



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

## **Masterarbeit**

Olaf Potratz

Ein Framework für physikbasierte 3D Interaktion mit großen  
Displays

Olaf Potratz

Ein Framework für physikbasierte 3D Interaktion mit großen  
Displays

Masterarbeit eingereicht im Rahmen der Masterprüfung  
im Studiengang Master 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 27.02.2014

**Olaf Potratz**

**Thema der Masterarbeit**

**Stichworte**

Mensch-Maschine-Interaktion, Motion Tracking, Gesten, Gestenerkennung, Physik-Engine

**Kurzzusammenfassung**

In dieser Arbeit wird ein Framework zur Interpretation von Gesten für große Displays entwickelt. Die Interpretation findet mithilfe einer Physiksimulation statt und soll von den verwendeten Sensoren unabhängig sein. Die Anforderungen für ein solches Framework werden beschrieben und analysiert. Die exemplarische Umsetzung erfolgt für zwei optische Motiontracker. Darauf aufbauende Verfahren zur Erkennung der Hände und der eigentlichen Gestenerkennung werden in der Theorie beschrieben und zum Teil umgesetzt. Für die Gestenerkennung selbst kommt eine Physik-Engine zum Einsatz. Die Teilsysteme kommunizieren über einen Message Broker miteinander. Im Anschluss an erste Versuche erfolgt eine Usability Untersuchung, die über die Benutzbarkeit des Systems hinaus noch klären soll, inwieweit eine Simulation von realen Gegenständen zur Interaktion, den Einstieg in ein solches System vereinfacht. Diese Ergebnisse werden dokumentiert und bewertet.

**Olaf Potratz**

**Title of the paper**

A framework for physics-based 3D interaction with large displays

**Keywords**

Human-Computer-Interface, Motion Tracking, Gestures, Gesture Recognition, Physics-Engine

**Abstract**

This paper deals with a framework for the interpretation of gestures for large displays. The interpretation takes place using a physics simulation, and should be independent of the sensors used. The requirements for such a framework are being described and analysed. An exemplary realization is carried out for two different optical motion tracker. Based procedures for the detection of hands and the actual gesture recognition are described in theory and are being partially implemented. For the gesture recognition itself, a physics engine is employed. The subsystems communicate with the help of a Message Broker. Following initial trials, a usability study is carried out. Apart from clarifying the system's usability, the study also aims to explain to what extent the simulation of real objects simplifies the user's handling of such a system. These results are being documented and evaluated.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Zielsetzung . . . . .	3
1.3	Gliederung der Arbeit . . . . .	4
<b>2</b>	<b>Analyse</b>	<b>5</b>
2.1	Mensch-Maschine-Kommunikation . . . . .	5
2.1.1	Gestenerkennung . . . . .	5
2.1.2	Angewandte Gestenerkennung in 2D . . . . .	6
2.1.3	Angewandte Gestenerkennung in 3D . . . . .	6
2.1.4	Herausforderungen in der 3D Gestenerkennung . . . . .	7
2.2	Themenverwandte Arbeiten . . . . .	8
2.2.1	A User Interface Framework for Multimodal VR Interactions . . . . .	8
2.2.2	Vision-based 3D Finger Interactions for Mixed Reality Games with Physics Simulation . . . . .	8
2.2.3	HoloDesk: Direct 3D Interactions with a Situated SeeThrough Display . . . . .	9
2.2.4	weitere Arbeiten . . . . .	10
2.3	Szenario . . . . .	11
2.4	Schalttafel . . . . .	12
2.4.1	Schieberegler . . . . .	13
2.4.2	Schaltknopf . . . . .	14
2.4.3	Griff . . . . .	15
2.4.4	Hebel . . . . .	16
2.4.5	Rad . . . . .	17
2.4.6	Joystick . . . . .	18
2.5	Gesten . . . . .	19
2.5.1	Kontext . . . . .	19
2.5.2	Start- und Endgeste . . . . .	19
2.5.3	Steuerungsgesten . . . . .	20
2.6	Funktionale Anforderungen . . . . .	24

---

2.6.1	Abgrenzung . . . . .	24
2.7	Erste Untersuchungen . . . . .	24
<b>3</b>	<b>Algorithmen</b>	<b>26</b>
3.1	Aufbereitung der ART-Trackerdaten . . . . .	26
3.2	Abstraktion auf Aktoren (ART) . . . . .	28
3.3	Aktorzurordnung . . . . .	31
3.3.1	Anzahl der Hände . . . . .	31
3.3.2	Hypothese: Normale Posen . . . . .	32
3.3.3	Erwartungshaltung an Aktorbewegung . . . . .	33
3.3.4	Startposition der Aktoren . . . . .	34
3.4	Gestenerkennung (ART) . . . . .	34
<b>4</b>	<b>Design</b>	<b>36</b>
4.1	Funktionalität . . . . .	36
4.1.1	Abgrenzung der Sensorik . . . . .	36
4.1.2	Gestenerkennung in der Physiksimulation . . . . .	37
4.1.2.1	Handlungsgruppen für Gesten . . . . .	37
4.1.2.2	Joints . . . . .	37
4.1.2.3	Kontextwahl . . . . .	38
4.1.2.4	Start- und Endgeste . . . . .	39
4.1.2.5	Steuergesten . . . . .	40
4.2	Design . . . . .	41
4.2.1	Sensoren . . . . .	41
4.2.2	Physikalische Ausprägung der Artefakte . . . . .	43
4.2.3	Visualisierung . . . . .	44
4.2.4	Steuerung für externes Programm . . . . .	45
4.3	Software-Architektur . . . . .	46
4.3.1	Kommunikation Message Broker . . . . .	46
<b>5</b>	<b>Realisierung</b>	<b>49</b>
5.1	Message Broker . . . . .	49
5.2	Sensorenadapter . . . . .	49
5.2.1	ART-Tracker . . . . .	50
5.3	Physik-Engine . . . . .	51
5.3.1	Wahl der Physik-Engine . . . . .	51
5.3.2	Entwicklung eigener Joints . . . . .	51
5.4	Umsetzung der Aktoren & Kontextbestimmung . . . . .	52
5.4.1	Aktoren in der Physiksimulation . . . . .	52
5.4.2	Aktoren in der 3D-Simulation . . . . .	52

---

5.5	Umsetzung des Schaltpults . . . . .	53
5.5.1	Umsetzung der Artefakte . . . . .	53
5.5.2	Umsetzung der Anzeiger . . . . .	60
5.6	Grafische Ausgabeverfahren . . . . .	62
5.7	Akustische Ausgabe . . . . .	64
5.8	Fazit . . . . .	65
5.8.1	Änderungen bei erneuter Umsetzung . . . . .	66
5.9	Erweiterungsmöglichkeiten . . . . .	66
5.9.1	Weitere Sensoren . . . . .	66
5.9.2	Weitere Ausgabemedien . . . . .	68
<b>6</b>	<b>Usability Untersuchung</b>	<b>69</b>
6.1	Aufbau . . . . .	69
6.1.1	Versuchsaufbau I: Die Schalttafel . . . . .	69
6.1.2	Versuchsaufbau II: Das virtuelle Cockpit . . . . .	70
6.2	Durchführung . . . . .	73
6.2.1	Testdurchlauf für die Schalttafel . . . . .	73
6.2.2	Durchführung des Tests für das virtuelle Cockpit . . . . .	75
6.3	Ergebnisse und Diskussion . . . . .	77
6.3.1	Ergebnisse Schalttafelversuch . . . . .	77
6.3.2	Ergebnisse des virtuelle Cockpits . . . . .	78
<b>7</b>	<b>Schluss</b>	<b>79</b>
7.1	Zusammenfassung . . . . .	79
7.2	Fazit . . . . .	80
7.3	Ausblick . . . . .	81
	<b>Literaturverzeichnis</b>	<b>82</b>
	<b>Abbildungsverzeichnis</b>	<b>86</b>
	<b>Glossar</b>	<b>88</b>
<b>A</b>	<b>Fragebögen</b>	<b>90</b>
<b>B</b>	<b>Ergebnisse</b>	<b>107</b>

# Kapitel 1

## Einleitung

Gesten sind ein großer Bestandteil der menschlichen Kommunikation. Im alltäglichen Gebrauch meist unbewusst benutzt, unterstützen sie häufig die verbale Kommunikation. Verbale Kommunikation kann allerdings gänzlich von Gesten ersetzt werden. Dies können Gesten unter der Verwendung von Händen und Armen sein. Auch kombinierte Gesten von zwei Händen und Armen sind möglich, letztlich kann aber auch ein einfaches Handzeichen als vollständige Geste verstanden werden.

Dieses geschieht häufig in Bereichen, in denen ein verbaler Kommunikationsweg versperrt ist, oder aber zur dauerhaften Benutzung zu umständlich oder zu ungenau ist. Letzteres betrifft auch die Kommunikation mit Computersystemen. Selbst wenn unter den gegebenen Umständen verbale Kommunikation denkbar wäre, ist sie meist zu umständlich, da für komplexere Handlungen detailreiche verbale Kommandos notwendig wären. Hier offenbart sich ein Betätigungsfeld für die Gesten und somit auch für die Gestenerkennung. Ein alltägliches Beispiel hier für sind die von der Polizei verwendeten Handzeichen zur Regelung des Straßenverkehrs<sup>1</sup>.

Umso umfangreicher die Interaktionen werden, welche mit Hilfe von Gesten ausführbar werden sollen, desto komplexer muss auch das auf dem Computer arbeitende System werden, um alle Gesten korrekt erkennen zu können.

Gleichzeitig werden Regeln für die korrekte Ausführung der Gesten unerlässlich. Mehrdeutig interpretierbare Bewegungen sollen möglichst unterbunden werden, damit es nicht zu Missinterpretationen kommt. Diese Regeln umfassen in den meisten Fällen ein Gestenalphabet, welches Gesten bestimmten Funktionen zuordnet. Dieses Alphabet muss im Regelfall vom Benutzer erlernt werden.

Es gilt ein Gleichgewicht zwischen der Komplexität eines Computersystems und der Anzahl der Regeln, die ein Benutzer beherrschen muss, zu finden. Hierbei gilt es, beide Teile in ihren Umfängen wenn möglich klein zu halten. Aus Sicht der Benutzer sollte es im besten Fall keine Regeln oder keine offensichtlichen Regeln, die es vorher zu lernen gilt, geben.

---

<sup>1</sup>[http://www.gesetze-im-internet.de/stvo\\_2013/\\_36.html](http://www.gesetze-im-internet.de/stvo_2013/_36.html)

Ein möglicher Weg dem Benutzer Regeln für die Gesten aufzuerlegen ist, diese Regeln in Form einer simulierten Umgebung darzustellen. Der Benutzer kann mit den Gegenständen interagieren. Wählt man Gegenstände, die dem Benutzer aus der Realität und seinem Umfeld bekannt sind, entfällt das Erlernen des Umgangs mit diesen Gegenständen. Es bedarf also nicht dem Erlernen neuer Regeln, da die Regeln aus der Realität bereits bekannt sind. Die in der simulierten Umgebung verwendeten Objekte haben zwei Aufgaben. Sie sind eine Art Scharnier zwischen Mensch und Maschine. Zum einen dienen sie dem Benutzer zur Interaktion und sorgen hier auch für visuelles Feedback. Zum Anderen ermöglichen diese Objekte dem Computersystem auf die Interaktionen des Benutzers zu schließen, indem der Zustand dieser Objekte betrachtet werden kann. Im Idealfall entfällt für diese Form der Kommunikation von Anwender und Computer jegliche Form von Controller. Die virtuelle Simulation selbst ist als Eingabegerät ausreichend.

Dieses Vorgehen steht im Sinne der „seamless interaction“, welche besagt, dass sich die Maschinen den Menschen und nicht die Menschen der Maschine anzupassen haben. In dem hier erläuterten Vorgehen wäre dies so zu verstehen, dass kein neues Wissen erlangt werden muss, da der Umgang mit den realen, zu einer oder mehreren Maschinen gehörenden Gegenständen, bereits erlernt wurde.

Die Gestenerkennung wird in dieser Arbeit durch die Beobachtung der simulierten Gegenstände, welche dem Anwender jedoch noch sichtbar gemacht werden, verwirklicht. Es geschieht keine Interpretation einzelner Teilbewegungen des Benutzers, sondern ausschließlich die, durch sein Handeln verursachten, Änderungen an zuvor genannten virtuellen Gegenständen.

## 1.1 Motivation

Zurzeit werden für die Kommunikation zwischen Mensch und Maschine meist reale Controller benötigt. Für die Kommunikation im Nahbereich sind Kommunikationsformen durch Berührung, wie Tastatur oder Touchdisplay, sicherlich konkurrenzlos. Sofern man auf eine distanzierte Kommunikation angewiesen ist, muss meist auf Fernsteuerungen oder ähnliches zurückgegriffen werden. In einer optimalen Kommunikation zwischen Mensch und Maschine sollte allerdings generell auf Controller verzichtet werden und die Interaktion sollte möglichst einfach und intuitiv gestaltet werden. Aufgrund der Verbreitung immer größerer Displays (Flat-Screens), aber auch dem Ausbau von Medienfassaden, rückt diese Herausforderung weiter in den Mittelpunkt.

Durch den rasanten technischen Fortschritt in Bereichen der Kerasensorik werden immer bessere, aber auch preislich erschwingliche Systeme auf den Markt gebracht. Somit erschließt sich dieser Forschungs- und Arbeitsbereich einer größeren Gruppe von Menschen. Ein interessanter Ansatz hierzu wird an dem Forschungsgebiet der Imaginären Interfaces

am Hasso Plattner Institut<sup>2</sup> erforscht und umgesetzt.

Der Ansatz dieser Arbeit versucht die Erkennung von Gesten vom eigentlichen Handeln des Anwenders zu trennen, indem das System sich nur die Auswirkungen der Gesten anschaut. Dies soll unter der Berücksichtigung des schnellen Fortschreitens der Technik im Bereich der Sensorik geschehen, innerhalb dessen die Gestenerkennung unabhängig von einem bestimmten Sensor geschehen kann.

## 1.2 Zielsetzung

Das Ziel dieser Arbeit kann in zwei Teile aufgeteilt werden.

Zum einen soll das oben aufgeführte Verfahren zur Gestenerkennung analysiert und die nötigen Eigenschaften, die für eine softwaretechnische Umsetzung von Bedeutung sind, ermittelt werden. Der praktische Teil dieser Arbeit umfasst die Entwicklung eines Softwaresystems, das die in der Analyse ermittelten Anforderungen berücksichtigt. Dieses System soll als Framework entwickelt werden, um später erscheinende Sensoren und Ausgabemedien einfacher integrieren zu können.

Um die Gestenerkennung realisieren zu können, verfügt der Kern des Systems über eine Physik-gestützte 3D-Simulation. Die Physik-Engine bildet das Fundament für die Erkennung der Gesten, da es die Einhaltung der Eigenschaften und Regeln der einzelnen Gegenstände sicherstellt. Die vom Anwender erzeugten Gesten sollen dann durch die Interaktion mit der 3D-Simulation erkannt werden und nicht, wie häufig üblich, nur durch reine Verarbeitung der Bewegungsabläufe. Die Lösung des Start-Ende Problems soll aus der Gestenerkennung selbst ermittelt werden und, jeweils separat, von den Anbindungen an die Kamerasensoren gelöst werden.

Das System soll durch einheitliche, einfache Schnittstellen eine lose Kopplung zu unterschiedlichen Kamerasensoren ermöglichen. Ebenso soll das visuelle und akustische Feedback an den Benutzer über das Framework entkoppelt und verwirklicht werden. Zum Ende hin steht dem Benutzer eine Grafikausgabe zur Verfügung, die sein Handeln widerspiegelt. Bei dieser grafischen Ausgabe handelt es sich um eine dreidimensionale Darstellung der Simulation, die nicht fotorealistisch ist.

Als weiteres Ziel soll untersucht werden, inwieweit die Verwendung von, aus der Realität bekannten, Gegenständen die Interaktion mit dem System erleichtern kann. Konkret soll anhand der unterschiedlichen Gegenstände ermittelt werden, ob eine Analogie den Einstieg in die Benutzung des Systems erleichtert. Die Frage der Benutzbarkeit und inwiefern man dem Ideal eines „natural user interface“ näher kommen könnte soll durch Tests mit anschließenden Fragebögen ermittelt werden. Hierfür werden verschiedene Personen, mit unterschiedlichen Hintergründen, zum Test eingeladen.

---

<sup>2</sup><http://www.hpi.uni-potsdam.de/baudisch/home.html>

### 1.3 Gliederung der Arbeit

Das erste Kapitel enthält die Einleitung, in der die Motivation und die Zielsetzung dieser Arbeit näher erläutert werden.

Das Kapitel 2 beinhaltet die Analyse eines Systems zur Gestenerkennung. Zuerst wird auf den Bereich Gestenerkennung allgemein eingegangen, anschließend wird ein Anwendungsszenario beschrieben. Es folgen themenverwandte Arbeiten, die vorgestellt und im Kontext der eigenen Zielsetzung auf ihre Verwendbarkeit hin bewertet werden.

Danach wird die Schalttafel-Metapher vorgestellt. Es werden die für die Schalttafel infrage kommenden Benutzerobjekte betrachtet. Im Anschluss werden Gesten bestimmt, mithilfe derer eine Interaktion zwischen dem Benutzer und der Schalttafel ermöglicht werden soll.

Die Analyse der Gesten, unter Berücksichtigung des hier gewählten Ansatzes, dienen als Grundlage der funktionalen Anforderungen, die für eine spätere Umsetzung des Systems formuliert werden. Als Abschluss wird in der Analyse noch ein erster Test vorgestellt, der eine grundsätzliche Machbarkeit nicht ausschließt.

Im Kapitel 3 werden die für die Entwicklung des Systems benötigten Algorithmen vorgestellt. Diese Algorithmen sind zum Teil spezifisch für einzelne Sensoren, zum Teil betreffen die Algorithmen das Kernsystem.

Kapitel 4 befasst sich im ersten von drei Teilen genauer mit dem gewählten Ansatz zur Gestenerkennung, es werden Überlegungen theoretischer Natur zur Gestenerkennung durch eine Physiksimulation gemacht. Der zweite Teil des Kapitels beinhaltet die für die Entwicklung eines Softwaresystems benötigten Beschreibungen der Teilkomponenten. Der letzte Teil des Kapitels beschreibt die zugrunde liegenden Architekturmuster und zeigt die Kommunikation zwischen den Komponenten des Systems.

Das 5. Kapitel befasst sich mit der Realisierung eines Framework-Prototyps. In den einzelnen Unterkapiteln wird die Umsetzung der Komponenten beschrieben. Es wird ein Fazit gezogen, inwieweit die in der Analyse aufgeführten Gesten erkannt werden konnten. Außerdem findet eine kurze Bewertung der technischen Umsetzung statt. Zum Abschluss wird ein Ausblick dahin gehend gewährt, welche Erweiterungen denkbar wären und welche Sensoren und Ausgabemedien wohl in Kürze zur Verfügung stehen und Verwendung finden könnten.

Kapitel 6 beschreibt im ersten Abschnitt die für die Usability Untersuchungen verwendeten Aufbauten. Anschließend werden die Durchführungen für beide Tests beschrieben. Der letzte Teil des Kapitels bietet einen Überblick über die Ergebnisse und die daraus gezogenen Schlussfolgerungen.

Das Kapitel 7 bildet den letzten Teil dieser Arbeit und enthält eine Zusammenfassung der gesamten Arbeit. Zum Schluss erfolgt ein Ausblick.

# Kapitel 2

## Analyse

Zu Beginn soll das Thema Gestenerkennung beleuchtet werden. Die im Anschluss folgenden Arbeiten dienen als Inspiration und zum Teil als Grundlage für das entwickelte Szenario und die Umsetzung dieser Arbeit. Die hier vorgenommene Analyse nimmt auf das in Abschnitt 2.3 beschriebene Beispielszenario Bezug und führt eine reale Schalttafel mit möglichen dort verwendbaren Komponenten auf. Anschließend wird untersucht, welche Bewegungen beim Betätigen dieser Schalttafelkomponenten gemacht werden und wie sich daraus Gesten ableiten lassen.

Es werden die funktionalen Anforderungen aufgeführt und zum Abschluss werden die in Projektausarbeitung [Potratz (2013)] ausgeführten ersten Untersuchungen gezeigt.

### 2.1 Mensch-Maschine-Kommunikation

#### 2.1.1 Gestenerkennung

Es bedarf vor der Gestenerkennung einer Unterscheidung, um welche Art von Geste es sich jeweils handelt. Joachim Boetzer führt in seiner Arbeit [Heitsch (2008)] eine Klassifizierung der Gesten nach Vladimir Pavlovic auf, es erfolgt dort eine Aufteilung der Gesten in bewusste und unbewusste Gesten.

Bewussten Gesten wurden weiter in kommunikative und manipulative Gesten unterteilt. Es soll sich dieser Unterscheidung für diese Arbeit bedient werden. Es werden beide Arten der bewussten Gesten verwendet. Ein Großteil der Arbeit wird ausschließlich den manipulativen Gesten widmen.

Die Richtige, oder besser gesagt, die im jeweiligen Augenblick korrekte Bedeutung einer Geste zu erkennen, ist schwierig. Eine Geste kann viele unterschiedliche Aussagen haben. Welche genaue Aussage der Initiator verfolgt, lässt sich oft nur im Zusammenhang mit vielen Faktoren erschließen. Einige Faktoren sind unter anderem der Gesteninitiator selbst, die Situation, in der die Geste gemacht wird und der Empfänger der Geste. Darüber hinaus

spielen der Zustand der Person, seine Haltung sowie seine Mimik eine wichtige Rolle bei der korrekten Interpretation der Geste. Bei der Gestenerkennung spielt das sogenannte „Start-Ende-Problem“ oder auch „Midas-Touch Problem“ eine große Rolle. Die im Folgenden aufgeführten Verfahren zur Gestenerkennung werden nur kurz angesprochen. Da nach Analyse dieser Verfahren keine Verwendung zum Lösen der in dieser Arbeit gestellten Aufgabe gesehen wurde, wird auf eine tiefer gehende Beschreibung verzichtet und auf die entsprechenden Arbeiten verwiesen.

### **2.1.2 Angewandte Gestenerkennung in 2D**

Gestenerkennung in einer zweidimensionalen Umgebung bedeutet, dass die zu interpretierende Geste sich auf zwei Dimensionen beschränkt.

Ein beliebtes Eingabegerät sind Touchscreens, vor allem Mobilgeräten wie Smartphones oder Tablets wären ohne sie kaum vorstellbar. Wie in der Arbeit [Mohammadali Rahimi (2008)] beschrieben, kann als Eingabegerät für 2D-Gesten auch ein Touchscreen dienen. Das Start-Ende-Problem kann hier im Zusammenspiel mit dem Berühren des Touchscreens, dem Start der Geste, und dem Loslassen des Touchscreens, als Ende der Geste, behandelt werden. Ansatz zur Erkennung der Gesten liefern hier die einzelnen Positionen der Berührung über die Zeit. Als mögliche Verfahren zur Gestenerkennung werden in [Mohammadali Rahimi (2008)] das „Hidden Markov Modell“, die „Support Vector Machines“ und die „Space-Time-Linearisierung“ genannt. All diese Verfahren spielen für die Arbeit keine Rolle, und sollen somit auch nicht weiter betrachtet werden.

### **2.1.3 Angewandte Gestenerkennung in 3D**

Diese Form der Gestenerkennung findet räumlich statt. Die Geste selbst muss aber nicht zwingend alle drei Dimensionen ausnutzen.

Besondere Herausforderungen ergeben sich hier aus dem Start-Ende-Problem, da ein Tracking der Bewegungen kontinuierlich erfolgt. Johann Heitsch stellt in seiner Arbeit [Heitsch (2008)] ein Framework zur Erkennung dreidimensionaler Gesten vor. Das System welches dort entwickelt wurde enthält eine Support Vektor Machine die das Ermitteln der Geste übernimmt. Es handelt sich bei der Support Vektor Machine um ein Verfahren zur Mustererkennung, genauere Informationen kann der Arbeit [Heitsch (2008)] entnommen werden.

Die Arbeit von Malte Nogalski [Nogalski (2012)] befasst sich mit einem Softwaresystem zur Positionierung von Soundquellen mittels Gesteninterpretation. Die Arbeit umfasst sowohl die Entwicklung, die Implementierung sowie die Auswertung. Bestandteil ist die Betrachtung verschiedener Trackingsysteme und ein Überblick über die verwendete Wellenfeldsynthese-Anlage.

### 2.1.4 Herausforderungen in der 3D Gestenerkennung

Die hier im Folgenden aufgeführten Schwierigkeiten sind Eigenheiten der dreidimensionalen Gestenerkennung.

#### Midas-Touch Problem

Der Name Midas-Touch Problem ist auf den Mythos des König Midas zurückzuführen. Dieser König wünschte sich, dass alles was er berührte zu Gold würde. In die Gestenerkennung übertragen bedeutet das die, auch als Start Ende Problem bezeichnet, Herausforderung sich daraus ergibt, dass ein System bei kontinuierlicher Bewegung des Benutzers in die Lage versetzt werden muss, entscheiden zu können, wann eine bewusste Geste beginnt und wann diese wieder endet. Dieses Problem hat zwei Seiten, zum einen können bewusste Gesten nicht rechtzeitig erkannt werden, also der Anfang der Geste vom System verpasst werden, was eventuell verhindert, dass eine Geste, beziehungsweise die richtige Geste, erkannt wird. Zum anderen ergibt sich das Problem, dass jegliche auch ungewollte Bewegung des Benutzers als Geste interpretiert wird. Der Benutzer also ungewollt mit dem System interagiert. Nachzulesen unter anderem in [Hara (2011)], wo auf die Ausführung dieses Problem eingegangen wird.

#### Kontextauswahl

Genau wie das oben aufgeführte Midas-Touch Problem, treten die Herausforderungen mit dem Ermitteln des Kontextes, in dem eine Geste auftritt, erst im dreidimensionalen Bereich auf.

Im 2D-Bereich ergibt sich der Kontext bereits durch das Berühren der Fläche. Der darunter liegenden Desktop-Metapher zufolge wird durch das Berühren ein Element ausgewählt, in dessen Kontext man sich dann bewegt und welche ein bestimmtes Gestenrepertoire versteht.

Im 3D-Bereich hingegen kann man das Problem nur insoweit umgehen, in dem man den Benutzer stets nur in einem globalen Kontext hält. Dieses Verfahren ist aber zunehmend ineffizient. Da man bei komplexeren Steuermöglichkeiten für jede Steuerung eine eigene Geste, bzw. eine eindeutige Abfolge bestimmter Gesten benötigt. Will man dieses Problem lösen und dem Benutzer mehrere Kontexte zur Verfügung stellen, in denen man übergeordnet betrachtet ob, die gleichen Gesten unterschiedliche Auswirkungen haben, muss man der Gestenerkennung noch eine Kontextermittlung vorschalten. Verschiedene mögliche Verfahren zur Wahl des Kontextes werden in dieser Arbeit in Abschnitt 4.1.2.3 aufgeführt.

## 2.2 Themenverwandte Arbeiten

Im folgenden Unterkapitel sollen drei bereits durchgeführte Projekte vorgestellt werden auf die in der Arbeit Bezug genommen wurde und die sowohl in der Entwicklung des Szenarios, als auch bei der Umsetzung einer Lösung für das Szenario beteiligt waren. Die hier genannten Projekte wurden bereits detailliert in der Arbeit [Potratz (2012b)] vorgestellt. Auf Grund ihrer immer noch stark vorhandenen Relevanz werden sie hier noch einmal kurz aufgeführt. Für ausführlichere Informationen sei auf [Potratz (2012b)] verwiesen.

### 2.2.1 A User Interface Framework for Multimodal VR Interactions

Die Arbeit [Latoschik (2005)] aus dem Jahre 2005 von Marc Latoschik umfasste eine multimodale Interaktion in einer virtuellen Umgebung. Die Realisierung wurde mittels Gestik, Sprache und dort vorliegenden Kontextinformationen umgesetzt. Zusätzlich gab es noch eine KI die das eigentliche Framework unterstützte. Die Kommunikation zwischen Sensoren und Framework findet mittels Multicast statt. Die Gestenerkennung erfolgt abhängig von der Wissensbasis, der Umgebung und weiteren Informationen. Sowohl Kontext der Geste als auch von wem die Geste initiiert wird, haben Einfluss auf das Verhalten des Gesamtsystems.

Das System ähnelt dem eigenen Arbeitsumfeld. Es handelt sich hier auch um ein Framework zur Interaktion in einem virtuellen Raum. Aufgrund des Alters der Arbeit, welche aus dem Jahr 2005 stammt, sind lediglich noch grundsätzliche Ideen und Lösungsansätze verwendbar. Allerdings müssen diese auf den Stand der Technik angepasst werden. Auch die Gestenerkennung selbst ist vom Kern her anders gelöst und kommt ohne Physik-Simulation aus.

### 2.2.2 Vision-based 3D Finger Interactions for Mixed Reality Games with Physics Simulation

Die Arbeit „Vision-based 3D Finger Interactions for Mixed Reality Games with Physics Simulation“ [Song u. a. (2008)] beschreibt die Umsetzung von zwei Spielen. Bei den Spielen handelt es sich um „Finger Fishing“ und „Jenga“.

„Finger Fishing“ kann als Vorarbeit zu der Umsetzung von „Jenga“ betrachtet werden, da „Jenga“ alle Funktionen die für „Finger Fishing“ benötigt werden enthält und darüber hinaus geht. Aus diesem Grund gilt das weitere Interesse nur noch der Umsetzung von „Jenga“. Das Spiel selbst ist einfach, man muss solange Klötzchen aus dem Turm ziehen, bis dieser zusammenstürzt.

Als Eingabegerät wird eine Kamera mit zwei Objektiven verwendet. Die zwei Objektive ermöglichen eine Tiefenwahrnehmung des Systems, allerdings nur für kurze Distanzen. Man hält die Hand in relativ kurzem Abstand vor die Kamera und das System sucht nach einer

möglichst langen Kette von aneinanderhängenden Pixeln, um die Position des Fingers und vor allem der Fingerspitze aus dem aufgenommenen Bild zu ermitteln. Ist der Finger erkannt, wird die Position der Fingerspitze in die Physik-Simulation übertragen und durch einen Aktor repräsentiert. Wenn der Zeigefinger einen virtuellen Stein berührt, ist die Haltung des Daumens entscheidend. Bei gehobenem Daumen reagiert das System nicht. Ist der Daumen dagegen gesenkt, oder nicht weiter sichtbar, weil er von der Hand selbst verdeckt wird, führt eine Kollision von Aktor und virtuellem Stein zu einer Verbindung. Diese Verbindung wird mittels eines Joints aufgebaut.

Die Ergebnisse der Arbeit zeigen, dass der Aufbau grundsätzlich funktionierte. Testpersonen konnten mit der Simulation über die Finger interagieren. Die in der Arbeit dokumentierte Benutzer-Studie liefert wertvolle Hinweise, was zu beachten ist, wenn man selbst im Anschluss an die Umsetzung eine Benutzer-Studie durchführen möchte. Die im Rahmen der [Potratz (2012a)] und [Potratz (2013)] gemachten Erfahrungen, revidierten die in [Potratz (2012b)] gemachten Aussagen bezüglich der Relevanz der Arbeit, da sich herausgestellt hat, dass die Interaktion mittels Joint doch das bevorzugte Interaktionsverfahren darstellt.

### **2.2.3 HoloDesk: Direct 3D Interactions with a Situated SeeThrough Display**

Bei der Arbeit von Otmar Hilliges u.a. [Hilliges (2012)] handelt es sich um einen Aufbau mit durchsichtigem Display und einem Kinect Sensor. Die Positionen im virtuellen und realen Raum überschneiden sich, der Benutzer sieht das Objekt real genau an der Position, wo es sich auch in der Simulation befindet. Als Sensor kommt eine Kinect zum Einsatz, die mittels Spiegel so umgelenkt wird, dass sie von oben die Hände oder andere Objekte innerhalb des Versuchsaufbaus beobachten kann. Die Daten der Hände werden weiterverarbeitet. Die Verarbeitung erfolgt in mehreren Schritten, am Ende der Verarbeitungskette gibt es ein, oder mehrere physikalische Assamblies in der Physik-Engine. Diese Assamblies bestehen aus einzelnen Physikobjekten, in diesem Fall kleinen Kugeln, die durch Joints auf ihren Positionen gehalten werden.

Die Interaktion zwischen Mensch und Simulation findet durch diese Aktoren statt. Innerhalb der Simulation wird eine Kollisionserkennung und Reibung eingesetzt, um Aktoren und andere physikalische Objekte in Kontakt treten zu lassen.

Die in der Arbeit aufgeführten Ergebnisse des Aufbaus waren vielversprechend, nicht zuletzt durch das durchsichtige Display war die Interaktion mit den Physikobjekten sehr realitätsnah. Die Arbeit ist aktuell und veranschaulicht gut die zurzeit vorhandenen Möglichkeiten. Insbesondere das Physikmodell der Aktoren aus zusammengesetzten einzelnen Kugeln, und die darauf mögliche realitätsnahe Interaktion beim Greifen von Objekten ist sehr interessant, auch wenn sich dieses Verhalten in ersten Versuchen nicht exakt nachbauen ließ.

### 2.2.4 weitere Arbeiten

Mitunter durch die Veröffentlichung der Kinect und deren Möglichkeiten zum Erkennen von Gesten sind in den letzten Monaten viele neue Arbeiten im Bereich der drei-dimensionalen Gestenerkennung erschienen. Im folgenden sollen einige davon kurz aufgeführt werden.

Die Arbeit [Söhmelioglu (2011)] von Burak Söhmelioglu „Experimenteller Vergleich von Techniken zur intuitiven Interaktion in Augmented Reality-Anwendungen“ beschreibt grundlegende Methoden zur Interaktion in einem virtuellen Raum. In diesem Rahmen werden auch verschiedene Gestenerkennungen im 3D-Bereich vorgestellt.

In der Arbeit „Human Gesture Recognition Using Kinect Camera“ [Watanapa (2012)] von Orasa Patsadu und anderen, wird eine Gestenerkennung auf Basis von Data-Mining-Methoden vorgeschlagen und mit herkömmlichen Verfahren zur Gestenerkennung, wie „Support Vector Machines“, Entscheidungsbäumen verglichen.

Die Arbeit, „Human-computer Interaction using Pointing Gesture based on an Adaptive Virtual Touch Screen“ [Ye-peng (2013)] von Pan Jing und Guan Ye-peng, befasst sich mit der Interaktion zwischen einem Benutzer und einem großen Bildschirm. Bei den Gesten handelt es sich allerdings um ausschließlich kommunikative Zeigegesten.

Ohne eine Wertung vornehmen zu wollen, gilt es hier zu beachten, worin genau die Gestenerkennung und Steuerung besteht. Ob nur die mitgelieferten Funktionen des Sensors und deren Gestenrepertoire genutzt wird, sowie in den Arbeiten "3D Hand Gesture Recognition Based on Sensor Fusion of Commodity Hardware" [K. Denker (2012)], "Gesture Recognition based on 2D and 3D Feature by using Kinect Device" [u.a. (2012)] und „Gesture Based PC Interface with Kinect Sensor“ [ERAP (2012)] oder aber ob eine tiefer gehende Gestenerkennung aufgebaut wird, wie es mit dieser Arbeit der Fall ist.

## 2.3 Szenario

Das hier im folgenden beschriebene Szenario ist in Teilen von den in Abschnitt 2.2 aufgeführten Projekten inspiriert. Es soll ein System entwickelt werden, das eine Mensch-Maschine-Kommunikation ermöglicht, auf Basis einfacher Gesten im Rahmen bestimmter Kontexte.

Es handelt sich um ein zweigeteiltes System, erster Teil sind die sich im Raum befindlichen virtuellen Eingabegeräte die der Benutzer mit seinen Händen greifen und manipulieren kann. Dieses geschieht sobald sich der Benutzer innerhalb des Messbereichs des verwendeten Trackingsystems befindet. Es werden beide Hände des Benutzers erfasst und in den virtuellen Raum übertragen. Die Hände werden hier zu den Aktoren und als geöffnete oder geschlossene Handsymbole abgebildet. Der Benutzer ist nun in der Lage, mit den auf dem Schaltpult befindlichen Artefakten zu agieren, indem er nach ihnen greift. War das Greifen erfolgreich, befindet sich eine Greiffläche in Reichweite zu der sich schließenden Hand, wird der Aktor, stellvertretend für die Hand, mit dem Artefakt verbunden und der Benutzer kann dieses nun durch die Bewegung seiner Hand manipulieren. Das System selbst kann die virtuellen Eingabegeräte auch manipulieren, in dem es exemplarisch genannt, Schalter auf bestimmte Positionen stellt oder ein Steuerrad wieder auf eine vom System errechnete Position vor oder zurück dreht. Der Benutzer sieht über eine Anzeige das Schaltpult und die Positionen seine Hände auf, beziehungsweise über der Schaltfläche sowie die möglichen Manipulationen, die er dort ausführt.

