastung des Anwenders, die bereits bei der Visualisierung eines Cone Trees beobachtet werden konnte (2.5). Damit bei großen Datenmengen keine Unübersichtlichkeit entsteht, sollten mehrere Galerien aufgebaut werden, sobald eine gewisse Anzahl an Bildern erreicht ist. In diesem Zusammenhang müssen Übergänge zwischen den Galerien geschaffen werden, sodass dem Benutzer ein einfacher Zugriff auf die Daten gewährleistet wird. Des Weiteren soll es die Anwendung ermöglichen, ein Bild näher zu betrachten oder weiter in den Hintergrund zu schieben.

Für die Realisierung der externen Schnittstelle, welche zur Navigation innerhalb der Galerie dient, soll eine bewegungsbasierte Steuerung implementiert werden. Damit eine handbuchfreie Benutzung des Systems gewährleistet wird, stützt sich die Steuerung auf mentale Modelle (2.2). Da für den Menschen Gewichtsverlagerungen intuitiv sind (4), bietet sich es bei dieser Steuerung an, eine Bedienung in dieser Form zu wählen. Dabei soll der Benutzer sich durch eine Verlagerung seines Gewichts innerhalb der Galerie bewegen. Die Problema-

tik der X-Achse (4.3), die sich innerhalb dreidimensionaler Räume ergibt, soll hierbei durch Drehungen auf dieser Koordinate gelöst werden.

Für weitere Interaktionen mit dem System sollte ein zusätzliches Eingabegerät eingesetzt werden. Zur Übernahme der grundlegenden Aufgaben einer klassischen Maus, sollte das Eingabegerät in der Lage sein, verschiedene Dinge anzuvisieren, mit dem Ziel sie zu bewegen oder von einer Galerie in eine weitere zu wechseln. Diese Eingaben sollten ebenfalls bewegungsbasiert ablaufen, um die bisherige Interaktionsform beizubehalten.

Wie bereits in den Use Cases beschrieben (3.2.1), wird die Benutzung dieser Applikation dabei in zwei Phasen aufgeteilt.

Phase 1: Im ersten Abschnitt gibt der Benutzer über eine grafische Oberfläche bekannt, nach welchen Kriterien er den Informationsbestand filtern möchte. Dieser Teil ist als Einrichtungsphase zu verstehen.

Phase 2: Nachdem der erste Teil abgeschlossen ist, beginnt die Phase der Navigation innerhalb des dreidimensionalen Raumes. Hier betrachtet der Anwender die Daten und navigiert mithilfe von Gewichtsverlagerungen durch die Galerie.

3.3 Objektdarstellung und Navigation in dreidimensionalen Räumen

Hinsichtlich der Entstehung einer realistischen und einfach verständlichen dreidimensionalen Darstellung, werden hier einige Problemfelder diskutiert, die sich beim Aufbau und der Navigation durch dreidimensionale Landschaften ergeben.

3.3.1 Orientierung in dreidimensionalen Räumen

Bei Menschen, die zum ersten Mal mit dreidimensionalen Anwendungen konfrontiert wurden, konnten bisher allgemeine Orientierungsprobleme beobachtet werden (Fitzmaurice u. a., 2008). Eines dieser Probleme wird als die „3D Amnesie“ bezeichnet. Diese tritt auf, wenn z.B. eine dreidimensionale Ansicht eines Würfels ohne eine Animation in eine Seitenansicht wechselt. In diesem Fall entsteht für den Menschen das Gefühl, der Körper habe sich in ein einfaches Rechteck verwandelt. Außerdem kann der Benutzer seine Orientierung verlieren, wenn es keine Anhaltspunkte darüber gibt, wo er sich innerhalb des Raumes befindet oder welchen Teil eines Objektes er betrachtet. Aufgrund dessen sollten Ansichtswechsel durch Sequenzen begleitet werden und Zusatzinformationen über den Zustand des betrachteten Objektes dargestellt werden.

3.3.2 Interaktion mit Objekten in dreidimensionalen Räumen

Wie in [Bowman u. a. \(1999\)](#) erläutert wird, umfasst der Umgang mit dreidimensionalen Objekten drei Phasen:

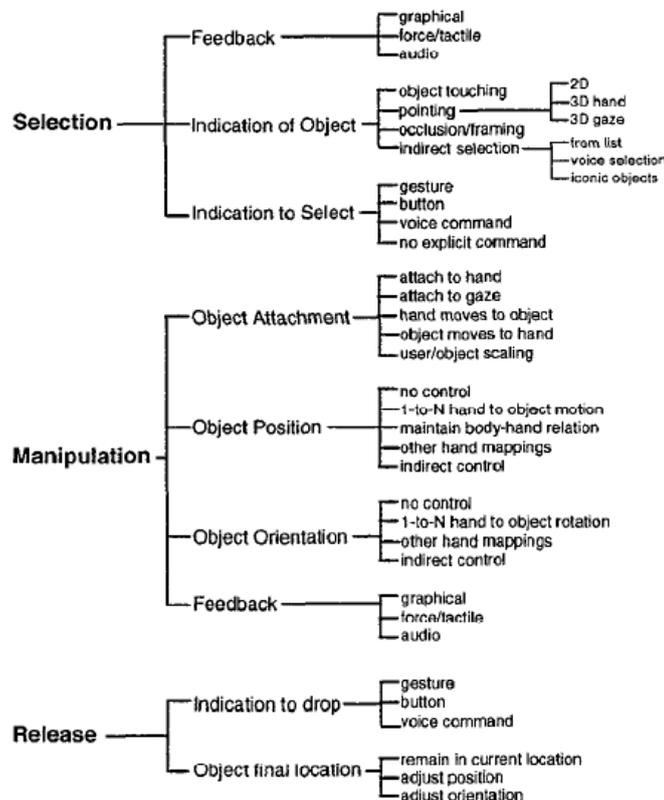


Abbildung 3.4: Taxonomie der Selektion/Bedienung von 3D-Objekten ([Bowman u. a., 1999](#))

1. Selektion: In dieser Phase gibt der Anwender an, mit welchem Objekt er beabsichtigt zu interagieren, indem er es selektiert.
2. Positionierung und Bedienung: Hier wird das selektierte Objekt in die relative Position zum Betrachter gesetzt und kann nun bearbeitet werden.
3. Freigabe: An dieser Stelle gibt der Anwender das Element frei, damit es wieder innerhalb des Raumes justiert wird. Dabei kann es sich um die aktuelle Position oder einen neuen Standpunkt handeln.

In [Teather und Stuerzlinger \(2007\)](#) werden Techniken zur Bewegung und Positionierung von Objekten innerhalb von dreidimensionalen Räumen vorgestellt. Es handelt sich um Richtlinien für Entwickler, die dabei helfen sollen, eine Navigation innerhalb solcher Räume für

Anwender intuitiv zu gestalten. Einige dieser darin enthaltenen Hinweise sollen im Folgenden etwas genauer betrachtet werden.

3.3.3 Bewegungstechniken

Relative Positionierung: Man sollte die Möglichkeit bieten, Objekte relativ voneinander zu positionieren, indem man sie miteinander in Verbindung setzt. Diese Funktion ist notwendig, falls der Anwender den Standpunkt eines Elementes relativ zu seiner Position verändern will. Ein Ansatz zur Realisierung solcher Interaktionen mit dem Objekt ist es, Vorbedingungen zu verwenden, die für eine Bewegung erfüllt sein müssen. Hierzu bietet es sich an, physikalische Regeln, wie z.B. die Schwerkraft, zu verwenden.

Sichtbarkeitsproblem: Anwender versuchen typischerweise nur Objekte zu bewegen, die sichtbar sind. Bei verdeckten Elementen versuchen sie ihren Standpunkt zu ändern, sodass das gewünschte Objekt sichtbar wird. Um zu entscheiden, welches Objekt dem Benutzer gezeigt wird, demnach ebenso innerhalb einer zweidimensionalen Anzeigefläche sichtbar wäre, gibt es u.a. den Raytrace Algorithmus. Dieser ist auf Appel, Goldstein und Nagel zurückzuführen und wird u.a. in [Jensen und Christensen \(2007\)](#) beschrieben. Er basiert auf Strahlen, die aus Sicht des Beobachters ausgesendet werden. Dabei wird jeder Strahl gegen alle geometrischen Formen getestet und jeweils die Entfernung gemessen. Das Objekt mit der geringsten Entfernung wird folglich das angezeigte Element.

3.3.4 Perspektiven

Bei der Navigation durch dreidimensionale Räume gibt es zwei verschiedene Kameraperspektiven, die besonders aus dem Spielbereich bekannt sind. Es handelt sich um die „First-Person“ und „Third-Person“ Perspektive, zwischen denen es elementare Unterschiede gibt ([Richard Rouse, 1999](#)). Bei einer First-Person-View erfolgt die Darstellung aus den Augen einer Figur die sich innerhalb der Landschaft befindet. Das bedeutet, solange sich der Anwender in einer gültigen Stelle innerhalb des Raumes befindet, liegt die Kamera relativ zum Betrachter gesehen, in der gleichen Position. Dies ist in der Third-Person-View nicht der Fall. Hier schaut man einer Figur, welche sich innerhalb des Raumes befindet, über die Schulter. Diese Figur ist also zu jeder Zeit sichtbar.

In diesem Zusammenhang hat die Third-Person Perspektive einen Nachteil gegenüber der First-Person Perspektive, wenn der dargestellte Raum mit Wänden eingegrenzt ist. In diesem Fall muss entschieden werden wie die Kamera reagiert, falls die Figur mit dem Rücken zur Wand steht. Dieses Problem wird mit der First-Person-View vollständig umgangen.

3.3.5 Mentale Modelle in dreidimensionalen Räumen

Damit die Interaktion und Navigation innerhalb dreidimensionaler Räume für den Anwender intuitiv gestaltet wird, sollte man sich an den mentalen Modellen des Benutzers orientieren. Um diese im System abzubilden ist es sinnvoll, ein realistisches Gefühl im Umgang mit den Objekten zu erzeugen. In [Agarawala und Balakrishnan \(2006\)](#) werden Ziele definiert, die hilfreich sind um eine Seamless Interaction zwischen dem Anwender und dem System zu ermöglichen:

- Wenn Objekte innerhalb eines Raumes bewegt werden, sollten sie sich analog zu Elementen in der realen Welt verhalten. Dadurch kann der Anwender sein Wissen über das Verhalten eines Objektes auf das System anwenden.
- In einigen Situationen kann es sich als nützlich erweisen, bestimmte physikalische Eigenschaften an Objekten abzuschalten. Dies kann z.B. zur Verhinderung von Kollisionen zwischen Objekten oder zur Vereinfachung der Steuerung dienen.
- Visuelle Übergänge sollten geschmeidig und gleichmäßig verlaufen, um den Anwender nicht zu überraschen oder zu verwirren.

3.4 Fazit

In diesem Kapitel wurden zwei Szenarien vorgestellt, mit denen die Anwendungsmöglichkeiten für die hier vorgestellte Applikation beschrieben wurden. Hieraus wurde erkennbar, dass es bei der Navigation innerhalb großer Informationsbestände eine intuitive Steuerung einen großen Vorteil bringt, da der Anwender sich vollständig den vorliegenden Daten zuwenden kann und die Interaktion mit dem Computer als natürlich empfunden wird.

Aus den Szenarien wurden daraufhin Bedingungen abgeleitet, die erfüllt werden müssen, um die Anwendungsfälle zu realisieren. Dabei ist neben der handbuchfreien Steuerung des Systems ein weiterer entscheidender Punkt, dass das System in der Lage sein muss, Daten zu filtern, um dem Benutzer ausschließlich relevante Informationen zu offerieren.

Vor dem Hintergrund Objekte in dreidimensionalen Räumen darzustellen sowie eine intuitive Navigation innerhalb dieser zu ermöglichen, wurden einige Problemfelder aufgezeigt, die im Design der Galerien zu beachten sind. Für die Steuerung in den Räumen ist es essentiell, dass Bewegungen des Anwenders interpretiert werden und von ihm erwartete Reaktionen des Systems auslösen. Das Thema Bewegungen wird im folgenden Abschnitt genauer betrachtet, mit dem Ziel diese Anforderung zu erfüllen.

4 Bewegungen

4.1 Begriffserklärung

In der Physik wird eine Bewegung als eine Änderung der Lage eines Körpers zu einem Bezugssystem beschrieben (Meyers, 2008). Da der Mensch sich zu keiner Zeit in einem Zustand vollständiger Ruhe befindet, ändert sich die Lage des menschlichen Körpers also kontinuierlich zu seinem Bezugssystem.

Menschen verwenden Bewegungen u.a. dazu sich fortzubewegen, zu schützen oder um etwas zu greifen. Dabei umfassen die Motorik, die Taxis und die Lokomotorik, welche alle aus der Wissenschaft der Kinesik stammen, das menschliche Bewegungsverhalten. Die Motorik beinhaltet u.a. die Körperhaltung, die Taxis entspricht der Körperausrichtung, folglich die Blick-, Kopf- und Rumpforientierung und die Lokomotorik ist während der Fortbewegung für die Distanzregelung zuständig.

Bei der Körperhaltung eines Menschen gehört die Gewichtsverlagerung zu den wesentlichen Faktoren, da jede menschliche Fortbewegungsaktivität die Notwendigkeit besitzt, das Gewicht zu verlagern (siehe Abbildung 4.1). Selbst das Schließen der Hand zu einer Faust erzeugt dabei anpassende Bewegungen im Körper und somit Gewichtsverlagerungen. Neben diesen unbewussten Veränderungen, verlagert ein Mensch sein Gewicht in bestimmten Situationen wissentlich, um Dinge auf diese Weise zu steuern. Beispiele hierfür sind Fahrradfahren und Gleitschirm fliegen, wo durch Gewichtsverlagerungen die Fahrt- bzw. Flugrichtung geändert werden kann.

4.2 Technologien

In diesem Abschnitt werden bewegungsbasierte Technologien vorgestellt, welche in den letzten Jahren entstanden sind. Hier werden Bewegungen, die zur Interaktion mit einem Computersystem dienen, interpretiert und in eine Reaktion auf das System abgebildet.



Abbildung 4.1: Gewichtsverlagerung bei einem Laufschrift ([Fitness, 2008](#))

4.2.1 EyeToy

Das EyeToy⁹ ist ein Zusatzgerät für die Playstation 2¹⁰ von Sony. Bei dieser Konzeption wird der Spieler mit einer Kamera beobachtet. Dabei überträgt die Kamera das eingefangene Bild der Person auf den Bildschirm, woraufhin sich der Anwender interaktiv innerhalb des Spiels sieht und dieses durch Bewegungen steuert.

4.2.2 Segway

Der Segway¹¹ Personal Transporter wurde im Jahr 2001 von der Segway Inc. entwickelt (siehe Abbildung 4.2). Dieses Fahrzeug, welches mit einem Elektromotor ausgestattet ist, wurde für eine sichere und einfache Fortbewegung konzipiert. Hierbei handelt es sich um ein Gerät, bei dem der Fahrer sich auf eine Plattform stellt, welche mit zwei Rädern verbunden ist und sich an einer Lenkstange festhält. Mit dem Ziel eine intuitive Steuerung zu gestalten, wird der Segway ausschließlich durch Gewichtsverlagerungen gesteuert. Das heißt, es werden keine weiteren Bedienelemente, wie Drehgriffe oder Bremshebel, verwendet.

Der Erfinder, Dean Kamen, hat sich für die Steuerung des Segways sehr an der Arbeitsweise des menschlichen Gehirns orientiert. Dieses reagiert, sobald sich ein Mensch so weit

⁹<http://www.eyetoy.com/shared/locale.asp?returnURL=/index.asp>

¹⁰<http://de.playstation.com/>

¹¹<http://www.segway.com/>

nach vorne lehnt, dass er das Gleichgewicht verliert, indem es die Person dazu veranlasst einen Schritt nach vorne zu gehen, um nicht zu fallen. Auf diese Weise funktioniert auch die Steuerung eines Segways. Wenn der Mensch sich auf dem Gerät nach vorne lehnt, fährt es vorwärts und bei einer Gewichtsverlagerung nach hinten, bremst er ab bzw. fährt rückwärts, sofern diese Verlagerung im Stillstand des Fahrzeugs auftritt. Um Kurven zu fahren, muss die Lenkstange in die entsprechende Richtung bewegt werden.

Zur Interpretation der Bewegungen des Anwenders werden Gyroskopsensoren eingesetzt. Basierend auf der Corioliskraft messen solche Sensoren die Bewegungsrichtung des menschlichen Körpers. Hierbei handelt es sich um die Kraft, die auf einen Körper wirkt, der sich in einem rotierenden Bezugssystem bewegt.

Der Erfolg, der durch solche intuitiven Steuerungen erreicht wird, lässt sich durch die verschiedenen Einsatzgebiete, in denen dieses Gerät eingesetzt wird beschreiben. Der Segway wird u.a. in Großstädten als Touristenfahrzeug eingesetzt, vor dem Hintergrund den Personen Sehenswürdigkeiten zu präsentieren und findet bereits in der Sicherheitsbranche und bei der Polizei Anklang.



Abbildung 4.2: Aufbau eines Segways (segway, 2008)

4.2.3 Neural Impulse Actuator

Der Neural Impulse Actuator (NIA) von OCZ (Abbildung 4.3) ermöglicht in Verbindung mit der klassischen Maus eine neue Art der Computersteuerung. Hierzu detektieren Sensoren über

ein Stirnband die Impulse der Gesichtsmuskeln sowie die Spannungen aus der elektrischen Aktivität des Gehirns ([Heise-Online, 2008](#)). Dieses Verfahren wird in der Medizin als Elektroenzephalografie (EEG) bezeichnet. Außerdem ist das Gerät in der Lage, die Richtung, in die der Anwender schaut, zu bestimmen. Neben dem Stirnband besteht das Gerät aus einer Box mit einem USB-Anschluss sowie einer Konfigurationssoftware für Windows. Mithilfe dieser Software lassen sich Profile erstellen, in denen Aktionen des NIA definiert werden können, um somit verschiedene Programme zu steuern.



Abbildung 4.3: Neural Impulse Actuator ([OCZ-Technology, 2008](#))

4.2.4 Nintendo Wii

Die Spielkonsole Nintendo Wii¹², die seit Ende 2006 auf dem Markt angeboten wird, arbeitet fast ausschließlich mit interaktiven Steuerungen in ihren Spielen. Dabei wird dem Anwender ermöglicht, durch Bewegungen und Gesten mit den erhältlichen Eingabegeräten, das Spiel zu steuern. Zwei elementare Geräte zur Steuerung dieser Konsole werden im Folgenden genauer erklärt.

Wii Remote

Die Wii Remote (Wiimote) ist das Haupt-Eingabegerät für die Nintendo Wii. Es handelt sich um ein kabelloses Gerät, das per Bluetooth mit der Konsole verbunden wird. Dabei wird mit einer Sensorleiste, die sich am TV befindet sowie einer Infrarotkamera im Gerät, die relative Position dieses Controllers ermittelt. Neben den Eingabeknöpfen wird hier mit einem 3-Achsen Beschleunigungssensor gearbeitet. Anhand diesem werden Bewegungen und Drehungen des Controllers erfasst und direkt in die Spielsteuerung abgebildet (Abbildung

¹²<http://wii.com/>

4.4). Des Weiteren enthält dieses Gerät einen 16 kByte großen EEPROM Memory, in dem u.a. Kalibrierungsdaten des Anwenders gespeichert werden können.

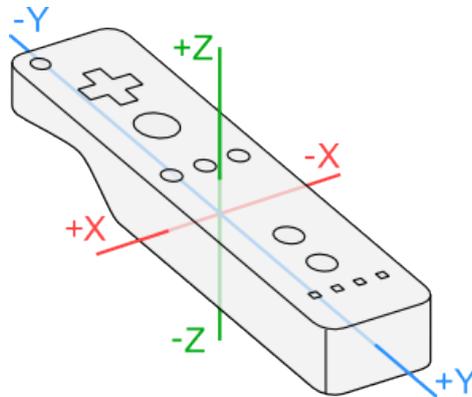


Abbildung 4.4: Wii Remote Achsen (wiibrew.org, 2008b)

Wii Balance Board

Das Wii Balance Board (Abbildung 4.5) ist ein weiteres Eingabegerät für die Nintendo Wii. Hierbei handelt es sich ebenfalls um ein Gerät, das per Bluetooth mit der Konsole verbunden wird. Es arbeitet mit vier Drucksensoren zur Ermittlung der Gewichtsverlagerung des Anwenders (wiibrew.org, 2008a) und verwendet diese für die Steuerung eines Spiels. Diesbezüglich stellt der Benutzer sich auf das Board und kann somit durch seine Körperneigung Einfluss auf das Spiel nehmen.

