



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

**Ausarbeitung zur Veranstaltung „Projekt 2“ im  
Masterstudiengang Informatik WiSe 2012/13**

Olaf Potratz

Erste Versuche mit einem 3D-Gestenframework

**Olaf Potratz**

**Thema der Ausarbeitung zur Veranstaltung „Projekt 2“ im Masterstudiengang Informatik WiSe 2012/13**

**Stichworte**

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

**Kurzzusammenfassung**

Die Ausarbeitung zur Veranstaltung „Projekt 2“ umfasst den jetzigen Entwicklungsstand des Frameworks. Darüber hinaus wurden erste Versuche mit verschiedenen Sensoren zum Steuern einer kleinen Anwendung durchgeführt und wurden hier dokumentiert.

**Olaf Potratz**

**Title of the paper**

Initial trials with a 3D gesture Framework

**Keywords**

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

**Abstract**

The preparation for the event "Project 2" includes the current state of development of the framework. Furthermore, initial trials conducted with various sensors for controlling a small application and are documented here.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Aufbau . . . . .	1
1.2	Motivation . . . . .	1
<b>2</b>	<b>Design und Realisierung</b>	<b>2</b>
2.1	Design . . . . .	2
2.1.1	Framework . . . . .	2
2.2	Realisierung . . . . .	3
2.2.1	Physik-Engine . . . . .	3
2.2.2	Kinect . . . . .	5
2.2.3	ART-Tracker . . . . .	6
2.2.4	Headup-Display (HUD) . . . . .	6
2.2.5	Powerwall . . . . .	6
<b>3</b>	<b>Erste Versuche</b>	<b>8</b>
3.1	Kinect . . . . .	8
3.2	ART-Tracker . . . . .	9
3.2.1	Mit Joystick-Attrappe . . . . .	9
3.2.2	Headup-Display . . . . .	10
<b>4</b>	<b>Zusammenfassung und Ausblick</b>	<b>11</b>
4.1	Zusammenfassung . . . . .	11
4.2	Ausblick . . . . .	11
	<b>Literaturverzeichnis</b>	<b>13</b>
	<b>Abbildungsverzeichnis</b>	<b>14</b>

# Kapitel 1

## Einleitung

### 1.1 Aufbau

Die Projektausarbeitung gliedert sich neben diesem Einleitungskapitel noch in drei weitere Kapitel. Das Kapitel 2 befasst sich mit den konkreten Design- und Realisierungsschritten, welche im Zuge der Projektarbeit umgesetzt wurden. Kapitel 3 soll dem Leser einen Überblick über den momentanen Entwicklungsstand des Projekts geben. Es werden hier erste Versuche mit verschiedenen Sensoren aufgeführt. Das Kapitel 4 gibt eine kurze Zusammenfassung der Arbeit wieder und zeigt im Ausblick, wohin die Entwicklung des Systems noch führen soll und bei welchen noch fehlenden Komponenten die Realisierung abgeschlossen werden soll.

### 1.2 Motivation

Diese Projektausarbeitung baut im Wesentlichen auf Vorarbeiten aus den vergangenen Semestern des Masterstudiums und der Bachelorarbeit Potratz (2011) auf. Im ersten Teil des Projekts, der im Potratz (2012a) zusammengefasst wurde, ging es hauptsächlich um den Aufbau einer Versuchsumgebung und der Suche welche Softwarekomponenten geeignet wären, diesen Aufbau zu verwirklichen. In dieser Ausarbeitung werden konkrete Umsetzungen einzelner Komponenten beschrieben und die Erfahrungen erster Versuche aufgeführt. Parallel zu dieser Veranstaltung fand außerdem noch die Seminar Ringvorlesung Potratz (2013) statt zu der es auch eine Ausarbeitung anzufertigen galt. Diese Arbeit befasst sich Schwerpunkt mäßig also mit dem Istzustand des Projekts wohin gegen die Ausarbeitung der Ringvorlesung sich eher mit der grundsätzlichen Entwicklung des Systems befassen und dem damit verbundenen Zielen.

# Kapitel 2

## Design und Realisierung

Dieses Kapitel soll sowohl das der Realisierung zugrunde liegende Design vorstellen, als auch einen Einblick auf die Art und Weise und den Stand der Umsetzung gewähren.

### 2.1 Design

Das oberste Ziel dieses Design soll eine möglichst lose Kopplung jeglicher Komponenten sein. Einerseits um der stetig wachsende 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 bislang alle Komponenten in C# zu erstellen, aber da sich diese möglichst breite Einbindung von Komponenten auch andere Programmiersprachen beinhalten soll, wurden auch einige Adapter für Sensoren wie auch Ausgabekomponenten in Java implementiert.