Das eigentliche Wirken des Benutzers, die Steuerung eines (zweiten) Softwaresystems, soll dem Benutzer über die große Projektionsfläche sichtbar gemacht werden. Dieses System reagiert nicht direkt auf die Gesten des Benutzers, sondern auf die durch ihn verursachten Manipulationen an einer Schalttafel. Somit wird von den eigentlichen Gesten abstrahiert und man spiegelt eher die intendierten Eingaben des Benutzers wider.

## 2.4 Schalttafel

Die virtuelle Installation verschiedener Artefakte wird im Begriff der Schalttafel zusammengefasst. Im weiteren Verlauf der Arbeit virtuelle Schalttafel, Schaltpult oder auch Schaltbord genannt. Die Metapher der Schalttafel ähnelt der in 2.1 zu sehenden Schalttafel. Die im Schaltpult befindlichen Geräte werden in dieser Arbeit als Schalttafel-Artefakt oder kurz Artefakte bezeichnet. Diese Artefakte umfassen zum Beispiel Schaltknöpfe, Hebel und Stellräder. Eine ausführliche Auflistung der Artefakte folgt im Anschluss. Jedes Schalttafel-Artefakt verfügt über eine oder mehrere Greifflächen, wie in 2.6(b) beispielhaft verdeutlicht. Bei einer Greiffläche handelt es sich um einen Bereich des Artefakts, mit dem sich ein Akteur verbinden kann. Dies ermöglicht eine Vereinfachung der Handhabung mit den Artefakten, da der Benutzer direkt erkennen kann, wo er anfassen kann und soll. Grundsätzlich kann sich nur ein Akteur mit einer Greiffläche verbinden zur Zeit verbinden. Es ist allerdings möglich, einen Schalttafelartefakt mit mehr als einer Greiffläche auszustatten um so eine beidhändige Aktion zu erzwingen. Als ein Beispiel hierfür ist der zweihändige Hebel, der später in 2.4.4 vorgestellt wird, zu nennen.



Abbildung 2.1: : Fotoausschnitt einer Schalttafel [Quelle: Schalttafel]

Die verwendeten Artefakte haben entweder diskrete oder kontinuierliche Zustände. Der einfachste Fall für diskrete Zustände sind „An“ und „Aus“. Wie im Beispiel des Schaltkopfes 2.4.2. Die Position des beweglichen Bestandteils des Artefaktes werden immer auf genau einen Zustand abgebildet wird. Der Hebel 2.4.4, Schieberegler 2.4.1 und ähnliche Artefakte

können über mehr als zwei Zustände verfügen.

Für Schieberegler und Räder können kontinuierliche Zustände in Betracht gezogen werden. Hier wird nicht ein bestimmter Zustand abgebildet, sondern ein prozentualer Wert, zwischen den beiden Extremwerten, die der bewegliche Bestandteil eines Artefakts annehmen kann. Hier im Anschluss werden die einzelnen Artefakte aufgeführt.

### 2.4.1 Schieberegler

Der Schieberegler wie in Abbildung 2.2 zu sehen besitzt einen Griff, und kann längs einer Achse bewegt werden. Es kommen sowohl die x als auch die y-Achse infrage. Es ist zu Beginn festzulegen, längs welcher Achse ein Schieberegler manipuliert werden soll. Eine abwechselnde oder gleichzeitige Beweglichkeit entlang beider Achsen ist nicht vorgesehen. Der Schieberegler kann grundsätzlich kontinuierliche oder diskrete Zustände besitzen, auch dies ist beim initialisieren des Schiebereglers zu Beginn anzugeben.

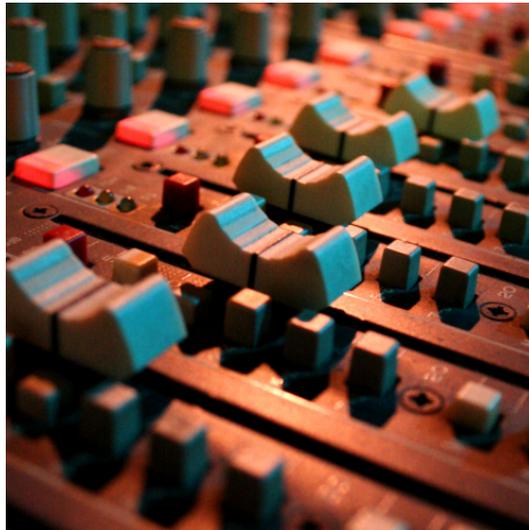


Abbildung 2.2: : Foto eines Schiebereglers [Quelle: Schieberegler]

## 2.4.2 Schaltknopf

Unter dem Oberbegriff Schaltknopf sollen zwei mögliche Artefakte aufgeführt werden, die sich wie folgt unterscheiden.

**Taster** Die erste soll ein Schaltknopf in Form eines Tasters sein. Eine Abbildung hier zu ist 2.3(a) zu sehen. Der Taster hat zwei diskrete Zustände, „An“ und „Aus“. Die Manipulation des Tasters findet längs der z-Achse statt. Grundsätzlich ist der Taster in der Position „Aus“. Wird der Taster gedrückt, wechselt er in den Zustand „An“. Nachdem der Taster losgelassen wird, springt der Taster nach kurzer Zeit wieder in den Zustand „Aus“ und kann von hier aus erneut gedrückt werden.

**Kippschalter** Als weitere Möglichkeit für ein Artefakt das die Drückgeste behandelt, soll hier ein Kippschalter genannt werden. Wie auch der Taster, besitzt der Kippschalter die zwei diskreten Zustände „An“ und „Aus“. Die Manipulation findet auch durch Drücken statt, allerdings bewegt sich das Artefakt selbst um die y-Achse. Die Besonderheit des Kippschalters ist seine wechselnde Fläche, an der gegriffen werden kann. Die hervorstehende Fläche kann gegriffen und rein gedrückt werden. Dabei ändert der Kippschalter seinen Zustand, von „An“ nach „Aus“ oder umgekehrt. Ist der Zustand geändert, löst der Kippschalter eigenständig die Verbindung mit der Hand und verweilt in seinem Zustand. Nun kann mit der Hand die andere aktiv gewordene Fläche gegriffen werden, und der Benutzer kann den Zustand des Kippschalters durch erneutes Drücken wieder zurück setzen. Eine Abbildung des Kippschalters findet sich unter 2.3(b).

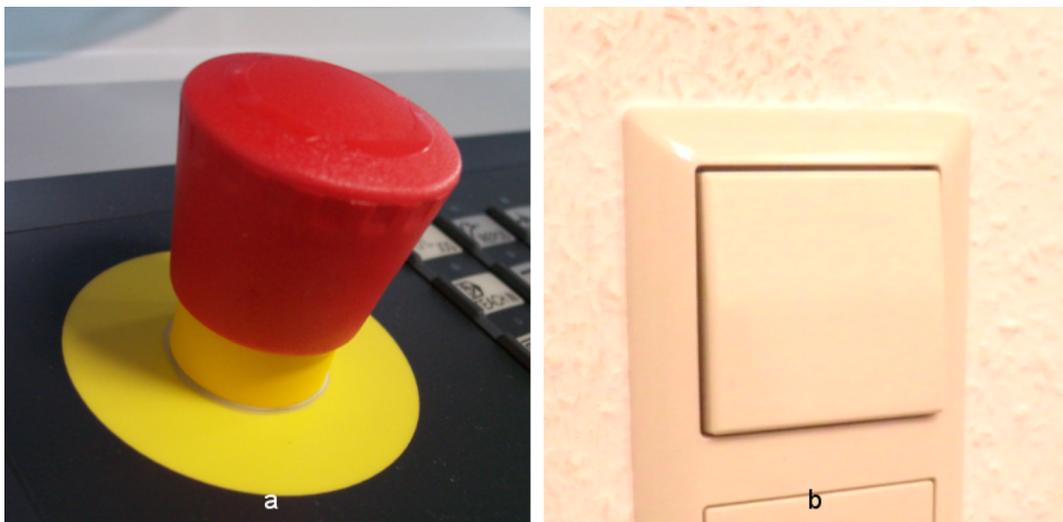


Abbildung 2.3: : (a) Abbildung eines Tastschalters [Quelle: Notaus]  
(b) Abbildung eines Kippschalters

### 2.4.3 Griff

Der Griff stellt ein Artefakt dar, welches entlang einer Achse bewegt wird. Der Griff wird für gewöhnlich entlang der y- oder der z-Achse vom Benutzer durch eine Hand- / Armbewegung zu sich gezogen. Abbildung 2.4 zeigt einen in der Realität vorkommenden Griff. Der Griff besitzt in der dortigen Ausprägung zwei diskrete Zustände und kann nach unten gezogen werden.



Abbildung 2.4: : Abbildung eines Griffes [Quelle: Griff]

### 2.4.4 Hebel

Der Hebel stellt das vierte Artefakt in der hier aufgeführten Liste an denkbaren Bestandteilen einer virtuellen Schalttafel dar. Die Abbildung 2.5 zeigt verschiedene Formen eines Hebels. Der Hebel ist in seiner einfachsten Form mit einer Hand, sprich einem Akteur zu bedienen. Es ist aber denkbar, auch Hebel mit zwei Greifflächen zu erstellen um eine Art Sicherung der Aktion zu erzeugen. Man muss demnach bewusst beidhändig zugreifen und den Schalter umlegen.

Der Hebel in seiner einfachsten Form hat zwei Zustände, in die eine Richtung gekippt, oder eben in die andere Richtung. Darüber hinaus sind aber weitere Modelle durchaus überlegenswert. Ein Hebel mit drei Zuständen, in dem der mittlere Zustand, der zwischen den Zuständen „an“ und „aus“ als undefiniert gilt und der Hebel beispielsweise in den ursprünglichen Zustand zurückfällt. Eine andere Möglichkeit den Hebel zu verwenden ist, ihn mit einer kontinuierlichen Ausgabe zu versehen, analog zu einem Gashebel.

Die Bewegung des Hebels erfolgt über eine Achse. Der Akteur selbst beschreibt, seitlich betrachtet, eine bogenförmige Bewegung.



Abbildung 2.5: : (a) Foto eines Einhandhebels [Quelle: Schalttafel]

### 2.4.5 Rad

Bei dem Rad, wie in 2.6 gezeigt, handelt es sich hierbei um ein kreisförmiges Artefakt, welches zentral an einer Achse aufgehängt ist und somit die Beweglichkeit auf eine Rotation um eine Achse einschränkt. Auch dieses Artefakt bietet die Möglichkeit, in mehreren Ausprägungen aufgebaut zu werden. Einmal für den einhändigen Betrieb, eventuell mit Knauf zum Anfassen und die Variante für beiden Hände, wo das Rad selbst dann mehrere greifbare Flächen besitzen muss.

Eine Besonderheit des Rades liegt darin, das dem Benutzer die Grenzen zum Drehen nicht offen ersichtlich sind. Ob das Rad mehrere Vollkreise vollziehen oder nur noch einen Halbkreis in die eine oder andere Drehrichtung vollziehen kann, ist vom Artefakt selbst nicht zu erkennen. Ist das Rad an dem einen oder anderen Ende der Bewegungsmöglichkeit angekommen, muss das dem Benutzer auf jeden Fall angezeigt werden.



Abbildung 2.6: : (a) Rad für eine Hand [Quelle: Handkurbel]  
(b) Rad für zwei Händige Benutzung [Quelle: Schalttafel]

### 2.4.6 Joystick

Der Joystick, siehe 2.7, als Letztes hier aufgeführtes Artefakt, weist einige Besonderheiten gegenüber den anderen Artefakten auf. Beim Joystick handelt es sich um einen Hebel, der als einziges Artefakt zwei Achsen besitzt, um die er manipuliert werden kann. Die Manipulation geschieht um die x- und z-Achse und kann gleichzeitig stattfinden. Darüber hinaus hat der Joystick eine Ruheposition, in die er nach kurzer Zeit zurückkehrt, sobald er losgelassen wird. Das Artefakt wird ausschließlich im diskreten Betriebsmodus betrieben. Es gibt insgesamt 9 Stadien, die das Artefakt besitzt. Zum einen die Ruheposition in der Mitte sowie acht weitere Stadien, jeweils bei voller positiver wie auch negativer Auslenkung über beide Achsen.



Abbildung 2.7: : Foto eines Joysticks mit diskretem Schaltverhalten

## 2.5 Gesten

Die Hände des Anwenders werden als Aktoren bezeichnet. Bei speziellem Bezug auf eine bestimmte Hand, wird die linke Hand als "Left-Hand" und entsprechend die rechte Hand als "Right-Hand" bezeichnet. Es können die jeweiligen Gesten der linken und rechten Hand unabhängig von einander ausgeführt werden. Jede Hand wird als einzelner Akteur im System abgebildet. Die Aktoren können sich dreidimensional bewegen. Ihre Bewegungen haben im Zusammenhang mit den weiter unten aufgeführten Gesten, Effekte auf die virtuelle Umgebung. Es werden vom Framework nur zwei Gesten benötigt um in Verbindung mit der Bewegung der Aktoren die Manipulationen im virtuellen Raum zu bewerkstelligen. Diese beiden Gesten werden im Weiteren aufgeführt.

### 2.5.1 Kontext

Genau wie das oben aufgeführte Midas-Touch Problem, treten die Herausforderungen mit dem Ermitteln des Kontextes, in dem eine Geste auftritt, erst im dreidimensionalen Bereich so richtig auf.

Im 2D-Bereich ergibt sich der Kontext bereits durch das Berühren der Fläche. Der darunter liegenden Desktop-Metapher zufolge wird durch das Berühren ein Element ausgewählt, in dessen Kontext man sich dann bewegt und welches ein bestimmtes Gestenrepertoire versteht.

Im 3D-Bereich hingegen kann man das Problem nur insoweit umgehen, indem man den Benutzer stets in nur einem globalen Kontext hält. Dieses Verfahren ist aber zunehmend ineffizient. Da man bei komplexeren Steuermöglichkeiten für jede Steuerung eine eigene Geste, bzw. eine eindeutige Abfolge bestimmter Gesten, benötigt. Will man dieses Problem lösen und dem Benutzer mehrere Kontexte zur Verfügung stellen, in denen man dann übergeordnet betrachtet, ob die gleichen Gesten unterschiedliche Auswirkungen haben, muss man der Gestenerkennung noch eine Kontextermittlung vorschalten.

### 2.5.2 Start- und Endgeste

Wie im Abschnitt 2.1.4 beschrieben, stößt man bei der Gestenerkennung im Raum auf die Problematik, dass Start und Ende einer Geste nicht ohne Weiteres erkannt werden können. Die folgende Herangehensweise zum Lösen sieht vor, dass zwei gesonderte Gesten definiert werden, eine Start- und eine Endgeste. Es handelt sich hierbei um kommunikative Gesten, da sie dem System getrennt von der eigentlichen Simulation etwas mitteilen. Sie haben keine direkte Auswirkung auf die Artefakte innerhalb der Simulation.

### 2.5.3 Steuerungsgesten

Alle hier folgenden Steuergesten werden von einer Start- und einer Endgeste umschlossen. Es ist hier hierbei allerdings zu Beachten, dass mehrere einzeln für sich geltende Steuergesten hintereinander geschaltet werden können, ohne dass sie jeweils von einer Endgeste beendet werden und die nächste Geste durch eine Startgeste eingeleitet wird. Als veranschaulichendes Beispiel hierfür sind der Schieberegler 2.4.1 oder der Joystick 2.4.6 zu nennen. Eine Abfolge kann in einzelne Steuergesten herunter gebrochen werden, und wird so auch vom System verarbeitet. Die Gestenerkennung selbst läuft aber kontinuierlich weiter.

#### Drückgeste

Die erste hier zu beschreibende Geste soll die Drück-Geste sein. Im weiteren Verlauf auch als „Drücken“ bezeichnet. Die Geste wird mit einer Hand ausgeführt. Welche Hand, ob die linke oder rechte Hand, ist unerheblich. Sie startet mit dem Schließen der Hand, also dem Greifen 4.1.2.4. Anschließend bewegt sich die Hand in gerader Linie vom Körper weg. Ein Abbruch der Geste ist möglich, indem einfach die Hand während der Bewegung geöffnet (siehe:4.1.2.4). Die Geste ist wiederholbar, das bedeutet, dass nach Ausstrecken der geschlossenen Hand, die Hand wieder an den Körper gezogen werden kann, um erneut dann in die gleiche Richtung ausgestreckt zu werden, welches ein zweites Mal „drücken“ zur Folge hätte. Abschließend wird die Hand geöffnet, also die Endgeste 4.1.2.4 ausgeführt. Die Abbildung 2.8(a) zeigt einmal die Drückgeste wie sie ausgeführt wird, auf Abbildung 2.8(b) ist das Zustandsdiagramm für die Drückgeste abgebildet.

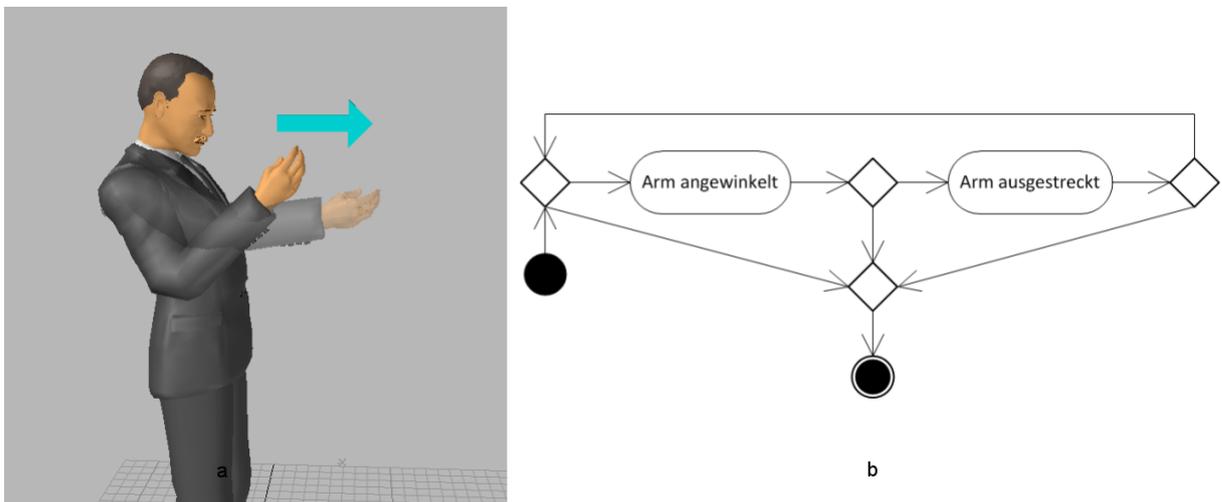


Abbildung 2.8: : (a) Drückgeste(b) Zustandsdiagramm für die Drückgeste

## Ziehgeste

Die Ziehgeste, in Kurzform als „Ziehen“ bezeichnet, kann von einer beliebigen Hand ausgeführt werden. Die Geste startet mit ausgestrecktem Arm und dem Schließen der Hand 4.1.2.4. Die Hand wird dann, wie in der Grafik 2.9 anschaulich dargestellt, an den Körper gezogen. Diese Geste ist nicht wiederholbar, was bedeutet, dass nach der Ziehgeste die Endgeste folgen muss. Nach dem Öffnen und dem anschließenden Ausstrecken der Hand kann die Geste wiederholt werden. Solange die Ziehgeste nicht vollendet wurde, ist ein Abbrechen, durch erneutes Öffnen der Hand, möglich. Eine Erweiterung der Ziehgeste besteht darin, dass sie mit beiden Händen gleichzeitig ausgeführt werden kann. Auch hier erfolgt, parallel zur ersten Variante, bei ausgestreckten Armen ein Schließen der Hände, ein synchrones an den Körper ziehen der Arme und anschließend das Öffnen beider Hände. In Abbildung 2.9 (a) und (b) werden beide Varianten in einem Zustandsdiagramm dargestellt.

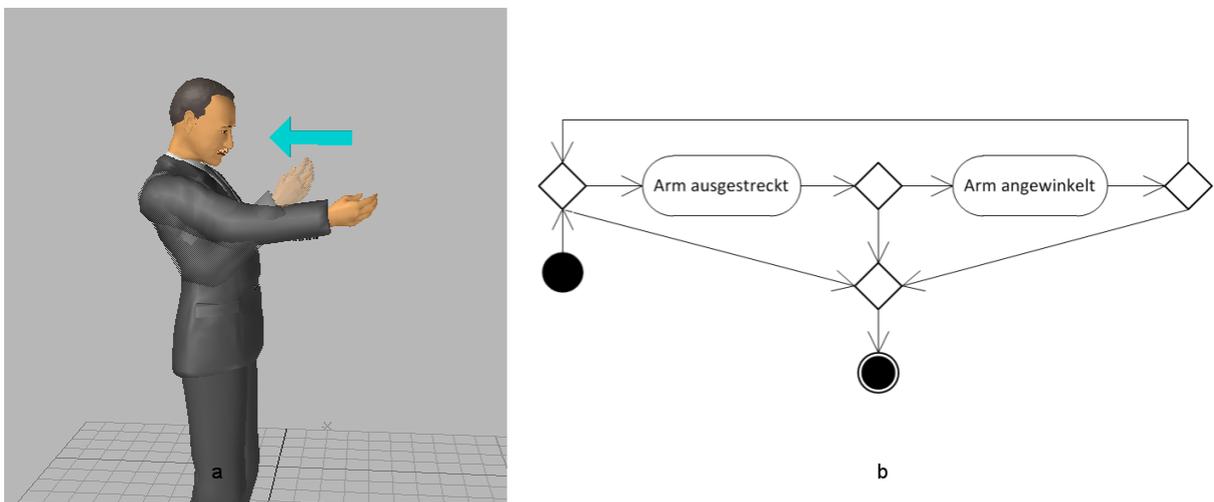


Abbildung 2.9: : Ziehgeste

## Schiebegeste

Die Schiebegeste wird mit einer Hand ausgeführt. Die Geste selbst erfolgt längs der x-Achse im Raum. Die Abbildung 2.10 (a) zeigt die Schiebegeste entlang dieser Achse. Bei der Schiebegeste handelt es sich um eine kontinuierliche Geste, dabei ist es denkbar ein Stück weit zu schieben, inne zu halten und dann in dieselbe Richtung die Bewegung fortzuführen. Darüber hinaus ist es möglich, die Bewegungsrichtung zu wechseln und in die entgegengesetzte Richtung zu schieben, ohne zwischendurch die Gestenerkennung durch ein Öffnen der Hand zu beenden. Hier gilt es also, mehrere Schiebegesten hintereinander richtig zu erkennen. 2.10 (c) beinhaltet das erstellte Zustandsdiagramm zur Schiebegeste.

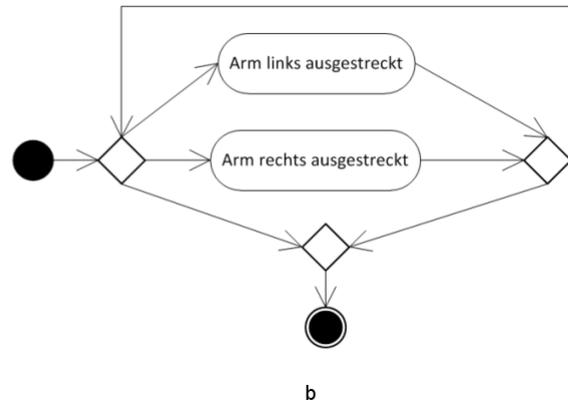


Abbildung 2.10: : Schiebegeste

**Hebe- / Senkgeste**

Eine weitere Geste ist die Hebegeste und ihr Pendant die Senkgeste, im weiteren Verlauf der Arbeit auch als „Heben“ und „Senken“ bezeichnet. Im Folgenden wird die Hebegeste genauer beschrieben. Diese Gesten kommen sowohl mit einer als auch mit beiden Händen vor. Wie in Abbildung 2.11(a) grafisch dargestellt, beginnt die Geste mit leicht ausgestreckten Armen, die in einem Winkel zum Boden zeigen. Wird nun die Hand geschlossen, und die Hand samt Arm nach oben bewegt, wird die Hebegeste vollführt.

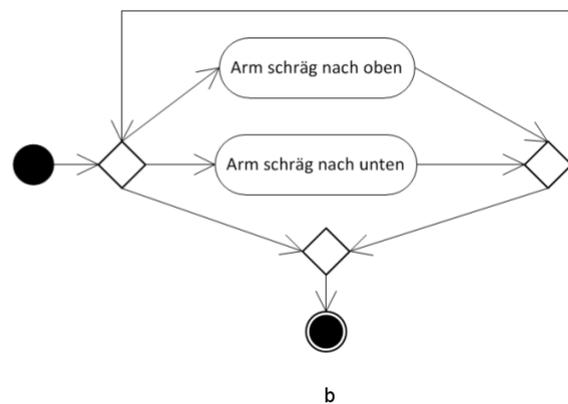
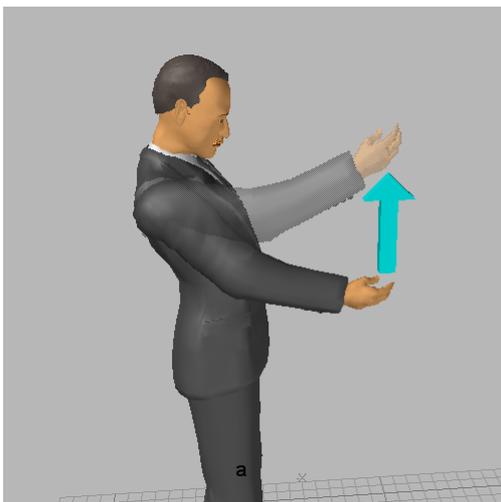


Abbildung 2.11: : Hebegeste

Die Geste ist nach dem Durchführen nicht sofort zu wiederholen, da die Arme erst wieder gesenkt werden müssen. Öffnet man die Hände hierbei nicht, wird die Senkgeste ausgeführt. Die Senkgeste ist das Gegenteil zur Hebegeste und verläuft analog zu der Hebegeste, nur das die Senkgeste bei geschlossene und erhobener Hand beginnt. Werden hingegen nach der Hebegeste die Hände geöffnet und die Arme gesenkt, wird keine Senkgeste vollführt. Der gesamte Ablauf der Hebe- und Senkgeste wird in 2.11(b) verdeutlicht.

### Drehgeste

Es handelt sich wie in Abbildung 2.12 zu erkennen ist, um eine kreisförmige Bewegung. Diese Bewegung kann sowohl im Uhrzeigersinn, wie auch gegen den Uhrzeigersinn ausgeführt werden. Die Drehgeste, im Weiteren auch als „Drehen“ bezeichnet, kann sowohl als einhändige wie auch als beidhändige Variante vorkommen. Die Geste selbst kann von einem Teilkreis bis hin zu mehreren Vollkreisen ausgeprägt sein. Die Drehgeste kann in Teilgesten unterteilt werden. Eine Teilgeste umfasst die Drehbewegung in eine Richtung, bis diese Bewegung stoppt. Im Anschluss kann losgelassen werden, die Gestenerkennung stoppt, oder aber die Bewegung kann fortgeführt werden, im oder gegen den Uhrzeigersinn. In der Abbildung 2.12 (b) ist ein Zustandsdiagramm für die Drehgeste abgebildet. Prinzipiell gibt es bei der beidhändigen Variante keinen Unterschied zur Geste mit nur einer Hand, außer, dass zu beachten ist, dass ein Vollkreis mit beiden Händen ohne zwischenzeitigem Absetzen der Geste nicht möglich ist, da ansonsten die Hände über Kreuz liegen würden.

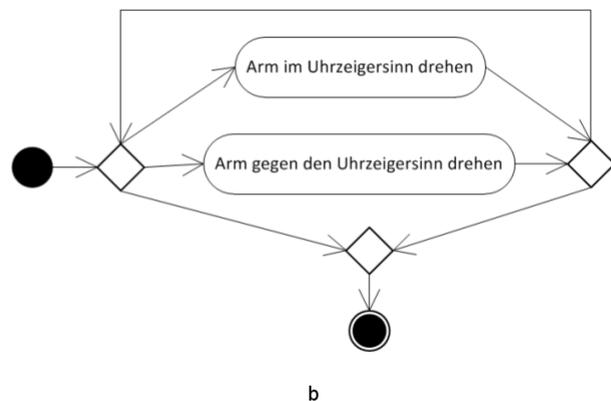
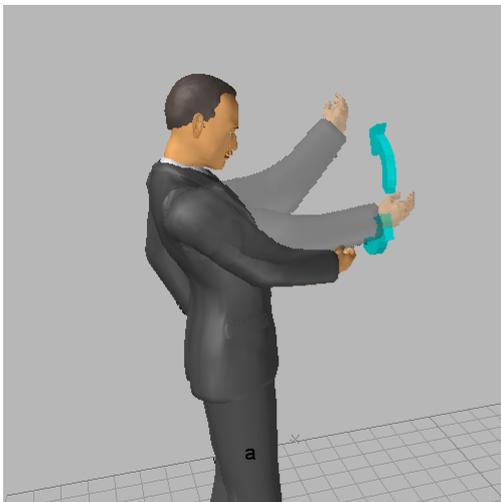


Abbildung 2.12: : Drehgeste

## 2.6 Funktionale Anforderungen

Es bestehen folgende Anforderungen an ein zu entwickelndes System

- Anbindung an verschiedene Sensoren
- Entwicklung einer physikgestützten Simulation
- Aufbereitung der Sensordaten um Hände in die Simulation übertragen zu können
- Getrenntes Erkennen der Start- und Endgeste
- Errichten der Schalttafel und ihrer Artefakte
- Erkennen der in 2.5.3 aufgeführten Gesten
- Eine grafische Ausgabe der Simulation

Die rasante Entwicklung in dem Bereich der Sensoren, der Physik- und Grafik-Engines, wie auch der Ausgabemittel, ist es geboten, ein Framework zu entwickeln, das diesem schnellen Wandel Rechnung trägt. Es muss die Möglichkeit geben, neue Sensoren oder andere Komponenten einfach und standardisiert in das System zu integrieren, ohne an dem Framework selbst große Anpassungen vornehmen zu müssen. Auch andere Teilsysteme müssen austauschbar. Hierfür wird eine lose Kopplung aller Komponenten benötigt, die über eine einheitliche Schnittstelle miteinander kommunizieren können.

### 2.6.1 Abgrenzung

Die Entwicklung eines Systems beschränkt sich auf den Anwendungsbereich im Innenraum eines Gebäudes. Zum einen sind hier die Bedingungen für Tests kontrollierbarer, zum anderen sind die meisten auf dem Markt erhältlichen Sensoren nur zur Benutzung im Innenraum gedacht.

Des Weiteren wird der Messbereich eingegrenzt, die Simulation ist ortsgebunden und kann nicht vom Benutzer bewegt werden.

Als letzte Abgrenzung ist noch zu erwähnen, dass von einem Aufbau für eine Person ausgegangen wird. Sämtliche zum Einsatz kommende Verfahren gehen davon aus, dass nur eine Person zu tracken und zu verarbeiten sein wird.

## 2.7 Erste Untersuchungen

In der Veranstaltung „Projekt 2“ im Wintersemester 2012/13 wurden erste praktische Tests durchgeführt. Die Details sind in der dazu angefertigten Ausarbeitung Potratz (2013) nachzulesen.

Für die ersten Versuche kam die Kinect for xBox als Sensor zum Einsatz. Der Versuchsaufbau umfasste zusätzlich zu dem Sensor noch eine Physiksimulation und ein extern zu steuerndes Programm.

Der Adapter der Kinect filtert alle Daten bis auf die Positionen der Hände aus. Diese Informationen wurden mittels eines UDP Broadcast Streams an die Physiksimulation übertragen und an die Repräsentanz der Hände, die Aktoren, weiter gegeben.

Die auf der Abbildung 2.13(a) zu sehende Joystick-Attrappe diente dem Benutzer als visuelles Feedback. An dieser Stelle befand sich der simulierte Joystick. Diese Attrappe verfügt weder über Anschlüsse, noch über Markierungen um vom Sensor erfasst zu werden.

Zu Beginn der Simulation wurde die Attrappe vor der Kinect positioniert. Der Benutzer fasste den Joystick wie üblich am oberen Ende an und diese, durch die Kinect ermittelte, Handposition wurde dann genutzt um den simulierten Joystick innerhalb der Simulation an dieser Stelle zu positionieren.

Nun konnte der Joystick mittels der Aktoren manipuliert werden, die so gewonnenen Steuerkommandos wurden an das externe Programm, in diesem Fall ein Computerspiel, auch mittels UDP-Paketen übertragen. Die Aussage erfolgte mittels eines Beamers, wie auf Abbildung 2.13(b) zu sehen ist.

Als Ergebnis dieses Versuchs stellte sich heraus, dass der Aufbau grundsätzlich funktionierte. Einzige Verzögerungen durch den Sensor kam als Kritikpunkt zum Tragen.

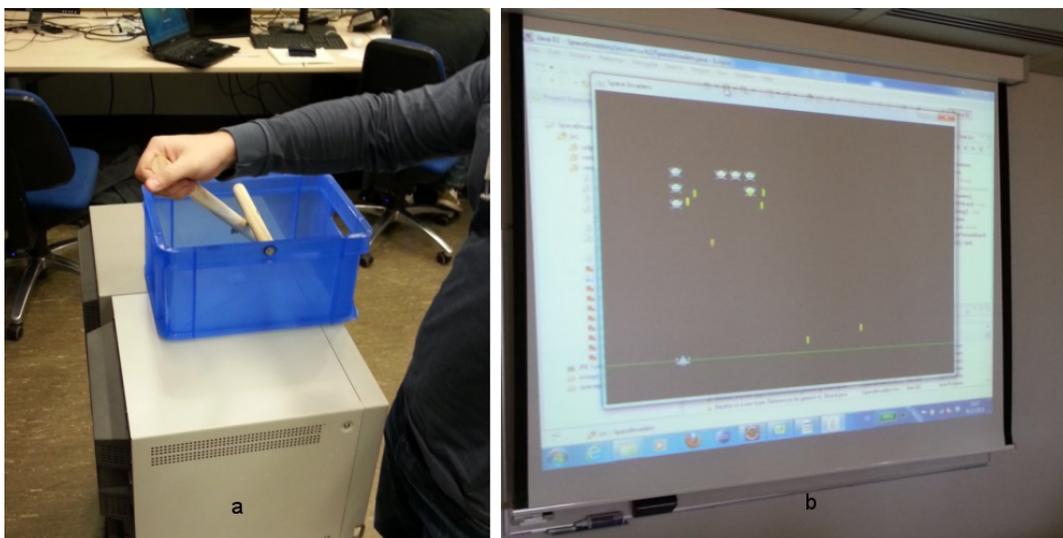


Abbildung 2.13: (a) Joystick-Attrappe  
(b) Sicht des Benutzers auf das zu steuernde Computerspiel

# Kapitel 3

## Algorithmen

Dieses Kapitel führt die zugrunde liegenden Algorithmen auf, die für diese Arbeit im Bereich des ART-Tracker Adapters und der Gestenerkennung Verwendung finden. Die Reihenfolge der verwendeten Verfahren ist so angeordnet, dass man die Handbewegung des Benutzers, über die verschiedenen Abstraktionslevel, bis hin zur Manipulationsgeste nachvollziehen kann. Dabei behandeln die Unterkapitel 3.1, 3.2 und 3.4 spezifische Verfahren, die es bei der Umsetzung mit den ART-Tracker Sensor zu implementieren galt. Die im Unterkapitel 3.3 aufgeführten Verfahren sind unabhängig vom ART-Tracking System und könnten auch bei anderen Sensoren Verwendung finden.