Bei den verwendeten Sensoren, mittels derer die Neigung des Körpers bestimmt wird, handelt es sich um 16-Bit Drucksensoren, die im linken und rechten Teil sowie oben und unten im Board angebracht sind.

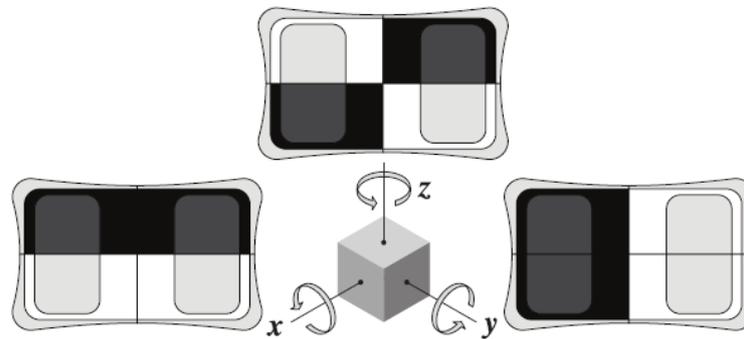


Abbildung 4.5: Gewichtsverlagerung auf die schwarzen Quadrate verursacht eine Rotation der dargestellten Achsen (de Haan u. a., 2008)

4.3 Interpretation von Bewegungen

Bei einer intuitiven Steuerung auf Basis von Bewegungen, ist es erforderlich, Aktionen des Anwenders so zu interpretieren wie er es erwartet. Bei der Navigation durch einen dreidimensionalen Raum gibt es drei Achsen, über welche man sich bewegen kann (Abbildung 4.6). Dabei bietet es sich an, die Z-Achse zu verwenden, mit der Absicht in den Raum hinein und über die Y-Achse nach oben und unten zu gehen. Allerdings steht man bei der X-Achse vor der Frage, ob diese dazu verwendet wird, um auf ihr zu gehen oder sie zur Rotation innerhalb des Raumes genutzt wird. Für eine Rotation spricht, dass das räumliche Gefühl für den Anwender steigt, wenn es ihm ermöglicht wird sich innerhalb eines Raumes umzuschauen.

Ein weiteres Problem, welches sich bei einer Steuerung innerhalb eines dreidimensionalen Raumes anhand von Gewichtsverlagerungen ergibt, sind die Interaktionsmöglichkeiten. Der Mensch kann über einfache Verlagerungen seines Gewichts nur zwei Achsen steuern. Zur Bewegung auf der dritten Achse, benötigt man also eine weitere Bewegung als Interaktionsmöglichkeit, welche ebenfalls interpretiert werden muss.

4.3.1 Beginn und Ende einer Bewegung

Bewegungen zu interpretieren und demzufolge entsprechend zu reagieren, verlangt die Definition von Beginn und Ende einer Bewegung. Zunächst muss zur Festlegung des Beginns einer Bewegung entschieden werden, welche Situation als Ruheposition bezeichnet werden soll. Auch wenn ein menschlicher Körper nie einen Zustand völliger Ruhe erreicht (4.1),

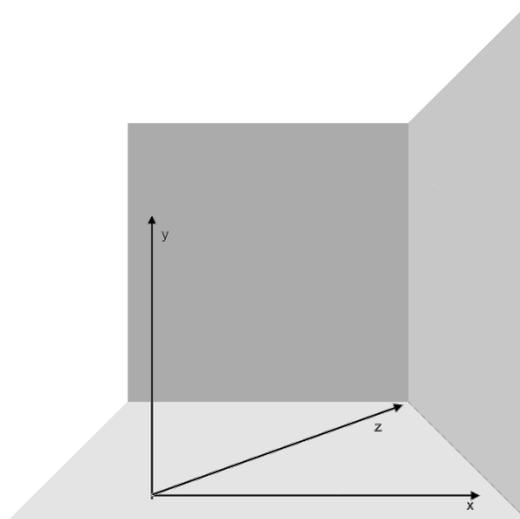


Abbildung 4.6: Koordinatensystem im dreidimensionalen Raum

muss es für das System eine Situation geben, welche die Ausgangslage darstellt und keine Reaktion erfordert.

Sobald der Anwender sich aus dieser Position herausbewegt, ergibt sich eine neue Bewegung. Dies könnte z.B. eine Gewichtsverlagerung um eine festgelegte Prozentzahl oder eine Handbewegung sein. Das Ende einer Bewegung muss ebenfalls vom System erkannt werden. Hierbei könnte es sich um einen direkten Übergang in eine weitere Bewegung oder zurück in die Ruheposition handeln.

Bei einer auf Bewegungen basierenden Computerinteraktion gibt es ein Problemfeld das sich „Midas Touch Problem“ (Jacob, 1990) nennt. Der Name dieses Problems ist auf eine Geschichte über einen König namens Midas zurückzuführen. Sie beschreibt den Wunsch dieses Königs, der besagt, dass alles was er anfasst, zu Gold wird. Als ihm dieser Wunsch erfüllt wird, droht er an Durst und Hunger zu sterben, weil nun auch Essen und Trinken zu Gold werden. Diese Situation lässt sich auch auf eine bewegungsbasierte Steuerung abbilden. Wenn ein Mensch eine Bewegung macht, um ein System zu bedienen, reagiert dieses mit einer entsprechenden Reaktion. Allerdings könnte es sich auch um eine Bewegung handeln, die keine Reaktion des Systems hervorrufen sollte. Hierin liegt die Problematik, welche das „Midas Touch Problem“ betrachtet: Ein Mensch ist nicht daran gewöhnt seine Bewegungen ausschließlich zur Steuerung eines Systems einzusetzen. Er erwartet sich bewegen zu können, ohne dass die Anwendung darauf reagiert. Dabei kann der Computer nicht unterscheiden, ob es sich dabei um eine absichtliche Aktion des Benutzers zur Steuerung handelt, oder ob die Bewegung in einem anderen Kontext stattgefunden hat. Auf diese Weise könnte der Anwender in eine analoge Schwierigkeit gelangen, in der auch König Midas steckte, wenn nämlich jede neue Bewegung immer wieder eine weitere Reaktion des Systems her-

vorrufen, ohne dass sie vom Benutzer gewollt ist.

Zur Lösung dieses Problems bietet sich an, für die Steuerung einen Zustandsraum zu schaffen. Ein Ansatz hierbei wäre, aus einem Zustand nur Übergänge in bestimmte Zustände zu ermöglichen. Auf diese Weise lässt sich ein unkontrolliertes Umschalten, zwischen allen möglichen Zuständen, die das System annehmen kann, verhindern. Eine weitere Herangehensweise wäre, einen Übergang von einem Zustand in jeden weiteren zu ermöglichen, wobei dieser allerdings vom Anwender anzukündigen ist. Das bedeutet, dass der Benutzer durch eine eindeutige, von allen anderen Bewegungen unterscheidbare Aktion, oder über ein externes Gerät ankündigt, dass eine Aktion folgt, auf die eine Reaktion erwartet wird. Eine dritte Variante wäre, die beiden eben genannten Ansätze zu verknüpfen. Hierbei könnte man dem Anwender den aktuellen Zustand und die möglichen Zustandsübergänge anzeigen, zwischen denen er dann per Knopfdruck oder einer eindeutig identifizierbaren Bewegung wählt. Bei diesem Lösungsansatz, bei dem die automatische Erkennung von Bewegungen eingegrenzt wird, lassen sich ungewollt bezweckte Reaktionen des Systems durch Aktionen des Benutzers zu einem großen Teil ausschließen. Allerdings würden sich hierbei Behinderungen im Interaktionsablauf ergeben, da der Benutzer in seinem natürlichen Bewegungsablauf gestört wird. Aus diesem Grund gibt es für diese Problemstellung keine allgemeine Lösung, sondern muss für verschiedene Systeme individuell angepasst werden, um eine möglichst nahtlose Interaktion zu erzielen.

4.4 Fazit

Im Zusammenhang mit der Realisierung einer Seamless Interaction zwischen Mensch und Computer, liegt ein interessanter Ansatz darin, die Bewegungen eines Anwenders in Reaktionen des Systems abzubilden, mit dem Ziel dadurch verschiedene Anwendungen zu steuern.

Bei dieser neuartigen Interaktionsform mit Computern gibt es bisher noch keine Konventionen die eine einfache Bedienung garantieren. Deshalb muss untersucht werden, welche Bewegungsabläufe sich anbieten, damit dem Benutzer eine intuitive Steuerung seines Gerätes gewährleistet wird.

Infolgedessen soll im Rahmen dieser Arbeit eine bewegungsbasierte Steuerung implementiert werden, die es nun gilt zu designen und zu realisieren.

5 Design

Im iFlat an der Hochschule für Angewandte Wissenschaften Hamburg (HAW) werden Applikationen erstellt, die dazu dienen, ein Intelligent Home aufzubauen. Dabei entstehen verschiedene Anwendungen, die dem Benutzer u.a. seinen Alltag erleichtern sollen. In diesem Bereich findet auch die hier vorgestellte Anwendung ihren Platz, welche in diesem Kapitel genauer präsentiert wird.

Basierend auf der Analyse haben sich Anforderungen und Erwartungen an ein System ergeben, die in diesem Kapitel in ein Design eingebettet werden, mit dem Ziel eine Anwendung mit einer bewegungsbasierten Steuerung zu realisieren und diese intuitiv nutzbar zu machen. Dabei werden die grundlegenden Funktionen, die das System erfüllen sollte herangezogen (3.2.3), um die Architektur der Applikation zu diskutieren (5.2). Zudem werden die einzelnen Aufgaben und die Komponenten des Systems untersucht und beschrieben. Außerdem werden Entwurfsmuster vorgestellt, deren sinnvoller Einsatz innerhalb des Designs erläutert wird (5.3).

5.1 Überblick des Systems

Das System verfügt über drei elementare Komponenten, welche für die hier vorgestellte Applikation erforderlich sind. Es handelt sich dabei um die grafische Oberfläche, einer externen Steuerung und einem Repository. Auf der Abbildung 5.1 erkennt man das Zusammenspiel der einzelnen Komponenten. Dabei liefert das Repository die anzuzeigenden Daten an die Darstellung. Über die externe Steuerung kann der Anwender daraufhin durch die Informationen, die ihm bereitgestellt werden, navigieren. Das Repository ist somit dafür zuständig, die vorliegenden Daten zu verwalten und muss entscheiden, welche dem Benutzer präsentiert werden. Zusätzlich enthält es die gespeicherten Suchanfragen des Anwenders.

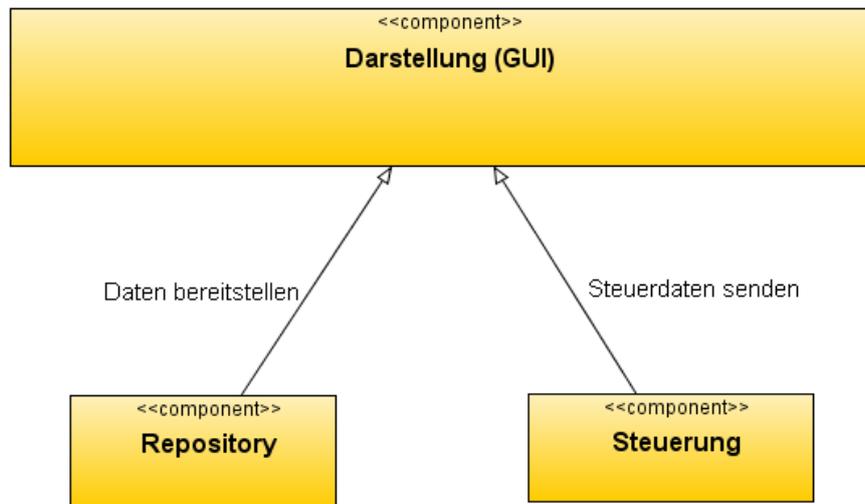


Abbildung 5.1: Systemüberblick

5.1.1 Zusammenspiel der Komponenten

In den folgenden Sequenzdiagrammen soll erläutert werden, wie die einzelnen Komponenten zusammenarbeiten, während der Anwender mit dem System interagiert.

Abbildung 5.2 beschreibt die interne Reaktion des Systems, wenn der Anwender die entscheidenden Kriterien für die Suche ausgewählt hat. In diesem Fall werden alle Bilder analysiert und für die Darstellungskomponente zur Verfügung gestellt. Diese liefert dem Benutzer daraufhin das visuelle Ergebnis auf dem Bildschirm.

Die Abbildung 5.3 beschreibt die Interaktion des Anwenders mit dem Repository. Dieses liest bereits die Daten der Bilder aus, bevor eine Auswahl der Kriterien stattgefunden hat, mit der Absicht Vorschläge zu machen, nach welchen Gesichtspunkten die Bilder gefiltert werden können. Nachdem die relevanten Daten bestimmt wurden, wird die Anfrage gespeichert, um sie für eine erneute Verwendung zur Verfügung zu stellen. Um die Ergebnisse der Suche zu speichern, bietet sich ein Intelligenter Ordner (3.2.1) an. Nach dem Filtervorgang wird der Anwender darüber benachrichtigt, ob Daten zur Ansicht zur Verfügung stehen.

Abbildung 5.4 zeigt die internen Systemabläufe während der Navigation durch den dreidimensionalen Raum. Dabei werden die Bewegungen über eine externe Schnittstelle abgefangen und interpretiert. Falls es sich um eine für das System bekannte Bewegung handelt, wird eine entsprechende Nachricht an die Darstellungskomponente gesendet, damit diese in eine Reaktion des Systems abgebildet wird.

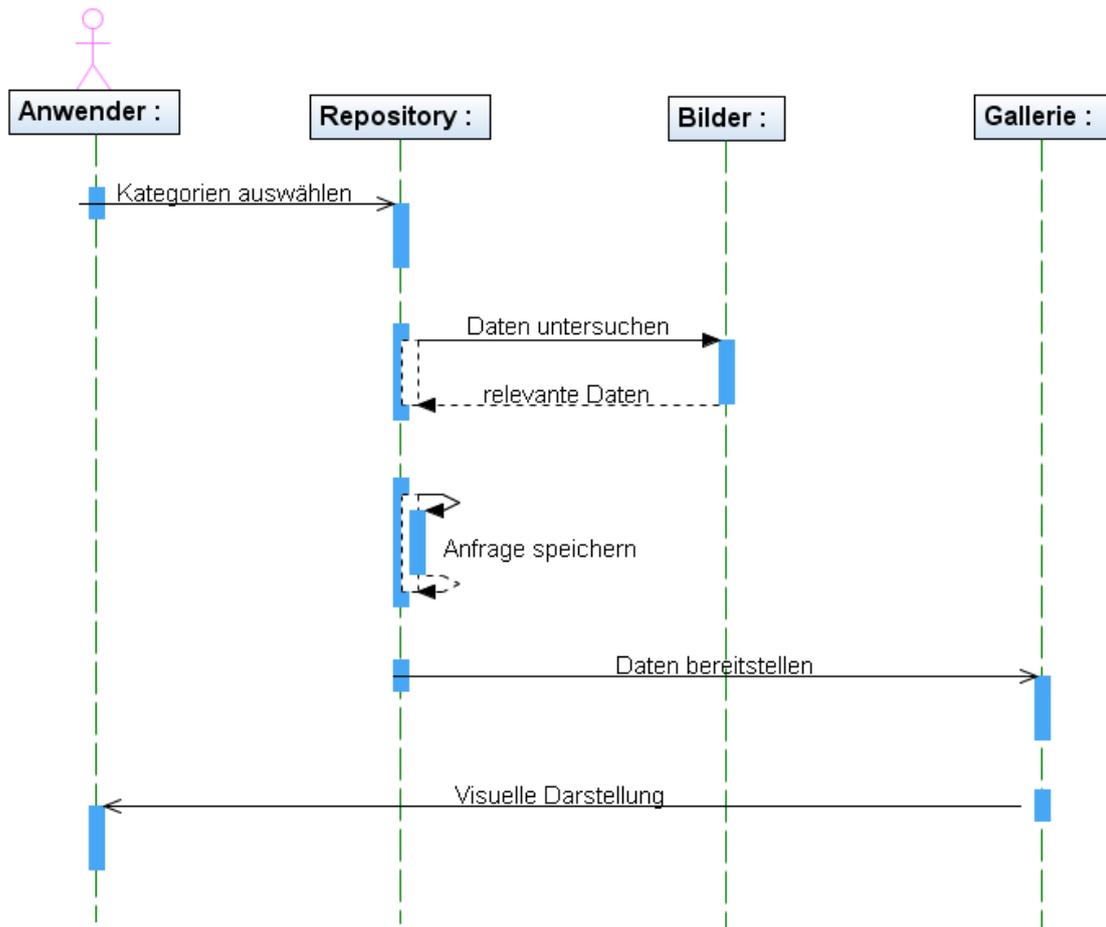


Abbildung 5.2: Ablauf vom Untersuchen der Daten bis zur visuellen Darstellung der Galerie