#### 2.1.1 Framework

##### Istzustand

Das in Abbildung 2.1 dargestellte Framework ist in großen Teilen umgesetzt und die programmtechnische Umsetzung entspricht diesem Framework. Die Sensoren liefern ihre nur rudimentär aufbereiteten Daten an den ActiveMQ<sup>1</sup>. Von dort aus erhält die Physiks simulation ihre Daten. Innerhalb der Physiks simulation werden diese Daten dann aufbereitet und verarbeitet. Sämtliche Positionen werden für eine grafische Darstellung zurück an den ActiveMQ

---

<sup>1</sup><http://activemq.apache.org/>

versandt. Darüber hinaus werden Manipulationen durch die Aktoren an den Physikobjekten registriert und durch die Gestenerkennung ausgewertet. Auch diese Ergebnisse werden dann zur Programmsteuerung an den ActiveMQ übermittelt.

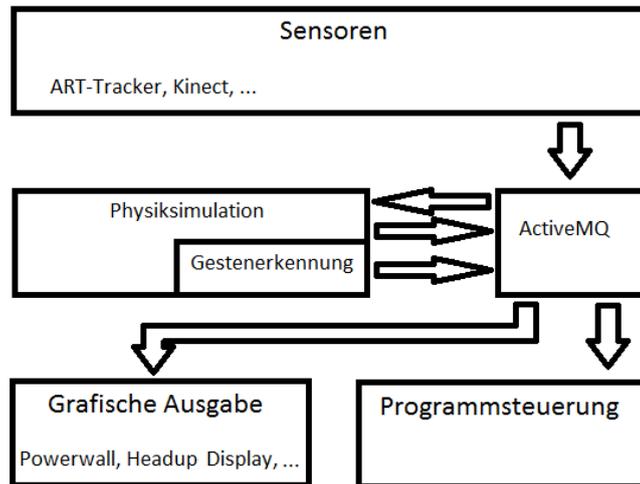


Abbildung 2.1: : Jetztiger Aufbau des Frameworks

### Sollzustand

Im Laufe der Umsetzung des Designs und dem Fortschreiten des Entwicklungsprozesses haben sich bereits notwendige Änderungen ergeben. Vor allem soll die Aufbereitung der Sensordaten und die Erkennung der Greifgeste zur entsprechenden Sensorik verlagert werden. Somit wäre eine strikte Trennung von Physik und Sensorik gegeben was den Einbau weiterer Sensoren erleichtern würde. Weiterführende Angaben hierzu und auch eine ausführliche Grafik können aus dem Seminarbericht Potratz (2013) entnommen werden.

## 2.2 Realisierung

Im folgenden Abschnitt wird kurz auf die Umsetzung der bereits vorhandenen Komponenten einzeln eingegangen.

### 2.2.1 Physik-Engine

Wie bereits in der Ausarbeitung Potratz (2012a) zu Projekt 1 beschrieben wurde, fiel die Entscheidung eine Umsetzung der Physik auf eine Einbettung in XNA 4.0<sup>2</sup> entschieden, als

<sup>2</sup><http://msdn.microsoft.com/de-de/library/bb200104%28v=xnagamestudio.40%29.aspx>

eigentliche Physik-Engine kommt Nvidia PhysX<sup>3</sup> zum Einsatz. Die Verbindung von PhysiX und XNA 4.0 wird durch das PhysX.Net<sup>4</sup> Framework gewährleistet. Abbildung 2.2 zeigt den für die Physik-Engine benötigten Aufbau eines Joysticks. Einzelne Gesten, wie links, rechts oder Feuer, werden mittels Kollisionserkennung der entsprechenden farblichen Flächen mit den eingefärbten Kugeln ermittelt.

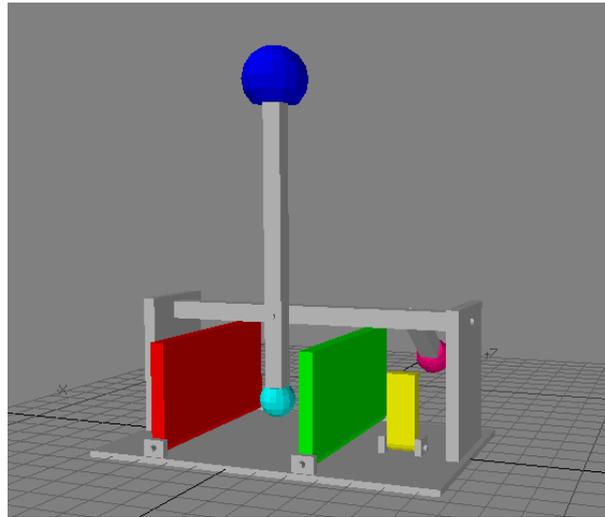


Abbildung 2.2: : Aufbau des Physik-Joysticks

Um eine Interaktion zwischen dem Akteur<sup>5</sup>, 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.