### 3.1 Aufbereitung der ART-Trackerdaten

Die Daten des Motiontrackers werden in einzelnen Paketen übermittelt. Jedes dieser Pakete beinhaltet alle zu einem Zeitpunkt erkannten Punkte. Sie bilden also eine Momentaufnahme. Die übermittelten Daten innerhalb der ART-Tracker Pakete sind verrauscht. Verrauscht bedeutet in diesem Fall, dass die Datenpakete Punkte enthalten können, die nicht von den Handschuhen stammen. Überdies kommt es vor, dass nicht alle Punkte der Handschuhe bei jedem Abtastzyklus erkannt oder deren Position korrekt erkannt wurden. Der ART-Tracker durchläuft bis zu 30 mal pro Sekunde diesen Abtastzyklus, der im weiteren Verlauf auch als Frame bezeichnet wird. Bei der Aufbereitung der Daten gilt es vor allem dieses Rauschen herauszufiltern und so gut wie möglich zu minimieren. Die folgenden hier aufgeführten Verfahren werden unter dem Begriff Disambiguierung zusammengefasst.

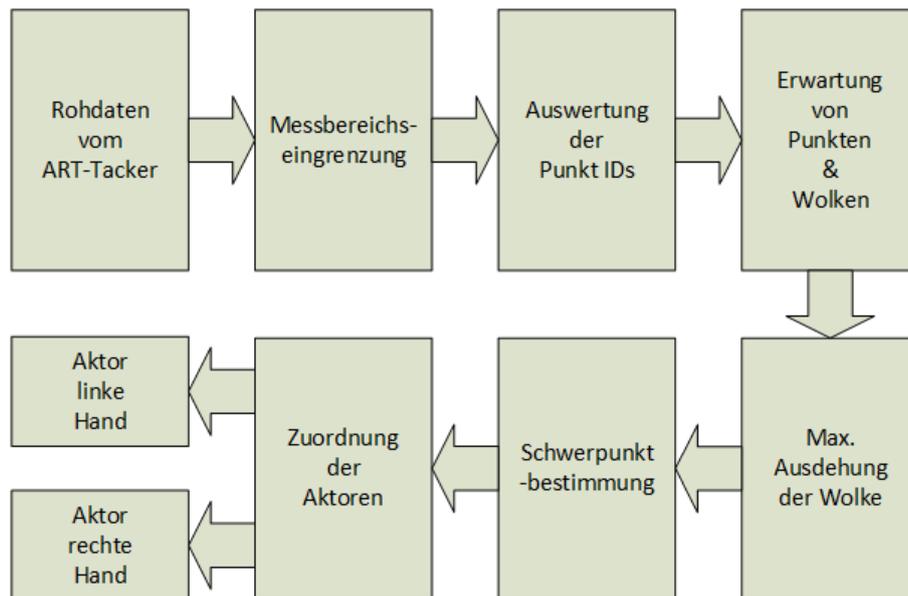


Abbildung 3.1: : Darstellung der maximalen Ausdehnung der Finger

**Messbereichseingrenzung** In einem ersten Schritt sollen alle ermittelten Punkte entfernt werden, bei denen sichergestellt ist, dass sie nicht von den Handschuhen stammen können. Der relevante Messbereich wird zuerst in Form eines Quaders abgegrenzt. Alle Punkte, die außerhalb dieses Bereichs liegen, können verworfen werden. Dieses Vorgehen erlaubt die Beseitigung von Punktdaten, die durch Reflexion oder Spiegelung an Oberflächen entstehen oder aber auch das unfreiwillige Tracking von ART-Trackerartefakten, die sich im Sichtfeld der Kameras befinden.

Der Messbereich ist mit der später verwendeten Simulationsumgebung abzustimmen. So wird sichergestellt, dass nur Messpunkte verarbeitet werden, die auch im Simulationsraum dargestellt werden können. Die Handschuhe befinden sich im Regelfall im Laufe einer Messung relativ mittig im Messbereich, da dort der Proband steht. Somit ist der relevante Bereich für die Messung klar abzugrenzen.

**Auswertung der Punkt IDs** Das einzige Hilfsmittel, welches vom ART-Tracking System zur Verfügung gestellt wird, sind die IDs der einzelnen Punkte. Die im weiteren Verlauf als Punkt IDs bezeichneten Werte sind die eindeutigen Identifikationsnummern der Punkte. Diese Eindeutigkeit bleibt über alle Frames erhalten. Solange die Trackingsoftware von sich aus sicherstellen kann, dass ein Punkt über mehrere Frames immer derselbe ist, wird dieser Punkt in den Daten auch mit der selben ID versehen. Kann die Software dies nicht sicherstellen, bekommt der Punkt eine neue, noch nicht verwendete ID zugewiesen. Die Verfolgung

der Punkte über mehrere Frames durch die Trackingsoftware ist allerdings verhältnismäßig unzuverlässig. Punkte bekommen häufig neue IDs zugewiesen. Somit scheiden die Punkt IDs als alleiniges Identifikationsmerkmal aus. Sie können aber für die nachfolgenden Verfahren als Stütze dienen, sofern IDs über mehrere Frames vorhanden sind.

## 3.2 Abstraktion auf Aktoren (ART)

**Erwartungshaltung an Anzahl von Punkten und Wolken** Da das Gesamtsystem auf eine einzelne Person ausgelegt ist, siehe 2.6.1, muss als Erwartungswert von zwei Händen mit je fünf Markern ausgegangen werden. Pro Abtastdurchlauf können also maximal 5 Punkte für jede Hand erkannt werden. Auf Grund der in Abschnitt 3.4 zu verwirklichenden Start- 4.1.2.4 und Endgeste 4.1.2.4 und den damit auftretenden Herausforderungen, wird auf eine Minimalgrenze an Punkten pro Frame verzichtet.

Als Grundlage der Wolkenfindung wird auf Datensätze mehrerer aufeinander folgender Frames gesetzt. Das Ganze ist als Ringpuffer umgesetzt, in diesem werden die Daten des ältesten Frames durch Daten des neuen Frames ersetzt. Die maximal zu erwartende Punktzahl steigt mit der Anzahl der verwendeten Frames. Bei einem Frame 5 Punkte pro Hand, bei zwei Frames 10 Punkte pro Hand und so weiter. Diese Obergrenze dient später dazu, bei einer errechneten Wolke festzustellen ob sie eventuell zu viele und somit nicht korrekte Punkte besitzt. Ist dies der Fall, muss noch einmal untersucht werden welche Punkte in der Wolke aussortiert werden müssen.

Wie anfangs beschrieben ist eine Untergrenze pro Frame nicht praktikabel. Allerdings wird eine Untergrenze benötigt, um die Wahrscheinlichkeit zu erhöhen, dass es sich bei den aufgezeichneten Punkten und deren Wolke tatsächlich um eine aufgezeichnete Hand handelt. Es wird daher eine Mindestanzahl von Punkten über alle gerade verwendeten Frames genommen. Punktwolken müssen über die fünf Frames hinweg mindestens 10 Punkte besitzen. Dies gilt für alle Wolken, die als mögliche Abbildung einer Hand infrage kommen wollen. Zusätzlich gibt es noch Kriterien, die die Güte der Punktwolken erhöhen. Besitzt eine Wolke mindestens einen Frame in dem 4 Punkte einer Wolke zugeordnet werden können, oder zwei Frames in denen mindestens 3 Punkte existieren, wird diese Güte erhöht. Somit eröffnet sich die Möglichkeit auch Frames zu akzeptieren, in denen durch ungünstige Umstände eventuell nur zwei oder weniger Punkte der Hand übertragen wurden. Die Güte spielt dann eine Rolle, wenn es mehr als zwei mögliche Punktwolken gibt und es zu entscheiden ist, bei welchen es sich um Hände handelt.

**Maximaler Ausdehnung der Wolke** Die Ausdehnung der zu erwartenden Punktwolke wird

durch mehrere Faktoren bestimmt. Zum einen spielt hier der maximale Abstand der am weitesten von einander entfernten Markern eine Rolle. Hierbei wird es sich im Regelfall um den Daumen und den kleinen Finger handeln, siehe Abbildung 3.2. Im ersten Schritt sollte man hier einen durchschnittlichen Wert annehmen, sich später aber die Option offen lassen, diesen Abstand individuell einzumessen. Ein weiterer Punkt, der bei der Ausdehnung betrachtet werden muss, ist, dass sich bei ersten Versuchen mit dem Handschuh und dem ART-Tracking System das Phänomen ergab, dass die Tiefenwahrnehmung nur bei geöffneter Hand wirklich zuverlässig arbeitet. Schließt man die Hand, ergeben sich Fehlmessungen längs der z-Achse im Raum. Damit man beim Greifen ein ähnlich zuverlässiges Tracking gewährleisten kann, bedarf es hier noch einiger Erweiterungen am Wolken Modell. Die Erwartungshaltung an die Wolke wird dahin gehend verändert, dass nicht mehr von einer klassischen Kugelform ausgegangen werden kann, sondern von einer Kugel die längs der z-Achse gedehnt ist. Somit können auch Punkte noch mit einbezogen werden, die fälschlicherweise einen zu hohen oder zu niedrigen z-Wert bekommen haben. Damit dieses Verfahren zielführend ist, bedarf es Festlegungen über Haltung der Hände des Benutzers, die später in diesem Kapitel aufgeführt werden, siehe dafür den Abschnitt über die erwartbaren Posen 3.3.2.

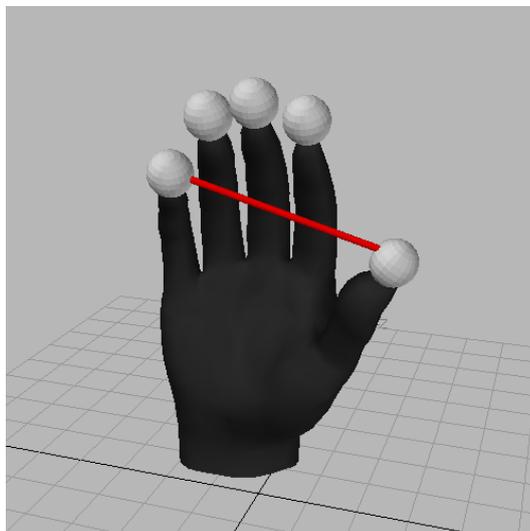


Abbildung 3.2: : Darstellung der maximalen Ausdehnung der Finger

**Schwerpunkte der Wolke bestimmen** Nachdem alle übertragenen Punkte zu Punktwolken zusammengefasst oder verworfen wurden, rücken die Punktwolken selbst in den Fokus. In einem ersten Schritt wird der Schwerpunkt der Punktwolke errechnet. Für

die Berechnung des Schwerpunkts wird das arithmetische Mittel aus den drei Raumachsen gebildet. Die x-, y- und z-Koordinaten der Punkte werden mit folgender Formel verarbeitet.

$$X_s = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$$

Die so erhaltene Punktkoordinate im Raum ist der Schwerpunkt der Wolke und wird für den Mittelpunkt des Aktors verwendet. Aufgrund der asymmetrischen Auslegung der Hand von 4 Fingern zu einem Daumen, ist der Schwerpunkt dichter an den Fingern gelagert. Dies führt zu einer Unschärfe bei der Bestimmung des eigentlichen Mittelpunktes der Hand. Diese Unschärfe verringert sich zunehmend, wenn man die Hand schließt. Da der Benutzer aber zu jeder Zeit über ein visuelles Feedback seiner Aktoren verfügt, kann diese Unschärfe vernachlässigt werden, da der Benutzer dieses unbewusst ausgleicht.

**Abbildung der Punktwolken auf Aktoren** Nachdem alle Schwerpunkte der Wolken berechnet wurden, werden alle Wolken potenziell zusammengefasst und dienen somit als möglicher Standort für die in der Simulation befindlichen Aktoren. Die Anzahl der Wolken spielt hier keine Rolle. Sollten mehr als zwei Punktwolken zur Verfügung stehen, wird im späteren Verlauf eine Aussortierung stattfinden.

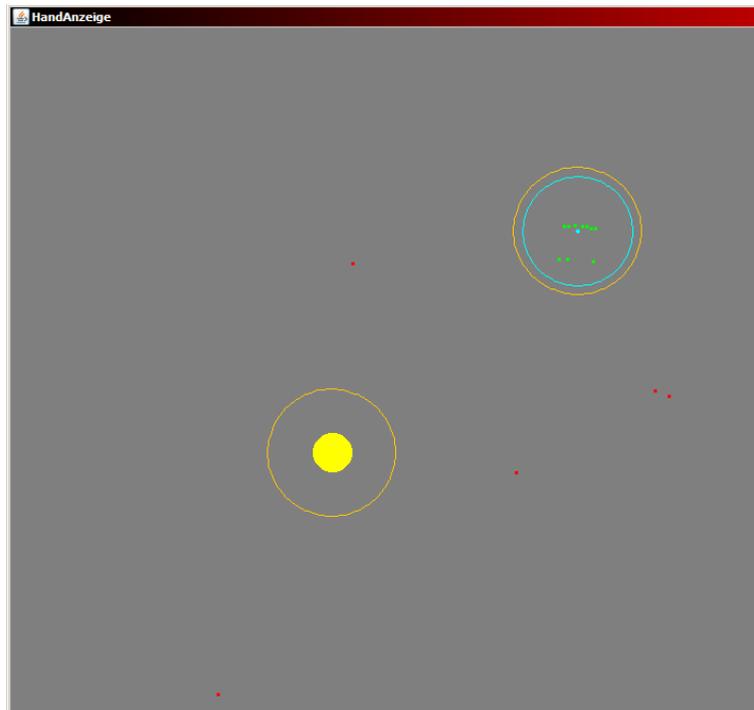


Abbildung 3.3: : Darstellung der Punktwolkenanalyse: links geschlossene Hand, rechts offene Hand mit einzelnen Punkten

### 3.3 Aktorzuordnung

Um die im vorherigen Abschnitt gefunden potenziellen Aktoren möglichst zuverlässig und korrekt den tatsächlichen Aktoren zuordnen zu können, bedarf es verschiedener Annahmen. Damit diese Annahmen auch möglichst häufig zutreffen, müssen bestimmte Voraussetzungen vom Versuchsaufbau und der entsprechenden Versuchsperson erfüllt werden. Im folgenden Abschnitt wird auf diese Voraussetzungen und Hypothesen genauer eingegangen.

#### 3.3.1 Anzahl der Hände

Der Versuchsaufbau ist auf eine Person ausgelegt. Folglich ergibt sich die Hypothese, dass maximal zwei Hände vorkommen können. Es muss also gewährleistet sein, dass null bis zwei Punktwolken zuverlässig den Aktoren zugeordnet werden können. Falls es mehr als zwei Positionen geben sollte, gilt es, die richtigen herauszufiltern.

Ist keine Versuchsperson innerhalb des Messbereichs können folglich auch keine Hände und somit Punktwolken gefunden werden. In dem Fall müssen alle bis dahin errechneten Daten gelöscht werden und das System in seinen algorithmischen Startzustand versetzt werden.

Werden ein oder zwei Punktwolken und somit potenzielle Aktoren ermittelt, kommen die hier weiter aufgeführten Verfahren zum Einsatz. Sollten sich mehr als zwei Punktwolken ermitteln lassen, werden alle die Punktwolken verworfen, die am ehesten nicht zu der Person gehören können. Auch hier kommen die im Anschluss aufgeführten Verfahren zur Anwendung.

### 3.3.2 Hypothese: Normale Posen

Es gibt bestimmte Verhaltensweisen, die von der Testperson erwartet werden. Dies umfasst vor allem die Anzahl möglicher zu erwartender Posen. Die Posen, die erwartet werden, und von denen zur späteren Berechnung ausgegangen wird, werden im folgenden aufgeführt.

**Stand der Person** Der Proband wird sich in aller Regel im aufrechten Stand oder aber sitzend im Messbereich befinden. Diese Vorhersage schränkt die mögliche Lage der Aktoren ein. Dies kommt besonders dann zum Tragen, wenn mehr als zwei Punktwolken existieren. Hier werden die Punktwolken die sich mittig im Messbereich befinden bevorzugt. Erst wenn sich keine Werte im zentralen Bereich des Messraums befinden, werden Punktwolken an den Rändern berücksichtigt.

**Abstand der Hände** Um die Berechnung und Zuordnung der Aktoren zu ermöglichen, ist es essentiell, dass die Hände mindestens eine Handbreite Abstand voneinander haben. Dieser Abstand verhindert, dass beide ermittelten Punktwolken zu einer zusammen gefasst werden und somit eine der beiden Hände keine Aktorzuordnung mehr bekäme. Darüber hinaus ist der maximale Abstand zweier Hände wichtig. Werden beide Arme seitlich, im rechten Winkel vom Körper weg gestreckt, ergibt die Distanz der Hände den maximal Abstand. Dieser Abstand ist abhängig von der Größe der Person. Zur Verwendung sollte anfangs von einem Wert ausgegangen werden, der einer durchschnittlich großen Person zuzuordnen ist. Dieser Anfangswert kann durch ein exemplarisches Einmessen ermittelt werden. Grundsätzlich sollte der Wert aber variabel gehalten werden, damit spätere Anpassungen, durch vorheriges individuelles Einmessen der Person zum Start eines Versuches, möglich bleiben.

**Bewegungsgeschwindigkeit der Hände** Trotz hoher Abtastrate des ART-Trackers von 30 Frames pro Sekunde (siehe ART-Anleitung[ARTtracker-Anleitung (2007)]), ist es notwendig, die Hände nicht zu schnell zu bewegen. Eine zu schnelle Bewegung führt dazu, dass ein zu großer Abstand zwischen den einzelnen erfassten Punkten entsteht und die Verfahren in 3.3.3 ins Leere laufen.

Unterschieden werden hier benötigte Posen mit und ohne Verwendung des Headtrackers, der am Headup-Display 5.11 befestigt ist.

### Ohne Position des Kopfes

**Blickrichtung** Die Hauptblickrichtung der Person ist auf die Powerwall. Es wird erwartet, dass sich die Testperson im Laufe eines Versuchs nicht umdreht und dann mit dem Rücken zur Powerwall steht, da ansonsten die Zuordnung von linker und rechter Hand nicht mehr zutrifft.

**Position und Haltung der Arme** Die Testperson sollte Hände und Arme nicht kreuzen, da ansonsten, unter der Voraussetzung der Blickrichtung in Richtung Powerwall die Zuordnung der Aktoren vertauscht wird. Im Regelfall werden die Hände maximal zur Körpermitte geführt. Sollte eine Bewegung der Hand über die Körpermitte hinaus gehen, ist darauf zu achten, dass die andere Hand mehr Abstand zum Körper hält, damit eine stimmige Zuordnung erhalten bleibt.

### Mit Position und Blickwinkel des Kopfes

Wird der Headtracker verwendet, ergeben sich daraus Vorteile zur Zuordnung von Aktoren. Es können darüber hinaus auch weitere Einschränkungen im Suchraum vorhandener Punktwolken vorgenommen werden.

**Blickrichtung** Eine fest vorgegebene Hauptblickrichtung ist in diesem Fall nicht nötig, da durch das Tracking die genaue Lage des Kopfes und somit auch eine genauere Lage der Person bestimmbar ist. Sollte sich der Proband um  $180^\circ$  um die eigene Achse drehen, würde das System dies registrieren und folglich die Erwartung wo linke und rechte Hand zu suchen sind, tauschen.

**Position und Haltung der Arme** Solange die beiden Hände und Arme nicht über Kreuz gelegt werden, kann die korrekte Zuordnung der Aktoren erfolgen. Sollten sich beiden Hände links oder rechts vom ermittelten Kopf befinden, wird der Aktor, der dichter am Kopf ist, der Hand der gegenüber liegenden Seite zugeordnet.

**Abstand der Hände zum Kopf** Wird die Position des Kopfes als Mittelpunkt der Person gesehen, kann von ihr aus mit dem Radius einer Armlänge eine Kugel errechnet werden, in der sich beide Hände befinden müssen. Diese Vorstellung ist etwas vereinfacht, da die Arme nicht direkt am Kopf beginnen und der genaue Abstand ohne spezielle Posen beim Einmessen nicht ermittelbar wäre. Außerdem erlaubt das Schultergelenk dem Arm keine halbkugelförmige Bewegungsfreiheit.

### 3.3.3 Erwartungshaltung an Aktorbewegung

**Verhalten der Hände in Bewegung** Davon ausgehend, dass es sich bei den zu verfolgenden Punktwolken um zwei Hände handelt, ergeben sich Erwartungshaltungen an diese

Wolken. Eine dieser Erwartungen ist, dass die beiden Punktwolken nicht ohne weiteres verschwinden können. Voraussetzung dafür ist, dass die Hände den Messbereich der Anlage nicht verlassen. Darüber hinaus werden die Hände einen kontinuierlichen Weg beschreiben. Die Hände und somit auch die Punktwolken können zwischen zwei Messungen nur eine bestimmte Wegstrecke zurückgelegt haben, daher springen die Punktwolken nicht. Misst man Bewegungsrichtung und Geschwindigkeit, kann man eine Aussage darüber machen, wo die Wolke sich ungefähr bei der nächsten Messung befinden wird.

### 3.3.4 Startposition der Aktoren

Allen Versuchen, eine möglichst reibungslose und korrekte Verfolgung der Hände zu erzielen, zum Trotz, kann es immer wieder vorkommen, dass entweder die Verfolgung der Hände oder aber die Zuordnung fehlschlägt. Um in diesem Fall möglichst schnell weiter arbeiten zu können, stellt sich der Benutzer gerade in seine Ausgangsposition und streckt beide Arme links, bzw. rechts vom Körper, in einem möglichst 90°-betragenden Winkel, ab. Die Hände befinden sich nun in besonders beobachteten Bereichen, werden hier Punktwolken erkannt, wird die gesamte Erkennung zurückgesetzt und davon ausgegangen, dass die dort vorhandenen Punktwolken die Hände des Benutzers sind.

Darüber hinaus, wenn das System die Bewegung einer Hand nicht mehr verfolgen kann und deren Spur verliert, der Aktor in der Simulation auf die initiale Position gesetzt. Dies soll den Benutzer dazu animieren dort wieder hin zugreifen und somit die Zuordnung von Punktwolke und Aktor sauber neu zu beginnen.

## 3.4 Gestenerkennung (ART)

Die Adapter der einzelnen Sensoren übernehmen den ersten Teil der Gestenerkennung. Es muss aus den vorhandenen Daten ermittelt werden, ob die Hand offen oder geschlossen ist, da die sogenannte Greifgeste 4.1.2.4 als Startgeste zur Gestenerkennung verwendet wird. Als Endgeste gilt das Öffnen der Hand, im folgenden Abschnitt werden beide Gesten gemeinsam genauer erläutert.

**Greif- und Loslassgeste** Es gibt prinzipiell mehrere Möglichkeiten anhand einer Punktwolke zu untersuchen, ob die Hand geöffnet oder geschlossen ist. In der Arbeit [Potratz (2011)] wird bereits ein Verfahren beschrieben, das den Flächeninhalt eines umspannenden Polygons berechnet. Dieses Verfahren hat den ConvexBottom-Algorithmus als Vorbild, welcher aus [GeoLab] entnommen wurde. Über den so entstehenden Flächeninhalt sollte dann ein Greifen oder Geöffnetsein, der Hand ermittelt werden.

Das im Weiteren verwendete Vorgehen sieht allerdings etwas anders aus. Basierend auf der Tatsache, dass der Daumen den größten Abstand zu einem errechneten Schwerpunkt einer Punktwolke hat, soll dieser Abstand zu Hilfe genommen werden um zu berechnen, wann gegriffen wird. In Abbildung 4.1(a) ist eine geöffnete Hand zu sehen und in Abbildung 4.1(b) die gleiche Hand, wenn sie greift. Im Fall des ART-Trackers sei noch erwähnt, dass das Greifen wie in Abbildung 4.1 anders aussehen sollte, da das Ballen einer Faust, die Reflektionskugeln einschließen und folglich ein Tracking verhindern würde. In dem Fall wäre ein Schließen der Hand mit ausgestreckten Fingern nötig.

Wird die geöffnete Hand nun zu einer sich schließenden Hand geformt, verringert sich der Abstand des Daumens zu den anderen vier Fingern und somit auch zum Schwerpunkt. Wird ein bestimmter Abstand unterschritten, gilt die Hand als geschlossen und der Aktor wird in den Status Gegriffen versetzt. Dabei gilt es zu beachten, dass die Grenzen von Gegriffen und Geöffnet nicht bei einander liegen. Es gibt im Abstand noch einen Bereich dazwischen, eine Grauzone, wo die Hand als undefiniert gilt. Solange sich der Abstand des Daumens in dieser Grauzone bewegt, wird der alte Zustand, in unserem Beispiel also geöffnete Hand, beibehalten. Erst wenn die Grenze zum Schließen unterschritten ist, wechselt der Status.

Wird die Hand wieder geöffnet, gilt die Hand solange als geschlossen, bis die Grenze zum Geöffnet sein überschritten wird. Das verwenden einer Grauzone soll ein andauerndes Öffnen und Schließen der Hand verhindern wenn sich der Daumen der Grenze nähert. Die zu wählenden Grenzwerte sind aus Versuchen zu ermitteln. Die Werte sind variabel zu halten, um sie später durch ein vorheriges Einmessen der Hände der Testperson verfeinern zu können.

# Kapitel 4

## Design

Im ersten Teil dieses Kapitels wird sich mit der Funktionalität befasst, welche ein System bieten muss, das den Anforderungen aus 2 genügen will.

Danach folgt die nähere Beschreibung einer Simulation, in der die Gestenerkennung stattfinden soll. Mögliche Gesten werden in Gruppen unterteilt und es werden unterschiedliche Verfahren zur Kontextbestimmung vorgestellt. Anschließend erfolgt die Erklärung, wie die eigentliche Gestenerkennung stattfinden soll.

Im zweiten Teil des Kapitels werden verschiedene Sensoren aufgeführt und deren Funktionsweise erläutert. Unterschiedliche Ausprägungen der Artefakte werden gezeigt, und auch wie diese jeweils der Gestenerkennung dienen können. Abschließend wird auf die Verfahren zur Visualisierung eingegangen und erklärt, wie die Steuerung eines externen Programms ermöglicht werden kann.

Das letzte Teilkapitel befasst sich mit der dem System zugrunde liegenden Software-Architektur. Es wird ein verteiltes System vorgestellt, das mittels Message Broker kommuniziert.

### 4.1 Funktionalität

Dieser Abschnitt umfasst die Funktionsweisen für die geforderte Gestenerkennung unter der Verwendung einer Physiksimulation. Nach einer Abgrenzung werden Gruppen erläutert, in die Gesten innerhalb einer Physiksimulation eingeordnet werden können. Anschließend werden Joints vorgestellt und anschließend erfolgt ein Überblick über unterschiedliche Verfahren der Kontextwahl. Als Abschluss wird auf Steuergesten näher eingegangen.

#### 4.1.1 Abgrenzung der Sensorik

Für die Umsetzung werden der ART-Tracker mit passiven Markern und die Kinect for Windows als Sensorkomponenten verwendet. Andere Sensortechniken mit ihren spezifi-

schen Eigenheiten und darauf angepassten Verfahren um dort die Hände und eine Start-Endegesten zu finden, werden im weiteren nicht berücksichtigt. Es soll sich bei der Umsetzung um einen Prototypen handeln und keinesfalls um ein fertiges Produkt.

## 4.1.2 Gestenerkennung in der Physiksimulation

### 4.1.2.1 Handlungsgruppen für Gesten

Wie in [Potratz (2011)] beschrieben, kann man die Gesten innerhalb einer Physiksimulation in vier Kategorien einteilen. Eine Gruppe umfasst die Handlungen die notwendig sind, um auf eine Geste schließen zu können. Diese vier Gruppen werden im Anschluss kurz aufgeführt.

**Konstruktion** Eine Konstruktionsgeste liegt vor, wenn zwei einzelne physikalische Gegenstände sich verbinden und so ein neues Assembly erschaffen. Im Gegensatz zu der Auslegung in [Potratz (2011)], ist es nicht nur möglich Assemblys durch Kollision zu erschaffen, sondern auch, wenn sie durch Joints miteinander verbunden werden.

**Destruktion** Es liegt eine Destruktionsgeste vor, falls ein bestehendes Assembly aufgelöst wird. Es ist dabei unerheblich, ob andere Assemblys durch dieses Handeln entstehen oder nicht.

**Transformation** Das Assembly als ganzes bleibt erhalten, nur die beteiligten Einzelkomponenten verändern ihre Position oder Anordnung zu einander.

**Manipulation** Die Manipulation kann die Position und die Rotation des Assemblys im Raum betreffen. Hierbei wird das gesamte Assembly bewegt.

### 4.1.2.2 Joints

Ein Joint ist ein innerhalb einer Physiksimulation verwendetes Verbindungsstück zwischen zwei einzelnen physikalischen Objekten. In diesem Zusammenhang sei auf die Arbeit von Russel Smith [Smith (2002)] verwiesen, in der verschiedene Arten von Joints in seiner Arbeit aufführt. Philipp Roßberger beschreibt Joints in seiner Arbeit [Roßberger (2008)] als Feder-Masse-Systeme.

Grundsätzlich umfasst der Begriff Joint sehr viele Arten von Möglichkeiten, wie Gegenstände miteinander verbunden werden können. Es besteht die Möglichkeit, die Aktionen, die zwei Objekte die mittels Joints verbunden sind, einzuschränken. Bestimmte Joints erlauben nur Rotationen um ausgewählte Achsen, andere erlauben nur innerhalb bestimmter Grenzen die Bewegung eines der Objekte, bevor das andere Objekt beeinflusst wird.

Für das zu entwickelnde System sind mehrere Arten von Joints nötig. Welche, die die Bewegungsmöglichkeiten einschränken, und welche die, ähnlich eines Gummibands, Kräfte von

einem Objekt auf das andere Objekt übertragen können. Es gibt keine einheitlichen Umsetzungen in den verschiedenen Physik-Engines. Somit gilt bei der Wahl einer Engine darauf zu achten, ob die dort vorrätigen Joints die Anforderungen erfüllen können.

#### **4.1.2.3 Kontextwahl**

Innerhalb einer Physiksimulation besteht die Kontextwahl darin, das entsprechende Artefakt auszuwählen, mit dem interagiert werden soll und das somit zur Gesteninterpretation herangezogen wird. Diese Artefakte bestehen nun selbst aus einzelnen Teilobjekten. Jedes Artefakt wird mit einem oder mehreren Objekten bestückt, mit denen ein Aktor sich verbinden kann. Diese Objekte werden Greifobjekte oder kurz Griffe genannt. Es wird der Kontext des Artefaktes gewählt, mit dessen Griff sich ein Aktor verbindet.

Im folgenden werden vier verschiedene Möglichkeiten aufgeführt wie so eine Verbindung entsteht und wie lange sie Bestand hat.

##### **Kontextwahl mittels Reibung**

Der Aktor berührt den Griff eines Artefakts, die Physik-Simulation errechnet anhand von Oberflächeneigenschaften und auftretenden Kräften die entstehende Reibung. Das Physikobjekt wird ausschließlich durch die errechneten Reibungskräfte manipuliert. Die Kontextwahl entsteht implizit durch die Berührung beider Objekte. Bei dieser Variante sollten die Griffe so geformt sein, dass möglichst eine große Oberfläche für die Reibung zur Verfügung steht. Entfernt sich der Aktor wieder und die Berührung zwischen den beiden Gegenständen endet, ist auch die Wahl des Kontextes wieder aufgehoben.

##### **Kontextwahl mittels Kollision**

Der Aktor schneidet das Greifobjekt. Beide Objekte werden nun mit einem Joint miteinander verbunden und der Kontext ist gewählt. Der Joint ermöglicht die Kräfteübertragung von Aktor an den Griff und somit an das Artefakt selbst. Die Kontextwahl besteht solange, wie Griff und Aktor sich schneiden, sobald diese Kollision nicht mehr besteht, erlischt auch der Kontext und auch der Joint wird gelöst.

##### **Kontextwahl durch Distanzgrenze**

Ähnlich zu der Variante mit der Kollision, wird hier der Kontext bereits entschieden, wenn eine bestimmte Distanz zwischen einem Greifobjekt und einem Aktor unterschritten wird. Beide Objekte werden wieder mit einem Joint verbunden und diese Verbindung und der Kontext bestehen solange, wie der Aktor die Distanzgrenze in seinen Bewegungen nicht überschreitet.

### **Kontextwahl über Distanzgrenze und Startgeste**

Dieses Verfahren ist eine Kombination aus der eben aufgeführten Distanzunterschreitung und einem separaten Ereignis. In diesem Fall handelt es sich um die Startgeste, die die Kontextwahl anstößt. Erfolgt die Startgeste, und befindet sich der Akteur gleichzeitig innerhalb der Grenzwerte zu einem Physikobjekt, wird eine Verbindung (Joint) zwischen Akteur und Physikobjekt erstellt. In diesem Fall wird die Verbindung erst wieder gelöst, sobald die Endgeste erfolgt.

#### **4.1.2.4 Start- und Endgeste**

##### **Startgeste: Greifen**

Der Begriff "greifen" ist mehrdeutig belegt. Da noch häufiger auf die Arbeit Potratz (2011) verwiesen wird, soll der Begriff hier deutlich in Abgrenzung zu dieser Arbeit definiert werden. In der Arbeit Potratz (2011) wird das Greifen als eine Geste mit zwei Händen, die einen Gegenstand greifen, verstanden. Im Gegensatz dazu soll in dieser Arbeit „greifen“ als das Schließen einer Hand verstanden werden. Eine Abbildung dazu ist in Abschnitt 4.1 zu finden. Im weiteren Verlauf steht der Begriff des Greifens oder Greifgeste, sofern explizit nicht anders hervorgehoben, immer für das einhändige Greifen. Die Greifgeste löst bei dem "greifenden Akteur" die Funktion aus, das dieser Akteur sich mit einem Artefakt des Schaltpults verbinden will. Dies geschieht aber nur unter der Bedingung, dass sich ein solches Artefakt innerhalb einer vorher definierten Entfernung befindet. Ansonsten verbindet sich der Akteur nicht. Analog zu dem realen Schließen einer Hand, wenn man beim Schließen der Hand etwas greift, hält man dieses fest, ansonsten ist die Hand geschlossen und man hat eben nichts gegriffen. Es bedarf erst den nun folgenden Loslasse-Gesten, da mit einem erneuten Zugreifen eine Verbindung mit einem Artefakt ermöglicht wird.

##### **Endgeste: Loslassen**

Als Loslassen oder auch Loslasse-Geste wird bezeichnet, wenn eine geschlossene Hand geöffnet wird. Es entspricht der gegenteiligen Handlung des Greifens, und umfasst auch nur eine Hand. Zu sehen in Abbildung 4.1. Wird eine Hand geöffnet, wird ein etwaiger Gegenstand losgelassen. Gesetzt den Fall das vorher geschehene Greifen führte nicht zum Verbinden des Akteurs mit einem Artefakt des Schaltpults, passiert nichts weiter, andernfalls wird ein Gegenstand freigegeben. Beide Hände können diese Gesten unabhängig von einander ausführen.

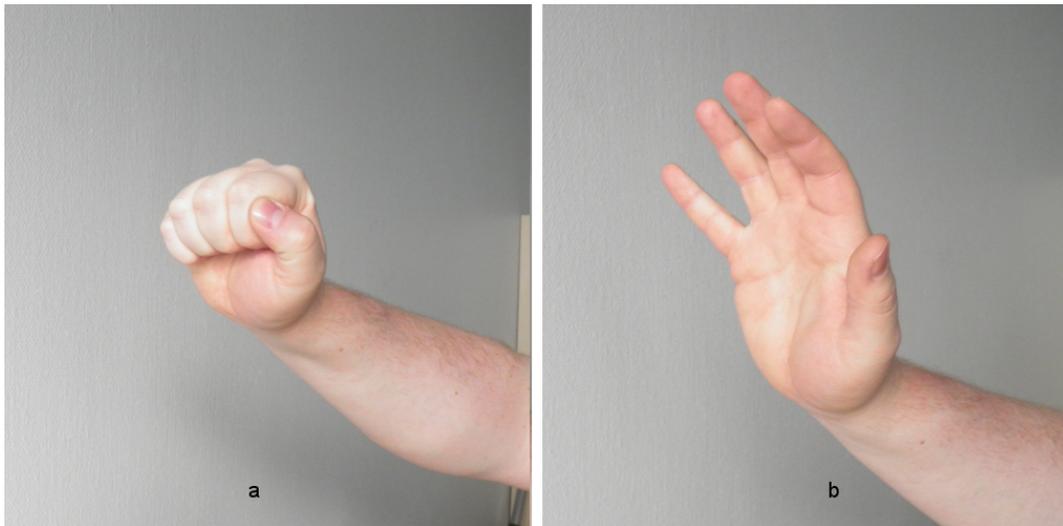


Abbildung 4.1: : Start- und Endgeste

#### 4.1.2.5 Steuergesten

Die eigentliche Gestenerkennung findet hier statt. Es handelt sich hierbei um manipulative und transformative Gesten. Diese Einstufung der Gesten auf Grundlage Handlungsgruppen für Gesten 4.1.2.1 geschieht unter Betrachtung der physikalischen Simulation. Die spätere Darstellung in der 3D-Simulation wird hierbei vernachlässigt, da die dort abgebildeten Objekte zum Teil wesentlich umfangreicher sind.