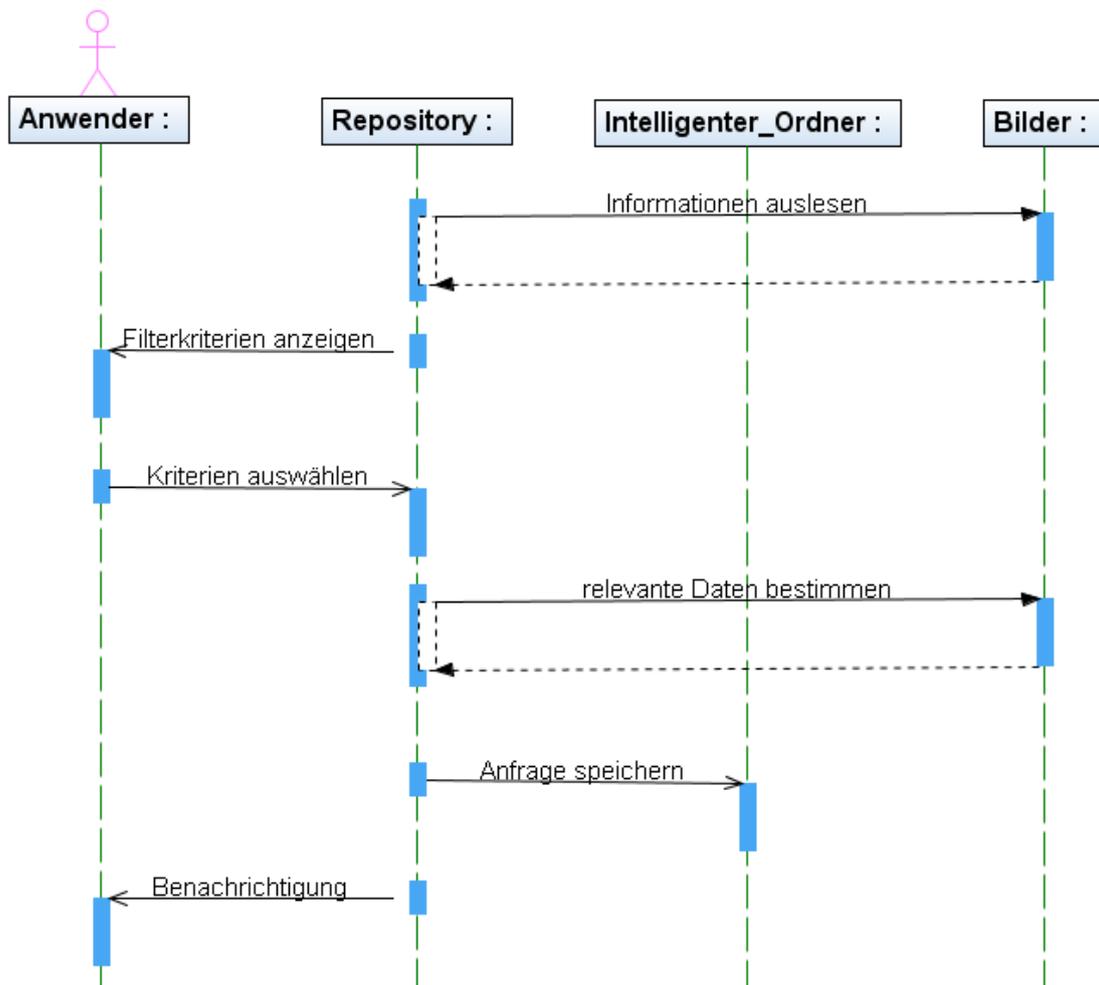


Abbildung 5.3: Ablauf der Einrichtungsphase

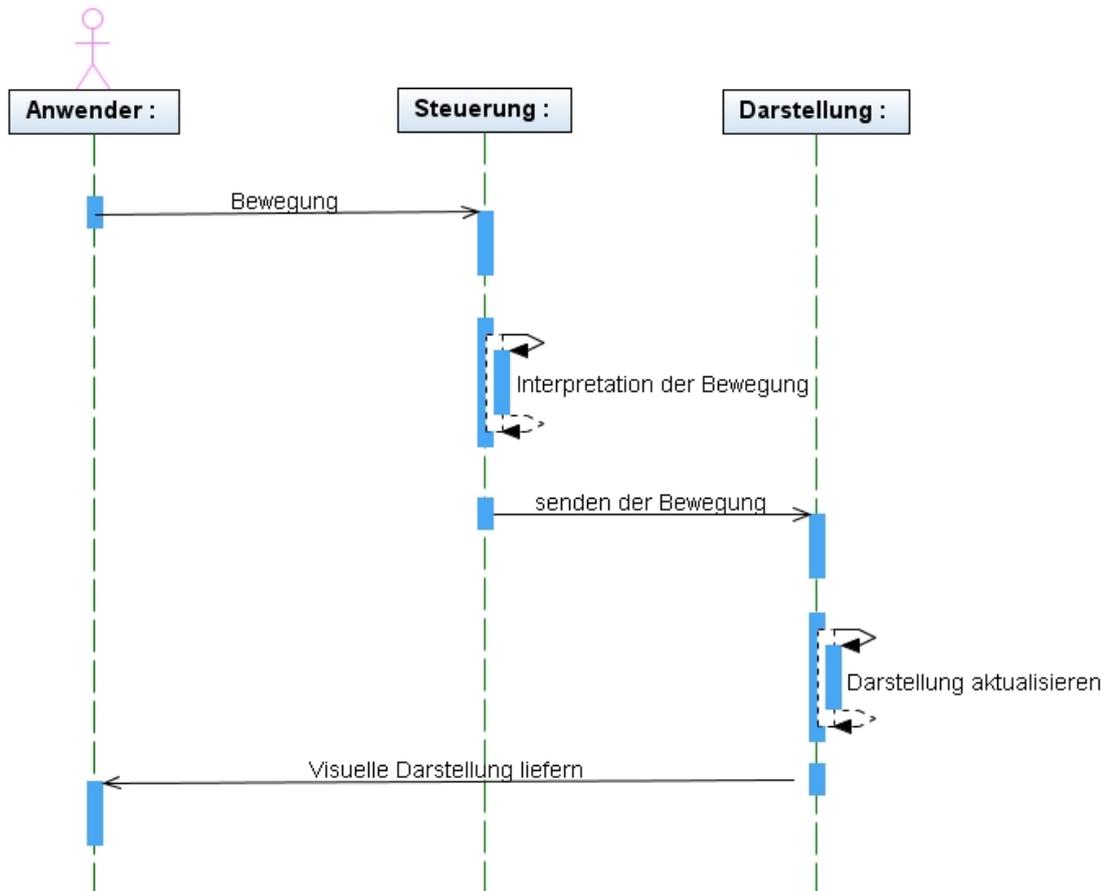


Abbildung 5.4: Navigation

5.2 Architektur

In diesem Teil der Arbeit wird die Architektur der zu realisierenden Anwendung vorgestellt. Die Architektur einer Software ist die Basis für den gesamten Entwicklungsprozess. Sie stellt den Rahmen für die Erfüllung der Anforderungen dar, die technische Basis für das Design sowie das betriebswirtschaftliche Fundament für die Abschätzung der Entwicklungskosten. Außerdem fungiert sie als Grundlage für die Wartung und der Analyse von Abhängigkeiten und Konsistenz innerhalb einer Software (Perry und Wolf, 1992).

Aus der Analyse lassen sich drei primäre Anforderungen identifizieren. Zum einen muss ein Repository implementiert werden, über welches Kriterien bestimmt werden, die bei der

Auswahl der Bilder eine Rolle spielen. Zudem muss eine Komponente diese Bilder entgegen nehmen und in eine übersichtliche Form fügen. Des Weiteren muss eine Interaktion mit dem System über die externe Schnittstelle ermöglicht werden.

Für die Realisierung eines solchen Systems bietet sich das Architekturmuster Model View Controller an.

5.2.1 Model View Controller

Das MVC-Pattern, welches 1979 erstmals beschrieben wurde, dient dazu, einen flexiblen Programmentwurf zu erstellen, in dem eine strikte Aufgabenteilung innerhalb der Anwendung vorliegt. Folglich wird u.a. die Wartbarkeit der Software erhöht.

Bei diesem Konzept wird die Applikation in drei Einheiten aufgeteilt, denen jeweils feste Aufgaben zuzuordnen sind ([Reenskaug, 1979](#)).

Model

Das Model repräsentiert Wissen. Entweder handelt es sich um ein einzelnes Objekt, oder einer Struktur von Objekten. Einerseits sollte es eine eindeutige Beziehung zwischen dem Model und seinen Bestandteilen und andererseits zur dargestellten Welt geben. Diese Komponente ist unabhängig von den beiden weiteren Bestandteilen des MVC.

View

Die View-Komponente stellt die visuelle Repräsentation des Modells dar und ist fest mit ihm verbunden. In der Regel schiebt sie gewisse Attribute des Modells in den Vordergrund, während andere in den Hintergrund gerückt werden. Demzufolge arbeitet die als Präsentationsfilter.

Die erforderlichen Daten, welche sie zur Präsentation benötigt, erhält sie vom Model, indem es Fragen daran stellt. Die Realisierung solcher Fragestellungen der View erfordert, dass das Model die Fragen interpretieren kann und die View die Semantik der gelieferten Antworten versteht, um diese darzustellen.

Controller

Der Controller fungiert als Schnittstelle zwischen Benutzer und dem System. Er empfängt Benutzereingaben, bringt diese in eine Form, in der sie das System verarbeiten kann und liefert diese an ein oder mehrere View-Komponenten. Dabei sollte er die View in keiner

Weise ergänzen, indem er z.B. die Darstellung erweitert. Demgegenüber sollte die View nichts von Benutzereingaben wissen, die über verschiedene Interaktionsmöglichkeiten an das System gesendet werden. Hierdurch ist es zu jeder Zeit möglich, den Controller um Funktionen zu erweitern, welche Eingabesequenzen des Anwenders interpretieren und entsprechende Nachrichten an die View senden.

5.2.2 MVC im System

Die eben vorgestellte 3-Schichten-Architektur soll nun innerhalb des hier vorgestellten Systems umgesetzt werden (Abbildung 5.5).

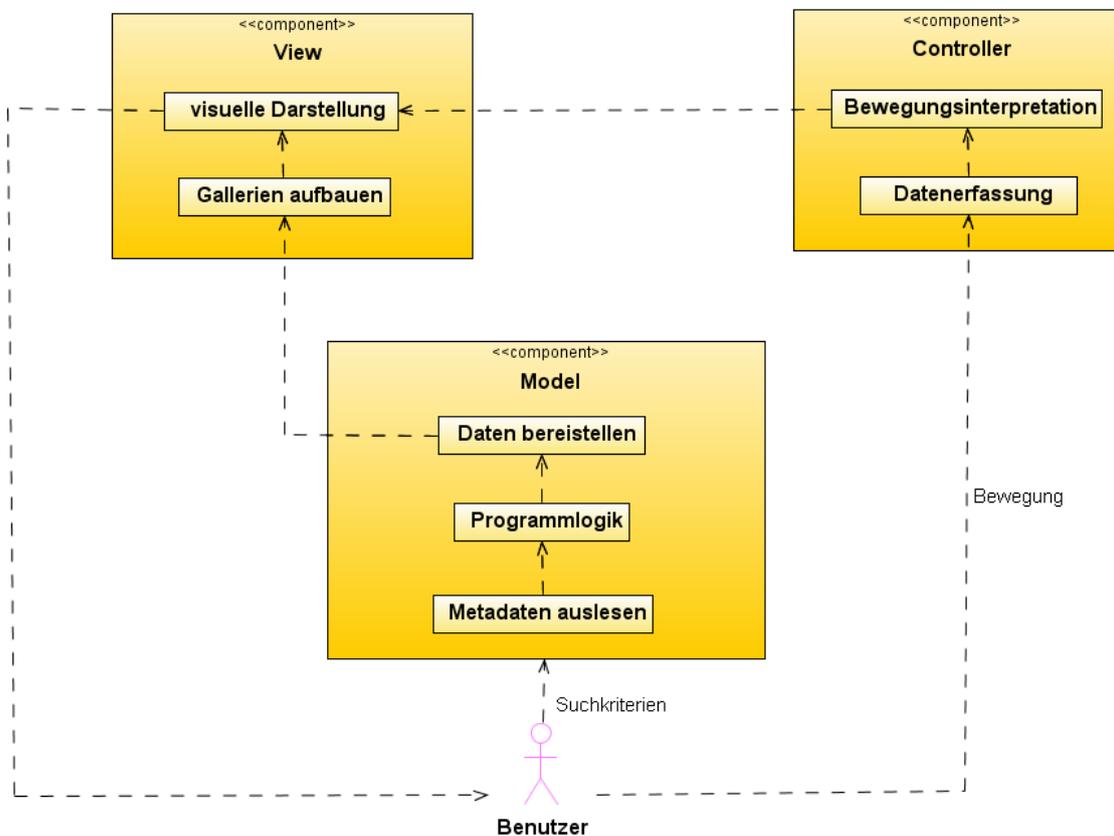


Abbildung 5.5: Komponentenübersicht des MVC innerhalb des Systems

Model

Das Model übernimmt in diesem System die Programmlogik. Das bedeutet, diese Komponente ist dafür zuständig, die Fotos zu analysieren und für die View bereitzustellen, falls sie den Suchkriterien des Anwenders entsprechen. Damit die relevanten Daten bestimmt werden können, muss diese Komponente Zusatzinformationen auslesen und interpretieren. Die Realisierung erfordert eine einheitliche Form, in der die auszulesenden Daten vorliegen. Außerdem muss es eine Schnittstelle zwischen dem Model und der View-Komponente geben. Über diese wird vom Model die Positionierung und die Ausrichtung der darzustellenden Objekte an die View übermittelt. Bei der Präsentation dieser Datenobjekte sind deren physikalischen Eigenschaften, wie z.B. das Gewicht und die Größe, zu beachten, damit dem Anwender eine intuitive Handhabung dieser Elemente geboten wird.

Die Persistenz der Daten spielt bei persönlichen und vertraulichen Daten eine tragende Rolle. Die Informationen müssen diesbezüglich nach einem Prozess oder nach einer Unterbrechung weiterhin existieren, ohne dass sich ihr Informationsinhalt geändert hat. Deshalb ist es empfehlenswert, den Einsatz von Datenbanken und Dateisystemen möglich zu machen. Da allerdings der Fokus dieser Arbeit auf einer bewegungsbasierten Steuerung liegt, wird dieser Aspekt hier nicht weiter betrachtet.

View

Die View-Komponente besitzt eine Schnittstelle zum Model und eine weitere zum Controller. Über erstere wird die visuelle Repräsentation der Model-Komponente realisiert, indem die aktuelle Lage sowie die Größe und Position des darzustellenden Objektes abgefragt wird. Mit dem Ziel diese Aufgabe zu erfüllen, muss die Komponente auf eine beliebige Anzahl von Bildern vorbereitet sein, um dem Anwender alle relevanten Daten präsentieren zu können.

Über die Schnittstelle zum Controller wird die Navigation durch die dargestellten Galerien realisiert. Dabei werden Nachrichten dieser Komponente empfangen und auf die grafische Darstellung abgebildet.

Controller

Der Controller empfängt die vom Anwender, über die zur Verfügung stehenden Eingabegeräte, gesendeten Bewegungen und interpretiert diese. Damit entschieden werden kann, welche Aktion der Benutzer ausgeführt hat, muss er jede Bewegung, die eine Reaktion des Systems auslöst, kennen. Deshalb wird ein bestimmtes Repertoire von Bewegungen bereitgestellt, aufgrund derer vordefinierte Nachrichten an die View-Komponente gesendet

werden. Wie eine dynamische Gesten-Bibliothek aussehen könnte, beschreiben Mohammad Rahimi und Matthias Vogt ([Rahimi und Vogt, 2008](#)). Da es sich bei dieser Anwendung um eine Steuerung handelt, die im Rahmen dieser Arbeit nicht modifiziert wird, werden feste Bewegungen definiert und auf eine dynamische Bibliothek verzichtet.

Anhand der elementaren Interaktionen aus der Analyse (3.2.1) ergeben sich verschiedene Bewegungen die der Controller erkennen muss, um eine Navigation durch die Galerie zu ermöglichen:

vor/zurück: Vor- und Zurückgehen stellen eine elementare Fortbewegungsmöglichkeit zur Navigation durch den Raum dar. Zur Realisierung einer intuitiven Steuerung durch Bewegungen, soll die Komponente eine Gewichtsverlagerung des Anwenders bemerken und entsprechend darauf reagieren. Hier soll eine Verlagerung des Gewichts nach vorne oder hinten ausreichen, um sich innerhalb des Raumes in die entsprechende Richtung zu bewegen.

rechts/links: Auch hier soll die Verlagerung des Körpergewichts in eine der Richtungen eine Reaktion auslösen, indem der Controller die Nachricht an die View sendet, sich um die eigene Achse zu drehen. Aufgrund der begrenzten Möglichkeiten des Menschen im Stehen sein Gewicht zu verlagern, werden rechts und links über Drehungen realisiert. Weitere Bewegungen auf der X-Achse werden somit nicht möglich sein.

Aktionen mit Objekten: Zur Interaktion mit den Objekten, muss der Controller weitere Nachrichten des Benutzers verstehen. Dabei sollen die Objekte anvisiert werden können, um dann Aktionen, wie z.B. Drehen oder Heranzoomen, durchzuführen. Mit dem Ziel dies zu erreichen, soll ein Cursor auf dem Bildschirm gesteuert werden. Dazu bietet sich an, dem Benutzer die Möglichkeit zu verschaffen, über einfache Handbewegungen den Cursor zu kontrollieren. An dieser Stelle sollte ebenfalls eine einfache Bewegung, wie links, rechts, hoch und runter, genügen, damit das System reagiert und den Cursor bewegt.

Diese Bewegungen muss der Controller verstehen und interpretieren, sodass eine Seamless Interaction entstehen kann und folglich eine intuitive Navigation durch die dreidimensionale Galerie erzielt wird.

5.3 Pattern

Design Pattern dienen als Lösungsschablonen für immer wiederkehrende Probleme der Softwareentwicklung. Dabei benennen, abstrahieren und bestimmen sie die Schlüsselaspekte einer allgemeinen Design-Struktur, mithilfe dieser ein wiederverwendbares objektorientiertes Design kreiert wird. Dazu bestimmen sie die beteiligten Klassen und Instanzen sowie ihre

Rollen, Beziehungen und Zuständigkeiten innerhalb des Designs ([Gamma u. a., 1995](#)). Im Folgenden werden Entwurfsmuster beschrieben, welche für die Realisierung der Applikation nützlich sind.

5.3.1 Observer Pattern

Das Observer Pattern gehört zu der Gruppe der Verhaltensmuster. Es wird angewendet, falls es Abhängigkeiten zwischen Objekten gibt, welche beinhalten, dass bei einer Zustandsänderung eines Objektes alle abhängigen Elemente benachrichtigt werden sollen, um sich automatisch zu aktualisieren.

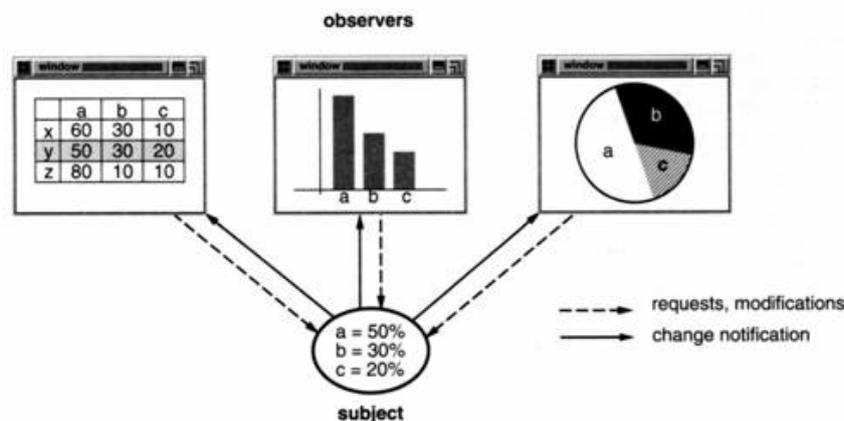


Abbildung 5.6: Beispielanwendung eines Observer Patterns ([Gamma u. a., 1995](#))

Hierzu bietet das beobachtbare Objekt (Subjekt) eine Schnittstelle für die An- und Abmeldung der Observer an und sendet eine Benachrichtigung an die Beobachter, sobald sich sein Zustand ändert. Der Beobachter selbst implementiert ein Interface, an welches die Nachricht der Zustandsänderung geschickt wird. Des Weiteren speichert er den aktuellen Zustand des betrachteten Objektes, um mit diesem konsistent zu sein.

Durch die Verwendung dieses Patterns erreicht man eine lose Kopplung unter den Komponenten. Somit benötigt das Subjekt keine Kenntnisse über die Struktur seiner Observer, da diese nur die angebotene Schnittstelle verwenden. Außerdem erhält ein abhängiges Element Änderungen automatisch.

Das Observer Pattern lässt sich in diesem Design sehr gut einsetzen. Hier findet es Verwendung zwischen der View und dem Model. Wobei der Vorteil darin besteht, dass

das Model keine Kenntnisse über die registrierten Komponenten und deren Logik benötigt. Dabei ist das Repository, also die Model-Komponente, das zu beobachtende Objekt und die Darstellung arbeitet als Observer. Hierdurch kann der Anwender eine neue Suche starten, ohne die bisherige Galerie zu schließen. Diese erfährt nun automatisch von den Änderungen, welche durch die erneute Filterung entstanden sind und kann die neuen Informationen in die darzustellenden Datenbestände aufnehmen. Mit dieser Lösung wird es dem Anwender ermöglicht, den dreidimensionalen Raum dynamisch zu füllen, falls die Suchkriterien erweitert oder eingeschränkt werden sollen.

5.3.2 Singleton

Das Singleton Pattern gehört zu der Kategorie Erzeugungsmuster. Es stellt sicher, dass von einer Klasse nicht mehr als ein Objekt erzeugt wird und bietet einen globalen Zugriffspunkt für diese Instanz an. Dieses Muster wird eingesetzt, wenn es exakt eine Instanz einer Klasse geben muss, welche von mehreren Clients verwendet wird. Außerdem findet es Verwendung, falls eine alleinige Instanz durch Unterklassenbildung spezialisiert werden soll. Der Vorteil dieses Entwurfsmusters liegt neben der Kontrolle darüber wie und wann eine alleinige Instanz verwendet wird darin, dass es eine Verbesserung gegenüber globalen Variablen darstellt, indem beim Singleton der Namespace nicht belegt werden muss.

Innerhalb dieses Designs lässt sich das Singleton Pattern für alle drei Komponenten einsetzen, da es jeweils nur eine Instanz dessen geben soll. Im Zuge der Gewährleistung, nur relevante Daten zu offerieren, sollte zu jeder Zeit nur ein Repository existieren, welches die Darstellung mit Daten versorgt. Des Weiteren sollte es nur eine Steuerung gleichzeitig geben, um sicherzustellen, dass die empfangene Bewegung von den Eingabegeräten des Anwenders stammt. Zudem sollte nur ein Raum, in dem sich die Galerien befinden bestehen, damit sichergestellt ist, wem das Model die relevanten Daten zur Verfügung stellt. Da hier alle drei Komponenten als Singleton realisiert werden, könnten auch direkte Referenzen zwischen den Klassen verwendet werden. Im Hinblick auf die Erweiterbarkeit der Software ist es allerdings sinnvoll dieses Pattern zu implementieren.

Detaillierte Informationen zu den hier vorgestellten Pattern findet man in der Literatur „Elements of Reusable Object-Oriented Software“ ([Gamma u. a., 1995](#)).

5.4 Gestaltung der dreidimensionalen Galerie

Aufgrund der Problematiken, welche sich bei der Gestaltung von 3D-Räumen ergeben (3.3), wird im folgenden Abschnitt beschrieben, wie die Galerien zur Darstellung der Bilder konstruiert werden.