Grundsätzlich gibt es sechs Freiheitsgrade, längs derer sich ein Objekt bewegen kann. Drei Bewegungsgrade, entlang der Raumachsen und drei Rotationsmöglichkeiten auch jeweils um die drei Achsen im Raum.

Der Benutzer kann nun, je nachdem wie diese Freiheitsgrade für ein Artefakt gewählt und konfiguriert sind, dieses Artefakt manipulieren. Das Ergebnis dieser Manipulation wird vom System ausgewertet. Werden bei diesen Ergebnissen Zustandsänderungen des Artefaktes entdeckt, werden diese weiter gegeben. Als Beispiel dient hier, das Umliegen eines Hebels. Der Hebel wird von unten nach oben bewegt. Dabei spielt es keine Rolle, wie lange es dauert, noch ob mitten in der Bewegung gestoppt wird. Auch ein Loslassen oder ein kurzzeitiger Richtungswechsel haben keinen Einfluss. Erst wenn ein bestimmter Grenzwert überschritten ist, wechselt der Anzeiger innerhalb des Artefaktes seinen Zustand, von beispielsweise „aus“ auf „an“.

Die Anzeiger legen das Schaltverhalten der Artefakte fest, ob diese diskret oder kontinuierlich sind. Außerdem werden die Grenzen, wann geschaltet wird, durch die Anzeiger bestimmt. Ein Artefakt legt mithilfe der Freiheitsgrade allerdings seine maximale Beweglichkeit fest. Sämtliche Schaltlogik der Anzeiger muss sich innerhalb dieser Freiheitsgrade befinden, um wirksam zu werden.

## 4.2 Design

Das oberste Ziel dieses Designs soll eine möglichst lose Kopplung jeglicher Komponenten sein. Zum einen um der stetig wachsenden Zahl an Sensoren eine Möglichkeit zu bieten, ohne Änderungen am Gesamtaufbau eingebunden zu werden. Zum anderen ist es somit möglich verschiedene Kombinationen von Sensoren und Ausgabemedien zu testen. Auch ist es möglich alle Komponenten, auch den erschaffenen Physikraum und die dazu verwendete Physik-Engine, auszutauschen falls der Austausch im Verlauf weiterer Versuche erforderlich werden sollte. Prinzipiell wäre es möglich gewesen alle Komponenten in C# zu erstellen, aber da diese möglichst breite Einbindung von Komponenten auch andere Programmiersprachen beinhalten soll, wurden auch einige Adapter für Sensoren wie auch Ausgabekomponenten in Java implementiert.

### 4.2.1 Sensoren

Die Sensoren selbst, beziehungsweise die Software, die das Einbinden des Sensors in das System ermöglicht muss individuell die spezifischen Daten des Sensors aufarbeiten. Die Ausarbeitung des sogenannten Adapters umfasst sowohl die Abstraktion der Hand auf eine dreidimensionale Koordinate, wie auch die Erkennung der rechten oder linken Hand.

Darüber hinaus muss spätestens aus den Rohdaten beim Adapter eine Gestenerkennung stattfinden. Sowohl die Greif- als auch die Loslassgeste müssen hier gefunden werden. Anschließend sendet der Adapter die aufbereiteten Informationen selbstständig an den Message Broker.

#### ART-Tracker

Der ART-Tracker von der Firma Advanced Realtime Tracking GmbH <sup>3</sup> kann sowohl für das Tracking der Hände verwendet werden, wie auch mithilfe eines ihrer starren Artefakte zur Bestimmung der Position und Lage des Kopfes genutzt werden.

Die für die Aufbereitung und Verarbeitung der Daten der aufgezeichneten Hände benötigten Algorithmen wurden im Kapitel 3 bereits besprochen.

---

<sup>3</sup><http://www.ar-tracking.com/contact/>



Abbildung 4.2: : Selbstgebauter Handschuh mit Markern

Das Foto in Abbildung 4.2 zeigt die später von den Benutzern getragenen Handschuhe mit den Markern an den Fingerspitzen. Der ART-Tracker kann darüber hinaus noch verwendet werden, um die Position und Lage eines Headup-Displays zu berechnen. Diese Aufgabe ist separat von den Aufgaben der Erkennung und Verarbeitung der Handdaten zu betrachten. Dem Handbuch für das ART-Tracking Systems ARTtracker-Anleitung (2007) ist zu entnehmen, dass die Ermittlung der Position so wie die Rotation von starren bereits eingemessenen ART-Trackerartefakte innerhalb des Messbereichs möglich ist.

### **XBox 360 Kinect**

Die Kinect von der xBox 360, die noch im ersten Versuch 2.7 zum Einsatz kam, spielt im weiteren Verlauf dieser Arbeit keine Rolle mehr, da sie nicht über die benötigten Fähigkeiten zum Erkennen des Greifens mit den Händen verfügt.

**Kinect for Windows** Die Kinect for Windows bringt einige Neuerungen mit sich. Die wichtigsten sind wohl das Erkennen unter anderem vom Greifen und Öffnen der Hände. Die Kinect for Windows wird für die Umsetzung später eine tragende Rolle spielen. Sofern in der Arbeit nicht anders beschrieben, wird mit dem Begriff Kinect, immer die Kinect for Windows gemeint.

Die Erkennung der Greif- und Loslassgeste erfolgt bereits durch die Kinect und ihre Treibersoftware selbst. Unter anderem die Arbeit von [Brüggemann (2013)] erläutert den Aufbau und die Funktion der Kinect. Die Übermittlung einer gefundenen Geste erfolgt eventbasiert. Dies bedeutet in diesem Fall, dass nur in einem Datenpaket das von der Kinect gesandt wird, steht, dass eine Greifgeste gefunden wurde. Die nachfolgenden Pakete enthalten keine Informationen über das Greifen. Sobald die Kinect feststellt, dass die Hand sich wieder geöffnet hat, wird innerhalb eines Datenpakets mitgeteilt, dass die Hand sich wieder geöffnet hat.

Nun stehen beide Varianten zur Verfügung. Der ART-Tracker, der grundsätzlich in jedem Datenpaket mitteilt, ob die Hand geschlossen oder geöffnet ist, und wiederum die Kinect die dieses nur jeweils einmal pro Änderung mitteilt. Es wird sich für eine spätere Umsetzung entschieden, die Variante des ART-Trackers als Standard zu wählen ist. Somit muss die Realisierung der Anbindung der Kinect dafür Sorge tragen, dass etwaige Zustandsänderungen bemerkt und kontinuierlich bis zu einer Änderung der Handgeste ausgegeben werden.

## 4.2.2 Physikalische Ausprägung der Artefakte

### Ausmodellierte Artefakte

Abbildung 4.3 zeigt den für die Physik-Engine benötigten Aufbau eines Joysticks. Einzelne Gesten, wie nach vorne, hinten, links oder rechts Drücken werden mittels Kollisionserkennung der entsprechenden farblichen Flächen mit den eingefärbten Kugeln ermittelt. Dieses Verfahren wurde in dem ersten Test 2.7 benutzt. Dieses Verfahren benötigt eine kontinuierliche Kollisionserkennung durch zugrunde liegende Simulation. Dieses Verfahren ist bildlich sehr gut darstellbar, erlaubt aber keine Möglichkeit der Konfiguration ohne jeweiligen Umbau des Physikmodells.

### Abstrakte Artefakte

Eine weitere Möglichkeit die Artefakte physikalisch auszumodellieren ist, sich auf das Nötigste zu beschränken. Abbildung 4.3 ist so ein abstraktes Artefakt, welches einen Joystick darstellt. Es umfasst ein Greifobjekt (gelb), einen Ankerpunkt (lila) und den beide verbindenden Joint (hellblau), zwischen diesen beiden Teilobjekten. Es wird nun das Schaltverhalten des Joysticks über die Rotation des Greifobjekts um den Ankerpunkt, beziehungsweise der Abstand zwischen Greifobjekt und Ankerpunkt bestimmt.

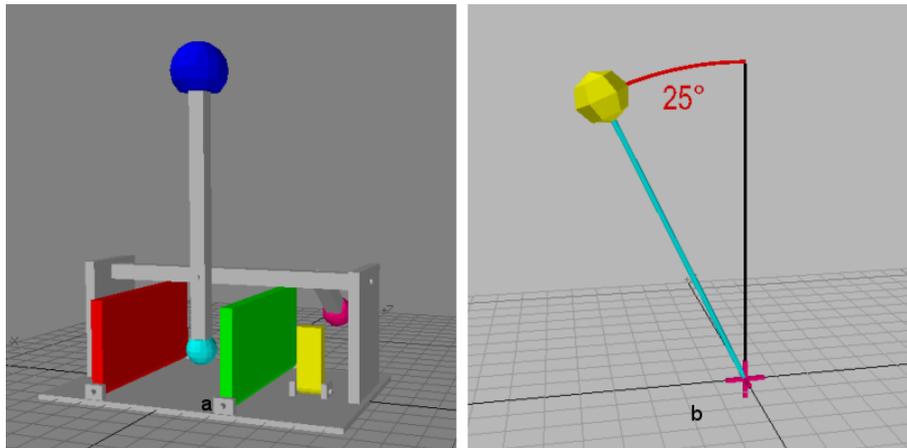


Abbildung 4.3: : Physik-Joysticks aus Potratz (2012a)(a); Abstrakter Joystick (b)

### 4.2.3 Visualisierung

Die Visualisierung dient als Feedback für den Benutzer. Alle Artefakte, und darüber hinaus, auch die Aktoren des Benutzers, werden durch Verwendung einer 3D-Engine dargestellt. Die Physikobjekte übertragen ihre Positionen an ihr entsprechendes Gegenstück in der 3D-Simulation. Die 3D-Simulation ist somit ein Abbild des Zustands der Physiksimulation, allerdings mit einer anderen Darstellungsform. Diese Entitäten für die grafische Abbildung sind meist detailreicher und sehen ihren realen Vorbildern aus Kapitel 2 sehr ähnlich.

Das somit mögliche Feedback an den Benutzer erfüllt mehrere Aufgaben. Zum einen sieht der Benutzer eine Abbildung seiner Hände und kann somit feststellen, wo er sich gerade mit seinen Händen innerhalb des Simulationsraums befindet.

Zum Zweiten bekommt er kontinuierlich eine Rückmeldung, welches und wie er ein Artefakt manipuliert. Dies ermöglicht ein Abbrechen einer Bewegung, bevor sie als Geste erkannt wird.

Außerdem ergibt die Visualisierung noch die Möglichkeit eine erkannte Geste, oder aber den Zustand einer Artefaktes, separat anzuzeigen. Dies geschieht über die verschiedenen Varianten von Anzeigern.

Die Darstellung der Entitäten durch die 3D-Engine soll grafisch einfach gehalten werden. Es werden keine Texturen verwendet. Auf optische Details wie Glanz, Reflexion oder Transparenz wird größtenteils verzichtet.

#### Großes Display

Ein großes Display kann als Darstellung der Schalttafel dienen. Dabei ist entscheidend, dass das Display selbst und die Simulation der Schalttafel zumindest was die räumliche Ausrichtung betrifft, identisch sind. Der Benutzer steht mit etwas Abstand vor diesem großen Display

und manipuliert mit Gesten die Gegenstände, die er auf dem Display sieht. Das errechnete Bild, welches von der simulierten Schalttafel erzeugt wird, sollte eine Frontalsicht auf die Schalttafel bieten und aus der gleichen Blickrichtung erfolgen, wie der Benutzer auf das Display schaut.

### **Headup-Display**

Die Benutzung eines Headup-Displays ermöglicht es, die Artefakte nicht nur an einem Ort in der Simulation aufzubauen und diese wie im Fall der Schalttafel dann über ein fest positioniertes Display auszugeben. Es wird möglich, die Artefakte im Raum zu verteilen. Der Benutzer hat nun die Möglichkeit, die Artefakte immer von der jeweiligen Position aus zu betrachten, in der er sich befindet. Dies wird als Ich-Perspektive bezeichnet.

Der vorher feste Blickwinkel auf die Simulation wird durch die Position und Lage des Headup-Displays ersetzt und das so erzeugte Bild in das Headup-Display gespielt. Somit wird das Headup-Display ein passiver Teil dieser Simulation, in dem es die Kamera innerhalb der Simulation steuert. Um dies zu ermöglichen, müssen nun nicht nur die Hände, sondern auch das getragene Headup-Display, oder der Kopf des Benutzers von Sensoren erfasst werden, und diese Daten zur Verfügung gestellt werden.

Headup-Displays können sowohl für ein oder beide Augen ausgelegt sein. Grundsätzlich lassen sich Headup-Displays in zwei Kategorien einteilen. Die die entweder durchsichtig sind und das Bild, das der Benutzer wahrnimmt durch Informationen erweitern. Oder aber die Headup-Displays, die nicht durchsichtig sind und das reale Bild völlig durch eine Simulation ersetzen. Diese Variante, auf beide Augen ausgelegt, eignet sich nicht für einen Versuchsaufbau, in dem sich der Benutzer frei bewegen kann. Allerdings wäre eine Simulation im Sitzen durchaus denkbar.

#### **4.2.4 Steuerung für externes Programm**

Das eigentliche Ziel des Frameworks und aller seine Komponenten ist die Steuerung einer anderen Software, um dort die Eingabemedien, wie Tastatur, Maus, Joystick oder ähnliches abzulösen. Diese Programmsteuerung muss somit in zwei Teile aufgeteilt werden. Der erste Teil umfasst die Senderkomponente, von der aus die Kommandos an die Software verschickt werden. Diese Komponente befindet sich innerhalb des Frameworks und muss, entsprechend der zu bedienenden Software, passende Kommandos versenden. Jedes Artefakt bekommt per Dependency Injection eine individuell angepasste Teilkomponente des Senders. Diese Teilkomponente ist entsprechend der Art des Artefakts angepasst und verschickt selbstständig entsprechende Kommandos an den Message Broker.

Der zweite Teil der Komponenten ist in der zu steuernden Software integriert. Es handelt sich hierbei um eine Komponente, die die Nachrichten vom Message Broker abrufen und verarbeitet. Im besten Fall hat man bereits eine Software die über solch eine Umsetzung ver-

fügt, oder aber man muss diese dann für die Software umsetzen. Es empfiehlt sich diese Komponente zu erst zu entwickeln, da man im Regelfall bereits über eine herkömmliche Programmsteuerung verfügt und somit der Umfang und die erwartete Funktionalität gegeben ist. Erst im zweiten Schritt, sind dann die passenden Artefakte und Senderkomponenten zu verwirklichen.

### 4.3 Software-Architektur

Als zugrunde liegende Architektur wird ein verteiltes System gewählt, wobei jedes Modul, wie in Abbildung 4.4 zu sehen ist, nur über den Message Broker kommuniziert.

Geht man einen Schritt zurück und betrachtet das System als Ganzes, kann man erkennen, dass sich das System für eine Umsetzung, in Anlehnung an das MVC-Entwurfsmuster mit aktivem Model, eignet. Es wird also noch eine Observer-Komponente hinzugefügt. Eine genaue Erläuterung zum MVC-Entwurfsmuster kann der Arbeit Potratz (2011) entnommen werden. Im Unterschied zum klassischen MVC-Modell werden die Daten für den Controller unabhängig von der View-Komponente ermittelt. Die Daten stammen von den Sensoren. Die View zeigt diese Daten lediglich an, manipuliert diese aber nicht.

#### 4.3.1 Kommunikation Message Broker

Wird ein Message Broker als Kommunikationsmittel zwischen den Modulen benutzt, ergibt sich daraus der Vorteil der programmiersprachlichen Unabhängigkeit. Vorausgesetzt in der jeweiligen Programmiersprache existieren Erweiterungen, um mit diesem Message Broker zu kommunizieren. Die ausgetauschten Nachrichten sind standardisiert. Im Anschluss sollen die beiden verwendeten Nachrichtenformate für den Transport der Handdaten und der Daten über Position und Lage des Kopfes aufgeführt werden. Sobald ein Sensor, oder sein entsprechender Adapter eines oder beide dieser Nachrichtenformate beherrscht, kann er prinzipiell im Framework als Sensor benutzt werden.

**Model** Die im Model vorgehaltenen Daten umfassen die realen Positionen der Hände und ob diese offen oder geschlossen sind. Darüber hinaus werden hier noch die Daten über die Position und Lage des Kopfes vorgehalten, sofern sie zur Darstellung in der View benötigt werden. Die Informationen der Hände werden in Aktoren innerhalb der Physik-Engine übertragen. Die Physik-Engine enthält zusätzlich noch die Daten für alle weiteren in der Simulation vorhandenen Artefakte. Diese Daten umfassen sowohl Position wie auch die internen Zustände. Die Physik-Engine manipuliert ihre Daten selbstständig, lediglich die Position und Zustände der Aktoren werden von außen beigesteuert.

**View** Die View-Komponente entnimmt die Daten zur Anzeige dem Modell und stellt diese dar. Die View-Komponente ist abweichend zum MVC-Muster vollkommen passiv. Darüber hinaus kann man die Steuerung der externen Programme noch der View zuschreiben, da dieser Teil auch bestimmte Zustände innerhalb des Modells widerspiegelt.

**Controller** Die Adapter der jeweiligen Sensoren bilden den Kern des Controllers. Sie liefern die Daten der Hände und des Kopfes an das Modell.

**Observer** Der Observer prüft das Modell als Ganzes auf Veränderung. Bei Veränderung werden alle relevanten Daten, die für die grafische Ausgabe und die zum Steuern der externen Programme benötigten Befehle, sofern vorhanden, dem Modell entnommen und passend an die View weitergegeben.

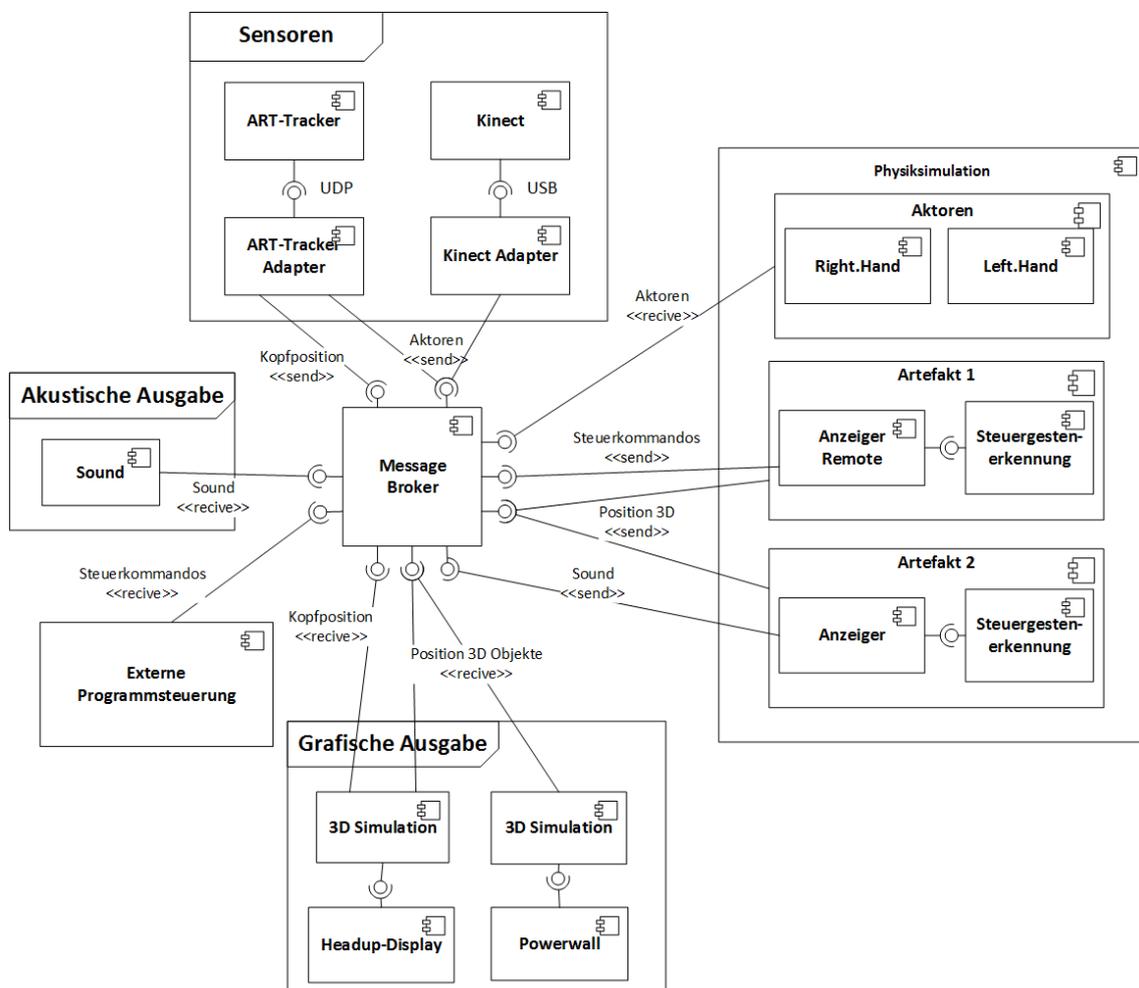


Abbildung 4.4: : Komponentendiagramm

## Nachrichtenformate

Zu ersetzende Symbole	Beschreibung
<float>	Es gilt diesen Platzhalter durch einen Float-Wert zu ersetzen. Es ist der komplette Wertebereich eines Float hier zu erwarten.
<boolean>	Dieser Platzhalter wird durch ein Boolean-Wert ersetzt. Es gibt „true“ und „false“.
+	Das Plus selbst gehört nicht zum Nachrichtenformat, es soll nur darauf hinweisen, dass die durch die Abbildung auftretenden Umbrüche zu entfernen sind, dass das eigentliche Format ohne Umbrüche auskommt.

Tabelle 4.1: Legende der Nachrichtenformate

**Nachrichtenformat Hand** Jeder Aktor bekommt seine eigene Nachrichten-Queue und wird separat versandt. Aus den Nachrichtenformaten selbst ist nicht ersichtlich um welchen Aktor es sich handelt. Hand[x=<float>;y=<float>;z=<float>;grab=<boolean>]

Hand[x=<float >;y=<float >;z=<float >;grab=<boolean >]

**Nachrichtenformat Kopf** Auch die Daten für den Kopf werden in einer separaten Queue abgelegt.

Kopf[x=<float >;y=<float >;z=<float >;eta=<float >;theta=<float >;+  
phi=<float >;rot=[<float >, <float >, <float >, <float >, <float >,+  
<float >, <float >, <float >, <float >]]

# Kapitel 5

## Realisierung

In diesem Kapitel soll die Realisierung gezeigt werden und welche Software hierbei zum Einsatz kam. Die Umsetzung der Softwarekomponenten erfolgte, bis auf die Ausnahme des Kinect Adapters, in Java 7. Alle verwendeten Frameworks oder andere fremde Software wird in den Unterkapiteln aufgeführt.

### 5.1 Message Broker

Für die Realisierung des Message Brokers wird auf eine Apache ActiveMQ <sup>4</sup> in der Version 5.9 zurück gegriffen. Wie im Design in Abschnitt 4.3.1 beschrieben erhält jeder Akteur eine eigene Queue. Die folgende Tabelle beschreibt die Namen der einzelnen Queues. Hält man sich an die in 4.3.1 und 4.3.1 spezifizierten Nachrichtenformate, kann über diese Queues die Kommunikation zwischen den Sensoradaptern und der Simulation, wie auch der Simulation mit einem externen Programm stattfinden. Die empfangen Nachrichten verfallen nicht. Es werden also, sobald ein Konsument vorhanden ist, der Reihe nach, alle Nachrichten ausgeliefert.

### 5.2 Sensorenadapter

Im folgenden Abschnitt wird kurz auf die Umsetzung der bereiten zur Verfügung stehenden Sensoren und ihre Anbindung an das Gesamtsystem im einzelnen eingegangen.

---

<sup>4</sup><http://activemq.apache.org/>

Name der Queue	Beschreibung
Hand.Links	Hält die Daten für die linke Hand vor.
Hand.Rechts	Hält die Daten der rechten Hand vor.
Kopf	Empfängt die Position und aus Ausrichtung des Headup-Displays.
SpaceSim.Steuerung	Spezielle Queue für die Kommunikation mit dem externen Programm für den Versuchsaufbau des virtuellen Cockpits [link]

Tabelle 5.1: Namen der Queues auf der ActiveMQ

## Kinect for Windows

Es werden die in Unterkapitel 4.2.1 aufgeführten Anforderungen umgesetzt. Die Umsetzung erfolgt mit Hilfe des Kinect SDK Version 1.7<sup>5</sup>, die Programmiersprache ist C# 4.0<sup>6</sup>. Die Basis für die Umsetzung bildet das Tutorial von Christian Kokott [Kokott (2013)] und das Tutorial [Duncan (2013)] und der dort enthaltene Sourcecode von Greg Duncan.

Die Anbindung an die ActiveMQ erfolgt die Erweiterung Apache.NMS.ActiveMQ Version 1.6.0<sup>7</sup>. Der Start des Adapters aktiviert die Kinect, sobald die Kinect aktive ist und eine Person erkennt, wird die Übertragung an die ActiveMQ gestartet.

### 5.2.1 ART-Tracker

Die Ursprünge des Sensoradapters gehen auf die Arbeit [Potratz (2011)] zurück und es kommen die gleichen Handschuhe (siehe 4.2) für den Anwender zum Einsatz. Es werden für die Verarbeitung der Daten von den Händen die in Kapitel 3 vorgestellten Algorithmen implementiert. Zusätzlich wird eine grafische Oberfläche entwickelt, um die Algorithmen zu überprüfen und besser einzustellen. Eine Abbildung der Ausgabe dieser grafischen Oberfläche kann unter 3.3 betrachtet werden.

Die Anbindung an die ActiveMQ erfolgt durch die in der ActiveMQ 5.9 mitgelieferten jars. Darüber hinaus werden alle aufgezeichneten Rohdaten durch den Adapter gespeichert. Es ist möglich, diese wieder einzuspielen. Dies ermöglicht eine genauere Feineinstellung der Filter und Greifgestenerkennung.

<sup>5</sup><http://www.microsoft.com/en-us/download/details.aspx?id=36996>

<sup>6</sup><http://www.microsoft.com/en-us/download/details.aspx?id=17718>

<sup>7</sup><http://activemq.apache.org/nms/>

## 5.3 Physik-Engine

Um eine Interaktion zwischen dem Aktor, also dem Repräsentanten der Hand des Benutzers, und den zu manipulierenden Physikobjekten zu ermöglichen, werden im Folgenden drei Verfahren vorgestellt, wobei zum jetzigen Zeitpunkt nur die Varianten a und b erfolgreich umgesetzt werden konnten.

### 5.3.1 Wahl der Physik-Engine

Im Projekt 1 [Potratz (2012a)] wurde auf eine Umsetzung mit XNA 4.0<sup>8</sup> und der Physik-Engine Nvidia PhysiX<sup>9</sup> gesetzt. Aufgrund der Entscheidung, die Kontextwahl mit über die Distanz und Startgeste zu lösen, konnten auf Eigenschaften der Physik-Engine verzichtet werden. Es wurde keine Reibung mehr benötigt. Somit konnten auch andere Physik-Engines und ihre Einbindungen in 3D-Grafik-Engines wieder in Betracht gezogen werden. Hier fiel vor allem die JmonkeyEngine 3 wieder in den Fokus, mit der Verwendung von der Bullet Physik-Engine<sup>10</sup> in C++. Die Verbindung zwischen JMonkey, welche in Java geschrieben ist, und der Bullet-Engine wird durch JBullet<sup>11</sup> sichergestellt. Mit dieser Engine wurden bereits ausgiebig Erfahrungen in der Umsetzung der Arbeit [Potratz (2011)] gesammelt.

Da JBullet in den ersten Versuchen passende Eigenschaften zeigte, die grundsätzlich eine Umsetzung ermöglichen und nicht zuletzt auch die Möglichkeit einfacher selbst erstellte 3D-Objekte in JMonkey einzuladen ließen die Wahl schlussendlich auf die JMonkey-Engine und JBullet fallen.

### 5.3.2 Entwicklung eigener Joints

Die über die JMonkey eingebundene JBullet Physik-Engine bieten von sich aus Joints an. Darunter fielen auch Joints, die zum Verbinden des Aktors mit den Artefakten dienen könnten. Bestimmte Varianten der Joints erfüllten prinzipiell die Anforderungen aus 4.1.2.2. Allerdings zeigte sich nach dem Fortschreiten der Umsetzung, dass diese Joints nicht immer zuverlässig waren, beziehungsweise ab und an ein inakzeptables Fehlverhalten an den Tag legten.

Da eine Fehlersuche im Rahmen des eigenen Programmcodes erfolglos blieb, musste ein eigener Joint implementiert werden. Obwohl für die Physik-Engine ODE geschrieben, bot die Arbeit von Russel Smith [Smith (2002)] einen guten Einstieg für die Entwicklung eines eigenen Joints. Die Implementierung dieses Joints ist ausschließlich in Java umgesetzt, er wird nicht, wie die vorgefertigten Joints, in die nativen implementierten Joints von Bullet übertragen. Der

<sup>8</sup><http://msdn.microsoft.com/de-de/library/bb200104%28v=xnagamestudio.40%29.aspx>

<sup>9</sup><http://www.nvidia.de/object/physx-9.12.1031-driver-de.html>

<sup>10</sup><http://bulletphysics.org/wordpress/>

<sup>11</sup><http://jbullet.advel.cz/>

selbst entwickelte Joint erfüllt die an ihn gestellten Anforderungen aus 4.1.2.2 und kommt jeweils zur Verbindung zwischen Aktor und Artefakt zum Einsatz. Andere Stellen innerhalb der Physik-Simulation sind aber weiterhin mit den originalen Joints aus JBullet versehen.

## 5.4 Umsetzung der Aktoren & Kontextbestimmung

Die beiden Aktoren sind die Abbildungen der Hände des Benutzers. Innerhalb der Simulation gibt es zwei Erscheinungsformen. Beide Formen werden in den folgenden Abschnitten beschrieben. Darüber hinaus fällt die Implementierung der Wahl des Kontextes auch in diesen Bereich. Sie wird somit auch hier beschrieben.

### 5.4.1 Aktoren in der Physiksimulation

Die Aktoren sind innerhalb der Physiksimulation als Kugel abgebildet. Diese Repräsentation bleibt dem Benutzer im Regelfall verborgen, da dieser ausschließlich die 3D-Simulation zu Gesicht bekommt. Die Varianten für die Bestimmung des Kontextes sind theoretisch im Abschnitt 4.1.2.3 aufgeführt. In diesem Fall findet eine Umsetzung der Variante 4.1.2.3 mittels Distanzbestimmung zu den möglichen Kontexten statt, ausgelöst durch die Greifgeste der entsprechenden Hand. In diesem Fall wird der Aktor unter Verwendung des in Abschnitt 5.3.2 entwickelten Joint mit der innerhalb des Suchradius am nächsten liegenden und freien Greiffläche verbunden und befindet sich nun in dem Kontext des dazu gehörigen Artefaktes. Die Wahl hat Einfluss auf die Ausgestaltung der physikalischen Kugel, die als Repräsentant des Aktors dient. Es kann hier auf eine Kollisionserkennung verzichtet werden, wie sie für eine der anderen möglichen Kontextwahlen benötigt worden wäre. Des Weiteren können hier Implementierungen zu Oberflächeneigenschaften, die eine Reibungsberechnung ermöglichen, weggelassen werden, wie sie eine mögliche Umsetzung nach 4.1.2.3 verlangt hätten.

### 5.4.2 Aktoren in der 3D-Simulation

Die Aktoren werden in der 3D-Simulation als Hände abgebildet. Für die drei Zustände eines Aktors gibt es, wie in Abbildung 5.1 zu sehen, ein dreidimensionales Gegenstück. In der Darstellung wird unterschieden welche Hand der Aktor repräsentiert und entsprechend eine linke oder rechte Hand in die 3D-Simulation geladen. Die Lage der Hand wird so gewählt, wie der Benutzer voraussichtlich seine Hand auch real hält. Beiden verwendeten Sensoren 4.2.1 und 4.2.1 ist es nicht möglich die Lage im Raum einer realen Hand zu bestimmen, somit ist eine dynamische Rotation der Handabbildungen innerhalb der 3D-Simulation nicht möglich.

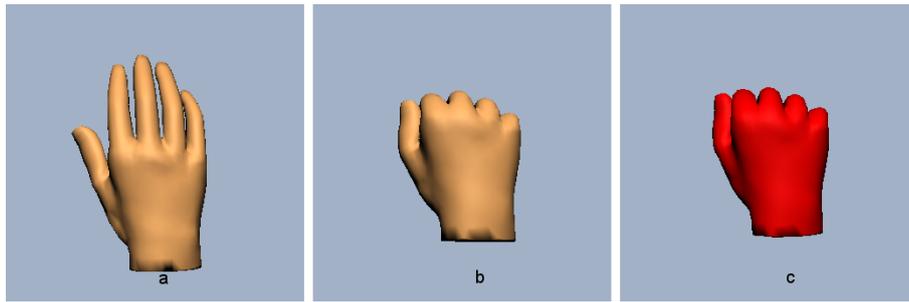


Abbildung 5.1: Hand offen (a), Hand gegriffen (b), Hand nicht gegriffen (c)

## 5.5 Umsetzung des Schaltpults

Die Umsetzung des Schaltpultes umfasst mehrere Bereiche. Zum einen geht es um die physikalische und grafische Modellierung der Artefakte, zum zweiten wird innerhalb der Artefakte die angestrebte Gestensteuerung umgesetzt. Die Artefakte selbst ändern ihren Zustand, wenn bestimmte Grenzwerte überschritten werden. Als Beispiel sei hier der Knopf genannt, der, wenn er weit genug rein gedrückt wird, von „nicht gedrückt“ auf den Zustand „gedrückt“ wechselt. Über die Schalttafel werden dann alle Zustände der verwendeten Artefakte gesammelt oder direkt an den Message Broker übermittelt. Darüber hinaus ist es möglich das Schaltpult anzusprechen, um die Artefakte in bestimmte Ausgangszustände zu versetzen. Im weiteren werden nun die Umsetzungen der einzelnen Artefakte beschrieben und Bezug darauf genommen, wie die einzelnen Gesten erkannt werden können.

### 5.5.1 Umsetzung der Artefakte

Bei der Umsetzung der Artefakte gilt es zu beachten, dass ihre maximale Auslenkung selbstbestimmt ist. Das Schaltverhalten der Artefakte hingehen, ob diskret oder kontinuierlich, hängt hingegen von dem beim Start eingesetzten Anzeiger ab. Die Artefakte werden mit ihrem für gewöhnlich eingesetzten Anzeiger beschrieben um ihr offensichtliches Verhalten einfacher zu beschreiben.

#### Hebel

Die Verhaltensweise des Einhand- und Zweihandhebels sind in ihrer Umsetzung nahezu identisch. Allein die Tatsache, dass beim Hebel für zwei Hände beide Griffe, in den Abbildungen 5.2 (a) und (c) grün dargestellt, gegriffen werden müssen, damit eine Manipulation erfolgen kann, ist unterschiedlich. Sobald eine Grifffläche von einem Aktor gegriffen wird, verfärbt sich diese Grifffläche rot und kann von keinem weiteren Aktor gegriffen werden. Die Hebel verfügen in dieser Umsetzung über einen Freiheitsgrad. Dieser Freiheitsgrad bezieht

sich auf die Rotation um die x-Achse. Die maximale Auslenkung um die x-Achse beträgt Plus-Minus  $80^\circ$  um die Ruheposition, die orthogonal zur Schalttafel ist.