5.4.1 Generierung der Bilderlandschaft

Da die Applikation die Begutachtung von beliebig vielen Fotos ermöglichen soll, muss definiert sein, in welcher Weise die Bilder innerhalb des Raumes angezeigt werden. Weil es sich bei Bildern um zweidimensionale Objekte handelt, werden diese in einem dreidimensionalen Rahmen gezeigt. Dadurch können digitale Fotos wie dreidimensionale Objekte verwendet werden. Mit der Absicht diese Bilder innerhalb einer begrenzten Landschaft anzuzeigen, muss die Anwendung vorgefertigte Räume zur Verfügung stellen, deren Größe sich an der Anzahl der darzustellenden Bilder orientiert. Dabei könnten die Bilder an animierten Wänden präsentiert werden (Abbildung 5.7 links), an denen sich der Anwender orientiert und die Bilder betrachtet. Der Innenraum der Galerie würde somit, wie es auch in realen Ausstellungen der Fall ist, als Freiraum zur Bewegung dienen. Aufgrund dessen würde für den Anwender das Gefühl entstehen sich in einem dreidimensionalen Museum zu bewegen.

Ein weiterer Ansatz wäre, die Daten an einer imaginären Wand darzustellen (Abbildung 5.8). In diesem Fall könnte der Anwender entlang der Bilder gehen und bekäme schnell eine gute Übersicht über die Menge der Bilder. Außerdem lassen sich in dieser Variante viele Fotos auf einem kleinen Raum verteilen. Mit der Internetbrowser-Erweiterung Cooliris wurde die Idee einer digitalen Bilderwand bereits erfolgreich implementiert (Cooliris, 2008). Hierbei handelt es sich um eine Erweiterung die derzeit für den Firefox, den Microsoft Internet Explorer sowie für den Safari erhältlich ist. Cooliris erlaubt es dem Benutzer, Bilder und Videos verschiedener Seiten in einer eleganten Diashow zu betrachten.

Abbildung 5.9 zeigt eine dritte Darstellungsvariante. Hier gibt es bis auf die angezeigten Bilder keine Orientierungsmerkmale innerhalb des Raumes. Dadurch entsteht für den Anwender das Gefühl, durch die Bilderlandschaft zu fliegen und nicht, wie bei den bisherigen Varianten, durch eine Ausstellung zu laufen. Dabei vergrößern sich die Bilder je näher man ihnen kommt bzw. verschwinden im Hintergrund, sobald man sie passiert hat. Allerdings muss hier festgelegt werden, wie das System reagiert, wenn der Anwender das Ende der Darstellung erreicht hat. In diesem Falle könnte er in einem leeren Raum stehen und somit die Orientierung verlieren. Deshalb wäre es sinnvoll, den Anwender über das Ende zu informieren, oder mit der Darstellung der Bilder von Neuem zu beginnen, um den Interaktionsfluss nicht zu unterbrechen.

In allen bisherigen Darstellungsformen könnte der Benutzer die Bilder ausschließlich in der Frontansicht betrachten und nicht um die Bilder herumlaufen, mit der Absicht z.B. weitere

Informationen auf der Rückseite des Bildes zu erhalten.

Aufgrund dessen werden die Fotos in dieser Anwendung frei im Raum angeordnet (Abbildung 5.7 rechts). Das bedeutet, dass die Bilderrahmen innerhalb der Landschaft in verschiedenen Abständen angeordnet werden und somit der gesamte Raum für die Darstellung ausgenutzt werden kann.

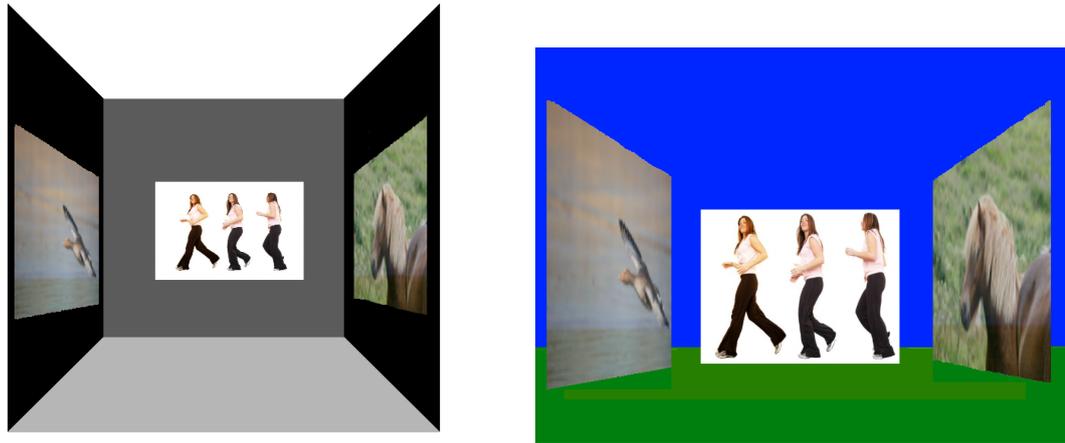


Abbildung 5.7: Beispielhafte Anordnung von Bildern. Links werden die Bilder in einem statischen Raum und rechts im freien Raum angezeigt.



Abbildung 5.8: Anordnung der Bilder in einer Bilderwand

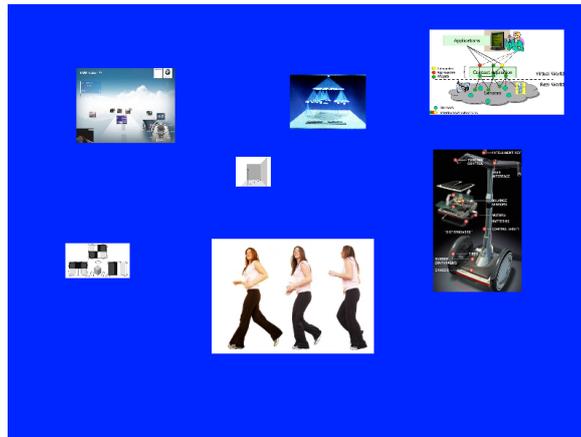


Abbildung 5.9: Anordnung der Bilder in einem freien Raum ohne Orientierungsmerkmale

5.4.2 Anordnung der Bilder

Für die Konstruktion der Galerien gibt es verschiedene Varianten zur Anordnung der Bilder. In diesem Kontext sollen verschiedene Räume entstehen, um die Übersicht bei einer großen Bilderanzahl weiterhin zu gewährleisten. Ein Ansatz hierzu wäre, die Bilder systematisch in verschiedenen Räumen aufzubauen. Diesbezüglich könnte man zur Anordnung der Bilder die vom Benutzer vorgegebenen Kriterien verwenden. Wenn ein Anwender die Fotos anhand des Datums eingrenzt, könnten aufgrund dieser Kategorie verschiedene Räume generiert werden. Dabei könnte ein Raum alle Bilder enthalten, die genau einem angegebenen Intervall entsprechen und zwei weitere die Bilder, die ein Datum kurz vor bzw. nach den relevanten Daten aufweisen. Zur Gestaltung übersichtlicher Räume, sollen in dieser Arbeit mehrere Galerien aufgebaut werden, sobald eine gewisse Anzahl an Bildern erreicht ist. Allerdings werden diese nicht weiter thematisch unterteilt, da durch das Repository nur relevante Daten in die Galerie aufgenommen werden und hier der Fokus auf dem generellen Aufbau von dreidimensionalen Galerien gelegt wird.

Für die Übersichtlichkeit innerhalb der Räume spielt die Größe der angezeigten Bilder eine tragende Rolle. Dabei soll eine einheitliche Größe für die Rahmen gewählt werden. Die Problematik, die sich hierbei ergibt ist, dass verschiedene Bilder häufig unterschiedliche Auflösungen aufweisen und folglich unterschiedlich groß innerhalb der Galerien dargestellt werden würden. Dadurch könnte für den Anwender der Eindruck entstehen, dass große Bilder eine höhere Priorität besitzen. Da dieses Empfinden nicht der Realität entspricht, sollen alle Bilder innerhalb des Raumes die gleiche Größe aufweisen. Aus diesem Grund muss die hier vorgestellte Applikation die Auflösung von Bildern analysieren können, um sie gegebenenfalls zu ändern, ohne einen zusätzlichen Aufwand für den Anwender zu produzieren.

5.4.3 Orientierung und Perspektive innerhalb der Galerie

Damit die Orientierung für den Anwender sichergestellt wird, sollte dieser zu jedem Zeitpunkt wissen, in welchem Raum er sich befindet. Deshalb bietet es sich an, die Landschaften visuell voneinander zu differenzieren. Außerdem wäre es sinnvoll eine Gesamtübersicht der erstellten Galerien anzubieten. Zur Betrachtung weiterer Objekte muss es dem Benutzer ermöglicht werden, zwischen den Galerien zu wechseln. Das bedeutet, die Räume müssen untereinander verbunden sein, um flüssige Übergänge zwischen diesen zu erreichen.

Eine 3D Amnesie (3.3.1) könnte auftreten, falls der Anwender ein Bild heranzoomt und somit ein zweidimensionales Objekt vor sich findet. Um dieser vorzubeugen, muss er intuitiv wissen, wie er zurück in die dreidimensionale Ansicht wechselt. Zum Heranzoomen von Objekten sollte es Animationen zwischen den Ansichten sowie eine eindeutige Interaktion mit dem System geben.

Die Perspektive bei der Navigation soll in dieser Anwendung dem Benutzer überlassen werden. Er soll also zwischen der „First Person“ und der „Third Person“ Perspektive wählen können. Während die Abbildungen 5.7 bis 5.9 die Ego-Perspektive, also die First-Person-Sicht zeigen, werden an dieser Stelle zwei weitere Third-Person-Perspektiven vorgestellt (Abbildung 5.10). In der linken Abbildung schaut man der Figur ausschließlich über sie Schulter. Bei dieser sogenannten Verfolgerperspektive befindet sich die Kamera in einer festen Position hinter dem Charakter und wird synchron zu diesem gelenkt. Dadurch weist diese Sicht eine hohe Ähnlichkeit zur Ego-Perspektive auf.

Rechts auf der Abbildung 5.10 wird die Third-Person-Perspektive aus der Vogelperspektive dargestellt. Hier befindet sich die Kamera in deutlich größerem Abstand zur gesteuerten Figur als bei der Verfolgerperspektive. Die Kamerasteuerung erfolgt hier entweder synchron zum Charakter oder durch ein separates Schema. Der Vorteil dieser Perspektive liegt darin, dass sie in einigen Fällen eine bessere Übersicht der Umgebung liefert und somit sehr gut für große Bilderbestände geeignet ist.

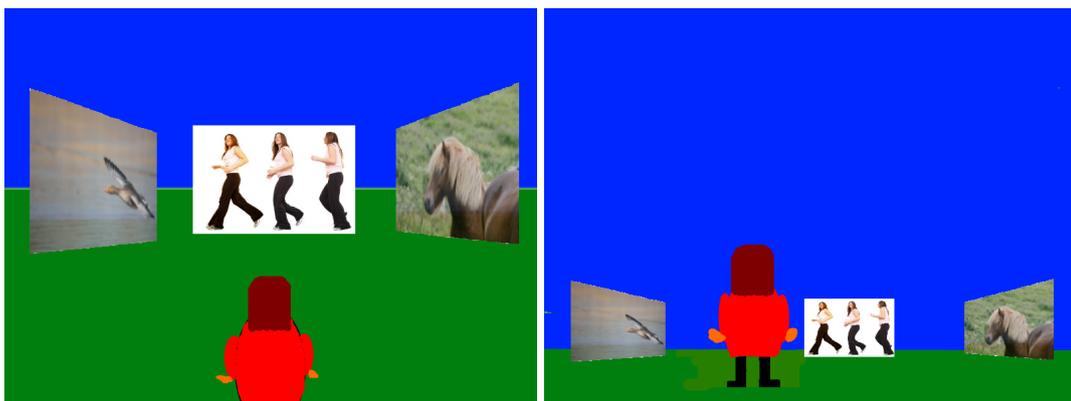


Abbildung 5.10: Beispielhafte Third-Person-Perspektiven

5.5 Agile Softwareentwicklung

Agile Methoden in der Informatik beschreiben den Einsatz von Agilität innerhalb des Softwareentwicklungsprozesses. Das Ziel hierbei ist, den Prozess flexibler und schlanker zu gestalten als es bei anderen klassischen Vorgehensweisen, wie z.B. beim V-Modell, der Fall ist. Dies soll dadurch erreicht werden, indem man den Schwerpunkt der Entwicklung auf die technischen sowie sozialen Probleme fokussiert. Eine der bekanntesten Methoden der agilen Softwareentwicklung ist das Extreme Programming, welches im folgenden Teil genauer beleuchtet werden soll.

5.5.1 Extreme Programming

Extreme Programming (XP) wird als eine Philosophie der Softwareentwicklung beschrieben und definiert Vorgehensmodelle, welche speziell auf Teamarbeit, Kommunikation, Feedback sowie Einfachheit basieren.

Dies wird durch viele und kleine Entwicklungsschritte erreicht, die ein frühzeitiges sowie konkretes und kontinuierliches Feedback an alle beteiligten Personen beinhalten. Durch diese kurzen Entwicklungsphasen entwickelt sich inkrementell ein Gesamtplan des Systems, welcher innerhalb jedes Schrittes weiterentwickelt wird. Auf diese Weise erhält man eine starke Flexibilität, falls sich während des Realisierungsprozesses etwas an der Funktionalität des Systems ändert. In diesem Vorgehensmodell wird also davon ausgegangen, dass die endgültigen Anforderungen an eine Software nur selten vollständig zu Projektbeginn bekannt sind. Des Weiteren werden die beschriebenen Entwicklungsphasen möglichst früh von kontinuierlichen Tests begleitet, um Programmfehler sowie eine fehlerhafte Integration von Softwarekomponenten zu vermeiden.

Weitere Informationen zu XP sind in der Literatur „Extreme Programming Explained“ ([Beck und Andres, 2004](#)) zu finden.

5.5.2 Test-Driven Development

Bei dieser Form der Softwareentwicklung werden die Tests konsequent vor den zu testenden Komponenten entwickelt. Dabei beinhaltet eine Sequenz innerhalb der Entwicklung sechs Aufgaben, die in [Desai u. a. \(2008\)](#) beschrieben werden:

1. Hinzufügen eines Tests für ein nicht implementiertes Element.
2. Ausführung aller bisher bestehenden Tests, um sicherzustellen, dass nur die neu hinzugefügten Tests fehlschlagen.

3. Neue Funktionalität im System implementieren.
4. Ausführung aller bisher bestehenden Tests, um sicherzustellen, dass alle Tests erfolgreich sind.
5. Refaktorisierung, um die Lesbarkeit oder Struktur des Codes zu verbessern.
6. Neubeginn bei Schritt 1.

Allerdings wird beim Test-Driven Development (TDD) nicht zwingend eine hundertprozentige Codeüberdeckung durch Tests angestrebt, da dies häufig zu zeit- und kostenintensiv ist. Aus diesem Grund müssen die Testintensität sowie der Testumfang in Abhängigkeit des Risikos und der Kritikalität analysiert und individuell festgelegt werden.

Der Erfolg dieses Entwicklungsprozesses wird im Rahmen einer Untersuchung ([Williams u. a., 2003](#)), bei der eine Fehlerreduktion von 40 Prozent bei einer Testgetriebenen Softwareentwicklung gegenüber einem Ad-Hoc Testansatz beobachtet werden konnte, beschrieben.

5.5.3 Prototyping

Der Ansatz des Prototyping wird u.a. dazu verwendet, bestehende Tests zu erfüllen sowie die Anforderungen an ein System auf ihre Realisierbarkeit hin zu untersuchen. Dazu werden Prototypen des Systems entwickelt, die bereits erste Ergebnisse liefern und ein frühzeitiges Feedback ermöglichen. Dabei repräsentiert der Prototyp ein ausführbares Modell des Systems, welches nicht zwingend alle finalen Anforderungen erfüllt, sondern sich auf ausgewählte Eigenschaften begrenzt.

Ein Ansatz hierbei ist das Rapid Prototyping, welches in [Kordon und Luqi \(2002\)](#) beschrieben wird. Diese Form des Prototyping wird innerhalb eines Projektes dazu verwendet, die Realisierbarkeit eines Systems zu prüfen oder weitere Anforderungen an das System zu spezifizieren. Dabei können Modelle entstehen, die entwickelt werden, um später vollständig verworfen zu werden und somit nicht als Teil des entwickelten Produktes einfließen. Bei diesem „Throw-Away Ansatz“ muss daher Aufwand betrieben werden, der letztlich nicht zur Implementation des finalen Produktes beisteuern wird. Beim „Evolutionären Prototyping“ hingegen wird eine Serie von Prototypen erstellt, welche sich immer weiter dem Produkt annähern und schließlich der finalen Software entsprechen.

Für die Realisierung einer bewegungsbasierten Steuerung werden im folgenden Abschnitt zwei Prototypen vorgestellt, anhand derer die zu verwendenden Eingabegeräte evaluiert werden. Es wird somit ein „Experimentelles Prototyping“ durchgeführt, welches die Möglichkeiten der Realisierung aufzeigt und weitere Aspekte der vorgestellten Ansätze aufgreift.

5.6 Machbarkeitsstudie

Mit der Absicht eine intuitive bewegungsbasierte Steuerung zu realisieren, werden an dieser Stelle zwei Prototypen präsentiert, die vor der eigentlichen Entwicklungsphase entstanden sind. Mit diesen Anwendungen sollen die vorgestellten Eingabegeräte evaluiert und exemplarisch dargestellt werden, wie die Steuerung der Anwendung aussehen soll. Dazu wurden zwei Beispielanwendungen implementiert, die eine Navigation innerhalb schon bestehender und frei verfügbarer Programme realisiert. Hierbei handelt es sich um Google Earth¹³ und Cooliris¹⁴. Bei Google Earth wurde eine Steuerung mit dem Wii Balance Board in Verbindung mit der Wii Remote und bei Cooliris mit dem Wii Balance Board und dem Neural Impulse Actuator implementiert. Die dabei entstandene Fragestellung, ob Nintendo Wii Geräte auf ein Wii-fremdes Gerät übertragbar sind, konnte bereits in verschiedenen Arbeiten beantwortet werden. Die Integration der Wii Remote in ein fremdes System wird in [Schlömer u. a. \(2008\)](#) und in der Masterarbeit von Christian Fischer ([Fischer \(2007\)](#)) durchgeführt. Mit der Einbindung des Wii Balance Board in ein anderes System als der Nintendo Wii, befasst sich [de Haan u. a. \(2008\)](#).

Die Abbildung 5.11 zeigt ein generelles Schema der zu implementierenden Steuerung. Dabei senden die Eingabegeräte über eine entsprechende Schnittstelle Bewegungsdaten, welche von den erstellten Applikationen abgefangen und interpretiert werden müssen. Des Weiteren muss die Bewegungserkennung in der Lage sein, Nachrichten an die zu steuernden Programme zu schicken, damit die Bewegungen auf Reaktionen innerhalb des Programms abgebildet werden können.

¹³<http://earth.google.de/>

¹⁴<http://www.cooliris.com/>

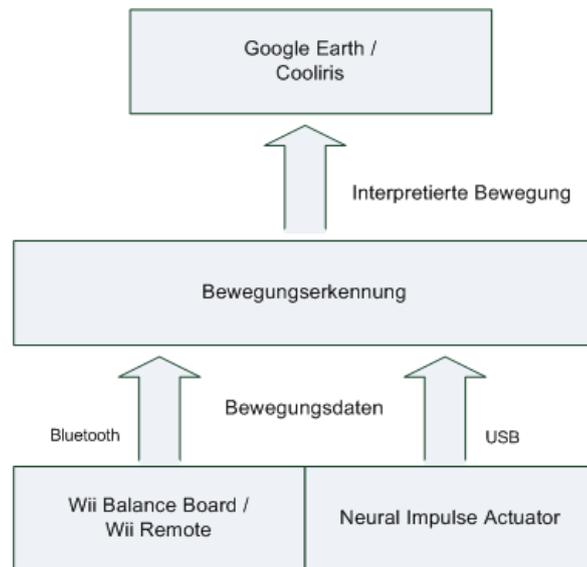


Abbildung 5.11: Schema der Steuerung

Integration der Wii-Geräte

Eine Integration der Wii Remote und des Wii Balance Board in ein fremdes System, wie einem PC, lässt sich über die Bluetooth-Schnittstelle realisieren. Dabei findet die Kommunikation ohne eine Verschlüsselung statt und kann somit problemlos durchgeführt werden. Zum Einbinden der Geräte, müssen sie lediglich eingeschaltet sein und sich im Synchronisationsmodus befinden. Hierin stellen die Geräte einen Dienst zur Verfügung, um von einer externen Stelle, z.B. einem Betriebssystem (in diesem Fall Windows), eingebunden werden zu können.