Für das Umschalten von einem Zustand müssen jeweils 90% des gesamten Wertes des Freiheitsgrads überwunden werden. Ist der Schalter aus, ist er nach unten geklappt. Von diesem Status „aus“, muss er also einen Wert von  $+64^\circ$  überschreiten, damit der Status auf „an“ wechselt. Soll der Hebel wieder auf „aus“ wechseln, muss der Hebel einen Winkel von  $-64^\circ$  unterschreiten. Die Abbildungen 5.2 (b) und (d) zeigen die physikalische Repräsentation der Hebel, mit Umrissen der grafischen Darstellung. Das pinkfarbene Kreuz ist der Mittelpunkt eines Artefaktes, um den eine Rotation stattfindet. Darüber hinaus ist das gelbe Griffobjekt zu sehen, welches jeweils mit einem Actor mittels Joint eine Verbindung zu einem Actor aufbaut, sofern ein Greifen der Hand erkannt wird, die durch diesen Actor repräsentiert wird.

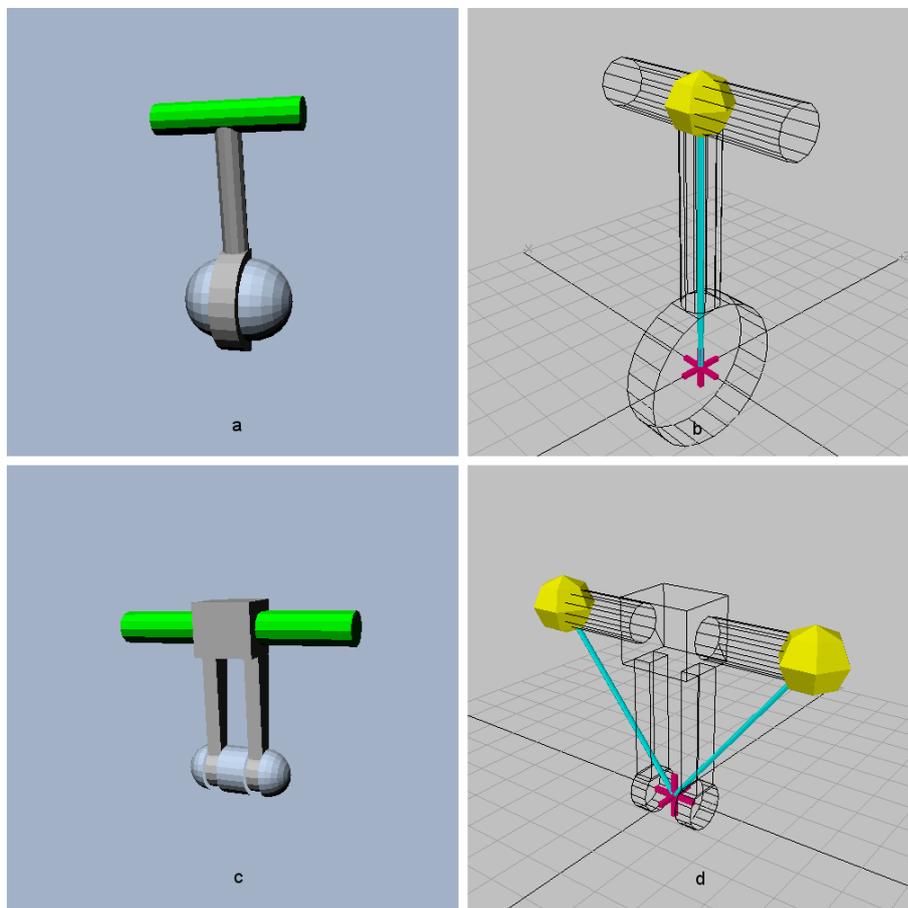


Abbildung 5.2: Einhand-Hebel : 3D-Ansicht (a), Physik-Ansicht (b)  
Zweihand-Hebel : 3D-Ansicht (c), Physik-Ansicht (d)

## Rad

Die Umsetzung des Rads umfasst ebenfalls eine Variante für eine Hand (Abbildung 5.3 (a)) und eine Variante für zwei Hände (Abbildung 5.3 (c)). Bis auf die unterschiedliche Anzahl der benötigten Hände zur Benutzung der Artefakte, und die damit verbundene unterschiedliche Verwirklichung der physikalischen Eigenschaften und der grafischen Darstellung, sind beide Artefakte gleich im Verhalten.

Beide Räder verfügen über einen identischen Freiheitsgrad für die Rotation um die z-Achse. Der Rotationsumfang beträgt  $1080^\circ$  bzw.  $6 \cdot \pi$ . Das Rad kann in diesem Fall also drei Mal um die eigene Achse gedreht werden. Die Zustandsänderung ist kontinuierlich von 0-100. Bei 0 ist das Rad vollständig rechtsherum gedreht. Es gilt also als „geschlossen“. Es kann nun links herum „auf“ gedreht werden.

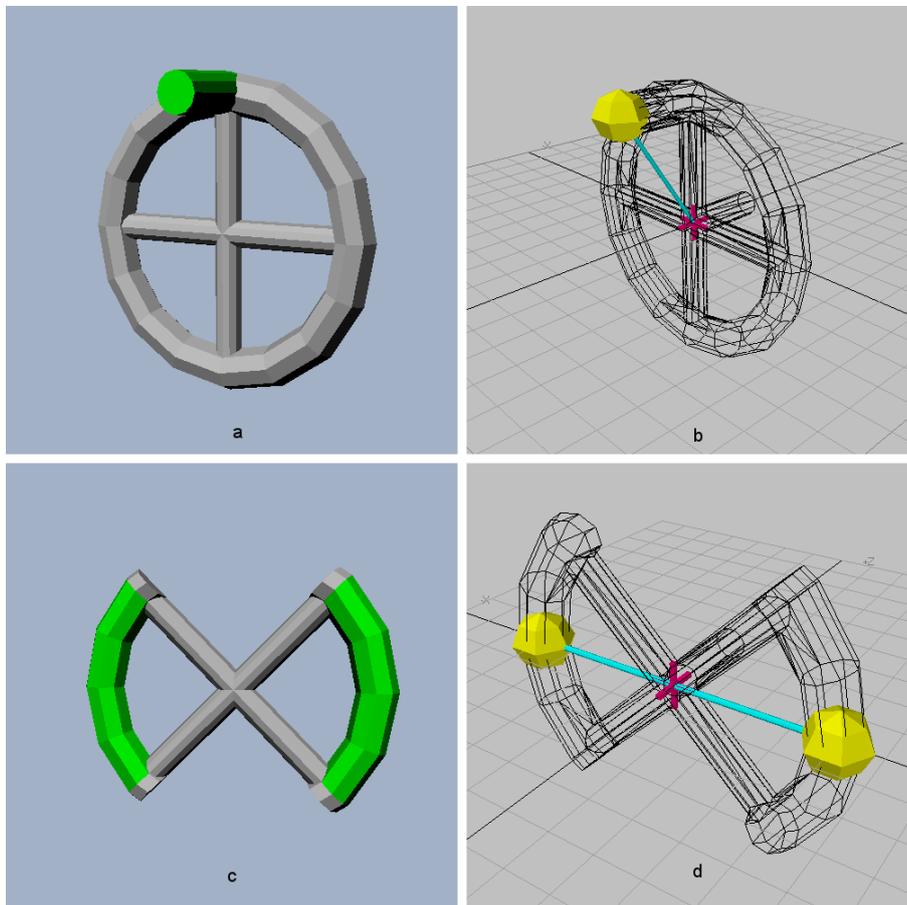


Abbildung 5.3: : Einhand-Rad : 3D-Ansicht (a), Physik-Ansicht (b)  
Zweihand-Rad : 3D-Ansicht (c), Physik-Ansicht (d)

Nach beispielsweise einer Rotation von drei Pi ist ein Wert von 50 erreicht. Es kann jederzeit wieder rechts herum gedreht werden, um den Vorgang rückgängig zu machen. Dreht man nach der ersten Rotation von drei Pi weiter nach links um zusätzliche drei Pi ist der Wert 100 erreicht und das Rad blockiert. Eine weitere Rotation nach links ist nun nicht mehr möglich.

## Schieberegler

Für die Realisierung des Schiebereglers wird nur ein gelbes Griffobjekt verwendet, da hier ausschließlich die Verwendung mit einer Hand vorgesehen ist. Es gibt zwei Ausprägungen des Schiebereglers. Eine Variante, wie sie in Abbildung 5.4 zu sehen ist, kann längs der y-Achse bewegt werden. Die andere Variante ist um 90° nach rechts gekippt und kann so entlang der x-Achse bewegt werden. Die maximale Beweglichkeit beträgt 3.0 Maßeinheiten. Der Zustand des Artefakts selbst ist im Regelfall kontinuierlich.

Der Mittelpunkt des Artefaktes wird wie gehabt durch das pinkfarbene Kreuz wie in Abbildung 5.4(b) zu erkennen ist, dargestellt.

Kontinuierliche Zustände werden von 0 bis 100 Prozent angegeben. Die Entfernung längs der zu manipulierenden Achse ist dafür entscheidend. Bei einem Abstand von 0 zum Mittelpunkt, entspricht dies einem Zustand von 0%. Die Entfernung von 3 entspricht hier einem Zustand von 100%.

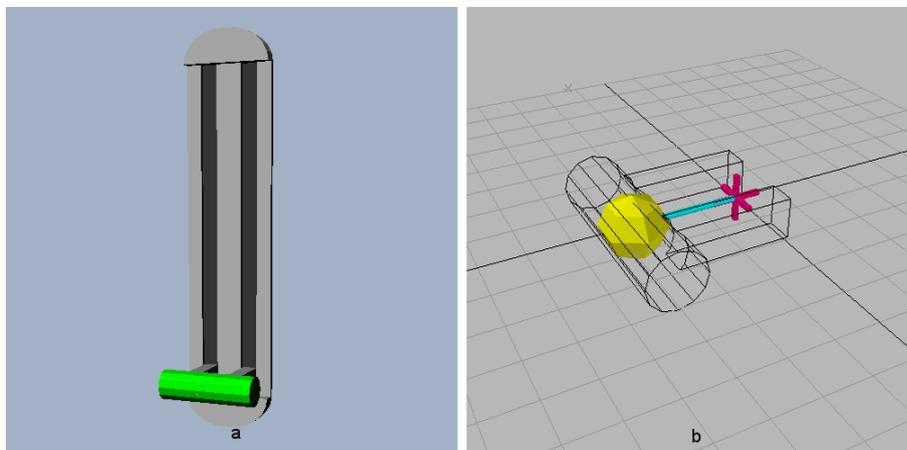


Abbildung 5.4: : Schieberegler: 3D-Ansicht (a), Physik-Ansicht (b)

## Schaltknopf

Das Schaltknopf-Artefakt besitzt ein Griffobjekt. Es kann längs der z-Achse bewegt werden. Die Zustände des Schaltknopfs sind diskret, er besitzt zwei Zustände. Hierbei ist zu beachten, dass der Schaltknopf beim Start auf „aus“ gestellt ist und sein Abstand zum Mittelpunkt

die maximale Ausdehnung beträgt.

Wird der Knopf gegriffen und dann „rein gedrückt“, ändert sich sein Zustand auf „an“ wenn der Abstand 20% der maximalen Ausdehnung unterschreitet. Zusätzlich löst der Schaltknopf die Verbindung zum Aktor in diesem Moment. Der Aktor schaltet auf seinen Zustand „nicht gegriffen“ und der Schaltknopf springt wieder auf seine maximale Ausdehnung zurück. Von hier kann der Schaltknopf erneut gegriffen werden und dann durch erneutes Drücken abermals seinen Zustand von „an“ wieder auf „aus“ ändern. Die Ausdehnung beträgt 0.8 Maßeinheiten, und sorgt dafür, dass der gedrückte Knopf nicht vollständig in der Schalttafel verschwindet, und immer ein Stück zu sehen ist.

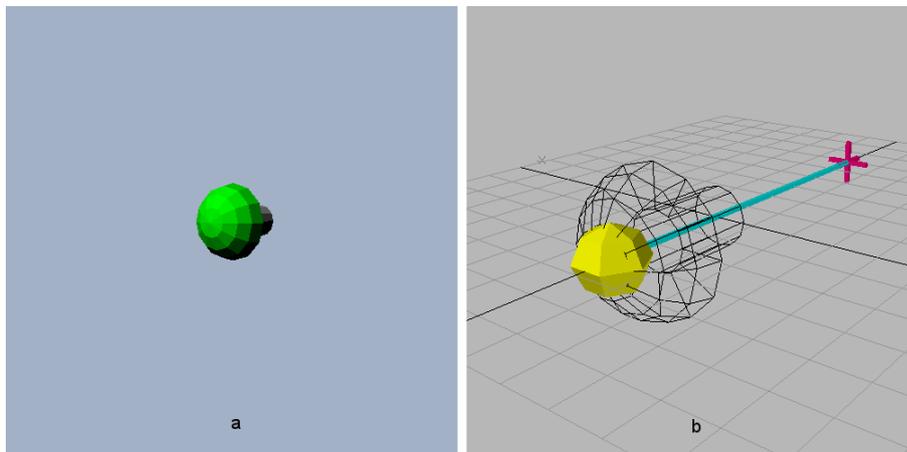


Abbildung 5.5: : Einhand-Hebel : 3D-Ansicht (a), Physik-Ansicht (b)

## Kippschalter

Der Kippschalter verfügt über einen Rotationsfreiheitsgrad um die y-Achse. Die Ruheposition des Artefakts liegt bei der Initialisierung genau zwischen beiden Zuständen. Die maximale Rotation in eine Richtung beträgt  $20^\circ$ . Zu Beginn wird der Status des Kippschalters auf „an“ oder „aus“ gestellt und so eine der beiden maximalen Auslenkungen gewählt.

Der Kippschalter verfügt über eine Besonderheit, er besitzt grundsätzlich zwei Griffe, wovon abwechselnd immer nur einer aktiv ist und gegriffen werden kann. Wie in Abbildung 5.6(a) zu sehen ist, ist vom Benutzer aus betrachtet, der linke Griff aktiv und grün gefärbt. Der aktive Griff verhält sich grundsätzlich wie jeder andere Griff auch und kann von einem Aktor gegriffen werden. Der Unterschied kommt dann zum Tragen wenn der Kippschalter gedrückt wird. Ändern sich der Status des Artefakts von „an“ nach „aus“ oder umgekehrt, löst sich die Verbindung von Griff und Aktor automatisch, vergleichbar mit dem Verhalten des Schaltknopfs 5.5.1. Der Aktor wechselt dann in den Zustand „nicht gegriffen“. Darüber hinaus wird der Griff deaktiviert und der andere Griff des Kippschalters wird aktiviert und kann nun gegriffen

werden. Als Benutzer muss man nun die Hand vorher öffnen und erneut den Kippschalter greifen, wenn man den Zustand des Artefaktes wieder ändern möchte.

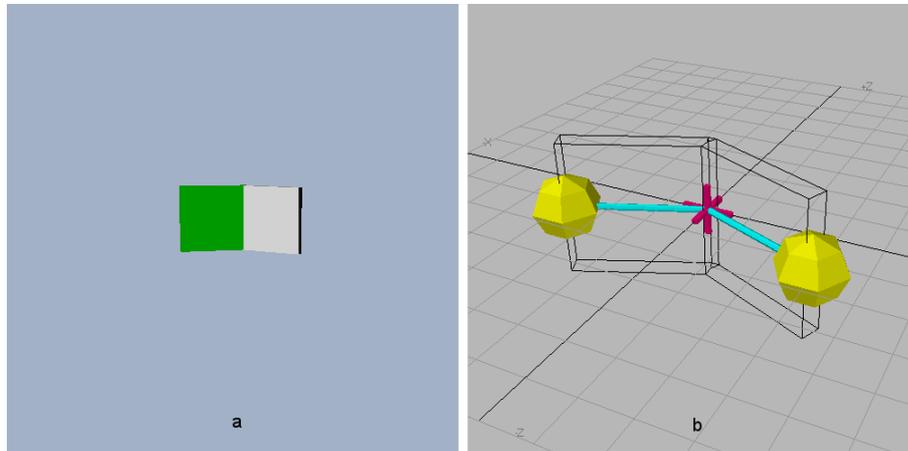


Abbildung 5.6: : Einhand-Hebel : 3D-Ansicht (l. oben), Physik-Ansicht (r. oben)

## Griff

Eine Realisierung des Griffs stellt, technisch gesehen, keinen großen Unterschied zum Schaltknopf 5.5.1 dar. Das Griffobjekt besitzt einen Freiheitsgrad längs der z-Achse. Die Ruheposition liegt beim Start auf dem minimalen Abstand zum Mittelpunkt des Artefaktes. Der Griff kann nun von einer Hand gegriffen und dann längs der z-Achse heraus gezogen werden.

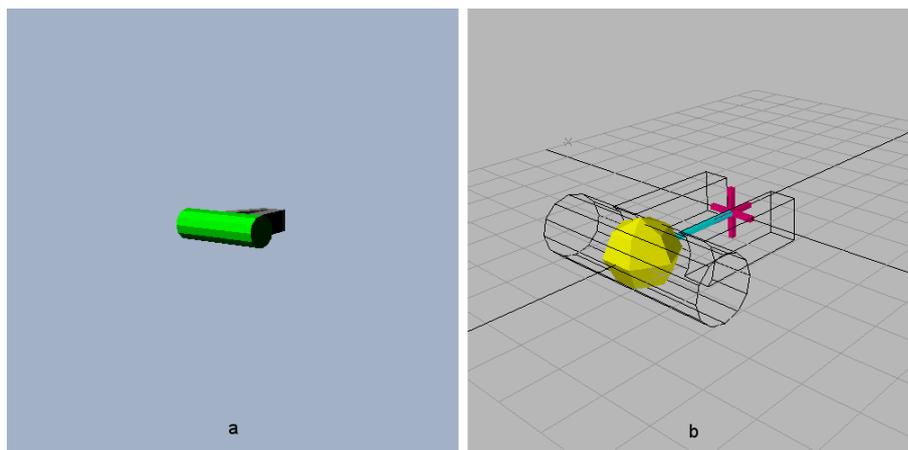


Abbildung 5.7: : Einhand-Hebel : 3D-Ansicht (l. oben), Physik-Ansicht (r. oben)

Vergleichbar mit dem Schaltknopf schaltet auch dieses Artefakt für gewöhnlich nach mindestens 80% des Maximalabstands, abhängig von dem installierten Anzeiger. Im Unterschied zum Schaltknopf bleibt der Aktor allerdings nach dem Zustandswechsel mit dem Griff verbunden. Mittels des Aktors kann der Griff nun wieder in seine ursprüngliche Lage zurückgeführt werden, wo sich abermals der Zustand dann ändert, wenn der aktuelle Wert 20% des Maximalwerts unterschreitet. Die maximale Auslenkung des Artefakts beträgt 1.5 Maßeinheiten.

## Joystick

Als letztes Artefakt ist hier der Joystick aufgeführt. Beim Joystick handelt es sich um das technisch anspruchsvollste Artefakt, da es zwei Freiheitsgrade besitzt. Der Joystick kann sowohl um die x-Achse, wie auch um die z-Achse rotieren. Es handelt sich dabei um eine Teilrotation mit jeweils 80°, von der Ruheposition aus gesehen, in alle Richtungen. In der Ruheposition steht der Joystick zentral nach oben, wie in Abbildung 5.8 zu sehen. Die Rotation um die beiden Achsen kann kombiniert erfolgen. Prinzipiell kann der Joystick eine diskrete wie auch kontinuierliche Statusänderung besitzen. In der jetzigen Umsetzung wurde, aufgrund des diskreten Joysticks 2.4.6, der als Beispiel diente, eine Umsetzung für diskrete Zustände gewählt.

Zu unterscheiden gibt es hier noch die Variante von fünf Zuständen zu der Variante mit neun Zuständen. Die Tabelle 5.2 zeigt die jeweiligen möglichen Zustände und ihre Codierung für eine spätere Weiterverarbeitung.

Code	Fünf Zustände	Neun Zustände
1		vorn-links
2	vorn	vorn
3		vorn-rechts
5	rechts	rechts
9	mitte	mitte
6		hinten-rechts
7	hinten	hinten
8		hinten-links
4	links	links

Tabelle 5.2: Tabelle mit Zuständen verschiedener diskreter Joystick Umsetzungen

Übergänge für die Variante mit fünf Zuständen wurden nach 60% der maximal Auslenkung gewählt. Für die Version mit neun Zuständen reagierte das System bereits nach 40% der Maximalauslenkung. Zu beachten ist hier noch in Abbildung 5.8(a) das auf eine Ver-

kleidung für den Joystick nach ersten Tests verzichtet wurde, da bei dieser zusätzlich der Blick auf den Aktor verdeckt wurde, wenn der Benutzer, räumlich gesehen, unterhalb des Joysticks mit seiner Hand war.

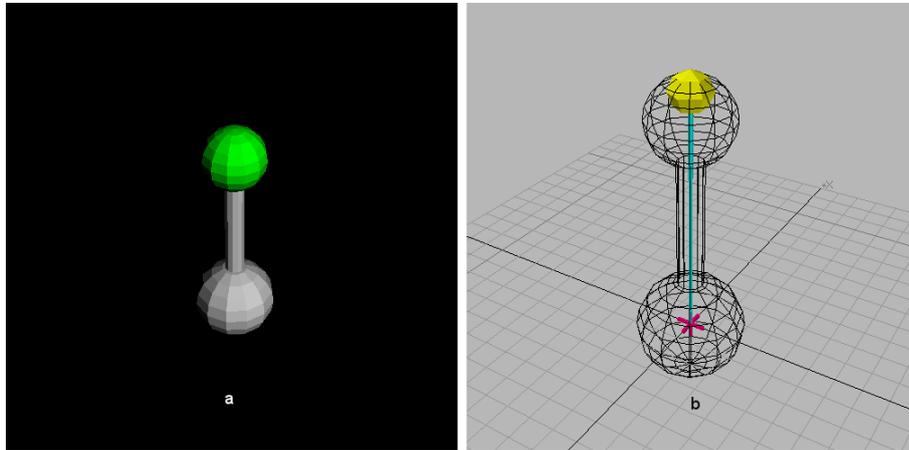


Abbildung 5.8: : Joystick : 3D-Ansicht (a), Physik-Ansicht (b)

## 5.5.2 Umsetzung der Anzeiger

Die Realisierung der Anzeiger umfasst sowohl solche, die über eine eigene grafische Darstellung verfügen, also auch solche, die die ermittelten Werte und Zustände der Artefakte nur weiter leiten können. Die Anzeiger werden bei der Initialisierung der Artefakte übergeben. Die realisierten Anzeiger werden nun im Einzelnen aufgeführt.

### Lampe

Die Lampe stellt die erste Umsetzung eines Anzeigers dar. Es handelt sich hierbei um einen diskreten Anzeiger mit zwei Zuständen. Abbildung 5.9(a) zeigt die Lampe im Zustand „aus“ und die Abbildung 5.9(b) ist die Darstellung, wenn die Lampe an ist. Das bereits in der Umsetzung der Artefakte aufgeführte Schaltverhalten mit der 80 20 Regel ist durch diesen Anzeiger geprägt. Sind Änderungen für einzelne Artefakte, die mit einem Lampenanzeiger verknüpft sind, erwünscht, gilt es eine weitere anders implementierte, beziehungsweise konfigurierte Instanz dieses Anzeigers zu erschaffen.

### Wasserstandanzeiger

Der Wasserstandanzeiger ist die zweite Variante einer Umsetzung für einen Anzeiger. Dieser Anzeiger verfügt über eine kontinuierliche Darstellung von 0 bis 100. Die Anzeige erfolgt

durch einen simulierten Wasserpegel. Der Pegelstand steigt mit einer kleinen Verzögerung und pro Zeiteinheit nur um ein bestimmtes Maß. Ein schlagartiger Wechsel von null auf hundert oder umgekehrt ist somit nicht möglich. Einerseits soll dies eine gewisse Natürlichkeit simulieren, andererseits wird damit auch ein mögliches Zittern des Artefakts, welches durch ein Zittern des Aktors verursacht wird für die Anzeige unterbunden. Der Wert 0 wird durch einen nicht mehr sichtbaren Pegel gezeigt. Der Wert 100 durch eine bis zum oberen Rand gefüllte Wassersäule. Die Darstellung dazwischen ist linear, sprich bei einem Wert 50, ist die Wassersäule bei der Hälfte angekommen. Abbildung 5.9(c) und 5.9(d) zeigen den Wasserstandanzeiger mit unterschiedlichen Werten.

## Anzeigemeter

Das Anzeigemeter ist inspiriert durch ein Tacho- oder Voltmeter. Hier wird aber der innere Zustand eines Artefakts abgebildet und keine Messgröße wie Geschwindigkeit oder elektrische Spannung. Die fünf Markierungen auf dem Zeigerblatt stellen die Werte 0, 25, 50, 75 und 100 dar. Auch diese Umsetzung eines Anzeigers besitzt eine Verzögerung bei der Darstellung des momentanen Wertes, die Gründe hierfür sind identisch zu denen beim Wasserstandanzeiger. In den Abbildungen 5.9(e) und 5.9(f) sind zwei unterschiedliche Werte eines Artefakts visualisiert.

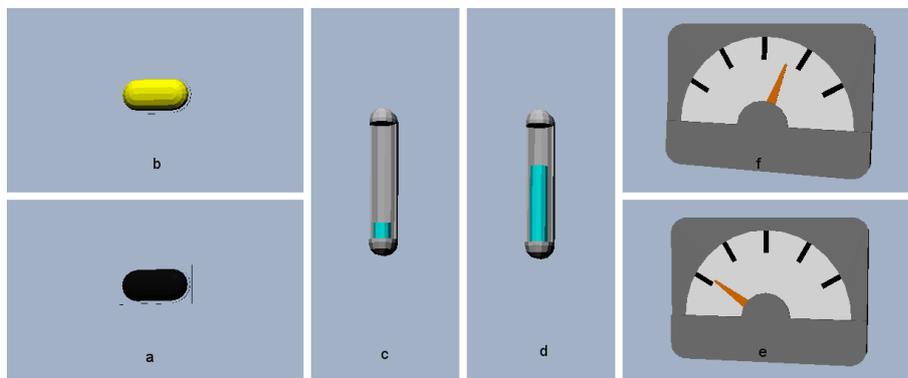


Abbildung 5.9: : Einhand-Hebel : 3D-Ansicht (l. oben), Physik-Ansicht (r. oben)

## Message Broker Anzeiger

Diese Komponente arbeitet ähnlich wie ihre Vorgänger. Allerdings besitzt diese Variante des Anzeigers, keine eigene grafische Ausgabe um einen diskreten Zustandswechsel oder kontinuierliche Wertänderungen anzuzeigen. Sie verfügt über eine Anbindung an die ActiveMQ. Die Werte werden vom Artefakt an den Anzeiger weitergeleitet. Hier werden diese entsprechend der Ausprägung des Anzeigers weiterverarbeitet und dann an den Message Broker

weiter gesandt. Dieser Vorgang wird kontinuierlich wiederholt. Die Queue an die diese Nachrichten versandt werden, wird in der Initialisierungsphase individuell festgelegt und ist über die Laufzeit nicht änderbar.

## 5.6 Grafische Ausgabeverfahren

Wie im Diagramm 4.4 beschreiben, bekommen die Ausgabemedien die Daten für die grafische Ausgabe über den Message Broker. Die Ausgabemedien müssen selbst über eine grafische Repräsentation der einzelnen Objekte verfügen. Mittels Übertragung wird nur der Zustand und die Position der einzelnen Objekte übertragen und ständig aktuell gehalten.

Die Realisierung erfolgt ebenfalls mit Hilfe des JMonkeyEngine in der Version 3. Alle in der 3D-Welt vorhandenen Objekte sind eindeutig identifizierbar und können somit gezielt über ihren Schlüssel angesprochen werden. Dieser Schlüssel liegt den einzelnen Nachrichten des Message Brokers bei, somit können einzelne Objekte gezielt verändert werden.

In der praktischen Umsetzung hat sich gezeigt, dass es sich nicht lohnt die Physikberechnung und die grafische Ausgabe getrennt zu starten, da beide wie beschrieben das gleiche Framework nutzen. Hierfür kann die grafische Ausgabe auch direkt über die Physik angesprochen werden. Dies erleichterte die Entwicklung und Fehlersuche. Das eigentlich angestrebte Verfahren über den Message Broker bleibt aber erhalten und man kann beliebig viele weitere grafische Ausgaben anbinden.

### Powerwall

Die Powerwall, wie in Abbildung 5.10 zu sehen, ist mit 2 „ATI Radeon™ HD 5870 Eyefinity 6“<sup>12</sup> Grafikkarten ausgestattet und ermöglicht so eine maximale Auflösung von 5760 \* 2160 Pixeln. Für die anstehenden Versuche wird die Powerwall sowohl zur Berechnung der Physik als auch zur grafischen Anzeige der Schalttafel benutzt. Wie dem Diagramm in 6.1 zu entnehmen ist, wird auch die Kinect direkt an die Powerwall angeschlossen. Somit befindet sich der benötigte Adapter auf der Powerwall und wird dort gestartet. Auch der Message Broker in Form einer ActiveMQ liegt bei den Versuchen auf der Powerwall, da diese bei allen Versuchen nie an ihre rechnerischen Leistungsgrenzen gekommen ist, lohnte es sich nicht, einzelne Komponenten des Frameworks auf andere Rechner auszulagern. Die Kommunikation erfolgt aber weiterhin über die ActiveMQ, und somit wäre eine Auslagerung bei Engpässen in der Rechenkapazität jederzeit gewährleistet.

<sup>12</sup><http://www.amd.com/de/products/desktop/graphics/ati-radeon-hd-5000/hd-5870-eyefinity-6-edition/Pages/overview.aspx>



Abbildung 5.10: : Powerwall Stand 06.02.2014

## Sekundär PC

Für Teile des Systems kann es nötig oder aber sinnvoll sein, diese auf einem zweiten Computer ausgeführt zu werden. Dies betrifft in erster Linie Komponenten die unter Verwendung der 3D- und Physik-Engine arbeiten, da hier immer nur eine Anwendung zurzeit im Fullscreen-Modus laufen kann. Darüber hinaus ermöglicht das Auslagern bestimmter Komponenten eine separate Überwachung. Besonders im Fall des Adapters für den ART-Tracker kann es von Vorteil sein, hier die Verfahren zum Erkennen der Greif- und Loslassgeste über einen separaten Monitor zu überwachen.

## Headup-Display (HUD)

Das Headup-Display (Abbildung 5.11), im weiteren Verlauf der Arbeit auch mit HUD abgekürzt, stellt in seiner Hardwareform eine Art Hybrid dar, einerseits ist es ausschließlich zur Wiedergabe der Grafik gedacht, andererseits benötigt es zur Darstellung der Grafik selbst seine spezifische Position im Raum, damit das passende Bild für den Träger des Headup-Displays errechnet werden kann. Das Headup-Display benötigt somit auch eine eigene Sensorenkomponente, die es mit den benötigten Werten versorgt. In diesem Fall wurde das ART-Trackingsystem zu Hilfe genommen um die Position zu bestimmen. Das Headup-Display wurde mit einem ART-Tracker Artefakt vom Typ Brille5 kombiniert. Das Artefakt wurde passend auf seine Ruheposition (Kopf gerade, Blickrichtung zur Powerwall) ins ART-Trackingsystem eingemessen. Die vom ART-Tracker errechneten Positions und Winkeldaten wurden empfangen, weiterverarbeitet und dann zur weiteren Verwendung ebenfalls der ActiveMQ zugesandt.

Um das Bild für das Headup-Display zu errechnen wird das in der 3D-Simulation vorhandene Kamera-Objekt mit den verarbeiteten Daten aus der ActiveMQ versorgt. Headup-Display

und Kamera-Objekt werden gleichgeschaltet. Somit kann über die Bewegung des Headup-Displays die Kamera gesteuert werden. Zusätzlich wird das erzeugte Bild noch auf dem PC ausgegeben, an dem das Headup-Display angeschlossen wird. Somit können weitere Personen verfolgen, was der Träger des Headup-Displays sieht. Die beiden Screenshots 5.11(b) und 5.11(c) zeigen die Artefakte die über das Headup-Display zu sehen sind, wobei noch zu beachten ist, dass die hier schwarzen Pixel, in der Darstellung im Headup-Display transparent sind und somit nicht abgebildet werden.



Abbildung 5.11: : Links: HUD mit ART-Artefakt  
Rechts: Grafische Darstellung des Joysticks aus der Physiksimulation

## 5.7 Akustische Ausgabe

Für die exemplarische Umsetzung des Systems wurde zwar eine mögliche Entkopplung zu Gunsten einer Kommunikation über die ActiveMQ vorgesehen. Jeder abzuspielende Sound würde mit einer ID codiert werden, die dann verschickt wird. Der Empfänger, der die entsprechenden Nachrichten abrufen muss, muss somit über alle Sounddateien verfügen und unter Verwendung der zugesandten ID die entsprechende Sounddatei abspielen. Aufgrund dessen, dass die Ausgabe wiederum über Teile der JMonkeyEngine realisiert wird, und es sich nicht empfiehlt mehrere JMonkey-Anwendungen parallel auf einem Rechner zu starten, hätte dies einen weiteren PC ausschließlich zur Ausgabe des Sounds bedurft. Daher wurde darauf verzichtet die Daten erst über JMS zu verschicken und dann auszugeben. Stattdessen wird die Soundausgabe gleich über die in JMonkey realisierte Physik&Grafik-Anwendung umgesetzt. Die Ausgabe des Sound erfolgt dann einheitlich über eine Schnittstelle zur Soundkarte, wo einfache Sounddateien in wav-Format abgespielt werden.

## 5.8 Fazit

In der Software konnten folgende Punkte umgesetzt werden:

- Aufarbeitung der ART-Tracker Daten der Hände und Erkennen der Greif- und Loslassgeste
- Auswertung der ART-Tracker Daten für das Headup-Display
- Anbindung der Kinect und somit das zur Verfügung stellen zweier Unterschiedlicher Sensoren
- Einheitliche Kommunikation der verteilten Komponenten über eine ActiveMQ
- Umsetzung einer physikalischen Simulationsumgebung
- Erstellen der Artefakte aus Kapitel 2
- Erkennen der in Kapitel 2 aufgeführten Gesten
- Ausgabe der Simulation, wahlweise mittels Headup-Display oder Powerwall

**Sensoren** Beide Sensoren, der ART-Tracker wie auch die Kinect funktionieren zum Tracken der Hände. Ihre Verwendung ist prinzipiell für das System möglich, allerdings haben beide Sensoren ihre Schwächen und Tücken. Vor allem die Qualität der ART-Tracker Daten ist stark vom Umfeld abhängig. Zu viele Reflexionen oder Sonnenlicht stören hier erheblich.

**Physik-Engine** Die Ergebnisse der Simulation sind zufriedenstellend, allerdings war die Umsetzung aller Funktionalitäten wesentlich aufwändiger als gedacht.

**Message Broker** Das System durch die Kommunikation über einen Message Broker voneinander zu entkoppeln war erfolgreich. Die befürchteten zeitlichen Verzögerungen sind so minimal, dass sie nicht ins Gewicht fallen. Allerdings zeigte sich in der Umsetzung, dass im ersten Schritt nicht alle Komponenten immer über diese Kommunikation getrennt werden müssen. Die Physik-Simulation und die für den Benutzer sichtbare 3D-Simulation wurden nicht getrennt, da dies einen Mehraufwand bedeutet hätte, ohne einen Vorteil für den gewählten Aufbau bereitzuhalten.

### 5.8.1 Änderungen bei erneuter Umsetzung

Wird sich die Frage gestellt, was man ändern würde, hätte man die Gelegenheit die Umsetzung des Frameworks erneut zu beginnen, ist hier folgender Punkt zu nennen.

Trotz der umfangreichen Recherche und dem Ausprobieren mehrerer Physik-Engines kam es dann bei der Umsetzung immer noch zu unvorhersehbaren Problemen, die nur teilweise zufriedenstellend gelöst werden konnten. Nicht zuletzt das Bedürfnis der Entwicklung eines eigenen Joins 5.3.2 zeigt diesen Umstand. Häufig sind nicht nur die Physik-Engines schuld an diesem Problem, auch die angebotenen Anbindungen an bestehende 3D-Grafik-Engines, sind, sofern überhaupt vorhanden häufig bei genauer Betrachtung, nicht befriedigend.