Nachdem die Socket-Verbindung zur Kommunikation mit den Geräten aufgebaut ist, lässt sich u.a. mit der WiimoteLib 1.5.2 (Peek, 2008) testen, ob die Integration des Gerätes vollständig gelungen ist. Dazu lassen sich, basierend auf den vorhandenen Sensoren, verschiedene Daten abfragen. Bei der Wii Remote handelt es sich dabei um den Status aller zwölf Buttons, welche sich auf dem Controller befinden sowie die aktuellen Werte des 3-Achsen Beschleunigungssensors und den Batteriestand des Gerätes. Außerdem lassen sich Nachrichten an den Controller senden, um LED's oder die Vibrationsfunktion der Wii Remote anzusteuern. Für das Wii Balance Board lassen sich die Daten der vier Drucksensoren sowie ein aus den vorliegenden Werten errechneter Durchschnittswert, der das auf dem Board befindliche Gesamtgewicht widerspiegelt, abfragen.

5.6.1 Bewegungserkennung

Die Bewegungserkennung muss verschiedene Aufgaben erfüllen (siehe Abbildung 5.12). Um diesen nachzukommen, muss die Komponente in der Lage sein, die Daten der verschiedenen Eingabegeräte zu empfangen. Des Weiteren müssen die eingehenden Werte aufbereitet werden. Das bedeutet, sie muss den aktuellen Zustand des zu steuernden Programms kennen, wie z.B. die aktuelle Mauszeigerposition, um anhand dieser Daten zu entscheiden, welche gültigen Aktionen ausgeführt werden können. Dadurch kann u.a. verhindert werden, dass der Mauszeiger während der Navigation aus dem gültigen Bereich läuft oder weitere ungültige Aktionen ausgeführt werden. Falls es sich um eine erlaubte Bewegung handelt, wird aufgrund des aktuellen Zustands eine entsprechende Handlung ausgeführt.

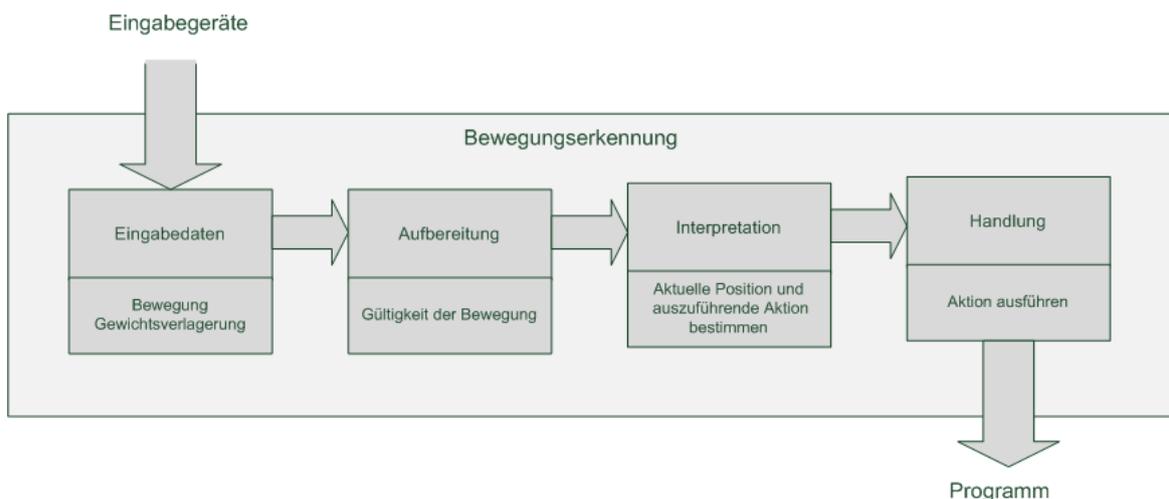


Abbildung 5.12: Schematische Darstellung der Aufgaben der Bewegungserkennung

5.6.2 Cooliris

Bei dieser Anwendung soll der Benutzer das Programm Cooliris mithilfe des Wii Balance Board sowie dem Neural Pulse Actuator steuern. Das Ziel dabei ist, die klassische Maus zur Steuerung des Programms durch diese Eingabegeräte zu ersetzen. Wie bereits im Abschnitt 5.4.1 erläutert wurde, werden in Cooliris beliebig viele Bilder als eine Bilderwand dargestellt (siehe Abbildung 5.13).



Abbildung 5.13: Bilddarstellung in Cooliris

Für eine Steuerung dieser Anwendung müssen folgende Funktionen implementiert werden:

Zoomen: Cooliris bietet die Möglichkeit, ein angezeigtes Bild per Mausklick heranzuzoomen. Dabei besteht eine Sequenz aus drei Klicks: Beim ersten Klick wird das Bild vergrößert angezeigt, beim zweiten wird es im Vollbild dargestellt und der dritte Mausklick beendet den Zoomvorgang.

Diese Funktion wird in diesem Prototypen von dem Neural Pulse Actuator übernommen. Da dieses Gerät Muskelbewegungen eines Gesichtes detektieren kann, werden diese Daten dazu verwendet, die Mausklicks zu simulieren. Das bedeutet, dass pro Muskelbewegung des Anwenders, welche eine gewisse Stärke aufweist, ein Mausklick ausgeführt wird.

Navigieren: Die Navigation nach links und rechts wird in dieser Anwendung ebenfalls durch die Maus realisiert. Dazu muss die linke Maustaste dauerhaft gedrückt werden. Während die Taste gedrückt wird, kann der Anwender durch Bewegungen auf der X-Achse die Richtung bestimmen, in die er an der Wand entlang läuft.

Im hier vorgestellten Prototypen wird diese Funktion vom Wii Balance Board übernommen. Dabei soll es dem Anwender ermöglicht werden, durch einfache Gewichtsverlagerungen seines Körpers in die entsprechende Richtung zu navigieren. An dieser Stelle werden also nur Bewegungen auf der X-Achse implementiert.

Die hier erstellte Applikation muss für die Aufgabe also dazu imstande sein, die Daten des Boards zu empfangen und zu interpretieren. Anhand dieser erhaltenen Daten wird analysiert, welche Körperhaltung der Benutzer eingenommen hat. Wenn er sich nach links oder rechts beugt, wird ein linker Mausklick sowie eine Bewegung des Mauszeigers in die entsprechende Richtung simuliert. Dementsprechend wird die Navigation durch die Bilderbestände ausschließlich anhand von Bewegungen des Benutzers realisiert.

Auswertung

Für die Steuerung über den Neural Pulse Actuator (NIA) wurde ein Event-Heap ([Johanson u. a., 2001](#)) eingesetzt, damit verhindert werden kann, dass Cooliris und das Gerät auf derselben Einheit installiert sein müssen. Dies hätte zu Problemen geführt, da der NIA über ein USB-Kabel angeschlossen wird und der Anwender sich somit jederzeit in unmittelbarer Nähe des PC befinden müsste, auf dem die zu steuernde Anwendung läuft. Durch den Einsatz des Event-Heaps kann die Applikation auf einer mobilen Einheit, wie z.B. einem Laptop laufen und erleichtert somit erheblich die Steuerung in Verbindung mit dem Wii Balance Board.

Für diesen Prototypen, in dem die Aufgabe des NIA auf eine Sequenz von Mausklicks auf einem bestimmten Punkt des Bildschirms beschränkt ist, ist der Einsatz dieses Gerätes empfehlenswert. Allerdings bietet der Neural Pulse Actuator keine Möglichkeit den Mauszeiger über den Bildschirm gleiten zu lassen. Da dies eine elementare Funktion innerhalb des dreidimensionalen Raumes sein wird, um z.B. die Bilder anzuvisieren, wird dieses Gerät in der weiteren Realisierung der Anwendung nicht weiter betrachtet.

Nach der erfolgreichen Integration der Wii-Geräte über die Bluetooth Schnittstelle, konnten für diesen Prototypen die Bewegungen auf der X-Achse einwandfrei implementiert werden. In diesem Teil der Steuerung war zu berücksichtigen, dass der Mauszeiger nicht über den angezeigten Raum hinauslaufen sollte. Dazu wird die aktuelle Bildschirmauflösung ausgelesen und der Zeiger automatisch in die Mitte gesetzt, wenn er sich dem Ende des Bildschirms nähert, damit der Anwender nicht im Interaktionsfluss gestört wird und keine Einschränkungen in der Steuerung hinnehmen muss.

5.6.3 GoogleEarth

In diesem Prototypen soll das Zusammenspiel des Wii Balance Board mit der Wii Remote für eine bewegungsbasierte Steuerung evaluiert werden. Dazu soll eine Navigation innerhalb von Google Earth implementiert werden. Diese Anwendung mithilfe des Wii Balance Board zu steuern, ist von dem Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) bereits erfolgreich implementiert worden ([DFKI, 2008](#)). Wie im vorherigen Prototypen beschrieben, soll auch hier die klassische Maus durch die beiden Eingabegeräte ersetzt werden und somit eine vollständige Steuerung der Anwendung über diese Geräte ermöglichen. Damit dieses Ziel erreicht werden kann, müssen folgende Funktionen implementiert werden:

Zoomen: In diesem Fall besteht das Zoomen nicht aus einer Sequenz von Mausklicks, sondern wird über das Mousrad durchgeführt. Das bedeutet, mit einer Drehung des Mousrads nach oben zoomt man weiter in die Welt hinein und nach unten drehen bewirkt das

Gegenteil, also aus der Welt heraus.

Für diese Funktion bietet sich das Steuerkreuz der Wii Remote an. Da der Anwender bereits vom Mausrad daran gewöhnt ist, werden hier die Knöpfe oben/unten des Kreuzes verwendet. Allerdings wird hier die Funktion der Maus etwas modifiziert. Denn hier reicht eine kurze Betätigung des Knopfes aus, um den Zoomvorgang einzuleiten und muss nicht wie bei dem Mausrad häufiger bedient werden. Wenn sich der Anwender an der gewünschten Position befindet, kann er durch das Betätigen des B-Knopfes, welcher sich auf der Rückseite des Controllers befindet, das Zoomen beenden.

Navigieren: Die Navigation in Google Earth ist in alle vier Himmelsrichtungen sowie in der Diagonalen möglich, wobei diese über die Maus oder die Tastatur durchgeführt werden kann. In dieser Applikation wird der Fokus auf die Steuerung über die Tastatur gesetzt. Die Aufgabe der Navigation soll auch in diesem Prototypen vom Wii Balance Board übernommen werden. An dieser Stelle wird sich die Steuerung jedoch nicht auf die X-Achse beschränken. Hier soll sich der Benutzer auch nach vorne, hinten sowie in die diagonale Richtung lehnen können, um in die entsprechende Richtung zu navigieren.

Auswertung

In diesem Prototypen ist die Aufgabe der Wii Remote auf das Zoomen beschränkt und ist zur Umsetzung dieser Funktionalität sehr empfehlenswert. Allerdings werden die weiteren Knöpfe und Sensoren dieses Gerätes in dieser Applikation nicht verwendet. Durch den 3-Achsen Beschleunigungssensor bietet sich dieses Gerät darüber hinaus an, in der weiteren Realisierung das Bewegen eines Mauszeigers zu übernehmen.

Das Wii Balance Board hat sich auch in dieser Applikation für die Navigation bewährt und wird in der weiteren Realisierung dazu verwendet werden, sich innerhalb des dreidimensionalen Raumes zu bewegen.

5.6.4 Ergebnis der Machbarkeitsstudie

Die Erstellung der Prototypen lieferte viele Erkenntnisse, die eine tragende Rolle bei der weiteren Implementation spielen. Neben der Evaluation der zu verwenden Eingabegeräte, konnten Erkenntnisse über die Intuitivität der realisierten Steuerungen gesammelt werden. Durch verschiedene Vorfürungen für Besuchergruppen die das iFlat der HAW Hamburg besucht haben, konnten die Applikationen an diversen Testpersonen evaluiert werden. Dabei wurde in den meisten Fällen der Prototyp Google Earth verwendet, da der Neural Pulse Actuator im Cooliris Prototypen für eine weitere Implementierung nicht in Betracht gezogen wird und für jede Person eine Kalibrierungsphase benötigt.

Durch die Rückmeldungen der Personen lässt sich darauf schließen, dass die Steuerungen

eine handbuchfreie Benutzung ermöglichen. Diese Schlussfolgerung ist neben Gesprächen darauf zurückzuführen, dass jede Person intuitiv wusste wie sie sich verhalten muss, um eine entsprechende Reaktion des Systems auszulösen. Allerdings traten während der Interaktion in einigen Fällen Irritationen beim Anwender auf, welche für eine Implementation der Steuerung weiterer Systeme sehr hilfreich sind. Dabei war auffällig, dass einige Personen während der Navigation mit dem Wii Balance Board parallel dazu versucht haben mit der Wii Remote zu zoomen. Die gleichzeitige Verwendung dieser Funktionen wird allerdings in Google Earth nicht angeboten. Das bedeutet, dass dies auch bei einer Steuerung mit den ursprünglichen Eingabegeräten nicht möglich ist.

Aufgrund dieser entstandenen Irritationen sowie den einzugehenden Kompromissen in der Steuerung der hier vorgestellten Prototypen, soll im weiteren Verlauf eine Applikation entwickelt werden, in der diese Probleme aufgegriffen und gelöst werden. Außerdem sollen die Möglichkeiten der Eingabegeräte weiter ausgereizt werden, indem ihnen mehr Funktionalität zugetragen wird. Des Weiteren wurde der Einsatz des Wii Balance Board sowie der Wii Remote zur Navigation innerhalb eines dreidimensionalen Raumes hier nicht implementiert.

6 Realisierung

In diesem Kapitel soll unter Berücksichtigung des Designs und den Anforderungen aus der Analyse eine konkrete Umsetzung der Applikation durchgeführt werden. In diesem Zusammenhang soll ein Filter entstehen, der Daten verschiedener Fotos ausliest und für die visuelle Darstellung zur Verfügung stellt. Die Darstellungskomponente ist dafür zuständig, die Bilder innerhalb von Galerien zu präsentieren, durch die dann über die externe Steuerung navigiert wird.

6.1 Filter

Wie bereits in den grundlegenden Funktionen (3.2.3) beschrieben wurde, muss der Filter in der Lage sein, die Daten verschiedener Fotos auszulesen. Das Speichern der gestellten Anfrage wurde hier nicht implementiert, da der Fokus hier auf dem Auslesen der Daten sowie deren Bereitstellung liegt.

Die ausgelesenen Eigenschaften der Fotos sollen visuell dargestellt werden, sodass Anwender ohne Kenntnisse über die Attribute zwischen allen Merkmalen wählen und anhand dieser filtern können. Die Voraussetzung, diese Daten der Fotos auslesen zu können ist, dass diese bereits mit entsprechenden Eigenschaften bestückt sind. Die Art und Weise wie man Bilder mit solchen Metadaten versehen kann, wird im folgenden Abschnitt beschrieben.

6.1.1 Bilder mit Metadaten versehen

Um Bilder mit Attributen zu versehen, bieten sich das Exchangeable Image File Format (Exif) und der IPTC-NAA-Standard an. Hierbei handelt es sich um Formate, die Daten direkt in die Datei von Bildern schreiben. Exif befindet sich bereits in der Version 2.2 und wurde von der „Japan Electronic and Information Technology Industries Association (JEITA)“ standardisiert (JEITA, 2002). Diese Daten, wie z.B. Datum, Uhrzeit oder der GPS-Koordinaten, werden als sogenannte Tags im Header des Bildes abgelegt. Der IPTC-NAA-Standard, welcher von der „International Press Telecommunications Council (IPTC)“ zusammen mit der „Newspaper Association of America (NAA)“ entwickelt wurde (Council, 2008), arbeitet nach einem ähnlichen Prinzip wie Exif. Auch hier lassen sich verschiedene Parameter als Metadaten an Bilder

anfügen. Mithilfe dieses Formats können Bilder u.a. mit dem Copyright sowie mit Schlagwörtern versehen werden.

Da innerhalb eines Fotobestandes häufig nicht alle Bilder mit solchen Daten versehen sind, gibt es Programme, die es ermöglichen, Fotos nachträglich mit Metadaten zu bestücken. Hierzu wurde in dieser Arbeit der Exifer¹⁵ verwendet. Dieses Programm bietet die Möglichkeit die Exif Daten sowie die IPTC-NAA Daten eines Bildes zu betrachten und ggf. zu ändern. Aufgrund dieser Funktionen ist das Programm sehr geeignet für diese Anwendung.

6.1.2 Auslesen der Metadaten

Um dem Anwender die möglichen Filterkriterien zu präsentieren und letztlich anhand dieser zu filtern, müssen die Metadaten der Fotos ausgelesen werden. Dies lässt sich in Java durch den Einsatz der Metadata-Extractor-Bibliothek¹⁶ realisieren. Diese bietet den Zugriff auf die Exif und IPTC-NAA Daten von Bildern, die in Form von JPEG-Dateien vorliegen. Zum Abfragen dieser Eigenschaften werden Methoden bereitgestellt, die es erlauben zu ermitteln, ob ein Bild bestimmte Tags enthält und wie diese aussehen.

6.1.3 Verwendete Metadaten

Die hier verwendeten Daten sollen eine möglichst große Auswahl an Filterkriterien ermöglichen. An dieser Stelle werden fünf verschiedene Eigenschaften zur Verfügung gestellt, zwischen denen der Anwender wählen kann:

Datum: Hier kann ein Anfangs- und ein Enddatum festgelegt werden. Dabei werden alle Bilder ausgewählt, die sich innerhalb dieses Zeitintervalls befinden.

Ort: Dieses Kriterium untersucht die Bilder hinsichtlich ihres Aufnahmeortes. Es kann sich hierbei sowohl um Städte als auch um andere Ortsbeschreibungen handeln.

Objekte: Diese Eigenschaft umfasst die Objekte, die sich auf dem Foto befinden. Wobei es sich hier um ein oder mehrere Elemente handeln kann.

Kategorie: Die Kategorie von Bildern beschreibt den Kontext der Bilder. Diesbezüglich ist es möglich eine Galerie mit z.B. allen geschossenen Urlaubsbildern zu generieren.

Fotograf: Das Attribut erlaubt es, alle Bilder, die von einer bestimmten Person geschossen wurden, auszulesen.

¹⁵<http://www.friedemann-schmidt.com/software/exifer/>

¹⁶<http://www.drewnoakes.com/code/exif/releases/>

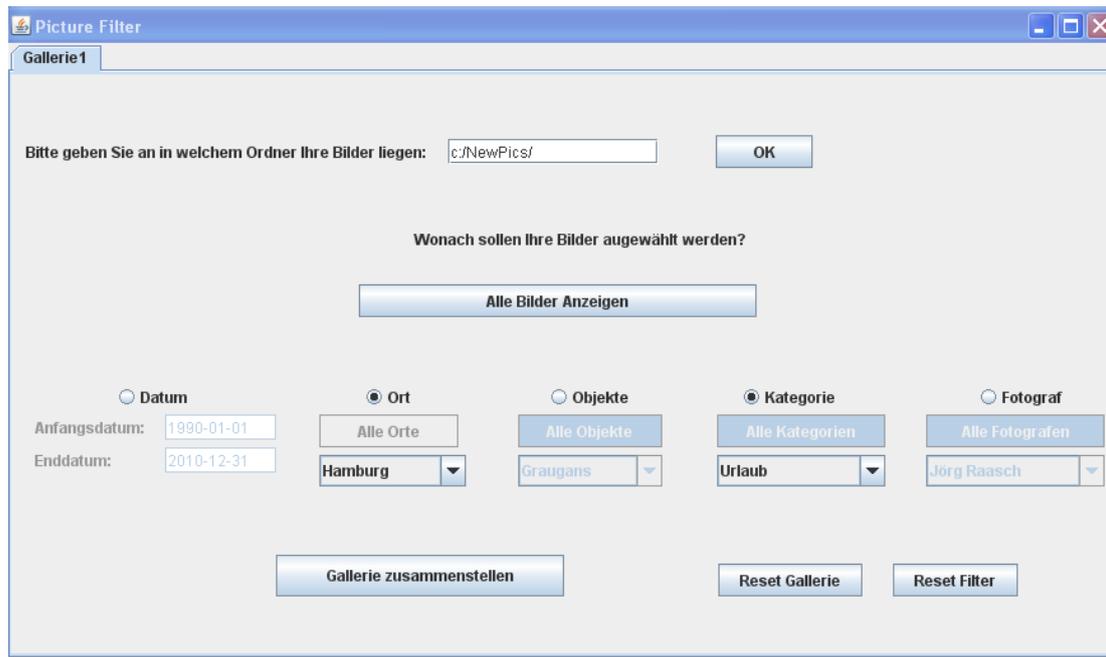


Abbildung 6.1: Darstellung des implementierten Filters

6.1.4 Arbeitsweise des Filters

Der hier vorgestellte Filter (siehe Abbildung 6.1) wurde durch mehrere kleine Entwicklungsschritte implementiert, mit dem Ziel dadurch im Laufe der Zeit um weitere Funktionalitäten erweitert zu werden.