Bei einer erneuten Umsetzungen eines solchen Frameworks, beziehungsweise der Implementierung der Physiksimulation, wäre sicherlich die Entwicklung einer eigenen kleinen Physik-Engine zu überdenken. Der Umfang, den diese Physik-Engine haben müsste, wäre gering, da an diese ja nur wenige Anforderungen gestellt würden. Vorteil hier wäre, dass man im vollen Umfang Zugriff auf die Berechnungen erhält, und einen Funktionalitäten, die andere Physik-Engines besitzen einen nicht in der Umsetzung behindern. Mit einer Eigenen vollständig test und auch anpassbaren Physik-Engine, wo sämtliches „unerwartetes Verhalten“ oder etwaige Fehler in der eigenen Verantwortlichkeit liegen, wäre das Gesamtsystem, in seinem Kern, um eine große Unbekannte ärmer. Somit entfällt ein beträchtlicher Risikofaktor, der einer erfolgreichen Umsetzung und Weiterentwicklung im Wege steht.

## 5.9 Erweiterungsmöglichkeiten

Für die Physiksimulation selbst liegen im Moment keine ernsthaften Ideen für eine Weiterentwicklung vor, allerdings kann das System durch neue Sensoren, wie auch andere Ausgabemedien, erweitert werden, so das auch ganz andere Anwendungsszenarien denkbar sind.

### 5.9.1 Weitere Sensoren

Im Folgenden werden zwei Sensoren vorgestellt die bereits an der HAW, wenn auch nur teilweise zu ersten Testzwecken, in anderen Projekten zum Einsatz kommen.

#### Kinect 2

Bei der Kinect for Windows 2 handelt es sich um die nächste Generation von Kinect Sensor die im PC-Umfeld zum Einsatz kommen wird. Abgesehen von einer besseren Auflösung und etwas weniger Verzögerung bringt die Kinect zwei die Neuerung mit, dass sie feststellen kann, wie Körperteile gedreht sind. Man wäre mithilfe der Kinect 2 in der Lage, die Aktoren in 5.4.2 passend zur Rotation der Hände richtig auszurichten.

### Leap Motion

Die Leap Motion <sup>13</sup> ist in der Lage einzelne Finger zu erkennen und eindeutig zuzuordnen. Die Abbildung 5.12 zeigt eine dreidimensionale Interpretation der Daten, die von dem Sensor aufgezeichnet wurden. In dieser Darstellung ist der Sensor selbst in der Mitte als weißes Viereck abgebildet. In der Arbeit von Sobin Ghose Ghose (2014) wird mitunter die Leap Motion erfolgreich zur Steuerung einer Anwendung mittels Gesten eingesetzt. Die Reichweite der Sensors ist sehr gering, von daher wäre ein Tragen der Leap Motion am Körper zu überlegen. In Verbindung mit einem entsprechenden Headup-Display, könnte der Benutzer Sensor wie auch Ausgabegerät am Körper tragen.



Abbildung 5.12: : Projektion des Leap Motion

<sup>13</sup><https://www.leapmotion.com/>

## 5.9.2 Weitere Ausgabemedien

### Oculus Rift

Die Oculus Rift<sup>14</sup> ist ein Headup-Display, das über beide Augen geht und die Sicht des Benutzers auf die Realität vollkommen verdeckt. Es verfügt über ein drei Achsen Gyrometer und Beschleunigungssensoren, welche die Rotation des Kopfes feststellen können und die Rotation an das System für eine passende Bildverarbeitung weitergeben. Eine Bewegung des Kopfes längs der drei Achsen wird nicht registriert. Die Oculus Rift wird im Sitzen benutzt, da der Benutzer keine Sicht mehr auf die Realität hat. Das Headup-Display könnte im Bereich des virtuellen Cockpits 6.1.2 zum Einsatz kommen, allerdings wäre dann noch die grafische Repräsentation des externen Programms, in dem Fall der Flugsimulation, mit in die Oculus Rift zu übertragen.

### Google Glass

Die Google Glass<sup>15</sup> ist ein Headup-Display, durch welches der Benutzer schauen kann. Er kann also zusätzliche Informationen wahrnehmen, während er sich in der Realität frei bewegen kann. Die Google Glass verfügt von sich aus nicht über die Fähigkeiten, Lage und Rotation des Kopfes festzustellen. Wenn diese Aufgabe von einer anderen Komponente übernommen wird, könnte die Google Glass ein wesentlich handlicherer Ersatz für das benutzte Headup-Display (siehe 5.11) sein. Die Google Glass könnte die Rolle des Headup-Displays in Verbindung mit der Leap Motion 5.9.1 übernehmen.

---

<sup>14</sup><http://www.oculusvr.com/>

<sup>15</sup><http://www.google.com/glass/start/>

# Kapitel 6

## Usability Untersuchung

### 6.1 Aufbau

Für die einzelnen Versuche werden unterschiedliche Versuchsaufbauten benötigt. Diese Versuchsaufbauten und ihre Besonderheiten sollen hier im folgenden Abschnitt beschrieben werden. Es handelt sich hierbei jeweils um eine Konkretisierung des generellen Aufbaumusters aus dem vorherigen Kapitel 5.

#### 6.1.1 Versuchsaufbau I: Die Schalttafel

Bei dem Aufbau für den Versuch mit der Schalttafel werden bis auf den ART-Tracker Adapter alle Software Komponenten auf dem Powerwall PC gestartet. Diese Komponenten sind der Message Broker in Form der ActiveMQ. Außerdem wird die Physiksimulation auf der Powerwall gestartet, diese beinhaltet die simulierte Schalttafel. Die Ausgabe der Schalttafel erfolgt auf 4 von den 6 Monitoren der Powerwall. Als letzte Komponente wird noch der Kinect-Adapter auf der Powerwall gestartet, sofern das Tracking der Hände über die Kinect erfolgen soll. Dieser Vorgang ist mit dem ART-Tracker Adapter abzustimmen, da nur einer der beiden Adapteranwendungen zur gleichen Zeit laufen darf.

Der ART-Tracker Adapter wird auf einem sekundären Computer gestartet, damit man eine separate Bildschirmausgabe der Punktwolken bekommt. Ein Bild dieser Aussage ist der Abbildung 3.3 zu entnehmen. Das ART-Tracking System selbst besitzt seinen eigenen Server, der ohnehin getrennt von dem eigentlichen Versuchsaufbau vorhanden ist. Ein Überblick über diesen Aufbau kann dem Diagramm in 6.1 entnommen werden. Der Bereich vor den Bildschirmen der Powerwall ist freigeräumt, es befinden sich keine weiteren Gegenstände im Messbereich der Trackingsysteme. Aus diesem Grund wird an dieser Stelle von einem Foto des Versuchsaufbaus verzichtet und auf die Abbildung 5.10(a) verwiesen.

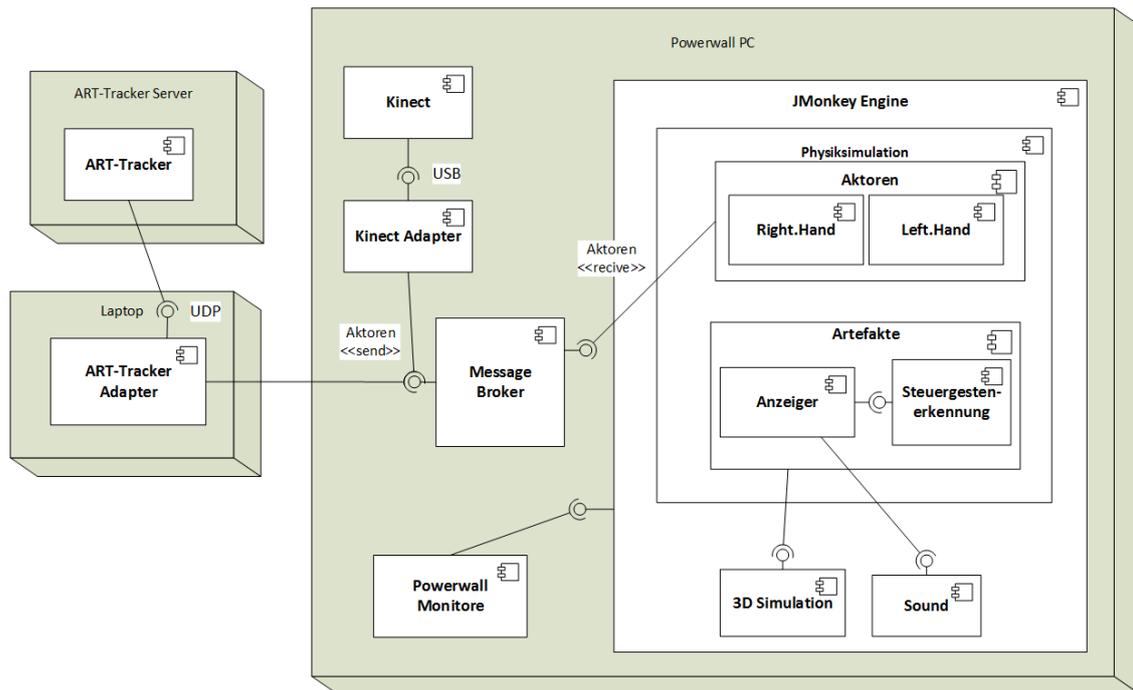


Abbildung 6.1: Verteilung der verwendeten Komponenten für den Schalttafelversuch

### 6.1.2 Versuchsaufbau II: Das virtuelle Cockpit

Wie aus der Abbildung 6.2 zu entnehmen ist, hat sich der Aufbau des Systems zum vorherigen Test stark geändert. Die Powerwall dient nun nicht mehr zur Gestenerkennung, hier laufen lediglich noch der Kinect-Adapter, die ActiveMQ und das externe zu steuernde Programm. In diesem Fall handelt es sich um eine kleine Flugsimulation, die mittels simulierten Artefakten gesteuert werden soll. Die eigentliche Physiksimulation findet auf dem Laptop statt. Von hier aus kann das erzeugte 3D-Bild aus Sicht des Piloten direkt an das angeschlossene Headup-Display ausgegeben werden.

In diesem Test wird das ART-Tracking System sowie auch der Kinect Sensor gleichzeitig verwendet, allerdings beschränkt sich der ART-Tracker auf die Positions- und Lagebestimmung des Headup-Displays. Der Kinect-Sensor übernimmt das Tracken der Hände.

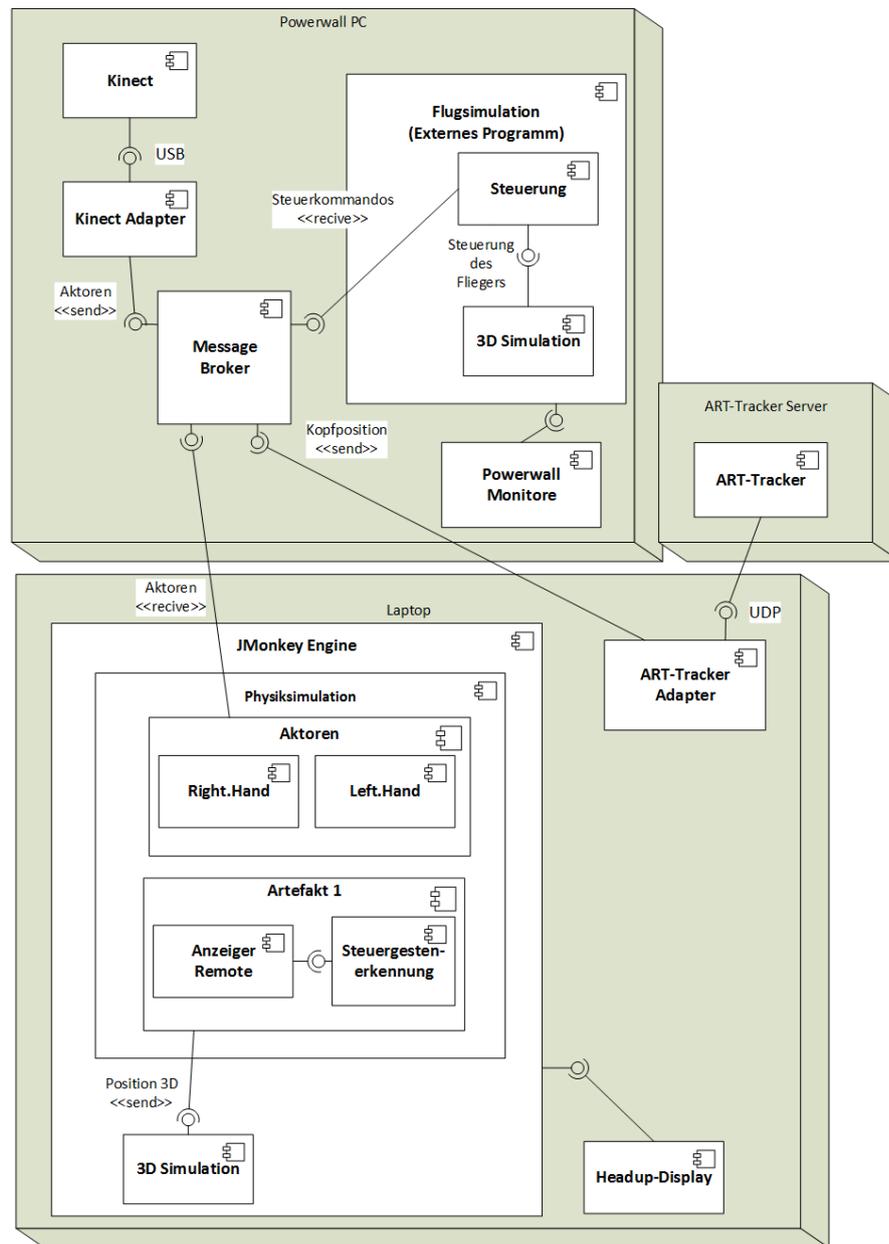


Abbildung 6.2: Verteilung der verwendeten Komponenten für das virtuelle Cockpit

Der Aufbau im Labor kann aus Abbildung 6.3 entnommen werden. Die Testperson nimmt auf dem Holzstuhl platz. Der Holzstuhl befindet sich in einem eingemessenen Standort, wie man an den Markierungen auf dem Fußboden erkennen kann. Dieses Einmessen ist notwendig, da passend zu der Position des Stuhls, auch die Artefakte in der Simulation positioniert werden müssen. Auf dem hinteren Stuhl nimmt der Testleiter platz. Alles was der Pilot über

das Headup-Display sieht, wird zugleich auf dem Laptop dargestellt. Der Testleiter überwacht von hier aus die ersten Schritte und leitet den Piloten zum Start des Tests an.



Abbildung 6.3: Versuchsaufbau für das virtuelle Cockpit

Der Joystick wird rechts von der Testperson simuliert, der Hebel für das Fahrgestell und der Schieberegler zum Schub geben befinden sich links vom Piloten. In Abbildung 6.4 sind die Artefakte abgebildet. Es ist hierbei zu beachten, dass das Headup-Display alle schwarzen Pixel transparent darstellt. Darüber hinaus befinden sich die Artefakte nicht direkt vor dem Benutzer, sondern seitlich, sodass diese Aufnahmen von einem gedrehten Kopf von ungefähr  $45^\circ$  nach links, beziehungsweise rechts und einen Blickwinkel nach unten von circa  $60^\circ$  entstanden sind.

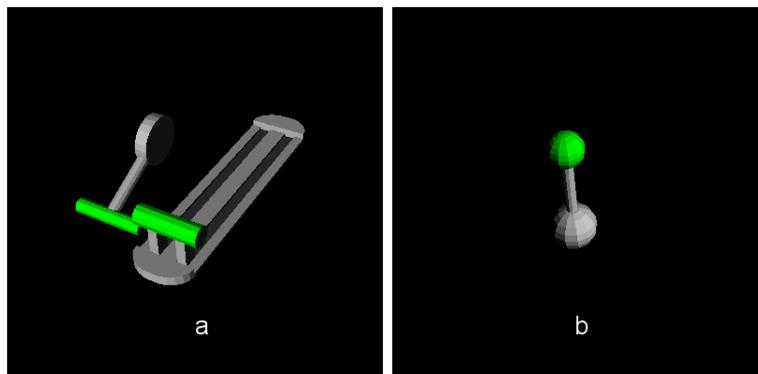


Abbildung 6.4: Sicht auf die Artefakte durch das Headup-Display

## 6.2 Durchführung

### 6.2.1 Testdurchlauf für die Schalttafel

Der erste Versuch wurde auf mehrere Tage verteilt durchgeführt. Für den Versuch mit einer Testperson wurden ungefähr 30 Minuten veranschlagt. Im Folgenden wird ein Testablauf exemplarisch beschrieben. Die Schritte sind in chronologischer Reihenfolge ausgeführt. Es ist wichtig, dass sich im Laufe eines Testdurchlaufs keine weiteren zukünftigen Testpersonen im Raum befinden, da ansonsten einige Ergebnisse durch Vorkenntnisse verfälscht werden würden.

#### Testvorbereitung

Zu Beginn eines Testdurchlaufs bekommt die Testperson den ersten Teil des Fragebogens um ihn auszufüllen. Nachdem dieser Teil ausgefüllt wurde, wird der Proband über die Funktionsweise des ART-Trackers aufgeklärt und in diesem Zuge gebeten, sämtlichen reflektierenden Schmuck, wie Ohrringe und Halsketten abzulegen. Im Anschluss zieht die Testperson die Handschuhe für den ART-Tracker an (siehe 4.2) und die Person begibt sich vor die Powerwall. Der Raum wird weitestgehend abgedunkelt.

#### Testdurchführung

Sollte die Schaltpult-Anwendung noch nicht gestartet worden sein, wird sie nun gestartet. Der Tester sieht bei dem ersten Aufbau den Hebel, ohne dass ihm explizit erklärt wird, dass es sich bei dem Artefakt um einen Hebel handelt. Darüber hinaus sind die Hände des Probanden nun ebenfalls in der Anwendung zu sehen. Es werden die drei möglichen Zustände seiner Abbildungen der Hände erklärt (siehe 5.4.2). Zugleich erfolgt noch der Hinweis wie mit den Handschuhen für den ART-Tracker am besten gegriffen werden kann. Zum Abschluss der kurzen Einführung wird dem Benutzer noch mal gezeigt, was passiert, wenn das System einmal das Tracking der Hände verliert und wie der Benutzer dann seine Hände wieder „einfangen“ oder bei den Wartepositionen „abholen“ kann. Nun beginnt der eigentliche Test. Dieser Teil wiederholt sich für alle Artefakte und wird in diesem Rahmen einmal exemplarisch für das erste Artefakt, den Hebel, durchgesprochen. Die Abbildung 6.5 zeigt eine Aufnahme mit Testperson und dem Zweihandrad-Artefakt unter Verwendung des ART-Tracker Systems.

Zuerst wird der Benutzer gefragt, was er dort genau auf der Schalttafel sieht. Sollte der Benutzer das Artefakt nicht erkennen, werden ihm Tipps gegeben. Nachdem die Art des Artefaktes geklärt wurde, wird die Funktionsweise zum Thema. In vielen Fällen ist der folgende Punkt unnötig gewesen, da die Testperson bereits anfangs nach dem Artefakt zu greifen und versuchte es zu bewegen. Sollte dies nicht der Fall gewesen sein, wird der Proband gefragt,



Abbildung 6.5: Testperson vor Powerwall mit Hebel-Artefakt

was mit einem Hebel im Regelfall gemacht werden kann und ermuntert nach dem grünen Griff zu greifen und diesen zu bewegen. Nachdem die zugrunde liegende Funktionsweise geklärt ist, wird dem Probanden Zeit gegeben, das Artefakt ausgiebig zu testen. Treten hierbei Fragen auf, werden diese geklärt und gegebenenfalls Hilfestellung geleistet. Im Laufe dieser Ausführung werden auf dem Aufgabenbogen Notizen gemacht. Zum Abschluss werden in Abstimmung mit der Testperson die Versuche mit dem Artefakt beendet und in der Anwendung auf das nächste Artefakt geschaltet. Dieser Vorgang wiederholt sich nun bis alle neun Artefakte durchlaufen wurden. Im Anschluss daran erscheint eine vollständige Schalttafel, wie in Abbildung 6.5 zu sehen ist. Hier bekommt die Testperson noch einmal die Möglichkeit alle Artefakte zu benutzen. Sie kann hier auch gleichzeitig mit zwei Artefakten interagieren. Hat die Testperson einen für sie ausreichenden Eindruck gewonnen, wird der Testdurchlauf beendet und zum Test mit dem Kinect-Sensor gewechselt. Für den Test mit der Kinect wird als erstes der ART-Tracker und der entsprechende Adapter für den ART-Tracker beendet. Darüber hinaus kann die Testperson die Handschuhe ausziehen. Es wird nun die Kinect einschließlich des dazugehörigen Adapters gestartet und der Proband passend vor der Kinect platziert, so dass diese ihn erkennt und anfängt die Positionsdaten der Hände zu übermitteln. Anschließend wird die Testanwendung erneut gestartet und die gesamte Schalttafel wieder gezeigt. Es erfolgt nun eine kurze Einweisung für den Benutzer, wie man mit dem Kinect-Sensor umzugehen hat. Das umfasst das geänderte Greifen und die etwas andere Positionierung vor der Powerwall. Ist die Einführung abgeschlossen, kann der Tester erneut mit der Schalttafel arbeiten und alle Artefakte noch einmal mithilfe der Kinect benutzen.

### **Testnachbereitung**

Nach Absprache mit der Testperson wird der Versuch mit der Kinect beendet. Der Testperson wird der zweite Teil des Fragebogens gereicht. Hier eventuell auftretende Fragen werden geklärt und gegebenenfalls Punkte auf dem Fragebogen konkretisiert. Darüber hinaus findet eine Unterhaltung über die Anwendung statt und die zu der Software gemachten Anmerkungen werden notiert.

## **6.2.2 Durchführung des Tests für das virtuelle Cockpit**

Der zweite Versuch findet auch auf mehrere Tage verteilt statt. Für den Versuch mit einer Testperson wurden ungefähr 15 Minuten veranschlagt. Im folgenden Abschnitt wird der Testablauf aufgeführt und die einzelnen Schritte sind in zeitlicher Reihenfolge notiert. Bei diesem Test ist es unerheblich, ob sich weitere Personen im Raum befinden die zu einem späteren Zeitpunkt testen wollen.

### **Testvorbereitung**

Am Anfang eines Testdurchlaufs wird der Testperson der erste Teil des Fragebogens ausgehändigt und sie gebeten diesen auszufüllen. Anschließend nimmt die Testperson auf dem für sie vorgesehenen Stuhl Platz und setzt das Headup-Display auf. Hierbei gilt die Position des Headup-Displays zu beachten ob der Bildschirm richtig vor dem rechten Auge sitzt und ob eine mögliche Brille der Testperson im Wege ist, und ob diese möglicherweise für den Test abgenommen werden kann.

### **Testdurchführung**

Zu Beginn des Testes wird zu erst das ART-Tracking System gestartet, inklusive des Adapters der die Daten des ART-Artefakts, welches auf dem Headup-Display montiert ist, verarbeiten kann. Im Anschluss wird auf dem Laptop die Anwendung für das virtuelle Cockpit gestartet und die dort errechneten Daten werden an das Headup-Display übertragen. Es erfolgt eine Justierung des ART-Artefakts auf dem Headup-Display, damit die Kopfbewegungen des Benutzers korrekt übernommen werden. Die Sicht, die der Tester über das Headup-Display bekommt, kann auf dem hinter ihm stehenden Laptop verfolgt werden. Somit ist es möglich direkt Hilfestellungen zu leisten, sollten sich bei der Interaktion mit den Artefakten Fragen oder Probleme ergeben.

Der Benutzer wird auf den Joystick rechts von seiner Position in der 3D-Welt hingewiesen. Sofern noch keine Vorkenntnisse bezüglich des Umgangs mit den Artefakten existieren, wie zum Beispiel aus dem Ersten Versuchsaufbau 6.1.1, wird der Umgang nun erläutert. Sobald der Benutzer den Joystick zuverlässig greifen gelernt hat, wird die Flugsimulation auf der

Powerwall gestartet. Ein Bild von dieser Szene ist der Abbildung 6.6 zu entnehmen. Das Flugzeug schwebt in der Luft und durch Betätigen des Joysticks lässt es sich auf der Stelle bewegen. Nachdem erste Erfahrungen mit diesem Artefakt gesammelt wurden, werden der Testperson noch die beiden Artefakte zu seiner linken gezeigt. Es handelt sich dabei um den Schieber für den Schub und den Hebel zum Einziehen des Fahrwerks ganz links außen. Die Testperson wird aufgefordert durch Betätigen des Fahrwerkhebels das Fahrwerk einzufahren und dann mit der linken Hand nach dem Schubregler zu greifen. Nun wird die Kameraansicht von der Flugsimulation insofern geändert, dass die Kamera eine feste Position hinter dem Flugzeug einnimmt und der Tester aufgefordert wird, ein wenig Schub zu geben. Der eigentliche Flug beginnt nun und dem Benutzer wird freie Hand gelassen bei seinen Tätigkeiten. Sollte der Flieger den Boden berühren, startet das Flugzeug wieder auf seiner Ursprünglichen Position und es kann weiter geflogen werden.



Abbildung 6.6: Testperson vor Powerwall mit Headup-Display

### Testnachbereitung

In Rücksprache mit dem Probanden oder durch die Aufforderung vom Probanden selbst, wird der Test beendet und es wird der zweite Teil des Fragebogens ausgehändigt. Sollten Unklarheiten bei den Fragen auf dem Fragebogen vorhanden sein, werden diese geklärt. Außerdem findet noch eine kurze Unterhaltung über das Erlebte statt.

## 6.3 Ergebnisse und Diskussion

Ein kurzes Vorwort unter welchem Blickwinkel die folgenden Testergebnisse zu bewerten sind. Um die in 1.2 gefassten Ziele bezüglich bekannter Artefakte aus der Realität zu testen, wurden die Testpersonen im Unklaren gelassen, was genau getestet werden sollte. Ihnen wurde das Komplettsystem vorgestellt, das sie benutzen sollten und nicht einzeln auf die Punkte eingegangen, welche die Hauptziele dieser Arbeit umfassen. Hauptgrund dafür war gerade die Hypothesen zu prüfen, ob die Übernahme von Form und Funktion realer Gegenstände in die Virtualität einen Einstieg in die Benutzung dieser Gegenstände vereinfacht, und somit ein „Natural Interface“ geschaffen werden konnte. Auch ein expliziter Hinweis auf das Testen der Gestensteuerung an sich hätte diesem Ziel im Wege gestanden, da die Artefakte ja Hauptbestandteil dieser Gestenerkennung sind. Dies hatte nun zur Folge, dass viele ihr Augenmerk auf andere Teile des Komplettsystems richteten. Vor allem die zum Teil nicht zufriedenstellenden Sensorleistungen wurden somit Hauptbestandteil der Kritik am Gesamtsystem. Die Gestensteuerung, das Erkennen und das Benutzen der Gegenstände fiel dabei in den Hintergrund, oftmals auch deswegen, da vielen Testpersonen die Nutzung als selbstverständlich erschien und dies als natürlich gegeben hingenommen wurde.

### 6.3.1 Ergebnisse Schalttafelversuch

Die Ergebnisse des Versuchs liegen im Anhang B als Rohdaten B oder ausgewertet als Grafiken B vor. Wie aus den Daten zu entnehmen ist, sind die Probanden bunt gemischt. Die Ergebnisse zeigen keine Auffälligkeiten im Zusammenhang mit bestimmten Merkmalen der Probanden.

Die Ergebnisse zeigen, dass die überwiegende Zahl der Artefakte von der Mehrzahl der Testperson schnell erkannt wurde, und sich auch der Funktionalität dieser Artefakte dann bewusst waren. Einzig der Schaltknopf und der Griff fielen hier etwas aus dem Mittel. Der Schaltknopf wurde seltener erkannt. Sobald er erkannt wurde, konnte die überwiegende Mehrzahl der Personen den Knopf ohne große Hilfestellung bedienen. Beim Griff hingegen war es umgekehrt, er wurde zwar erkannt, doch die Funktion, ihn zu sich ziehen zu müssen war nicht immer ersichtlich. Auch die Ausführung verlief nicht ohne Pannen.

Durch die Ergebnisse des Testes und die Unterhaltungen mit den Probanden kann die Aussage getroffen werden, dass alle Artefakte, die nicht über die Tiefe bedient werden müssen, am besten für eine Benutzung geeignet sind. Darunter fallen, der Schalthebel, der Schieberegler und das Rad.

Die Artefakte, die mit zwei Händen zu bedienen waren, schnitten in Punkte Benutzbarkeit nicht so gut ab, was von den Beobachtungen her aber häufig auf Ungenauigkeiten mit den Sensoren zurück zu schließen war.

Als Sensor wurde die Kinect besser bewertet. Hauptargumente hier waren, dass keine Handschuhe benötigt wurden und die Kinect das Greifen häufig besser erkannt hat.

Abschließend kann man sagen, dass sich der überwiegende Teil der Kritik an den Sensoren festmachen ließ. Die Schalttafel selbst wurde als gegeben hingenommen. Auch der Aufgabenbogen, wo notiert wurde, wie viel Hilfe ein Proband benötigte zeigt, dass sobald erste Anfangsschwierigkeiten gelöst waren, die Mehrheit der Probanden gänzlich ohne Hilfe auskam.

Eine große Mehrheit der Testperson, kann sich den Einsatz einer so oder ähnlich gearteten Gestensteuerung in Zukunft vorstellen. Alles in allem kann der Versuch als erfolgreich betrachtet werden.

### **6.3.2 Ergebnisse des virtuelle Cockpits**

Die Gruppe von Probanden bestand für diesen Versuch ausschließlich aus Informatikstudenten. Dementsprechend war hier mit Benutzern zu rechnen, die viel Erfahrung im Umgang mit Computern haben.

Die Steuerung wurde als intuitiv betrachtet. Eine Mehrzahl der Benutzer wertete die Funktionsweise als gut oder sehr gut. Die Sicht durch das Headup-Display wurde als mittelmäßig befunden, hier ist durch die Kommentare und Gespräche hinzuzufügen, dass das Headup-Display ein zu kleines Sichtfeld bot.

Die Steuerung der Flugsimulation wurde durchschnittlich bewertet. Sobald man sich auf die Flugsimulation konzentrierte, empfanden eine Mehrheit der Probanden, die Steuerung als schwer oder mittelmäßig, nur wenige empfanden sie als gut oder sehr gut.

Alle Benutzer wünschen sich ein weiteres Feedback, die meisten davon ein haptisches Feedback, einige auch eine akustische Rückmeldung oder eben beides.

Abschließend kann man über diesen Test sagen, dass die Benutzung zu Anfang gewöhnungsbedürftig ist. Auch hätte ein kontinuierlich reagierender Joystick sicher dazu beigetragen, dass die Steuerung als einfacher empfunden wurde. Alles in allem kann man den Versuch als Erfolg sehen.

# Kapitel 7

## Schluss

### 7.1 Zusammenfassung

Die Ziele dieser Arbeit waren die Entwicklung einer dreidimensionalen Gestensteuerung und die Durchführung einer Usability Untersuchung. Die Gestensteuerung sollte anhand von virtuellen Gegenständen unter Verwendung einer Physik-Simulation, stattfinden. Die Untersuchung sollte zeigen, inwieweit Abbildungen von realen Gegenständen in Form und Funktionalität eine Benutzung solcher Systeme vereinfachen können.

Im Zuge der Arbeit wurden verschiedene Gestenerkennungen kurz beleuchtet und mehrere themenverwandte Arbeiten vorgestellt und für die eigene Arbeit als Grundlage und Ideengeber herangezogen. Im Anschluss an das Szenario folgte die Vorstellung der Schalttafel-Metapher sowie verschiedene Artefakte wie Hebel und Schalträder. Diese Artefakte sind alle realer Natur. Es werden Gesten gesucht, die beim Benutzer dieser Artefakte gebräuchlich sind. Diese Gesten wurden dann genauer untersucht. Es folgte die Analyse der funktionalen Anforderungen. Die Analyse schloß mit einer kurzen Beschreibung eines ersten Versuchs, der die grundsätzliche Machbarkeit überprüfte.

Im 3. Kapitel wurden die Algorithmen vorgestellt, die später im Design mit Verwendung fanden und anschließend in der Umsetzung auch zur Anwendung kamen. Hier gab es einen Schwerpunkt für die Erkennung von Händen und der Greifgesten für den ART-Tracker. Der erste Abschnitt im Designkapitel umfasst die nötigen Funktionalitäten und beschreibt mögliche Lösungen für die einzelnen Komponenten, die in dem System benötigt wurden. Im Bereich der Sensorik wurden zwei verschiedene Motion-Tracker vorgestellt.

Im Bereich der Simulation wurden verschiedene Wege der Interaktion vorgestellt, darüber hinaus wurden Anforderungen an die benötigte Physik-Engine formuliert. Ausgehend von der Erfüllung dieser Anforderungen wird die Gestensteuerung beschrieben. Es wurde verdeutlicht, dass Zustandsänderungen der simulierten Objekte als Gestenerkennung benutzt wurden. Anschließend werden noch die möglichen Ausgabemedien aufgeführt und deren Einsatzmöglichkeiten aufgezeigt. Im zweiten Teil wurde auf die für die Umsetzung zugrunde

gelegte Softwarearchitektur verwiesen. Es wurde die Verwendung eines verteilten Systems beschreiben, welches mittels Nachrichten kommuniziert, und dessen Aufbau erläutert.

Der nächste Abschnitt der Arbeit umfasste die Realisierung. Es folgten technische Details zur Anbindung der einzelnen Sensoren, als auch die Wahl einer Physik-Engine. Hier kamen Erfahrungen aus weiteren vorherigen Versuchsaufbauten zum Tragen. Im Rahmen der Umsetzung der Schalttafel wurden die Umsetzungen der einzelnen simulierten Artefakte genau beschrieben. Auch die für die Anzeige verwendeten simulierten Gegenstände wurden hier genau aufgeführt. Im Anschluss daran wurden die für die Ausgabe benutzten Geräte und deren Einbindung ins System genauer untersucht. Im Anschluss folgte ein technisches Fazit und verschiedene Möglichkeiten der Weiterentwicklung aufgezeigt. Darüber hinaus wurden noch andere Sensoren und Ausgabemedien vorgestellt, die Verwendung in diesem Aufbau finden könnten.

Es haben im Anschluss an die Realisierung zwei unterschiedliche Tests zur Untersuchung der Usability stattgefunden. Es wurden der Testaufbau und die folgende Ausführung der Tests beschrieben.

Anschließend fand eine Bewertung der Testergebnisse statt. Den Abschluss dieser Arbeit bildet dieses Kapitel, es enthält diese Zusammenfassung, ein Fazit und einen Ausblick.

## 7.2 Fazit

Die hier entwickelte Umsetzung des erdachten Frameworks bietet ein Beispiel für eine physikbasierte Variante einer dreidimensionalen Gestenerkennung. Die in der Umsetzung aufgetretenen Schwierigkeiten und Grenzen des Systems liegen alle in technischen Bereichen. Die limitierenden Faktoren waren zum einen die Schwächen in der Genauigkeit und Zuverlässigkeit der für das Tracking verwendeten Kameras. Dabei zeigten beide Systeme unterschiedliche Ausprägungen dieser Schwächen. Zum Anderen erwies sich die Umsetzungen mithilfe der Physik-Engine als überraschend mühsam, so dass eine Weiterentwicklung erst nach dem Austausch dieser Komponente in Betracht gezogen werden sollte. Dies alles sind aber Schwierigkeiten, die durch andere und neuere Systeme sicherlich in naher Zukunft gelöst werden können. Somit kann man das Gesamtsystem als funktionstüchtig betrachten.

Als weiteres Fazit kann man die Benutzertests hinzunehmen und hinsichtlich der aufgestellten Hypothesen aus Abschnitt 1.2 betrachten. Es ist gelungen durch die Verwendung von real existierenden Gegenständen wie Schaltern, Rädern und Hebeln, ein virtuelles System zu erschaffen, da, in den meisten Fällen keinerlei Anleitung bedarf. Der größte Teil des Erlernens bezieht sich auf die oben angesprochene Kamerasensorik und ihre spezifischen Eigenheiten, was für die Fragestellung aber unerheblich ist.

### 7.3 Ausblick

Die Weiterentwicklungen im Bereich der Kamerasensorik werden die noch bestehenden Probleme von Latenz und Ungenauigkeiten lindern. Diese Schwierigkeiten sind sicherlich als einer der begrenzenden Faktoren eines solchen Systems anzusehen.

Darüber hinaus eröffnet der Fortschritt und der damit verbundene Preisverfall in diesem Bereich ein breiteres Anwendungsspektrum für diese Art von Kommunikation und Steuerung von Computersystemen. Der Bereich der Computerspiele und allgemeiner Unterhaltungselektronik nimmt hier bereits eine Vorreiterrolle ein. Ein gutes Beispiel ist hier die Xbox One<sup>16</sup>, die standardmäßig über ein Motiontrackingsystem verfügt. In diesem Bereich sind Steuerungen mit einfachen Gesten bereits alltäglich geworden, dies wird sich auch auf andere Bereiche ausdehnen.

Eine Form der berührungslosen und controllerfreien Kommunikation mit Computersystemen ist vor allem in Bereichen des öffentlichen Raums denkbar. Vorstellbar ist ein solches System überall dort, wo nicht allzu komplexe Tätigkeiten mit einem System geplant sind. Andererseits dort, wo Kommunikationsmedien wie Touchscreens nicht praktikabel sind, da zu schnelle Abnutzung, durch Schmutz und Staub, oder aber auch Vandalismus einen solchen Einsatz nur schwer vorstellbar machen.

Fällt die Wahl der Artefakte auf welche, die dem Benutzer aus der Realität vertraut sind, kann dies zu einer Steigerung des Grades der Immersion führen.

Aber auch im nicht öffentlichen Raum besteht Bedarf, unter anderem im Umfeld der „Smart Environment“ Entwicklung. Hier bietet vor allem der große Einzugsbereich von Systemen zum Motiontracking die Chance, dass eine solche Kommunikation nicht weiter ortsgebunden stattfinden muss. Die HAW Hamburg ist seit 2009 mit der Entwicklung eines „Smart Home“ befasst, welches unter dem Namen 'living place'<sup>17</sup> läuft. Eine Installation eines solchen Systems zur Gestensteuerung wäre hier sicher vorstellbar.

Prinzipiell kann ein solches System in allen Bereichen Einzug halten, wo keine zeitkritische Interaktion erfolgen muss, oder aber das Wiederholen einer Geste keine kritische Auswirkungen hätte. Inwieweit solche Systeme aber tatsächlich in Zukunft in den alltäglichen Gebrauch kommen werden, gilt abzuwarten, da neben der technischen Herausforderung, nicht zuletzt der Mensch über dessen Akzeptanz entscheiden wird. Hier wird die Zukunft zeigen, wo Gesten eine alltagstaugliche Alternative zur herkömmlichen Steuerungen ergeben.

---

<sup>16</sup><http://www.xbox.com/de-DE/xbox-one/meet-xbox-one>

<sup>17</sup><http://livingplace.informatik.haw-hamburg.de/blog/>

# Literaturverzeichnis

- [GeoLab ] : *Geometry Lab*, letzter Abruf: 20.02.2014. – URL <http://web.informatik.uni-bonn.de/I/GeomLab/RandomPolygon/index.html.en#convex>
- [Handkurbel ] : *Manivelle du tour Somua - Capucins*, letzter Abruf: 24.02.2014. – URL [http://upload.wikimedia.org/wikipedia/commons/c/c8/Manivelle\\_du\\_tour\\_Somua\\_-\\_Capucins,\\_Brest.jpg](http://upload.wikimedia.org/wikipedia/commons/c/c8/Manivelle_du_tour_Somua_-_Capucins,_Brest.jpg)
- [Schieberegler ] : *Mischpult*, letzter Abruf: 24.02.2014. – URL [http://upload.wikimedia.org/wikipedia/commons/8/85/Audio\\_mixer\\_faders.jpg](http://upload.wikimedia.org/wikipedia/commons/8/85/Audio_mixer_faders.jpg)
- [Notaus ] : *Not-Aus Betätiger*, letzter Abruf: 24.02.2014. – URL [http://upload.wikimedia.org/wikipedia/commons/f/f9/Not-Aus\\_Bet%C3%A4tiger.jpg](http://upload.wikimedia.org/wikipedia/commons/f/f9/Not-Aus_Bet%C3%A4tiger.jpg)
- [Griff ] : *Notbremse*, letzter Abruf: 24.02.2014. – URL <http://upload.wikimedia.org/wikipedia/commons/1/1d/RB-Notbremse.jpg>
- [Schalttafel ] : *Schalttafel im Schiffshebewerk Henrichenburg*, letzter Abruf: 24.02.2014. – URL [http://upload.wikimedia.org/wikipedia/commons/3/3f/Henrichenburg\\_old\\_boat\\_lift\\_06.jpg](http://upload.wikimedia.org/wikipedia/commons/3/3f/Henrichenburg_old_boat_lift_06.jpg)
- [ARTtracker-Anleitung 2007] *ARTtrack und DTrack Benutzerhandbuch v1.24.3 - A.R.T. GmbH, 2007*
- [Barnkow 2012] BARNKOW, Lorenz: *Kollaboration an Multitouch-Tischen: Visualisierung von Besitz bei Gruppenarbeit*. 2012. – URL [users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/barnkow.pdf](http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/barnkow.pdf)
- [Brüggemann 2013] BRÜGGEMANN, Daniel: *Konzept und prototypische Entwicklung eines Assistenzsystems für Präsentationen auf Grundlage von Microsoft Kinect*. 2013. – URL [www.cs.uni-paderborn.de/fileadmin/Informatik/FG-Szwillus/Bachelor-Studienarbeiten/Br%C3%BCggemann\\_\\_Daniel.pdf](http://www.cs.uni-paderborn.de/fileadmin/Informatik/FG-Szwillus/Bachelor-Studienarbeiten/Br%C3%BCggemann__Daniel.pdf)

- [Duncan 2013] DUNCAN, Greg: *Using the InteractionStream without WPF*. 2013. – URL <http://channel9.msdn.com/coding4fun/kinect/Using-the-InteractionStream-without-WPF>
- [ERAP 2012] ERAP, Samet: *Gesture Based PC Interface with Kinect Sensor*. 2012. – URL [http://comserv.cs.ut.ee/forms/ati\\_report/downloader.php?file=D6F29AA32480E9B57525B6CA46923DAE9BA3F97B](http://comserv.cs.ut.ee/forms/ati_report/downloader.php?file=D6F29AA32480E9B57525B6CA46923DAE9BA3F97B)
- [Erich Gamma 2001] ERICH GAMMA, Ralph Johnson John V.: *Entwurfsmuster*. Bonn : Addison-Wesley, 2001
- [Ghose 2014] GHOSE, Sobin: *Konzeption und Evaluation eines interaktiven Badezimmerspiegels*. 2014. – URL [users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/ghose.pdf](http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/ghose.pdf)
- [Hara 2011] HARA, Tenshi C.: *Realisierung der Erkennung menschlicher Bewegungsmuster durch Wireless Sensor Nodes*. 2011. – URL <http://hara.tc/publications/CaptureWSN.pdf>
- [Heitsch 2008] HEITSCH, Johann: *Ein Framework zur Erkennung von dreidimensionalen Gesten*. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/heitsch.pdf>
- [Hilliges 2012] HILLIGES, Otmar u.: *HoloDesk: direct 3d interactions with a situated see-through display*. 2012. – URL <http://doi.acm.org/10.1145/2208276.2208405>
- [K. Denker 2012] K. DENKER, u.a.: *3D Hand Gesture Recognition Based on Sensor Fusion of Commodity Hardware*. 2012. – URL <http://cg.cs.uni-kl.de/~denker/publications/pdf/mucl2.pdf>
- [Kokott 2013] KOKOTT, Christian: *Using the Kinect InteractionStream in XNA*. 2013. – URL <http://getgeeksome.wordpress.com/2013/05/09/using-the-kinect-interactionstream-in-xna/>
- [Latoschik 2005] LATOSCHIK, Marc E.: *A User Interface Framework for Multimodal VR Interactions*. 2005. – URL <http://doi.acm.org/10.1145/1088463.1088479>
- [Mohammadali Rahimi 2008] MOHAMMADALI RAHIMI, Matthias V.: *Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie*. 2008. – URL [http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/rahimi\\_vogt.pdf](http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/rahimi_vogt.pdf)

- [Nogalski 2012] NOGALSKI, Malte: *Gestengesteuerte Positionierung von Klangquellen einer Wellenfeldsynthese-Anlage mit Hilfe eines kamerabasierten 3D-Tracking-Systems*. 2012. – URL [users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/nogalski.pdf](http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/nogalski.pdf)
- [Potratz 2011] POTRATZ, Olaf: *Ein System zur physikbasierten Interpretation von Gesten im 3D-Raum*. 2011. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/potratz.pdf>
- [Potratz 2012a] POTRATZ, Olaf: *Entwicklung eines Gestenframeworks*. 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2012-proj1/potratz.pdf>
- [Potratz 2012b] POTRATZ, Olaf: *Physikbasierte Interaktion im virtuellen Raum*. 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2012-aw2/potratz/bericht.pdf>
- [Potratz 2013] POTRATZ, Olaf: *Erste Versuche mit einem 3D-Gestenframework*. 2013. – URL [users.informatik.haw-hamburg.de/~ubicomp/projekte/master12-13-proj2/potratz.pdf](http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master12-13-proj2/potratz.pdf)
- [Roßberger 2008] ROSSBERGER, Philipp: *Physikbasierte Interaktion in kollaborativen computergestützten Umgebungen*. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/rossberger.pdf>
- [Söhmelioglu 2011] SÖHMELIOGLU, Burak: *Experimenteller Vergleich von Techniken zur intuitiven Interaktion in Augmented Reality-Anwendungen*. 2011. – URL [http://www.cs.uni-paderborn.de/fileadmin/Informatik/FG-Szwillus/Bachelor-Studienarbeiten/BA\\_S%C3%B6hmelioglu\\_\\_Burak.pdf](http://www.cs.uni-paderborn.de/fileadmin/Informatik/FG-Szwillus/Bachelor-Studienarbeiten/BA_S%C3%B6hmelioglu__Burak.pdf)
- [Smith 2002] SMITH, Russell: *How to make new joints in ODE*. 2002. – URL <http://www.ode.org/joints.pdf>
- [Song u. a. 2008] SONG, Peng ; YU, Hang ; WINKLER, Stefan: *Vision-based 3D finger interactions for mixed reality games with physics simulation*. 2008. – URL <http://doi.acm.org/10.1145/1477862.1477871>
- [u.a. 2012] U.A., R. Paul M.: *Static Hand Gesture Recognition with 2 Kinect Sensors*. 2012. – URL <http://www.cs.uky.edu/~mihail/personal/d.php?file=pubs/kinect.pdf>
- [Wahlster 2002] WAHLSTER, Wolfgang: *Disambiguierung durch Wissensfusion: Grundprinzipien der Sprachtechnologie*. 2002. – URL <http://www.dfki.de/>

~wahlster/Publications/Disambiguierung\_durch\_Wissensfusion\_  
Grundprinzipien\_der\_Sprachtechnologie.pdf

[Watanapa 2012] WATANAPA, Orasa Patsadu; Chakarida Nukoolkit; B.: *Human Gesture Recognition Using Kinect Camera*. 2012. – URL [http://www.ee.unlv.edu/~blmorris/ecg795/docs/patsadu\\_jcsse2012\\_kinect\\_gesture.pdf](http://www.ee.unlv.edu/~blmorris/ecg795/docs/patsadu_jcsse2012_kinect_gesture.pdf)

[Ye-peng 2013] YE-PENG, Pan Jing; G.: *Human-computer Interaction using Pointing Gesture based on an Adaptive Virtual Touch Screen*. 2013. – URL [www.sersc.org/journals/IJSIP/vol6\\_no4/7.pdf](http://www.sersc.org/journals/IJSIP/vol6_no4/7.pdf)

# Abbildungsverzeichnis

2.1	: Fotoausschnitt einer Schalttafel [Quelle: Schalttafel]	12
2.2	: Foto eines Schiebereglers [Quelle: Schieberegler]	13
2.3	: (a) Abbildung eines Tastschalters [Quelle: Notaus] (b) Abbildung eines Kippschalters	14
2.4	: Abbildung eines Griffes [Quelle: Griff]	15
2.5	: (a) Foto eines Einhandhebels [Quelle: Schalttafel]	16
2.6	: (a) Rad für eine Hand [Quelle: Handkurbel] (b) Rad für zwei Händige Benutzung [Quelle: Schalttafel]	17
2.7	: Foto eines Joysticks mit diskretem Schaltverhalten	18
2.8	: (a) Drückgeste (b) Zustandsdiagramm für die Drückgeste	20
2.9	: Ziehgeste	21
2.10	: Schiebegeste	22
2.11	: Hebegeste	22
2.12	: Drehgeste	23
2.13	: (a) Joystick-Attrappe (b) Sicht des Benutzers auf das zu steuernde Computerspiel	25
3.1	: Darstellung der maximalen Ausdehnung der Finger	27
3.2	: Darstellung der maximalen Ausdehnung der Finger	29
3.3	: Darstellung der Punktwolkenanalyse: links geschlossene Hand, rechts offene Hand mit einzelnen Punkten	31
4.1	: Start- und Endgeste	40
4.2	: Selbstgebauter Handschuh mit Markern	42
4.3	: Physik-Joysticks aus Potratz (2012a)(a); Abstrakter Joystick (b)	44
4.4	: Komponentendiagramm	47
5.1	Hand offen (a), Hand gegriffen (b), Hand nicht gegriffen (c)	53
5.2	Einhand-Hebel : 3D-Ansicht (a), Physik-Ansicht (b) Zweihand-Hebel : 3D-Ansicht (c), Physik-Ansicht (d)	54

5.3	: Einhand-Rad : 3D-Ansicht (a), Physik-Ansicht (b)	
	Zweihand-Rad : 3D-Ansicht (c), Physik-Ansicht (d) . . . . .	55
5.4	: Schieberegler: 3D-Ansicht (a), Physik-Ansicht (b) . . . . .	56
5.5	: Einhand-Hebel : 3D-Ansicht (a), Physik-Ansicht (b) . . . . .	57
5.6	: Einhand-Hebel : 3D-Ansicht (l. oben), Physik-Ansicht (r. oben) . . . . .	58
5.7	: Einhand-Hebel : 3D-Ansicht (l. oben), Physik-Ansicht (r. oben) . . . . .	58
5.8	: Joystick : 3D-Ansicht (a), Physik-Ansicht (b) . . . . .	60
5.9	: Einhand-Hebel : 3D-Ansicht (l. oben), Physik-Ansicht (r. oben) . . . . .	61
5.10	: Powerwall Stand 06.02.2014 . . . . .	63
5.11	: Links: HUD mit ART-Artefakt	
	Rechts: Grafische Darstellung des Joysticks aus der Physiksimulation . . . . .	64
5.12	: Projektion des Leap Motion . . . . .	67
6.1	Verteilung der verwendeten Komponenten für den Schalttafelversuch . . . . .	70
6.2	Verteilung der verwendeten Komponenten für das virtuelle Cockpit . . . . .	71
6.3	Versuchsaufbau für das virtuelle Cockpit . . . . .	72
6.4	Sicht auf die Artefakte durch das Headup-Display . . . . .	72
6.5	Testperson vor Powerwall mit Hebel-Artefakt . . . . .	74
6.6	Testperson vor Powerwall mit Headup-Display . . . . .	76

# Glossar

<b>Disambiguierung</b>	Disambiguierung beschreibt das Auflösen von Mehrdeutigkeiten unter Verwendung von Selektionsrestriktionen und weiteren „globalen“ Informationen. Siehe auch [Wahlster (2002)]
<b>Dependency Injection</b>	Beschreibt ein Konzept der objektorientierten Programmierung, in dem einem Objekt zum Beispiel bei der Initialisierung ein anderes Objekt übergeben wird. Dieses Übergabeobjekt kann für das Verhalten des Objekts bestimmt sein.
<b>Framework</b>	Ein Framework, im Deutschen als Rahmenwerk bezeichnet, bestimmt die Architektur der Anwendung. Es stellt dem Entwickler eine Reihe von Grundfunktionalitäten in Form von Klassen und Methoden zur Verfügung, sodass der Entwickler sich auf die spezifischen Details seiner Anwendung konzentrieren kann. Siehe dazu auch: [Erich Gamma (2001)].
<b>Fullscreen</b>	Fullscreen, deutsch: Vollbild, beschreibt den Fall wenn eine Anwendung den Bildschirm komplett ausfüllt und auch der Fensterrahmen von Betriebssystem wegfällt.
<b>Gyrometer</b>	Ein Gyrometer ist ein Sensor der Drehbewegungen registriert. Er kann die Rotationsgeschwindigkeit des sich drehenden Körpers messen.
<b>Immersion</b>	Immersion ist ein Bewusstseinszustand. Um so höher der Grad der Immersion ist, desto mehr fühlt sich der Benutzer als Teil einer simulierten Umgebung.

**UDP**

User Datagram Protocol - UDP ist ein verbindungsloses Netzwerkprotokoll. Es können damit Daten über ein Netzwerk versendet werden, ohne das dem Sender der oder die Empfänger nicht bekannt sein müssen.

# **Anhang A**

## **Fragebögen**

Die hier aufgeführten Fragebögen kamen bei dem Versuchsaufbau der Schalttafel und dem Versuch des virtuellen Cockpits zum Einsatz. Bei den personbezogenen Fragen in den Fragebögen wurde sich an dem Fragebogen von Lorenz Barnkow aus der Arbeit Barnkow (2012) orientiert.

# Fragebogen zur Schalltafel Teil 1

Anleitung:

Bitte die folgenden Fragen nur genau eine Antwort ankreuzen. Möchten Sie sich zu einer Frage nicht äußern, dann kreuzen Sie bitte „Enthaltung“ an. Füllen Sie diese Fragen bitte **vor** dem Versuchsdurchlauf aus.

Ihre Antworten sind vertraulich, tauschen Sie sich nicht mit anderen Teilnehmern darüber aus. Bei Fragen stehe ich Ihnen gerne zur Verfügung.

## 1. Fragen zur Person

(1.1) Bitte geben Sie Ihr Alter an:

18-20	21-24	25-29	30-35	36-45	45 +	Enthaltung
<input type="radio"/>						

(1.2) Bitte geben Sie Ihr Geschlecht an:

Männlich	Weiblich		Enthaltung
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(1.3) Sind Sie zur Zeit (WiSe 2013/2014) eingeschriebener Student einer Hochschule?

Ja	Nein	Enthaltung
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Falls Ja:**

a) Welchen Abschluss streben Sie aktuell an? (z.B. B. Sc., M. Sc., Ph.D., ...)

b) Welcher Studienrichtung gehören Sie an? (z. B. Informatik, ...)

c) Im wievielten Studiensemester befinden Sie sich (Gesamtzahl)?

1 – 2	3 -4	5 – 6	7- 9	10 – 12	12 +	Enthaltung
<input type="radio"/>						

**Falls Nein:**

d) Welchen Beruf gehen Sie derzeit nach?

## 2. Computer Kenntnisse

(2.1) Wie lange verwenden Sie bereits Computer? (in Jahren)

Keine Vorkenntnisse	Bis zu einem Jahr	1 – 5	5 +	Enthaltung
<input type="radio"/>				

(2.2) Wie lange verwenden Sie einen Computer pro Woche? (in Stunden)

nie	0 -5	5 – 15	15- 25	25 – 40	40 +	Enthaltung
<input type="radio"/>						

(2.3) Haben Sie allgemeine Kenntnisse in Gestensteuerung? (Smartphone,...)

keine	wenig	mittelmäßig	gute Kenntnisse	Enthaltung
<input type="radio"/>				

(2.4) Haben Sie bereits Kenntnisse über 3D-Gestensteuerung (z.B. Smart TV, X-Box, ...)

keine	wenig	mittelmäßig	gute Kenntnisse	Enthaltung
<input type="radio"/>				

(2.5) Haben Sie bereits eine 3D-Gestensteuerung Verwendet?

nie	selten	gelegentlich	häufig	Enthaltung
<input type="radio"/>				



(3.3.3) Welcher Sensor gefiel Ihnen Besser?

keiner <input type="radio"/>	ART-Tracker <input type="radio"/>	Kinect <input type="radio"/>	Beide gleich <input type="radio"/>	Enthaltung <input type="radio"/>
---------------------------------	--------------------------------------	---------------------------------	---------------------------------------	-------------------------------------

(3.3.4) Warum?









**(5 Allgemeine Fragen zur 3D Gestensteuerung)**

(5.1) Oft bestehen Zweifel des Benutzers, ob Seine Interaktionen/Gesten vom Computer richtig verstanden wurden. In wie weit, Ihrer Meinung nach, konnte mit diesem System da gegen gewirkt werden? (Sensorfehler außen vor gelassen)

(5.2) Können Sie sich eine alltägliche Benutzung von einer 3D-Gestensteuerung vorstellen?

# Bewertungsbogen für einen Probanden im Testdurchlauf

## **Hebel**

(1.1) Hilfestellung beim Erkennen des Hebels

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(1.2) Hilfestellung bei Funktionsweise des Hebels

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **Zweihand-Hebel**

(2.1) Hilfestellung beim Erkennen des Zweihand-Hebels

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(2.2) Hilfestellung bei Funktionsweise des Zweihand-Hebels

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **Rad**

(3.1) Hilfestellung beim Erkennen des Rads

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(3.2) Hilfestellung bei Funktionsweise des Rads

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **Zweihand-Rad**

(4.1) Hilfestellung beim Erkennen des Zweihand-Rads

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(4.2) Hilfestellung bei Funktionsweise des Zweihand-Rads

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **Senkrechten Schieber**

(5.1) Hilfestellung beim Erkennen des senkrechten Schieber

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(5.2) Hilfestellung bei Funktionsweise des senkrechten Schieber

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **Waagerechten Schieber**

(6.1) Hilfestellung beim Erkennen des waagerechten Schieber

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(6.2) Hilfestellung bei Funktionsweise des waagerechten Schieber

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **Knopf**

(7.1) Hilfestellung beim Erkennen des Knopf

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(7.2) Hilfestellung bei Funktionsweise des Knopf

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **Kippschalter**

(8.1) Hilfestellung beim Erkennen des Kippschalter

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(8.2) Hilfestellung bei Funktionsweise des Kippschalter

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **Griff**

(9.1) Hilfestellung beim Erkennen des Griffs

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(9.2) Hilfestellung bei Funktionsweise des Griffs

nein	Ein wenig	viel
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

# Fragebogen zum virtuellen Cockpit Teil 1

Anleitung:

Bitte die folgenden Fragen nur genau eine Antwort ankreuzen. Möchten Sie sich zu einer Frage nicht äußern, dann kreuzen Sie bitte „Enthaltung“ an. Füllen Sie diese Fragen bitte **vor** dem Versuchsdurchlauf aus.

Ihre Antworten sind vertraulich, tauschen Sie sich nicht mit anderen Teilnehmern darüber aus. Bei Fragen stehe ich Ihnen gerne zur Verfügung.

## 1. Fragen zur Person

(1.1) Bitte geben Sie Ihr Alter an:

18-20	21-24	25-29	30-35	36-45	45 +	Enthaltung
<input type="radio"/>						

(1.2) Bitte geben Sie Ihr Geschlecht an:

Männlich	Weiblich		Enthaltung
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(1.3) Sind Sie zur Zeit (WiSe 2013/2014) eingeschriebener Student einer Hochschule?

Ja	Nein	Enthaltung
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Falls Ja:**

a) Welchen Abschluss streben Sie aktuell an? (z.B. B. Sc., M. Sc., Ph.D., ...)

b) Welcher Studienrichtung gehören Sie an? (z. B. Informatik, ...)

c) Im wievielten Studiensemester befinden Sie sich (Gesamtzahl)?

1 – 2	3 -4	5 – 6	7- 9	10 – 12	12 +	Enthaltung
<input type="radio"/>						

**Falls Nein:**

Welchen Beruf gehen Sie derzeit nach?

(1.4) Tragen Sie eine Brille?

ja	nein	Enthaltung
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 2. Computer Kenntnisse

(2.1) Wie lange verwenden Sie bereits Computer? (in Jahren)

Keine Vorkenntnisse	Bis zu einem Jahr	1 – 5	5 +	Enthaltung
<input type="radio"/>				

(2.2) Wie lange verwenden Sie einen Computer pro Woche? (in Stunden)

nie	0 -5	5 – 15	15- 25	25 – 40	40 +	Enthaltung
<input type="radio"/>						

(2.3) Haben Sie allgemeine Kenntnisse in Gestensteuerung? (Smartphone)

keine	wenig	mittelmäßig	gute Kenntnisse	Enthaltung
<input type="radio"/>				

(2.4) Haben Sie bereits Kenntnisse über 3D-Gestensteuerung?

keine	wenig	mittelmäßig	gute Kenntnisse	Enthaltung
<input type="radio"/>				

(2.5) Haben Sie bereits eine 3D-Gestensteuerung Verwendet? (z.B. Smart TV, X-Box, ...)

nie	selten	gelegentlich	häufig	Enthaltung
<input type="radio"/>				

(2.6) Haben Sie bereits am Versuch zum Schaltbord teilgenommen?

ja	nein	Enthaltung
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(2.7) Haben Sie Erfahrungen mit (Spiel-) Flugsimulationen ?

keine	wenig	mittelmäßig	gute Kenntnisse	Enthaltung
<input type="radio"/>				

## Fragebogen zum virtuellen Cockpit Teil 2

Anleitung:

Bitte die folgenden Fragen nur genau eine Antwort ankreuzen. Möchten Sie sich zu einer Frage nicht äußern, dann kreuzen Sie bitte „Enthaltung“ an. Füllen Sie diese Fragen bitte **nach** dem Versuchsdurchlauf aus.

Ihre Antworten sind vertraulich, tauschen Sie sich nicht mit anderen Teilnehmern darüber aus. Bei Fragen stehe ich Ihnen gerne zur Verfügung.

### 3. Handsteuerung und Greifgeste

(3.1) Wie intuitiv war für Sie die Steuerung der Hände

gar nicht	schwer	mittelmäßig	gut	sehr gut	Enthaltung
<input type="radio"/>					

(3.2) Wie Bewerten Sie deren Funktionsweise

sehr schlecht	schlecht	mittelmäßig	gut	sehr gut	Enthaltung
<input type="radio"/>					

(3.3) Wie Bewerten Sie die Sicht durch das Headup-Display?

gar nicht	schwer	mittelmäßig	gut	sehr gut	Enthaltung
<input type="radio"/>					

(3.4) Wie bewerten Sie die Steuerung der Flugsimulation?

sehr schlecht	schlecht	mittelmäßig	gut	sehr gut	Enthaltung
<input type="radio"/>					

(3.5) Wie war die Steuerung, wenn man sich auf die Flugsimulation konzentrierte?

sehr schwer	schwer	mittelmäßig	gut	sehr gut	Enthaltung
<input type="radio"/>					

(3.6) Würden weitere Feedback Kanäle eine Nutzung für Sie vereinfachen?

akustisches	haptisches	beides	keins	Enthaltung
<input type="radio"/>				

#### **4. Abschließende Fragen**

(4.1) Können Sie sich eine regelmäßige Nutzung eines solchen System mit Artefakt-Darstellung über Headup-Display vorstellen? (mit einem handlicheren Headup-Display versteht sich).

(4.2) Wenn Ja? Welche Einsatzbereiche könnten Sie sich vorstellen?

# **Anhang B**

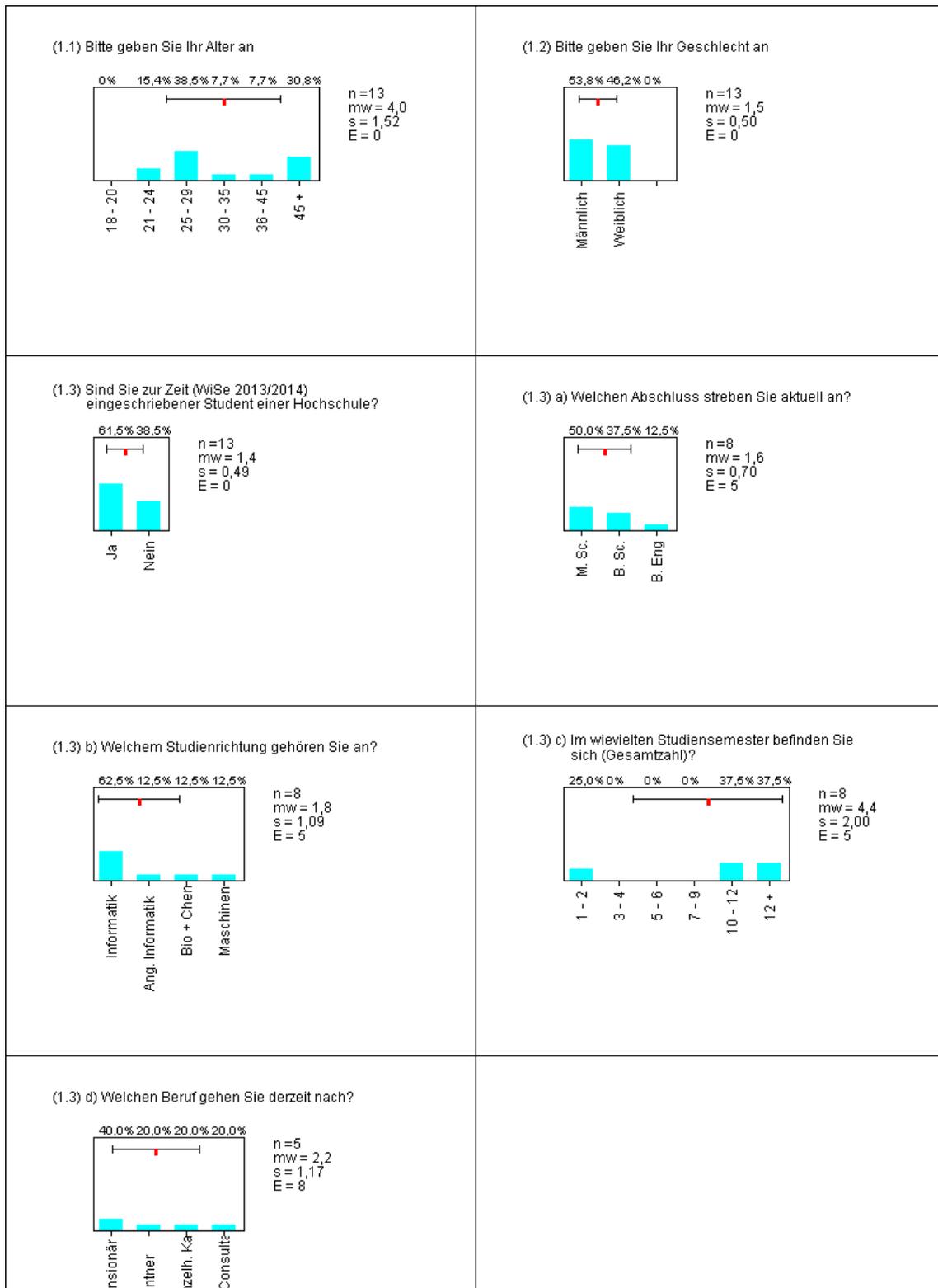
## **Ergebnisse**

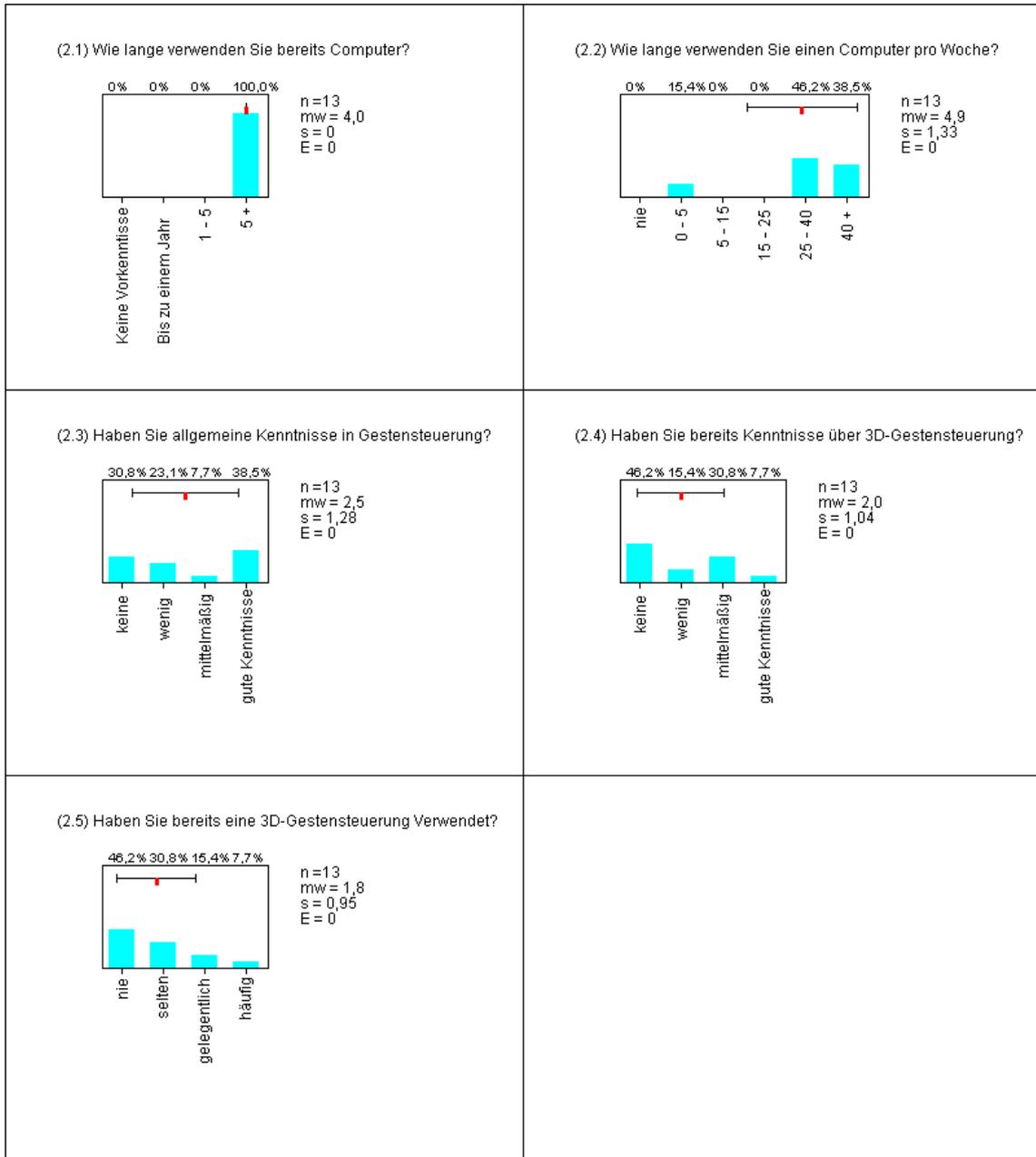
Die Ergebnisse der Fragebögen wurden händisch in den Computer übertragen. Die Diagramme wurden mit einem selbst entwickelten Programm in Java erstellt.