Die Komponente wurde so entwickelt, dass sich die Interaktion in folgende Schritte gliedert:

1. **Ordner auswählen:** Bevor der Anwender die Filterkriterien wählen kann, muss der Ordner angegeben werden, in dem sich die zu filternden Fotos befinden. In diesem Kontext spielt es keine Rolle, ob sich in dem Ordner Bilder ohne Metadaten befinden, da Fotos ohne geeignete Daten während des Filtervorgangs nicht beachtet werden. Nachdem der Anwender den Pfad zum entsprechenden Ordner eingegeben hat und diesen über den OK-Button bestätigt, beginnt die Komponente im Hintergrund die Metadaten der Fotos auszulesen. An dieser Stelle werden alle Eigenschaften, die in den Bildern gefunden werden dem entsprechenden Kriterium zugeordnet, vor dem Hintergrund sie dem Benutzer innerhalb eines Drop-Down-Menüs anzuzeigen.
2. **Kriterien bestimmen:** An dieser Stelle wird entschieden, nach welchen Kriterien die Fotos gefiltert werden. Allerdings lassen sich auch alle vorhandenen Bilder anzeigen,

falls keine Sortierung der Bilder stattfinden soll.

Die verschiedenen Kriterien sind miteinander kombinierbar, sodass nur Bilder in die Galerie einfließen, die mehrere Attribute erfüllen. Jedes der fünf Filterkriterien bietet außerdem die Möglichkeit, alle Bilder in den dreidimensionalen Raum aufzunehmen, welche einen entsprechenden Tag gesetzt haben. Beispielsweise lassen sich alle Bilder darstellen, die das Attribut „Fotograf“ gesetzt haben.

Wenn der Anwender den Button „Galerie zusammenstellen“ betätigt, beginnt das Java-Programm im Hintergrund die Daten auszuwerten und kopiert den Pfad der Bilder, welche die Filterkriterien erfüllen in dafür vorhergesehene Textdateien. Diese Datei wird daraufhin von der View-Komponente ausgelesen, um die Bilder darzustellen. Folglich können diese Textdateien als Kommunikationsschnittstelle zwischen der View-Komponente und dem Model verstanden werden. Nach diesem Vorgang wird der Benutzer über die Anzahl der Bilder, die den Galerien hinzugefügt wurden, informiert. Sollte es keine Übereinstimmung mit den ausgewählten Kriterien geben, wird er ebenfalls entsprechend benachrichtigt.

3. **Reset:** Hierbei handelt es sich um einen optionalen Vorgang. Der „Reset Filter“-Button dient dazu, den Filter zurück in den Urzustand zu versetzen, um eine neue Angabe für eine Generierung der Galerien zu machen. Der „Reset Galerie“-Button dient dazu, die bereits aufgebauten Räume zu leeren. Nach Betätigung des Buttons befinden sich somit keine Bilder mehr den Galerien.

6.2 Darstellung

Für die Erstellung der Darstellungskomponente wurde eine Software-Plattform gesucht, die es ermöglicht, die mentalen Modelle des Anwenders innerhalb eines dreidimensionalen Raumes (3.3.5) und die Problemfelder, die bei der Gestaltung von Räumen dieser Art (3.3) entstehen, aufzugreifen und zu lösen.

In Betracht gezogen wurden dabei die Open-Source-Engine für 3D-Grafikdarstellung Ogre¹⁷ und die frei verfügbare Software Croquet¹⁸. Dabei stellte sich Croquet schnell als eine positive Lösung heraus, da in dieser Technologie bereits vorgefertigte dreidimensionale Landschaften sowie weitere 3D-Objekte zur Verfügung gestellt werden.

Da die Einarbeitung in Ogre sich als weitaus zeitintensiver herausgestellt hat, wird der Fokus für die weitere Realisierung auf Croquet gesetzt.

¹⁷<http://www.ogre3d.org/>

¹⁸http://www.croquetconsortium.org/index.php/Main_Page

6.2.1 Croquet

Croquet ist eine Software-Architektur, in der 3D Welten dargestellt werden können und deren Teilnehmer durch einen Avatar repräsentiert werden. Das Prinzip dieser Architektur liegt darin, dass ein System entsteht, das keine Grenzen besitzt (Smith u. a., 2003). Diesbezüglich kann ein Anwender innerhalb seiner eigenen digitalen dreidimensionalen Umgebung alles kreieren und verändern. Durch ein Ad hoc multi-user Netzwerk besteht die Möglichkeit, verschiedene Umgebungen miteinander zu verbinden, um eine vernetzte 3D-Welt zu generieren. Croquet spiegelt dabei in vielerlei Hinsicht das World Wide Web wider, indem auch hier eine eigene „Home World“ aufgebaut werden kann, welche dann in einer anderen Umgebung verlinkt wird oder selbst auf weitere Welten verweist. Demzufolge können Benutzer sich in ihrer eigenen Umgebung oder in jeder anderen Welt innerhalb des Netzwerkes bewegen. Die erstellten Welten sind dynamisch und somit zu jeder Zeit veränderbar. Diese Veränderungen müssen bei jedem Teilnehmer in der gleichen Reihenfolge ausgeführt werden und rekonstruierbar sein. Aus diesem Grund wird von Croquet eine eindeutige Zeit des Systems sichergestellt und entstandene Änderungen von einem Router synchronisiert, bevor sie an die angemeldeten Teilnehmer geschickt werden (siehe Abbildung 6.2). Die Grundidee dieser Architektur ist, dass jedes einzelne Objekt innerhalb des Systems gemeinschaftlich nutzbar gemacht wird.

Croquet ist in Squeak, einer modernen Variante der Programmiersprache Smalltalk, entwickelt worden und ist demzufolge ein rein objektorientiertes System. Die dreidimensionalen Welten werden diesbezüglich ebenfalls in Squeak-Smalltalk programmiert.

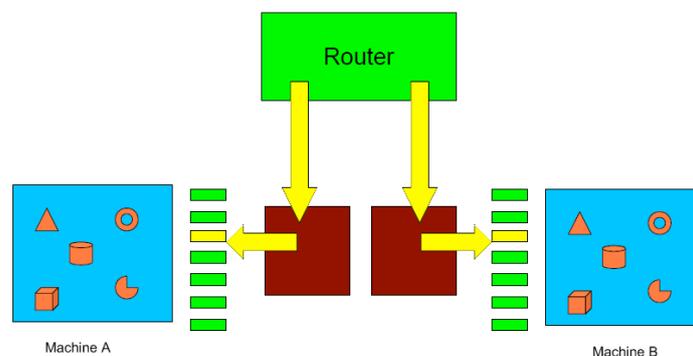


Abbildung 6.2: Synchronisation einer Veränderung innerhalb einer 3D Welt (Smith u. a., 2005)

Entwickeln mit Croquet

Bei der Entwicklung mit Croquet wird mit einem Image File gearbeitet, das alle Klassen und Objekte enthält. Des Weiteren besteht die Möglichkeit neue Images anzulegen und auf diesen zu arbeiten. Um die Audio- und Grafikerunterstützung zu erhalten, müssen auf dem Computer die OpenGL¹⁹ und die OpenAL²⁰ Bibliotheken installiert sein. Während der Entwicklung hat man jederzeit die Möglichkeit die dreidimensionale Umgebung zu öffnen, damit beispielsweise Auswirkungen von neuem Code überprüft werden können.

Croquet stellt bereits vorgefertigte Objekte wie dreidimensionale Fenster bereit, welche innerhalb eines Raumes positioniert werden können. Diese Fenster können so programmiert werden, dass sie Bilder anzeigen oder als ein sogenanntes Portal in eine weitere Umgebung dienen. Diese Portale entsprechen den bereits erwähnten Verlinkungen in eine andere Umgebung, wobei es sich sowohl um einen Link innerhalb der Home World als auch eine Verbindung zu einer Welt im Netzwerk handeln kann. Eine Auflistung weiterer Objekte sowie hilfreiche Code-Beispiele findet man in der Literatur „Croquet Programming - A Concise Guide“ (Smith u. a. (2006)).

6.2.2 Umsetzung der Galerien

In dem hier vorgestellten System wird die Anbindung an ein Netzwerk nicht weiter betrachtet, da sich aus den Szenarien (3.1) ableiten lässt, dass es sich hier um eine Home World handelt, zu der keine weiteren Teilnehmer Zugriff haben sollen. Dies ist damit zu begründen, dass es sich bei den Bildern um persönliche und vertrauliche Daten handelt, die ohne einer Erlaubnis keiner weiteren Person zur Verfügung stehen sollen.

Wie in den Szenarien beschrieben wird, soll es dem Anwender ermöglicht werden, beliebig viele Bilder in den Galerien darzustellen. Dies wird durch maximal fünf Galerien realisiert, welche alle aus der Hauptgalerie erreichbar sind und sich visuell voneinander unterscheiden.

Außerdem enthält jede Galerie jeweils zwei Verlinkungen, welche auf die benachbarten Räume verweisen. Dabei zeigt der erste Raum mit einem Link auf den letzten Raum und letzterer besitzt genauso eine Verbindung zum ersteren. Auf diese Weise ist es möglich, alle Räume zu betrachten, ohne zurück in die Hauptgalerie zu gehen, also im Kreis durch die Galerien zu navigieren. Allerdings besteht in jedem Raum die Möglichkeit über ein „Home-Portal“ zurück in die Hauptgalerie zu gelangen.

¹⁹<http://www.opengl.org/>

²⁰<http://www.openal.org/>

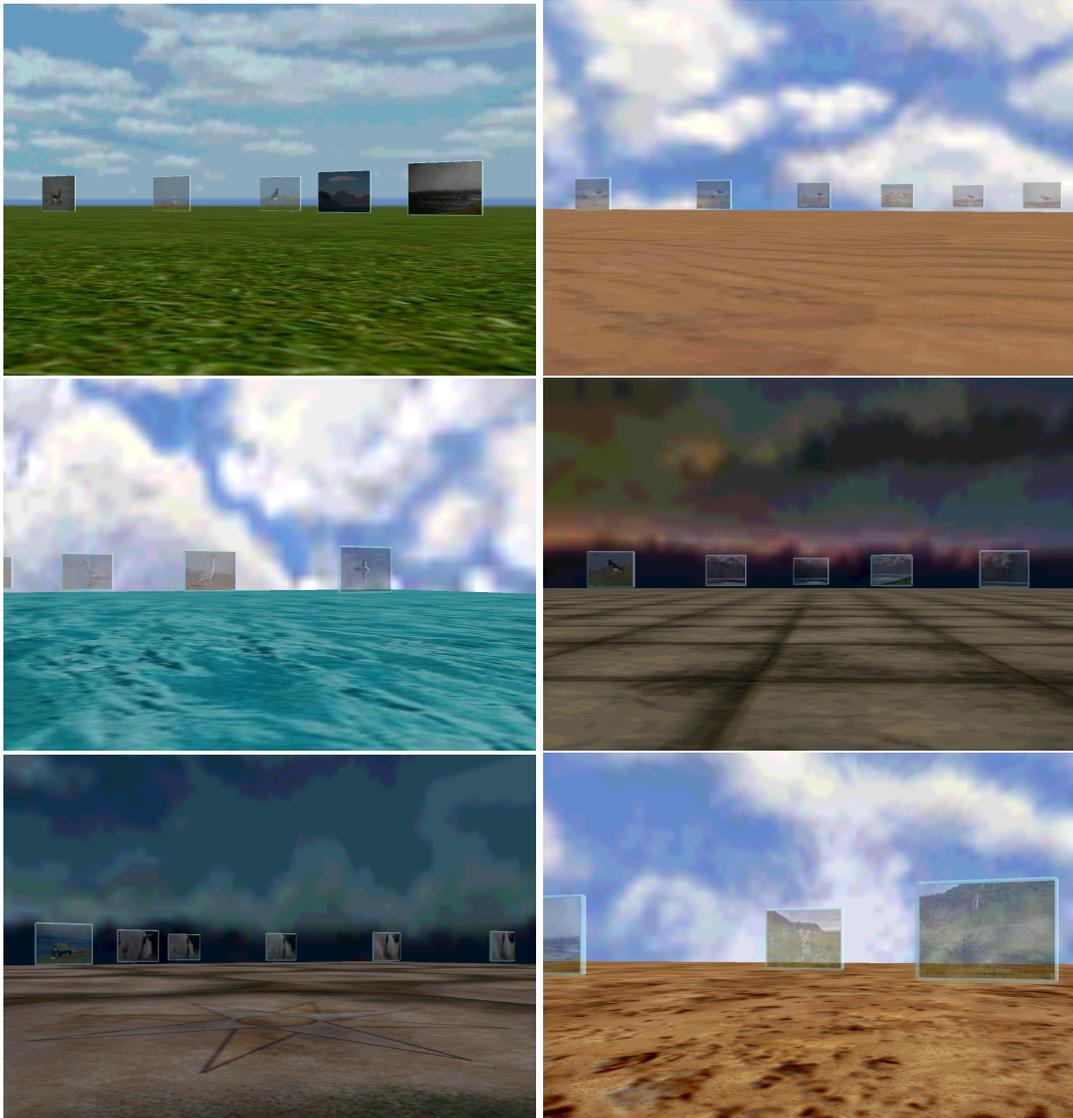


Abbildung 6.3: Darstellung der Hauptgalerie sowie der weiteren fünf Galerien

Kommunikation mit dem Model

Um die Kommunikation zwischen der View-Komponente und dem Model etwas genauer zu betrachten, wird an dieser Stelle die Arbeitsweise des Filters erneut aufgegriffen. In der Ordnerstruktur von Croquet ist für jede der möglichen fünf Galerien sowie der Hauptgalerie ein Ordner hinterlegt, in welche der Filter jeweils eine Textdatei mit den Pfaden der darzustellenden Bilder ablegt. Die Anzahl der Pfade pro Textdatei ist in diesem Zusammenhang von der Bildanzahl abhängig. Dazu wird die Zahl der relevanten Bilder durch fünf geteilt, damit eine Ausgeglichenheit der Fotos pro Galerie erreicht wird. Sollte es beispielsweise acht darzustellende Bilder geben, wird eine Galerie mit fünf und eine weitere mit drei Bildern generiert. Dabei wird bei jedem Wechsel zu einer weiteren Galerie eine Kopie des Pfades in die Datei der Hauptgalerie geschrieben.

Darstellung der Galerien

Croquet liefert bereits verschiedene Texturen und Bilder, die dem Entwickler ermöglichen, innerhalb kurzer Zeit einen Grundaufbau eines dreidimensionalen Raumes zu konstruieren, der daraufhin individuell gestaltet werden kann. Zur Generierung der Hauptgalerie wird per Stream die entsprechende Textdatei ausgelesen. Zur Speicherung der Pfade wird als Datentyp eine Liste verwendet, welche dynamisch wächst. Da in diesem Raum maximal fünf Portale existieren können, wird die Anzahl der Bilder diese Zahl nicht übersteigen. Zu jeder Galerie wird ein Portal generiert, das mit einem relevanten Foto repräsentiert wird. Ein Mausklick auf dieses Bild leitet den Ladevorgang der Galerie ein, die sich hinter dem Portal verbirgt. Nun kann der Anwender durch den Bilderrahmen, wie durch eine Tür in eine neue Umgebung, in die Galerie navigieren. Diese Bilderrahmen werden an verschiedenen Stellen im Raum angeordnet, sodass die dreidimensionale Fläche ausgenutzt wird.

Die weiteren Galerien können beliebig viele Bilder darstellen, allerdings leidet die Übersicht bei Galerien mit mehr als 20 Bildern. Zur Präsentation der Fotos liest jede Umgebung ihre entsprechende Datei aus und ordnet die Bilder im Raum an. Außerdem werden die Portale zu den benachbarten Räumen erstellt. Um diese in einer einheitlichen Form zu präsentieren, werden hier die Fotos der Hauptgalerie verwendet. In dieser Realisierung ist statisch eingetragen, welche benachbarten Räume eine Galerie besitzt, was zur Folge hat, dass bei weniger als fünf Galerien der Kreis nicht geschlossen wird.

Eine Nahansicht der Bilder wird über einen Pfeil über dem digitalen Bilderrahmen ermöglicht. Zur Verhinderung einer 3D-Amnesie ([3.3.1](#)), wird der Ansichtswchsel von einer Sequenz begleitet. Nachdem der Benutzer die Nahansicht des Objektes beendet hat, befindet er sich wieder im Navigationsmodus. Während der Navigation lässt sich die Perspektive jederzeit ändern. Hierbei kann der Benutzer zwischen der First-Person und den vorgestellten

Third-Person Perspektiven (5.4.3) wählen.

Die Erfahrungen mit Croquet haben ergeben, dass Bilder, die eine Auflösung von 640 x 480 Pixeln aufweisen, die Bilderrahmen vollständig ausfüllen, ohne Ränder zu verursachen. Da Croquet keine Funktion anbietet, um Bilder in eine einheitliche Größe zu bringen, muss diese Aufgabe von einer weiteren Software übernommen werden. Ein Programm, das sich hierfür anbietet, ist „Der Grandiose Bildverkleinerer²¹“. Hier muss lediglich derjenige Ordner, der die zu verändernden Bilder enthält sowie eine gewünschte Auflösung angegeben werden. Das Programm legt daraufhin einen Unterordner an, in dem die neuen Bilder abgelegt werden. In diesem Zusammenhang ist zu beachten, dass die neuen Bilder keine Exif und IPTC-NAA Informationen beinhalten und nachträglich eingefügt werden müssen.

6.3 Steuerung

Aufgrund der Ergebnisse und Erfahrungen, die aus den erstellten Prototypen (5.6) entstanden sind, werden für die bewegungsbasierte Steuerung dieser Anwendung das Wii Balance Board und die Wii Remote verwendet.

Wii Balance Board

Das Wii Balance Board besitzt in dieser Anwendung die Aufgabe, die Steuerung des Avatars zu übernehmen, um die Navigation durch die erstellten Galerien als intuitiv zu gestalten. Dabei soll, wie in den grundlegenden Funktionen des Systems (3.2.3) definiert ist, die X-Achse dazu dienen, sich auf dieser zu drehen. Dies hat zur Folge, dass die Laufrichtungen auf vorwärts und rückwärts begrenzt sind. Die weiteren Richtungswechsel werden dementsprechend durch Drehungen realisiert.

Da die WiimoteLib 1.5.2 (Peek, 2008) eine Bibliothek für C# ist, wird die Funktionalität der Eingabegeräte in dieser Sprache programmiert. Weil die Werte der Drucksensoren des Boards nicht exakt Null betragen, wenn sich niemand auf diesem befindet, sondern sich permanent im Grammbereich ändern, würde das implementierte Programm in diesem Fall von einer Zustandsänderung ausgehen und entsprechende Reaktionen einleiten. Aufgrund dessen wird erst auf die Werte reagiert, falls das errechnete Gewicht mehr als 10 kg aufweist, mit dem Ziel sicherzustellen, dass sich jemand auf dem Board befindet.

Damit die Anwendung individuell auf verschiedene Personen, also unabhängig vom Gewicht reagieren kann, wird hier mit relativen Gewichtsveränderungen gearbeitet. Das bedeutet, das Gewicht des einzelnen Anwenders wird verwendet, um auf seine Gewichtsverlagerungen zu schließen. Dabei wird davon ausgegangen, dass der Anwender sich nach vorne

²¹<http://www.foto-freeware.de/der-grandiose-bildverkleinerer.php>

lehnt, falls sich das Gewicht auf den beiden vorderen Drucksensoren um das 2,8-fache des auf dem Board befindlichen Gewichts erhöht. Diese Annahme wird ebenso für hinten, rechts und links verwendet.

Die Abbildung der Daten in Bewegungen innerhalb der Croquet-Anwendung verlangt eine Simulation eines Tastendrucks für jede der vier Bewegungsmöglichkeiten. Dabei kann es zu Komplikationen führen, falls der Anwender sich für längere Zeit in eine bestimmte Richtung neigt, um an einen gewünschten Ort zu navigieren. In diesem Fall werden zu viele Events an das System gesendet, sodass diese noch abgearbeitet werden, während der Benutzer bereits eine neue Position eingenommen hat. Da diese verzögerten Reaktionen zu einem Abbruch des Interaktionsflusses führen und somit die Intuitivität der Steuerung verloren geht, werden „Sleep-Times“ implementiert, in denen die Steuerungskomponente keine Daten des Wii Balance Board aufnimmt. Da es sich hierbei um 120 Millisekunden handelt, wird der Benutzer nicht in der Interaktion mit dem System beeinträchtigt und es werden nur Events gesendet, die direkt verarbeitet werden können.

Wii Remote

Die Wii Remote soll in dieser Applikation alle weiteren elementaren Interaktionsmöglichkeiten zur Verfügung stellen, welche nicht vom Wii Balance Board abgedeckt werden. Hierbei handelt es sich um das Anvisieren von Objekten sowie Mausclicks, mit der Absicht ein Portal zu öffnen bzw. ein Bild heranzuzoomen.

Die WiimoteLib bietet die Möglichkeit, innerhalb eines Programms, die Werte mehrerer Wii-Geräte zu empfangen. Aus diesem Grund konnten die Aufgaben der Wii Remote für die Croquet Steuerung in die bereits bestehende Quellcode-Datei für die Steuerung mit dem Wii Balance Board implementiert werden. Zum Anvisieren der Bilder müssen Mausbewegungen realisiert werden, damit der Anwender die Möglichkeit erhält, verschiedene Bilder im Raum anzuwählen. Für diese Steuerung wurden drei Geschwindigkeitsstufen des Mauszeigers realisiert, welche vom Neigungsgrad des Eingabegerätes abhängen. Die empfangenen Werte des in der Wiimote verbauten 3-Achsen Beschleunigungssensors können zwischen Bewegungen auf der X-Achse sowie der Y-Achse unterschieden werden. Die Werte liegen zwischen -1 und +1, wobei diejenigen im Minusbereich eine Neigung nach links bzw. oben und im Plusbereich nach rechts bzw. unten repräsentieren.

Die Mausbewegungen werden anhand der aktuellen Cursorposition errechnet, indem eine von der Geschwindigkeitsstufe abhängige Pixelanzahl addiert bzw. subtrahiert wird. Dadurch entsteht für den Anwender die Möglichkeit schnell neue Bilder zu fixieren oder den Mauszeiger langsam zu bewegen, wenn er präzise Eingaben macht.

Damit Portale geöffnet und Bilder anvisiert werden können, müssen neben den Mausbewegungen auch Mausclicks simuliert werden. Hierzu dienen die Buttons A und B des Controllers. Falls der Anwender den Cursor auf einem Portal oder einem Pfeil über dem Bild

platziert hat, genügt ein Klick auf den A-Button, um eine Reaktion des Systems auszulösen, also entweder ein Portal zu öffnen oder ein Bild heranzuzoomen. Dabei entstehen an dieser Stelle ähnliche Probleme wie im ersten Teil der Steuerung. Hier müssen ebenfalls Sleep-Times eingebaut werden, mit der Absicht nicht zu viele Events an das System zu senden. Sollte der Anwender sich in der Nahansicht eines Fotos befinden, gelangt er durch die Betätigung des B-Buttons jederzeit zurück in den Navigationsmodus.

Dieses Programm sollte erst geöffnet werden, wenn die entsprechenden Geräte über die Bluetooth-Schnittstelle eingebunden sind, da diese Applikation nur Daten von Geräten empfangen kann, die bereits beim Start erreichbar sind. Nachdem die Steuerungskomponente gestartet ist, öffnet sich eine Anzeige, die den Anwender über die Anzahl der integrierten Wii-Geräte informiert.

6.4 Evaluation

Um die realisierte Anwendung auf Funktionalität und Intuitivität zu überprüfen, konnten immer wieder Testläufe mit den Besuchergruppen des iFlat der HAW Hamburg durchgeführt werden. Darüber hinaus wurde im Usability Labor ein systematischer Test der Benutzbarkeit vorgenommen.

6.4.1 Bewertung der Umsetzung

Die hier vorgestellte Applikation erfüllt mit Ausnahme der Speicherung von Suchanfragen alle grundlegenden Funktionen (3.2.3), welche aus den funktionalen und nicht-funktionalen Anforderungen ermittelt wurden. Der Aspekt der Anfragenspeicherung wurde nicht verfolgt, da der Fokus auf dem Vorgang des Auslesens sowie der Bereitstellung der Daten lag.

Das Problem der 3D Amnesie (3.3.1) konnte dadurch umgangen werden, indem das Heranzoomen von einer Sequenz begleitet wird und dieser Modus per Knopfdruck auf der Wii Remote oder durch eine Gewichtsverlagerung beendet werden kann.

Die im Design vorgestellten Komponenten konnten entsprechend ihrer gestellten Aufgaben implementiert werden. Durch den Einsatz des MVC-Patterns kann der Controller um weitere Interaktionstechniken zur Steuerung erweitert werden, ohne Änderungen an den übrigen Komponenten zu erfordern.

Infolge des Prototyping konnten schon frühzeitig Erfahrungen über bewegungsbasierte Steuerungen gesammelt werden, die bei der weiteren Realisierung der Steuerung sehr hilfreich waren. Reaktionen auf die Benutzerfreundlichkeit der entwickelten Anwendung werden im weiteren Verlauf dieses Kapitels beschrieben.

6.4.2 Besuchergruppen

Aufgrund der Vorführungen der Applikation für die Besuchergruppen des iFlat konnte beobachtet werden, wie Benutzer mit dem System interagieren. Eine wichtige Erkenntnis war, dass die meisten Personen den Umgang mit dem Wii Balance Board ohne eine Einführung beherrschten und schnell verstanden haben, welche Bewegung die gewünschte Reaktion im System auslöst. Da dies auch in den Prototypen der Fall war, lässt sich daraus schließen, dass die umgesetzten Bewegungen sehr nah an den mentalen Modellen der Anwender liegen. Bei der Wii Remote benötigten ein Großteil der Benutzer eine kurze Einweisung in die Bedienung des Gerätes. Hier handelte es sich häufig um eine kurze Erläuterung über Bewegungen des zu steuernden Mauszeigers in rechter und linker Richtung, da hierfür der Controller in die entsprechende Richtung gedreht werden muss. Nach einem kurzen Hinweis wurde diese Funktion sehr schnell angenommen und bereitete während der weiteren Navigation keine Irritationen.

6.4.3 Usability-Test

Im Usabilitylabor der HAW können Anwendungen mithilfe technischer Hilfsmittel auf ihre Benutzbarkeit untersucht werden. Dabei sollen Erkenntnisse gesammelt und aufgezeichnet werden, die während einer einfachen Beobachtung des Anwenders verloren gehen könnten. Zur Durchführung dieser Tests, ist das Labor in zwei Bereiche aufgeteilt (siehe Abbildung 6.4). Im Testraum findet die Untersuchung des Testobjektes durch den Probanden statt, während er aus verschiedenen Kameraperspektiven gefilmt wird. Über Mikrofone ist die Testperson über die gesamte Testphase mit den Beobachtern verbunden und kann somit verbal seine positiven und negativen Eindrücke zum Ausdruck bringen. In dem zweiten Bereich des Labors, dem Regieraum, beobachten die Testleiter die Ereignisse über Monitore und können über Lautsprecher von außen Kontakt zum Probanden aufnehmen. Des Weiteren stehen zusätzliche Anwendungen zur Verfügung, mithilfe derer weitere Informationen für die Auswertung gesammelt werden können.

Aufgrund dieser Struktur lassen sich Defizite der getesteten Anwendung leichter erkennen, da die Aufzeichnungen auftretende Probleme während des Testablaufs deutlich machen und wiederholt werden können, bis eine genaue Interpretation möglich ist.



Abbildung 6.4: links: Testraum, rechts: Regieraum des Usabilitylab

Durchführung der Usability-Tests

Für die Durchführung des Tests wurden verschiedene Aufgaben definiert, deren Erfüllung einem Testdurchlauf entspricht. Diese Aufgaben wurden von sechs Probanden bearbeitet, während für jede Bearbeitung eine Zeitmessung durchgeführt wurde, mit dem Ziel Aussagen darüber zu treffen, an welcher Stelle Irritationen und Probleme beim Anwender aufgetreten sind.

Im folgenden Abschnitt werden die einzelnen Aufgaben beschrieben sowie die Intuitivität ihrer Ausführung anhand der gemessenen Zeiten analysiert:

1. Sie stehen zu Beginn auf einer Wiese und um Sie herum sehen Sie verschiedene Bilder. Wie viele Bilder sind zu erkennen und was ist auf ihnen zu sehen?
2. Suchen Sie nun das Bild heraus, auf dem ein Pferd dargestellt ist. Versuchen Sie in die dahinter gelegene Galerie zu gelangen.
3. Versuchen Sie ein Bild Ihrer Wahl heranzuzoomen, damit es fast den gesamten Bildschirm ausfüllt. Was ist auf dem Bild zu erkennen?
4. Als nächstes versuchen Sie diesen Modus aufzuheben, um wieder in den Navigationsmodus zu wechseln.
5. Versuchen Sie nun in die Galerie zu gelangen, die sich rechts neben dieser befindet. Visieren und klicken Sie dazu Bilder an, hinter der Sie eine Galerie vermuten.
6. Zum Schluss versuchen Sie zurück in die Ausgangswelt zu gelangen.

Auf der Abbildung 6.5 wird dargestellt, wie lange jeder Proband für eine Aufgabe benötigt hat. Dabei kennzeichnen die Ausrufezeichen besonders erwähnenswerte Daten. Diese gemessenen Zeiten geben Aufschluss darüber, ob einzelne Aufgaben für die Probanden

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Aufgabe 6	Summe Durchschnitt Erfolgsquote
TP1	erfolgreich 1:20 Min	erfolgreich 0:57 Min	erfolgreich 0:32 Min	erfolgreich 0:18 Min	erfolgreich 2:50 Min	erfolgreich 0:43 Min	6:40 1:06 100%
TP2	erfolgreich 1:36 Min	erfolgreich 0:59 Min	erfolgreich 0:54 Min	erfolgreich 0:18 Min	erfolgreich 3:16 Min	erfolgreich 1:36 Min	8:39 1:26 100%
TP3	erfolgreich 1:01 Min	erfolgreich 0:39 Min	erfolgreich 2:23 Min	nicht erfolgreich 0:40 Min	erfolgreich 4:00 Min	erfolgreich 0:43 Min	9:26 1:34 83%
TP4	erfolgreich 0:57 Min	erfolgreich 0:30 Min	erfolgreich 0:51 Min	erfolgreich 0:06 Min	erfolgreich 1:41 Min	erfolgreich 0:27 Min	4:32 0:45 100%
TP5	erfolgreich 1:12 Min	erfolgreich 0:46 Min	erfolgreich 0:19 Min	erfolgreich 0:07 Min	erfolgreich 0:35 Min	erfolgreich 0:40 Min	3:39 0:36 100%
TP6	erfolgreich 1:05 Min	erfolgreich 0:38 Min	erfolgreich 0:20 Min	erfolgreich 0:14 Min	erfolgreich 1:22 Min	erfolgreich 0:23 Min	4:02 0:40 100%
Summe Durchschnitt Standardabweichung Erfolgsquote	7:11 1:11 0:14 100%	4:29 0:44 0:11 100%	5:19 0:53 0:46 100%	1:43 0:17 0:12 83%	13:44 2:17 1:17 100%	4:32 0:45 0:26 100%	

Abbildung 6.5: Auflistung der benötigten Zeit jeder Testperson pro Aufgabe

wesentliche Schwierigkeiten bereitet haben und somit nicht intuitiv zu lösen waren. Dabei konnten die meisten Aufgaben in einem angemessenen Zeitrahmen gelöst werden. Auffällig ist dabei, dass bei der 3. Aufgabe ein großes Zeitintervall zwischen den einzelnen Testpersonen zu erkennen ist. Daraus lässt sich schließen, dass nicht jeder Anwender problemlos das Symbol zum Zoomen identifizieren konnte. Deshalb sollte in zukünftigen Systemen ein eindeutiges Symbol, wie z.B. einer Lupe, für diese Funktion gewählt werden. Ein weiteres großes Intervall ist bei der 5. Aufgabe zu beobachten. Aus den Reaktionen der Probanden war zu erkennen, dass es nicht eindeutig ist, hinter welchen Bildern sich eine Galerie befindet. Aus diesem Grund gilt es zukünftig zu untersuchen, wie man die Galerien visuell von den übrigen Bildern unterscheidet. Ansätze dazu wären, diese in der Größe der Darstellung voneinander zu unterscheiden oder ein signifikantes Bild für jeden Eingang in eine Galerie zu wählen.

Nach Erfüllung der Aufgaben wurde mit den Testpersonen ein Nachgespräch geführt, welches aus vier Kernfragen besteht:

1. Wie hat Ihnen die Anwendung gefallen?

Diese Frage wurde von allen Probanden positiv beantwortet. Dabei wurde der Einsatz der beiden Eingabegeräte als neue Form der Navigation durch Informationen als sehr interessanter Ansatz empfunden.

2. Was ist Ihnen während des Testdurchlaufs positiv bzw. negativ aufgefallen?

Neben den neuen Interaktionsmöglichkeiten wurde die Darstellung von zweidimensionalen Objekten innerhalb einer 3D-Landschaft als sehr positiver Ansatz bezeichnet. Dabei war für einige Probanden die Vorstellung besonders reizvoll, ihre eigenen Bil-

derbestände in den Galerien betrachten zu können. Außerdem wurde die Möglichkeit durch Bilder zu gehen, um in eine weitere Galerie zu gelangen als spannend empfunden.

Als negativer Aspekt wurde erwähnt, dass das System nicht auf die Eingaben beider Eingabegeräte gleichzeitig reagieren kann. Dies liegt an der Aufteilung der klassischen Maus auf zwei Geräte und somit entweder auf die Eingaben der Wii Remote zur Steuerung des Mauszeigers oder des Wii Balance Board zur Navigation im Raum reagiert wird. Zukünftig wäre es neben dem Einsatz komplexerer Eingabegeräte ein Ansatz, verschiedene Modi einzusetzen, in denen jeweils verschiedene Geräte aktiv sind, mit dem Ziel dem Anwender zu verdeutlichen, welche Interaktionsmöglichkeiten zum aktuellen Zeitpunkt zur Verfügung stehen.

3. Wie empfanden Sie die Interaktion mit den Eingabegeräten?

Die Form der Interaktion wurde als sehr positiv empfunden. Dabei fiel den Testpersonen die Navigation innerhalb des Raumes mithilfe des Wii Balance Board sehr leicht. Lediglich eine Person hatte anfängliche Schwierigkeiten darauf zu schließen, wie sie sich auf dem Board bewegen muss, um eine Reaktion des Systems auszulösen. Diese Irritation konnte allerdings schnell und ohne weitere Hilfestellungen behoben werden. Die Steuerung des Mauszeigers über die Wii Remote bereitete den Probanden ebenfalls keine Probleme. Allerdings wurde von einigen Personen angemerkt, dass die Drehbewegung zur Bewegung des Cursors auf der X-Achse anfänglich nicht direkt einleuchtend war. Nachdem diese Funktion bemerkt wurde, bereitete sie während der weiteren Interaktion mit dem System keine Probleme.

4. Haben Sie Verbesserungs- bzw. Erweiterungsvorschläge für das System?

An dieser Stelle wurde angeregt, per Knopfdruck eine Bilderwand aller im Raum befindlichen Bilder zu erhalten, welche sich bei einem weiteren Knopfdruck wieder im Raum verteilen. Außerdem wurde hier erwähnt, dass die Galerien explizit gekennzeichnet werden sollten, mit der Absicht deutlich zu machen, hinter welchen Bildern sich weitere Räume befinden.

Die Abbildung 6.6 bietet einen Zusammenschnitt der einzelnen Kameraaufnahmen bei einem Testdurchlauf. Darauf ist eine der Testpersonen zu erkennen, die die hier vorgestellte Applikation steuert. Dabei zeigt eine Kamera, wie die Navigation innerhalb der Galerie aussieht, während die anderen Aufnahmen die Interaktion mit den beiden Eingabegeräten zeigen.

Da der Umgang mit der Ausrüstung des Usability Labors eine intensive Einarbeitungszeit benötigt, möchte ich mich an dieser Stelle bei Boris Klengel und Stefan Richter für ihre tatkräftige Unterstützung während des Usability-Tests bedanken. Ihre Erfahrung im Umgang mit den Geräten war sehr hilfreich bei der Durchführung der Tests.



Abbildung 6.6: Testperson bei der Interaktion mit der Anwendung

6.5 Auswertung

In dem hier vorgestellten System ist es gelungen, die im Design vorgestellte Architektur sowie viele Anforderungen aus der Analyse umzusetzen. Der Einsatz des MVC-Entwurfsmusters ermöglicht es, weitere Applikationen zu bauen, die zusätzliche Interaktionsmöglichkeiten bieten und auf ihre Intuitivität hin untersucht werden können.

Die Problemfelder, welche sich bei der Gestaltung dreidimensionaler Räume ergeben, wurden hier aufgegriffen und konnten zu einem großen Teil zufriedenstellend gelöst werden. Darüber hinaus stellte sich Croquet schnell als eine geeignete Plattform heraus, da diese bereits viele Funktionen sowie entsprechende Möglichkeiten mitbringt, individuelle Anwendungen zu konstruieren.

Anhand der vielen Testläufe der Prototypen, welche mit verschiedenen Besuchergruppen durchgeführt wurden, konnte ein frühzeitiges Feedback der Steuerung und zusätzliche Anregungen für die Implementierung einer intuitiv bedienbaren Anwendung gesammelt werden.

Aufgrund dieser Reaktionen und weiterführenden Gesprächen konnte eine Steuerungskomponente entwickelt werden, welche ebenfalls von verschiedenen Besuchern getestet wurde. Eine systematische Usability-Analyse hat darüber hinaus mit Mess- und Beurteilungsverfahren dazu beigetragen, die Benutzerfreundlichkeit und die Bedienung zu evaluieren sowie Ansätze für zukünftige Verbesserungen des Systems geliefert. Basierend auf diesen Rückmeldungen der Benutzer kann darauf geschlossen werden, dass die Steuerungskomponente eine Seamless Interaction mit dem System ermöglicht.

Die hier erstellte Applikation soll dem Anwender die Navigation innerhalb seiner Bilderbestände erleichtern, indem eine benutzerfreundliche Interaktion mit dem System sowie eine einfache Handhabung im Filtern der Datenbestände geboten wird.

Bei bewegungsbasierten Computersteuerungen handelt es sich um ein neuartiges Themenfeld, in dem der Entwicklungsprozess einer Computerinteraktion dieser Art noch nicht abgeschlossen ist. Damit eine Verbesserung dieses Systems sowie eine weitere Vertiefung dieses Themas erreicht werden kann, sind weiterführende Arbeiten in diese Richtung notwendig.

7 Schluss

7.1 Zusammenfassung

Im Rahmen dieser Arbeit wurde eine Anwendung entwickelt, die es ermöglicht mit Metadaten versehene Bilder anhand verschiedener Eigenschaften zu filtern. Außerdem präsentiert eine weitere Komponente die relevanten Bilder in dreidimensionalen Galerien, durch welche man mithilfe einer externen Steuerung navigiert. Zur Navigation reichen Gewichtsverlagerung des Körpers sowie Handbewegungen aus, um eine Reaktion des Systems auszulösen.

Zur Beschreibung des Anwendungskontextes dieser Arbeit, wurden im Grundlagen-Kapitel (2) verschiedene Begriffe erläutert, die zu einem besseren Verständnis des Themas und des weiteren Verlaufs beitragen.

In der Analyse (3) wurden zwei Szenarien definiert, aus welchen sich die Anforderungen an ein System ableiten lassen. Dabei wurden diese in funktionale und nicht-funktionale Anforderungen aufgeteilt. Anhand dieser konnten somit grundlegende Funktionen definiert werden, welche von der zu implementierenden Anwendung erfüllt werden sollten. Hier konnten drei Kernkomponenten herausgearbeitet werden, aus denen ein System dieser Art bestehen sollte. Es handelt sich hierbei um einen Filter, einer Darstellungskomponente sowie einer externen Steuerung.