Abkürzung	Bedeutung
n	Anzahl der Testpersonen
mw	Mittelwert
s	Standardabweichung
E	Anzahl der Enthaltungen

Tabelle B.1: Legende für die Diagramme der Fragebögen

## Diagramme Schalttafel Fragebogen





<p>(3.1.1) Wie intuitiv war für Sie die Steuerung der Hände</p> <p>7,7% 7,7% 7,7% 46,2% 30,8%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n=13 mw= 3,8 s= 1,17 E= 0</p>	<p>(3.1.2) Wie Bewerten Sie deren Funktionsweise</p> <p>0% 0% 23,1% 69,2% 7,7%</p> <p>sehr schlecht schlecht mittelmäßig gut sehr gut</p> <p>n=13 mw= 3,8 s= 0,53 E= 0</p>
<p>(3.2.1) Wie intuitiv war für Sie die Greif / Loslassgeste</p> <p>0% 0% 15,4% 30,8% 53,8%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n=13 mw= 4,4 s= 0,74 E= 0</p>	<p>(3.2.2) Wie Bewerten Sie deren Funktionsweise</p> <p>0% 7,7% 30,8% 46,2% 15,4%</p> <p>sehr schlecht schlecht mittelmäßig gut sehr gut</p> <p>n=13 mw= 3,7 s= 0,82 E= 0</p>
<p>(3.3.1) Umgang mit dem ART-Tracking Sensor</p> <p>7,7% 30,8% 23,1% 30,8% 7,7%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n=13 mw= 3,0 s= 1,11 E= 0</p>	<p>(3.3.2) Umgang mit dem Kinect Sensor</p> <p>0% 7,7% 23,1% 38,5% 30,8%</p> <p>sehr schlecht schlecht mittelmäßig gut sehr gut</p> <p>n=13 mw= 3,9 s= 0,92 E= 0</p>
<p>(3.3.3) Welcher Sensor gefiel Ihnen Besser?</p> <p>0% 15,4% 78,9% 7,7%</p> <p>keine ART-Tracker Kinect Beide gleich</p> <p>n=13 mw= 2,9 s= 0,47 E= 0</p>	

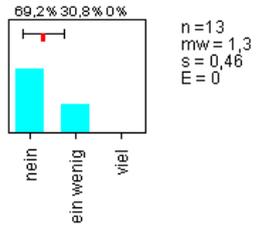
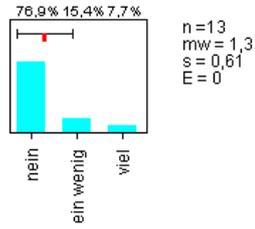
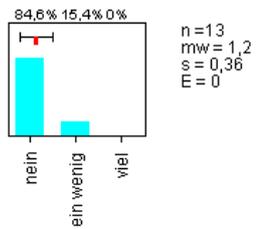
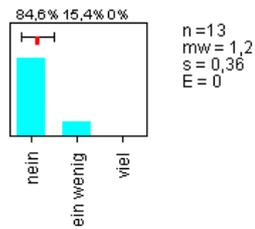
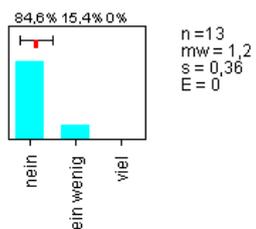
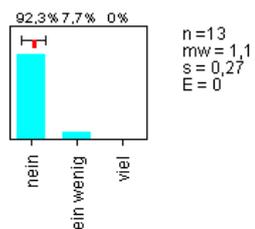
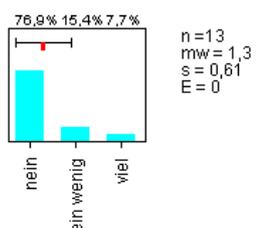
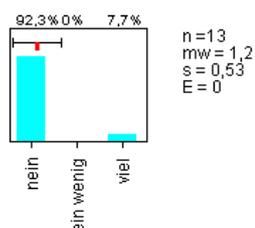
<p>(4.1.1) Haben Sie den Hebel erkannt?</p> <p>0% 0% 0% 30,8% 61,5%</p> <p>nein schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,7 s = 0,47 E = 0</p>	<p>(4.1.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>0% 0% 0% 30,8% 69,2%</p> <p>gar nicht schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,7 s = 0,46 E = 0</p>
<p>(4.1.3) Handhabung des Hebels</p> <p>0% 0% 7,7% 38,5% 53,8%</p> <p>unbenutzbar schlecht mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,5 s = 0,63 E = 0</p>	<p>(4.2.1) Haben Sie den Zweihand-Hebel erkannt?</p> <p>0% 7,7% 0% 30,8% 53,8%</p> <p>nein schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,4 s = 0,86 E = 0</p>
<p>(4.2.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>7,7% 7,7% 0% 30,8% 53,8%</p> <p>gar nicht schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,2 s = 1,23 E = 0</p>	<p>(4.2.3) Handhabung des Zweihand-Hebels</p> <p>7,7% 0% 30,8% 23,1% 38,5%</p> <p>unbenutzbar schlecht mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 3,8 s = 1,17 E = 0</p>
<p>(4.3.1) Haben Sie das Rad erkannt?</p> <p>0% 0% 0% 23,1% 69,2%</p> <p>nein schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,8 s = 0,43 E = 0</p>	<p>(4.3.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>0% 0% 7,7% 46,2% 46,2%</p> <p>gar nicht schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,4 s = 0,62 E = 0</p>

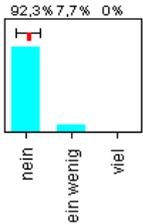
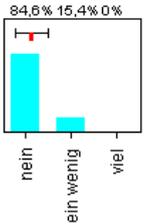
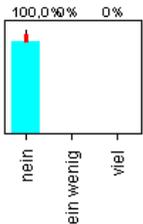
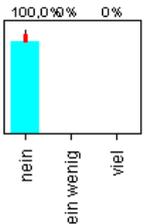
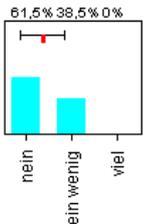
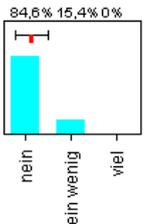
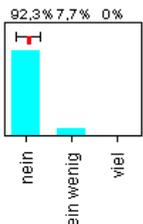
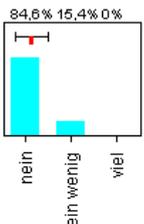
<p>(4.3.3) Handhabung des Rad</p> <p>0% 7,7% 7,7% 38,5% 46,2%</p> <p>unbenutzbar schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,2 s = 0,89 E = 0</p>	<p>(4.4.1) Haben Sie das Zweihand-Rad erkannt?</p> <p>0% 0% 7,7% 23,1% 61,5%</p> <p>nein schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,6 s = 0,64 E = 0</p>
<p>(4.4.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>0% 15,4% 7,7% 46,2% 30,8%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 3,9 s = 10,00 E = 0</p>	<p>(4.4.3) Handhabung des Zweihand-Rads</p> <p>0% 30,8% 30,8% 15,4% 23,1%</p> <p>unbenutzbar schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 3,3 s = 1,14 E = 0</p>
<p>(4.5.1) Haben Sie den senkrechten Schieber erkannt?</p> <p>0% 0% 0% 23,1% 69,2%</p> <p>nein schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,8 s = 0,43 E = 0</p>	<p>(4.5.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>0% 0% 0% 23,1% 76,9%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,8 s = 0,42 E = 0</p>
<p>(4.5.3) Handhabung des senkrechten Schieber</p> <p>0% 0% 0% 46,2% 53,8%</p> <p>unbenutzbar schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,5 s = 0,50 E = 0</p>	<p>(4.6.1) Haben Sie den waagerechten Schieber erkannt?</p> <p>0% 0% 0% 15,4% 76,9%</p> <p>nein schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,8 s = 0,37 E = 0</p>

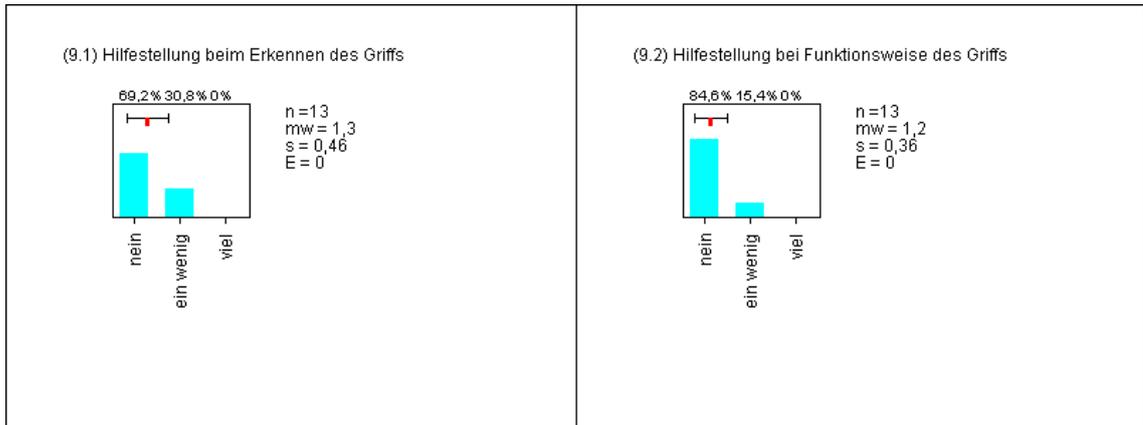
<p>(4.6.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>0% 0% 0% 23,1% 76,9%</p> <p>gar nicht schwer mittelmäÙig gut</p> <p>n = 13 mw = 4,8 s = 0,42 E = 0</p>	<p>(4.6.3) Handhabung des waagerechten Schieber</p> <p>0% 0% 0% 38,5% 61,5%</p> <p>unbenutzbar schlecht mittelmäÙig gut</p> <p>n = 13 mw = 4,6 s = 0,49 E = 0</p>
<p>(4.7.1) Haben Sie den Knopf erkannt?</p> <p>0% 0% 15,4% 69,2% 7,7%</p> <p>nein schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 3,9 s = 0,49 E = 0</p>	<p>(4.7.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>0% 0% 30,8% 46,2% 23,1%</p> <p>gar nicht schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 3,9 s = 0,73 E = 0</p>
<p>(4.7.3) Handhabung des Knopf</p> <p>0% 0% 23,1% 46,2% 30,8%</p> <p>unbenutzbar schlecht mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,1 s = 0,73 E = 0</p>	<p>(4.8.1) Haben Sie den Kippschalter erkannt?</p> <p>0% 0% 15,4% 23,1% 15,4%</p> <p>nein schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,0 s = 0,76 E = 0</p>
<p>(4.8.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>0% 0% 7,7% 23,1% 23,1%</p> <p>gar nicht schwer mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 4,3 s = 0,70 E = 0</p>	<p>(4.8.3) Handhabung des Kippschalter</p> <p>7,7% 7,7% 15,4% 0% 23,1%</p> <p>unbenutzbar schlecht mittelmäÙig gut sehr gut</p> <p>n = 13 mw = 3,4 s = 1,50 E = 0</p>

<p>(4.9.1) Haben Sie den Griff erkannt?</p> <p>0% 0% 15,4% 38,5% 38,5%</p> <p>nein schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,3 s = 0,72 E = 0</p>	<p>(4.9.2) Wie intuitiv war Ihnen die Funktionsweise?</p> <p>0% 0% 30,8% 30,8% 38,5%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,1 s = 0,83 E = 0</p>
<p>(4.9.3) Handhabung des Griff</p> <p>0% 15,4% 7,7% 46,2% 30,8%</p> <p>unbenutzbar schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 3,9 s = 10,00 E = 0</p>	<p>(4.10.1) Wie übersichtlich war das Schaltpult</p> <p>0% 0% 0% 53,8% 46,2%</p> <p>nein schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,5 s = 0,50 E = 0</p>
<p>(4.10.2) Gestellten Aufgaben gut verständlich?</p> <p>0% 0% 0% 38,5% 61,5%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,6 s = 0,49 E = 0</p>	<p>(4.10.3) Handhabung des Schaltpults</p> <p>0% 0% 15,4% 46,2% 38,5%</p> <p>unbenutzbar schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,2 s = 0,70 E = 0</p>
<p>(4.11) Wie hilfreich war die Visualisierung auf der Powerwall?</p> <p>0% 0% 0% 30,8% 69,2%</p> <p>gar nicht schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,7 s = 0,46 E = 0</p>	<p>(4.12) Wie hilfreich war das akustische Feedback?</p> <p>0% 0% 0% 38,5% 61,5%</p> <p>gar nicht schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,6 s = 0,49 E = 0</p>

## Diagramme Schalttafel Beobachtungsbogen

<p>(1.1) Hilfestellung beim Erkennen des Hebels</p>  <p>69,2% 30,8% 0%</p> <p>nein ein wenig viel</p> <p>n = 13 mw = 1,3 s = 0,46 E = 0</p>	<p>(1.2) Hilfestellung bei Funktionsweise des Hebels</p>  <p>76,9% 15,4% 7,7%</p> <p>nein ein wenig viel</p> <p>n = 13 mw = 1,3 s = 0,61 E = 0</p>
<p>(2.1) Hilfestellung beim Erkennen des Zweihand-Hebels</p>  <p>84,6% 15,4% 0%</p> <p>nein ein wenig viel</p> <p>n = 13 mw = 1,2 s = 0,36 E = 0</p>	<p>(2.2) Hilfestellung bei Funktionsweise des Zweihand-Hebels</p>  <p>84,6% 15,4% 0%</p> <p>nein ein wenig viel</p> <p>n = 13 mw = 1,2 s = 0,36 E = 0</p>
<p>(3.1) Hilfestellung beim Erkennen des Rads</p>  <p>84,6% 15,4% 0%</p> <p>nein ein wenig viel</p> <p>n = 13 mw = 1,2 s = 0,36 E = 0</p>	<p>(3.2) Hilfestellung bei Funktionsweise des Rads</p>  <p>92,3% 7,7% 0%</p> <p>nein ein wenig viel</p> <p>n = 13 mw = 1,1 s = 0,27 E = 0</p>
<p>(4.1) Hilfestellung beim Erkennen des Zweihand-Rads</p>  <p>76,9% 15,4% 7,7%</p> <p>nein ein wenig viel</p> <p>n = 13 mw = 1,3 s = 0,61 E = 0</p>	<p>(4.2) Hilfestellung bei Funktionsweise des Zweihand-Rads</p>  <p>92,3% 0% 7,7%</p> <p>nein ein wenig viel</p> <p>n = 13 mw = 1,2 s = 0,53 E = 0</p>

<p>(5.1) Hilfestellung beim Erkennen des senkrechten Schieber</p>  <p>92,3% 7,7% 0%</p> <p>n = 13 mw = 1,1 s = 0,27 E = 0</p> <p>nein ein wenig viel</p>	<p>(5.2) Hilfestellung bei Funktionsweise des senkrechten Schieber</p>  <p>84,6% 15,4% 0%</p> <p>n = 13 mw = 1,2 s = 0,36 E = 0</p> <p>nein ein wenig viel</p>
<p>(6.1) Hilfestellung beim Erkennen des waagerechten Schieber</p>  <p>100,0% 0% 0%</p> <p>n = 13 mw = 1,0 s = 0 E = 0</p> <p>nein ein wenig viel</p>	<p>(6.2) Hilfestellung bei Funktionsweise des waagerechten Schieber</p>  <p>100,0% 0% 0%</p> <p>n = 13 mw = 1,0 s = 0 E = 0</p> <p>nein ein wenig viel</p>
<p>(7.1) Hilfestellung beim Erkennen des Knopf</p>  <p>61,5% 38,5% 0%</p> <p>n = 13 mw = 1,4 s = 0,49 E = 0</p> <p>nein ein wenig viel</p>	<p>(7.2) Hilfestellung bei Funktionsweise des Knopf</p>  <p>84,6% 15,4% 0%</p> <p>n = 13 mw = 1,2 s = 0,36 E = 0</p> <p>nein ein wenig viel</p>
<p>(8.1) Hilfestellung beim Erkennen des Kippschalter</p>  <p>92,3% 7,7% 0%</p> <p>n = 13 mw = 1,1 s = 0,27 E = 0</p> <p>nein ein wenig viel</p>	<p>(8.2) Hilfestellung bei Funktionsweise des Kippschalter</p>  <p>84,6% 15,4% 0%</p> <p>n = 13 mw = 1,2 s = 0,36 E = 0</p> <p>nein ein wenig viel</p>



### Rohdaten Schalttafel Fragebogen

Nr.	1.1	1.2	1.3	a	b	c	d	2.1	2.2	2.3	2.4	2.5
1	3	2	1	B. Sc.	Bio + Chemie	1		4	5	3	2	2
2	2	2	1	B. Sc.	Ang. Informatik	1		4	5	2	1	1
3	3	1	1	B. Eng	Maschinenbau	5		4	5	2	1	1
4	6	1	2				Pensionär	4	2	1	1	1
5	6	1	2				Pensionär	4	2	1	1	1
6	6	2	2				Einzelh. Kauffrau	4	5	1	1	1
7	6	1	2				Rentner	4	5	1	1	1
8	3	1	1	B. Sc.	Informatik	5		4	6	4	4	4
9	3	2	1	M. Sc.	Informatik	6		4	6	4	2	2
10	4	1	2				IT Consultant	4	6	4	3	3
11	5	1	1	M. Sc.	Informatik	6		4	6	2	3	2
12	3	2	1	M. Sc.	Informatik	6		4	5	4	3	3
13	2	2	1	M. Sc.	Informatik	5		4	6	4	3	2

Tabelle B.2: Ergebnisse Fragebogen Schalttafel Teil1

Nr.	3.1.1	3.1.2	3.2.1	3.2.2	3.3.1	3.3.2	3.3.3	3.3.4
1	4	3	5	3	2	4	3	Die Gesten wurden schneller und zuverlässiger erkannt.
2	5	4	5	3	2	4	3	Hat besser getrackt
3	5	4	4	4	4	5	3	ruckelfrei, geschmeidiger
4	4	4	5	4	5	4	2	Ich konnte besser greifen, meine Hände gehorchten besser meiner Gesten
5	1	4	4	4	4	5	3	
6	3	4	3	3	3	4	3	Ich fand die Übertragung direkter
7	5	5	5	5	4	5	3	Mit ART-Tracker muß bewusst auf die Kamera-Einfallswinkel geachtet werden. Kinect hat größeren Erfassungsbereich.
8	2	3	5	4	1	3	3	
9	4	4	4	3	3	4	3	Die Handschuhe waren etwas groß und dadurch sind sie ständig abgerutscht. Man konnte die Hände nicht richtig schließen mit den Handschuhen
10	5	4	5	4	4	3	2	Bessere Tiefensteuerung, Kaum falsche Greifgesten, Verlor nicht so oft die Hände, Keine Ausrichtung vor einem speziellen Gerät nötig
11	4	3	3	2	3	3	3	Ohne Assecoirs (Handschuhe) ist es schöner & bequemer
12	4	4	5	5	2	5	3	genauer
13	4	4	4	4	2	2	4	keine deutlichen Unterschiede zw. Den Systemen. Allerdings ist die Handhabung mit der Kinect angenehmer (kein extra Zubehör)

Tabelle B.3: Ergebnisse Fragebogen Schalttafel Teil2 (1 von 5)

Nr.	4.1.1	4.1.2	4.1.3	4.2.1	4.2.2	4.2.3	4.3.1	4.3.2	4.3.3	4.4.1	4.4.2	4.4.3
1	5	5	4	4	4	4	5	5	4	3	4	4
2	5	5	4	5	5	3	5	5	4	5	5	3
3	5	5	5	5	4	4	5	4	5	5	5	5
4	4	5	5	4	5	3	5	3	3	5	4	3
5	5	5	5	5	5	5	5	5	5	5	5	5
6	4	4	4	2	1	1	4	4	4	4	3	2
7	5	5	5	5	5	5	5	5	5	5	5	5
8	6	4	4	6	2	3	6	4	4	6	2	2
9	4	4	4	4	4	4	4	4	4	4	4	4
10	4	5	5	5	5	5	4	5	5	4	2	2
11	5	4	3	5	4	3	5	4	2	5	4	2
12	5	5	5	4	5	5	5	4	5	5	4	3
13	5	5	5	5	5	5	5	5	5	5	4	3

Tabelle B.4: Ergebnisse Fragebogen Schalttafel Teil2 (2 von 5)

Nr.	4.5.1	4.5.2	4.5.3	4.6.1	4.6.2	4.6.3	4.7.1	4.7.2	4.7.3	4.8.1	4.8.2	4.8.3
1	5	5	4	5	5	4	4	4	4	5	5	1
2	4	5	4	5	5	5	4	5	5	6	6	6
3	5	5	5	5	5	5	3	4	4	4	6	6
4	5	5	5	5	5	5	4	4	4	6	6	6
5	5	5	5	5	5	5	4	5	5	6	6	6
6	4	4	4	4	4	4	4	4	4	6	6	6
7	5	5	5	5	5	5	5	5	5	6	6	6
8	6	4	4	6	4	4	6	3	3	6	4	3
9	4	4	4	4	4	4	4	4	4	3	3	3
10	5	5	5	5	5	5	4	3	4	4	5	5
11	5	5	4	5	5	4	4	3	3	4	4	2
12	5	5	5	5	5	5	3	4	5	3	4	5
13	5	5	5	5	5	5	4	3	3	5	5	5

Tabelle B.5: Ergebnisse Fragebogen Schalttafel Teil2 (3 von 5)

Nr.	4.9.1	4.9.2	4.9.3	4.10.1	4.10.2	4.10.3	4.11	4.12
1	5	5	4	5	5	4	5	5
2	4	3	4	5	5	4	5	5
3	5	5	5	5	5	5	5	4
4	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5
6	3	4	3	4	4	4	4	4
7	5	5	5	5	5	5	5	5
8	6	4	2	4	4	3	4	5
9	4	4	4	4	4	4	4	4
10	4	4	4	4	4	4	5	5
11	4	3	4	4	5	3	4	4
12	3	3	4	4	5	5	5	5
13	4	3	2	4	4	4	5	4

Tabelle B.6: Ergebnisse Fragebogen Schalttafel Teil2 (4 von 5)

Nr.	5.1	5.2
1	Anzeige über z.B. Füllstand, "Tacho", Lampen	Ja
2	Rote Hände und Stuff. Ein Errorgeräusch gab es auch oder?	Ja, nur keine Ahnung wo. Vielleicht mit dem heißen Backofen?!
3	Sehr hilfreich als Rückmeldung. Altbekannte Werkzeuge machen den Einstieg leicht.	Ja, z.B. Im auto für den Bordcomputer etc.
4	Gut dagegengewirkt, ich konnte jederzeit sehen was ich gerade mache	Sehr gut, z.B. Fernseher, Fahrkartenautomat, Auto
5	Aufbau und Wirkungsweise des Beispiele leicht verständlich	für den Betrieb und die Steuerungsämtlicher elektronischer Geräte im Haus; desweitern bei der Überwachung und Steuerung von Gebäuden
6	weiß nicht	Ist für mich noch gewöhnungsbedürftig
7	Hat Zweifel entgegengewirkt, da jederzeit erkennbar war, was das System "verstanden" hatte.	Ja, wenn noch etwas "Feinabstimmung" die Präzision erhöht.
8		
9	Greifen konnte ausprobiert werden und Handbewegungen	Ja, für den häuslichen Gebrauch
10	Gute Visualisierung (farben, Symbole); Gute Unterstützung durch den Sound; Es wurde mir bewusst wann ich etwas richtig gegriffen habe (Farbe, Sound)	Ja. Entertainment (Consumerbereich); Bewegung von Gefahrstoffen, die nicht direkt in Kontakt mit dem Benutzer kommen sollen
11	virtuelle Artefakte zu bedienen ist sehr hilfreich – man sieht was man tut. Nur die Einordnung der eigenen Person zum virtuellen Umfeld (telepräsenz) ist sehr gewöhnungsbedürftig.	eher nicht. Die Kombination Sehen, Riechen, Anfassen ist unmittelbarer
12	visuelles und akustisches Feedback waren hilfreich bei der Erkennung der Auswirkung einer Aktion. Somit wurde gut erkennbar, ob die ausgeführte Aktion erfolgreich war.	Ja :) z.B. Zur Steuerung von einfachen Geräten mit an / aus Funktion (z.B. Licht, Radio, etc.)
13	Da es klar definierten Gesten sind, scheint mir eine falsche Erkennung von vornherein eigentlich ausgeschlossen	Wenn es das in kleinerem (billigeren) Maße für den Hausgebrauch gibt, warum nicht?

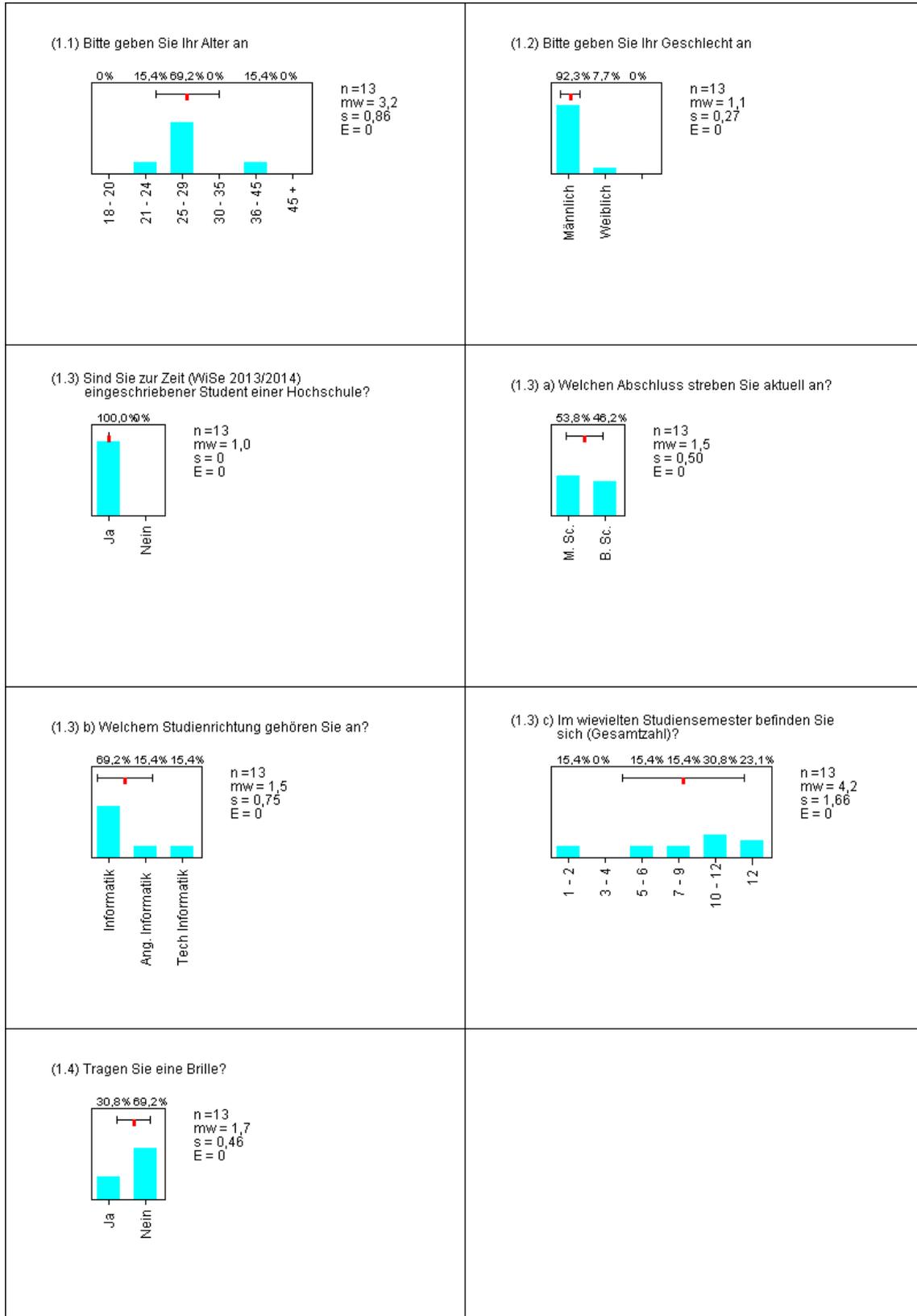
Tabelle B.7: Ergebnisse Fragebogen Schalttafel Teil2 (5 von 5)

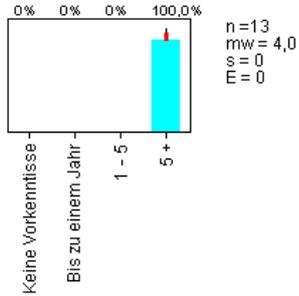
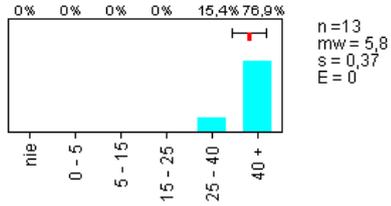
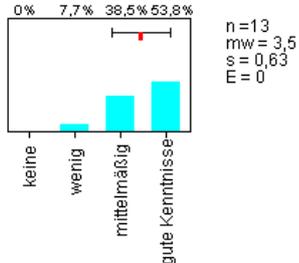
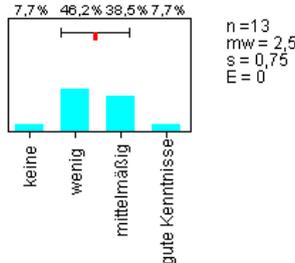
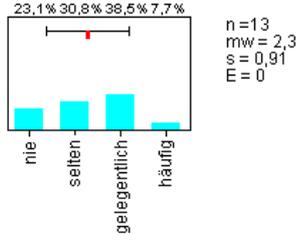
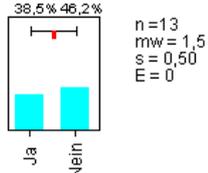
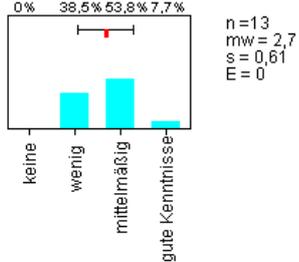
## Rohdaten Schalttafel Beobachtungsbogen

Nr.	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2	6.1	6.2	7.1	7.2	8.1	8.2	9.1	9.2
1	1	1	2	2	1	1	2	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1
4	2	3	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1
5	2	2	2	2	2	1	1	1	1	2	1	1	2	2	1	1	2	1
6	2	2	1	1	2	2	3	3	2	2	1	1	2	1	1	2	1	1
7	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	2	2
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabelle B.8: Ergebnisse des Notizbogen beim Schalttafelversuch

## Diagramme des virtuellen Cockpitsversuchs



<p>(2.1) Wie lange verwenden Sie bereits Computer?</p>  <p>0% 0% 0% 100,0%</p> <p>Keine Vorkenntnisse Bis zu einem Jahr 1 - 5 5 +</p> <p>n = 13 mw = 4,0 s = 0 E = 0</p>	<p>(2.2) Wie lange verwenden Sie einen Computer pro Woche?</p>  <p>0% 0% 0% 0% 15,4% 76,9%</p> <p>nie 0 - 5 5 - 15 15 - 25 25 - 40 40 +</p> <p>n = 13 mw = 5,8 s = 0,37 E = 0</p>
<p>(2.3) Haben Sie allgemeine Kenntnisse in Gestensteuerung?</p>  <p>0% 7,7% 38,5% 53,8%</p> <p>keine wenig mittelmaßig gute Kenntnisse</p> <p>n = 13 mw = 3,5 s = 0,63 E = 0</p>	<p>(2.4) Haben Sie bereits Kenntnisse über 3D-Gestensteuerung?</p>  <p>7,7% 46,2% 38,5% 7,7%</p> <p>keine wenig mittelmaßig gute Kenntnisse</p> <p>n = 13 mw = 2,5 s = 0,75 E = 0</p>
<p>(2.5) Haben Sie bereits eine 3D-Gestensteuerung Verwendet?</p>  <p>23,1% 30,8% 38,5% 7,7%</p> <p>nie selten gelegentlich häufig</p> <p>n = 13 mw = 2,3 s = 0,91 E = 0</p>	<p>(2.6) Haben Sie bereits am Versuch zum Schaltbord teilgenommen?</p>  <p>38,5% 46,2%</p> <p>Ja Nein</p> <p>n = 13 mw = 1,5 s = 0,50 E = 0</p>
<p>(2.7) Haben Sie Erfahrungen mit (Spiel-) Flugsimulationen?</p>  <p>0% 38,5% 53,8% 7,7%</p> <p>keine wenig mittelmaßig gute Kenntnisse</p> <p>n = 13 mw = 2,7 s = 0,61 E = 0</p>	

<p>(3.1) Wie intuitiv war für Sie die Steuerung der Hände</p> <p>0% 0% 23,1% 38,5% 38,5%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 4,2 s = 0,77 E = 0</p>	<p>(3.2) Wie bewerten Sie deren Funktionsweise</p> <p>0% 7,7% 30,8% 53,8% 7,7%</p> <p>sehr schlecht schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 3,6 s = 0,74 E = 0</p>
<p>(3.3) Wie bewerten Sie die Sicht durch das Headup-Display?</p> <p>0% 15,4% 53,8% 30,8% 0%</p> <p>gar nicht schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 3,2 s = 0,66 E = 0</p>	<p>(3.4) Wie bewerten Sie die Steuerung der Flugsimulation?</p> <p>7,7% 15,4% 53,8% 23,1% 0%</p> <p>sehr schlecht schlecht mittelmäßig gut sehr gut</p> <p>n = 13 mw = 2,9 s = 0,83 E = 0</p>
<p>(3.5) Wie war die Steuerung, wenn man sich auf die Flugsimulation konzentrierte?</p> <p>0% 38,5% 38,5% 15,4% 7,7%</p> <p>sehr schwer schwer mittelmäßig gut sehr gut</p> <p>n = 13 mw = 2,9 s = 0,92 E = 0</p>	<p>(3.6) Würden weitere Feedback Kanäle eine Nutzung für Sie vereinfachen?</p> <p>23,1% 46,2% 30,8% 0%</p> <p>akustisches haptisches beides keins</p> <p>n = 13 mw = 2,1 s = 0,73 E = 0</p>

## Rohdaten des virtuellen Cockpitsversuchs

Nr.	1.1	1.2	1.3	a	b	c	d	1.4	2.1	2.2	2.3	2.4	2.5	2.6	2.7
1	3	1	1	M. Sc.	Informatik	6		2	4	6	3	2	2	2	3
2	3	1	1	B. Sc.	Ang. Informatik	1		1	4	6	4	2	2	3	3
3	2	1	1	B. Sc.	Ang. Informatik	1		2	4	7	4	3	3	2	2
4	3	1	1	B. Sc.	Informatik	4		2	4	6	2	1	1	2	3
5	2	1	1	B. Sc.	Tech Informatik	3		2	4	6	4	3	3	2	3
6	3	1	1	M. Sc.	Informatik	4		2	4	6	3	2	1	2	3
7	3	1	1	B. Sc.	Tech. Informatik	3		2	4	5	3	2	3	3	2
8	3	1	1	M. Sc.	Informatik	5		2	4	6	4	2	2	2	4
9	3	1	1	B. Sc.	Informatik	5		1	4	6	4	4	4	1	2
10	3	1	1	M. Sc.	Informatik	5		2	4	6	4	3	1	1	3
11	3	2	1	M. Sc.	Informatik	6		1	4	5	4	3	3	1	2
12	5	1	1	M. Sc.	Informatik	5		1	4	6	3	2	2	1	3
13	5	1	1	M. Sc.	Informatik	6		2	4	6	3	3	3	1	2

Tabelle B.9: Ergebnisse Fragebogen virtuelles Cockpit Teil 1

Nr.	3.1	3.2	3.3	3.4	3.5	3.6	4.1	4.2
							Reaktive Steuerung (auf Ereignisse, Zustände)	
1	4	2	2	1	2	3	Ja, interaktive Bedienung	Reaktive Steuerung (auf Ereignisse, Zustände)
2	4	4	3	3	4	2	Ja!	Alltägliche Unterstützung z.B. Navi oder tragbarer Computer
3	4	4	3	3	3	2	Ja	Fahrzeugsteuerung
4	5	4	4	2	4	2	Ja	Jegliche Art der Steuerung von Fahrzeugen oder Maschinen; Videospiele; „Iron Man“ / Avatare
5	3	3	3	2	2	2	Ja könnte ich, wichtig wäre ein haptisches Feedback.	z.B. zur Steuerung autonomer Systeme.
6	3	4	3	3	3	1	Ja	Zusatzinformationen zu Produkten, Navigation, etc.
7	3	3	4	3	3	3	Ja!	Gaming; Produktvorschau
8	5	3	3	3	2	2	Ja	Berufsleben allgemein
9	4	4	2	3	2	2	-	-
10	4	4	3	4	2	1	Ja, falls sie keine Kopfschmerzen / Übelkeit verursachen würden	Überall wo eine Schaltzentrale benötigt wird, z.B. Fabriken zur Steuerung von Anlagen; Spielekonsolen
11	5	5	4	4	5	1	Ja	z.B zur Steuerung einfacher Haushaltsgeräte oder Heizung-/ Wassertemperatur, Lichtstärke, etc., Spielsteuerung, Navigation durch Fotoalben,...
12	5	4	3	4	3	3	Ja, mit Übung. Akustisches für Joystick	Roboterarme in einem Biolabor
13	5	3	4	3	3	3	Für eine Bedienung virtueller Bedienelemente würde ich mir großzügigere Gesten wünschen, um (hoffentlich) eine feinere Justierung realisieren zu können	Bedienung (von Apparaten) in sterilen Umgebungen. virtuelle Modellierung

Tabelle B.10: Ergebnisse Fragebogen virtuelles Cockpit Teil 2

*Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 27.02.2014 Olaf Potratz