Die letztere Komponente sollte in diesem Zusammenhang als eine bewegungsbasierte Steuerung implementiert werden. Aus diesem Grund befasst sich das Kapitel Bewegungen (4) intensiv mit diesem Begriff und stellt weiterhin einige Ansätze zur Interpretation von Bewegungen heraus.

Da die Darstellungskomponente dreidimensionale Räume generieren soll, wurden in der Analyse verschiedene Problemfelder, welche bei der Gestaltung solcher Räume auftreten können und zu berücksichtigen sind, beschrieben. Diese werden im Design (5) aufgegriffen und durch verschiedene Lösungsansätze behoben. Dazu wird u.a. die Anordnung von Bildern sowie die Perspektive des Betrachters genauer untersucht. Weiterhin wurde im Design mit dem MVC-Pattern eine geeignete Architektur gefunden, die in Kombination mit dem Observer Pattern und dem Singleton Pattern angewendet werden sollte. Außerdem wurde im Hinblick auf eine testgetriebene Entwicklung eine Machbarkeitsstudie anhand zwei implementierter Prototypen durchgeführt. An dieser Stelle wurden mit dem Wii Balance

Board, der Wii Remote und dem Neural Pulse Actuator drei Eingabegeräte evaluiert, wobei sich die beiden ersteren als die zu verwendenden Geräte herausgestellt haben.

Im Realisierungsteil (6) wurden viele Anforderungen aus der Analyse und dem Design in eine konkrete Anwendung umgesetzt. Dazu wurde in diesem Abschnitt die Funktionalität der einzelnen Komponenten sowie deren Realisierung diskutiert. In der anschließenden Evaluation konnten aufgrund der Reaktionen vieler Besuchergruppen sowie eines systematischen Usability-Tests Schlüsse auf die Intuitivität der Steuerung gezogen und verschiedene Anregungen für die Weiterentwicklung des Systems gesammelt werden.

Diese Anregungen werden im folgenden Ausblick erneut aufgegriffen. Außerdem wird ein Blick in die Zukunft von multimodalen Interaktionen geworfen.

7.2 Ausblick

7.2.1 Erweiterung des Systems

Mit dieser Arbeit konnte ein Ansatz zur intuitiven Computersteuerung erarbeitet werden. Anhand von Reaktionen der verschiedenen Testpersonen und den Beobachtungen, welche während der Steuerung gemacht wurden, lassen sich einige Verbesserungen und Erweiterungen für das System definieren.

Wie bereits in den grundlegenden Funktionen des Systems erwähnt wurde (3.2.3), sollte die Anwendung die Suchanfragen speichern. Dadurch wird es dem Anwender ermöglicht, gestellte Anfragen erneut an das System zu stellen, ohne die Anfrage ein wiederholtes Mal zu definieren. Dazu würde sich ein intelligenter Ordner anbieten (3.2.1), der Änderungen im Datenbestand bemerkt und diese Informationen bei der nächsten Darstellung der Ergebnisse miteinbezieht. So kann sichergestellt werden, dass der Benutzer zu jeder Zeit die aktuell relevanten Daten erhält, ohne die Anfrage ändern zu müssen.

In dieser Anwendung lassen sich Bilder eingrenzen, damit nur relevante Daten innerhalb eines dreidimensionalen Raumes präsentiert werden. Ein interessanter Ansatz wäre das Abrufen weiterer Informationen zum Bild per Handbewegung. Besonders im Hinblick auf das Szenario „Interaktives Museum“ (3.1.1) könnten die Besucher durch eine einfache Bewegung Informationen über den Maler, die Entstehung oder weitere Beschreibungen abrufen. Ein weiterer Ansatz wäre, auf den Rückseiten der Bilder zusätzliche Informationen zu positionieren.

Bei den dargestellten Daten handelt es sich in dieser Applikation ausschließlich um Fotos. Dabei ist es zukünftig denkbar, diverse Informationen in den 3D Landschaften zu

präsentieren. Dabei könnten diese Räume den klassischen Desktop eines heutigen Betriebssystems ablösen. Hier könnten die Galerien als Ordner dienen und die einzelnen Bilder würden als Portal in eine installierte Anwendung des Systems dienen.

Wenn das System verschiedene Informationen bereitstellt, würde sich der Einsatz einer dynamischen Bewegungsbibliothek anbieten. Hier könnte der Anwender u.a. neue Bewegungen und Gesten hinzufügen sowie nicht mehr benötigte Funktionen entfernen.

Weitere Ansätze für Erweiterungen des Systems haben sich während des Usability-Tests aufgrund der Gespräche mit den Testpersonen ergeben. Ein Aspekt dabei ist, verschiedene Modi für die Navigation einzurichten. Dabei sollten in den einzelnen Modi verschiedene Funktionen und Eingabegeräte zur Verfügung stehen. An dieser Stelle wäre es bedeutsam, dem Anwender zu visualisieren, in welchem Modus er sich zur Zeit befindet.

Außerdem könnten die für die Wii Remote implementierten Geschwindigkeitsstufen für das Bewegen des Mauszeigers auch auf das Wii Balance Board übertragen werden. Hier könnte die Stärke der Gewichtsverlagerung ausschlaggebend dafür sein, in welchem Tempo durch die Räume navigiert wird. Weiterhin könnte in zukünftigen Systemen die bereits in der Evaluation erwähnte Bilderwand implementiert werden (6.4.3). Dabei würde die Möglichkeit entstehen, alle im Raum enthaltenen Bilder an einer Wand zu betrachten und per Knopfdruck wieder im Raum zu verteilen.

7.2.2 Fazit

Die hier vorgestellte Applikation hat gezeigt, wie sich bewegungsbasierte Steuerungen zur Navigation innerhalb von Informationsbeständen einsetzen lassen. Diese Form der Kommunikation zwischen Mensch und Computer ist nur ein einzelner Bereich der multimodalen Interaktion. Dabei zeigt die steigende Anzahl an Veröffentlichungen und Forschungsbereichen den Bedarf an Systemen, die eine solche Interaktionsform anbieten.

Allerdings entstehen hier Fragestellungen, deren Antworten essentiell für den zukünftigen Erfolg dieser Interaktionsform sein werden: Wie wird sich die von Weiser beschriebene Schnittstelle ins Unterbewusstsein des Menschen und die daraus folgende natürliche Benutzung von Computern in den nächsten Jahren weiterentwickeln? Und welche Rolle spielen multimodale Interaktionen in dieser Entwicklung?

Zum jetzigen Zeitpunkt ist davon auszugehen, dass menschliche Bewegungen auch weiterhin ein wichtiger Bestandteil zur Vereinfachung der Computersteuerung sein werden. Dabei könnten in Zukunft Gewichtsverlagerungen in einem intelligent Home dazu dienen, den aktuellen Kontext zu erfassen: Sollte sich ein Mensch bei der Betrachtung eines Objektes auf einem Bildschirm nach vorne lehnen, könnte die aktuelle Darstellung vergrößert werden, da sich hieraus schließen lässt, dass die aktuelle Größe des Objektes zu klein ist.

Außerdem wird neben den Techniken zur Computersteuerung stetig an weiteren Technolo-

gien geforscht, in denen die Interpretation menschlicher Bewegungen durch Computer eine tragende Rolle spielt. Hierzu gehört u.a. das Performance-Capture, bei der neben Körperbewegungen auch Gesichtsausdrücke des Menschen aufgezeichnet werden, um sie auf Computer generierte 3D-Modelle zu übertragen. Zudem werden in der Medizin Technologien entwickelt, mit dem Ziel anhand von computergestützten Bewegungsanalysen z.B. Gleichgewichtsstörungen oder Gangstörungen zu diagnostizieren.

Aufgrund der verschiedenen Technologien, in denen menschliche Bewegungen eine tragende Rolle spielen, ist zu erwarten, dass auch in Zukunft viele interessante Geräte und Forschungsprojekte in diesem Bereich auftreten werden. Dabei wird weiterhin das Ziel verfolgt werden, die Allgegenwärtigkeit der Computer im menschlichen Alltag immer weiter Realität werden zu lassen.

Literaturverzeichnis

- [Agarawala und Balakrishnan 2006] AGARAWALA, Anand ; BALAKRISHNAN, Ravin: Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In: *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA : ACM, 2006, S. 1283–1292. – ISBN 1-59593-372-7
- [Beck und Andres 2004] BECK, Kent ; ANDRES, Cynthia: *Extreme Programming Explained: Embrace Change*. (2004). ISBN 0321278658
- [BMW 2008] BMW: *BMW Silverlight*. 2008. – URL <http://www.silverlight.bmw-tv.de.axprod.net/frontend/app/Default.html>. – Zugriffsdatum: 27.02.2009
- [Bowman u. a. 1999] BOWMAN, Doug A. ; JOHNSON, Donald B. ; HODGES, Larry F.: Test-bed evaluation of virtual environment interaction techniques. In: *VRST '99: Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA : ACM, 1999, S. 26–33. – ISBN 1-58113-141-0
- [Center 2008] CENTER, Ubiquitous I.: *T-Engine that supports a ubiquitous society*. 2008. – URL http://www.uidcenter.org/english/ubi_te.html#. – Zugriffsdatum: 27.02.2009
- [Cooliris 2008] COOLIRIS, Inc.: *Cooliris*. 2008. – URL <http://www.cooliris.com/>. – Zugriffsdatum: 27.02.2009
- [Council 2008] COUNCIL, International Press T.: *IPTC Standard Photo Metadata 2008*. 2008. – URL http://www.iptc.org/std/photometadata/2008/specification/IPTC-PhotoMetadata-2008_2.pdf. – Zugriffsdatum: 27.02.2009
- [Craik 1943] CRAIK, Kenneth: *The Nature of Exploration*. 1943
- [Dearman und Pierce 2008] DEARMAN, David ; PIERCE, Jeffery S.: It's on my other computer!: computing with multiple devices. In: *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2008, S. 767–776. – ISBN 978-1-60558-011-1

- [Desai u. a. 2008] DESAI, Chetan ; JANZEN, David ; SAVAGE, Kyle: A survey of evidence for test-driven development in academia. In: *SIGCSE Bull.* 40 (2008), Nr. 2, S. 97–101. – ISSN 0097-8418
- [Dey und Abowd 1999] DEY, Anind K. ; ABOWD, Gregory D.: *Towards a Better Understanding of Context and Context-Awareness*. 1999. – URL <http://www.it.usyd.edu.au/~bob/IE/99-22.pdf>. – Zugriffsdatum: 27.02.2009
- [DFKI 2008] DFKI: *Wii Balance Board interactions*. 2008. – URL <http://www.dfki.de/iui/advanti/lab/research.html>. – Zugriffsdatum: 27.02.2009
- [Fischer 2007] FISCHER, Christian: *Entwicklung eines multimodalen Interaktionssystems für computergestützte Umgebungen*. 2007
- [Fitness 2008] FITNESS, Professional: *Laufen als Gesundheitsbrunnen*. 2008. – URL <http://www.professionell-fitness.de/tipps-news/laufen/laufen-gesund.htm>. – Zugriffsdatum: 27.02.2009
- [Fitzmaurice u. a. 2008] FITZMAURICE, George ; MATEJKA, Justin ; MORDATCH, Igor ; KHAN, Azam ; KURTENBACH, Gordon: Safe 3D navigation. In: *SI3D '08: Proceedings of the 2008 symposium on Interactive 3D graphics and games*. New York, NY, USA : ACM, 2008, S. 7–15. – ISBN 978-1-59593-983-8
- [Gamma u. a. 1995] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; M., John: *Design Patterns, Elements of Reusable Object-Oriented Software*. (1995). ISBN 0-201-63361-2
- [de Haan u. a. 2008] HAAN, Gerwin de ; GRIFFITH, Eric J. ; POST, Frits H.: Using the Wii Balance Board™ as a low-cost VR interaction device. In: *VRST '08: Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. New York, NY, USA : ACM, 2008, S. 289–290. – ISBN 978-1-59593-951-7
- [Heise-Online 2008] HEISE-ONLINE: *Gedankensteuerung für Computerspiele*. 2008. – URL <http://www.heise.de/newsticker/Gedankensteuerung-fuer-Computerspiele--/meldung/104618>. – Zugriffsdatum: 27.02.2009
- [Ishii u. a. 1994] ISHII, Hiroshi ; KOBAYASHI, Minoru ; ARITA, Kazuho: Iterative design of seamless collaboration media. In: *Commun. ACM* 37 (1994), Nr. 8, S. 83–97. – ISSN 0001-0782
- [Ishii und Ullmer 1997] ISHII, Hiroshi ; ULLMER, Brygg: Tangible bits: towards seamless interfaces between people, bits and atoms. In: *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 1997, S. 234–241. – ISBN 0-89791-802-9

- [ISO/IEC 1991] ISO/IEC: *ISO/IEC 9126*. 1991. – URL <http://www.cse.dcu.ie/essiscope/sm2/9126ref.html>. – Zugriffsdatum: 27.02.2009
- [Jacko und Sears 2003] JACKO, Julie A. ; SEARS, Andrew.: *The Human-computer Interaction Handbook : Fundamentals, Evolving Technologies, and Emerging Applications*. (2003). ISBN 9780805838381
- [Jacob 1990] JACOB, Robert J. K.: *What you look at is what you get: eye movement-based interaction techniques*. In: *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 1990, S. 11–18. – ISBN 0-201-50932-6
- [JEITA 2002] JEITA: *Exchangeable image file format for digital still cameras: Exif Version 2.2*. 2002. – URL <http://www.exif.org/Exif2-2.PDF>. – Zugriffsdatum: 27.02.2009
- [Jensen und Christensen 2007] JENSEN, Henrik W. ; CHRISTENSEN, Per: *High quality rendering using ray tracing and photon mapping*. In: *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*. New York, NY, USA : ACM, 2007, S. 1
- [Johanson u. a. 2001] JOHANSON, Brad ; FOX, O ; HANRAHAN, Pat ; WINOGRAD, Terry: *The Event Heap: An Enabling Infrastructure for Interactive Workspaces* / Stanford University. 2001. – Forschungsbericht
- [Kordon und Luqi 2002] KORDON, Fabrice ; LUQI: *An Introduction to Rapid System Prototyping*. In: *IEEE Transactions on Software Engineering* 28 (2002), Nr. 9, S. 817–821. – ISSN 0098-5589
- [Meyers 2008] MEYERS, Lexikon: *Sachartikel Bewegung: Physik*. 2008. – URL <http://lexikon.meyers.de/wissen/Bewegung>. – Zugriffsdatum: 27.02.2009
- [Moeslund und Granum 2001] MOESLUND, Thomas B. ; GRANUM, Erik: *A survey of computer vision-based human motion capture*. In: *Comput. Vis. Image Underst.* 81 (2001), Nr. 3, S. 231–268. – ISSN 1077-3142
- [Norman 1983] NORMAN, Donald: *Some Observations on Mental Models*. 1983. – URL <http://www.cogs.susx.ac.uk/users/christ/crs/atcs/norman.html>. – Zugriffsdatum: 27.02.2009
- [OCZ-Technology 2008] OCZ-TECHNOLOGY: *OCZ Peripherals*. 2008. – URL http://www.ocztechnology.com/products/ocz_peripherals/nia-neural_impulse_actuator. – Zugriffsdatum: 27.02.2009
- [Parc 2008] PARC: *Palo Alto Research Center*. 2008. – URL <http://www.parc.com/>. – Zugriffsdatum: 27.02.2009

- [Peek 2008] PEEK, Brian: *Managed Library for Nintendo's Wiimote v1.5.2.0*. 2008. – URL <http://www.brianpeek.com/blog/pages/wiimotelib.aspx>. – Zugriffsdatum: 27.02.2009
- [Perry und Wolf 1992] PERRY, Dewayne E. ; WOLF, Alexander L.: Foundations for the study of software architecture. In: *SIGSOFT Softw. Eng. Notes* 17 (1992), Nr. 4, S. 40–52. – ISSN 0163-5948
- [Pierce 2008] PIERCE, Jeff: *Personal Information Environments*. 2008. – URL http://www.almaden.ibm.com/cs/projects/pie/#pie_infrastructure. – Zugriffsdatum: 27.02.2009
- [Pogue u. a. 2005] POGUE, David ; AULICH, André ; STEFFENS, Rudolf: *Mac OS X: Missing Manual, Tiger Ausgabe*. (2005). ISBN 3897214121
- [Rahimi und Vogt 2008] RAHIMI ; VOGT: *Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie*. 2008
- [Reenskaug 1979] REENSKAUG, Trygve: *Models - Views - Controllers*. 1979. – URL <http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf>. – Zugriffsdatum: 27.02.2009
- [Richard Rouse 1999] RICHARD ROUSE, III: What's your perspective? In: *SIGGRAPH Comput. Graph.* 33 (1999), Nr. 3, S. 9–12. – ISSN 0097-8930
- [Robertson u. a. 1991] ROBERTSON, George G. ; MACKINLAY, Jock D. ; CARD, Stuart K.: Cone Trees: animated 3D visualizations of hierarchical information. In: *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 1991, S. 189–194. – ISBN 0-89791-383-3
- [Schilit 1994] SCHILIT, Bill N.: Context-aware computing applications. In: *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, IEEE Computer Society, 1994, S. 85–90
- [Schlömer u. a. 2008] SCHLÖMER, Thomas ; POPPINGA, Benjamin ; HENZE, Niels ; BOLL, Susanne: Gesture recognition with a Wii controller. In: *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*. New York, NY, USA : ACM, 2008, S. 11–14. – ISBN 978-1-60558-004-3
- [segway 2008] SEGWAY: *segway*. 2008. – URL <http://www.slashgear.com/segway-i2-and-x2-personal-transporter-gets-fcc-approval-03818/>. – Zugriffsdatum: 27.02.2009
- [SIGCHI 1996] SIGCHI, ACM: *Definition of HCI*. 1996. – URL http://sigchi.org/cdg/cdg2.html#2_1. – Zugriffsdatum: 27.02.2009

- [Smith u. a. 2003] SMITH, David A. ; KAY, Alan ; RAAB, Andreas ; REED, David P.: Croquet - A Collaboration System Architecture. In: *Creating, Connecting and Collaborating through Computing, International Conference on O* (2003), S. 1–8
- [Smith u. a. 2005] SMITH, David A. ; KAY, Alan ; RAAB, Andreas ; REED, David P.: *Hedgehog Architecture*. 2005. – URL http://www.opencroquet.org/images/ee/2005_Hedgehog_Architecture.pdf. – Zugriffsdatum: 27.02.2009
- [Smith u. a. 2006] SMITH, David A. ; KAY, Alan ; RAAB, Andreas ; REED, David P.: *Croquet Programming - A Concise Guide*. 2006
- [Stanford 2008] STANFORD: *Stanford HCI Group*. 2008. – URL <http://hci.stanford.edu/>. – Zugriffsdatum: 27.02.2009
- [Teather und Stuerzlinger 2007] TEATHER, Robert J. ; STUERZLINGER, Wolfgang: Guidelines for 3D positioning techniques. In: *Future Play '07: Proceedings of the 2007 conference on Future Play*. New York, NY, USA : ACM, 2007, S. 61–68. – ISBN 978-1-59593-943-2
- [Weiser 1991] WEISER, Mark: *The Computer for the 21st Century*. 1991. – URL <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>. – Zugriffsdatum: 27.02.2009
- [Weiser 1993] WEISER, Mark: Some computer science issues in ubiquitous computing. In: *Commun. ACM* 36 (1993), Nr. 7, S. 75–84. – ISSN 0001-0782
- [wiibrew.org 2008a] WIIBREW.ORG: *a wiki dedicated to homebrew on the Nintendo Wii - Wii Balance Board*. 2008. – URL http://wiibrew.org/wiki/Wii_Balance_Board. – Zugriffsdatum: 27.02.2009
- [wiibrew.org 2008b] WIIBREW.ORG: *a wiki dedicated to homebrew on the Nintendo Wii - Wiimote*. 2008. – URL <http://wiibrew.org/wiki/Wiimote>. – Zugriffsdatum: 27.02.2009
- [Williams u. a. 2003] WILLIAMS, Laurie ; MAXIMILIEN, E. M. ; VOUK, Mladen: Test-Driven Development as a Defect-Reduction Practice. In: *ISSRE '03: Proceedings of the 14th International Symposium on Software Reliability Engineering*. Washington, DC, USA : IEEE Computer Society, 2003, S. 34. – ISBN 0-7695-2007-3

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 7. März 2009

Ort, Datum

Unterschrift