### Verfahren a - Reibung

Der Akteur berührt das Physikobjekt, die Physik-Simulation errechnet anhand von Oberflächeneigenschaften und auftretenden Kräften die entstehende Reibung. Das Physikobjekt wird ausschließlich durch die errechneten Reibungskräfte manipuliert.

### Verfahren b - Distanzunterschreitung

Sobald der Akteur wie in Abbildung 2.3b zu sehen ist, eine gewisse Distanz unterschreitet, wird zwischen dem Akteur und dem Physikobjekt eine Verbindung (Joint<sup>6</sup>) erstellt. Ab die-

<sup>3</sup><http://www.nvidia.de/object/physx-9.12.1031-driver-de.html>

<sup>4</sup><http://physxdotnet.codeplex.com/>

<sup>5</sup>Bei einem Akteur handelt es sich um die simulierte Repräsentation der Hand des Benutzers.

<sup>6</sup>Bei einem Joint handelt es sich um eine Gummiband ähnliche Verbindung, näheres dazu in Potratz (2011)

sem Zeitpunkt werden die vom Aktor ausgehenden Kräfte zur Manipulation über den Joint an das Physikobjekt übertragen. Sobald der Aktor, die Distanz wieder vergrößert wird, die Verbindung aufgehoben.

### Verfahren c - Greifgeste

Das in Abbildung 2.3c gezeigte Verfahren ist eine Kombination von Distanzunterschreitung und einem separaten Ereignis, in diesem Fall das Greifen, also Schließen der Hand. Wird die Hand geschlossen und befindet sich gleichzeitig innerhalb der Grenzwerte zu einem Physikobjekt wird eine Verbindung (Joint) zwischen Aktor und Physikobjekt erstellt. In diesem Fall wird die Verbindung erst wieder gelöst, sobald die reale Hand geöffnet wird.

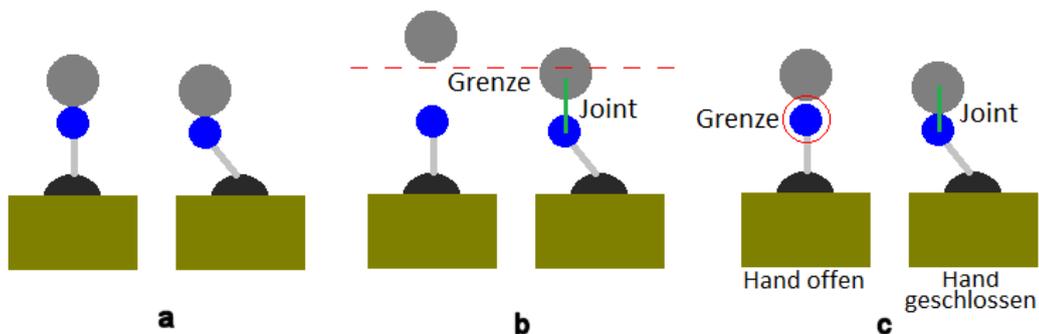


Abbildung 2.3: : Verfahren zur Interaktion zwischen Aktor und Physikobjekt

## 2.2.2 Kinect<sup>7</sup>

Die Umsetzung des Adapters für die Kinect erfolgte mit dem von Microsoft angebotene SDK<sup>8</sup>. Der Adapter filtert aus den Daten der Kinect die Positionen der Hände, alle weiteren Daten werden verworfen. In der ersten Variante des Adapters, der noch bei dem Versuch 1 3.1 zum Einsatz kam, wurde noch mittels eines UDP Broadcast Streams mit der Physiksimulation kommuniziert. Im weiteren Fortschreiten des Projekts und der Einbindung des JMS Frameworks<sup>9</sup> für C# erfolgt diese Kommunikation nun über den ActiveMQ.

<sup>7</sup><http://www.microsoft.com/en-us/kinectforwindows/>

<sup>8</sup><http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx>

<sup>9</sup><http://activemq.apache.org/nms/apachenmsactivemq.html>

### 2.2.3 ART-Tracker

Die Ursprünge dieses Sensoradapters gehen auf die Arbeit Potratz (2011) zurück. Die Funktionsweise ist nahezu gleich geblieben, da auch die gleichen Handschuhe für den Anwender benutzt werden. Die Kommunikation findet nun ausschließlich mittels JMS über den ActiveMQ statt. Die getrackten Reflexionen der Fingerspitzen werden zusammengefasst und ein Mittelpunkt gebildet, dieser Mittelpunkt wird dann für die Position des Aktors übertragen.

### 2.2.4 Headup-Display (HUD)

Das Headup-Display (Abbildung 2.4), im weiteren Verlauf der Arbeit auch mit HUD abgekürzt, stellt eine Art Hybrid dar, einerseits ist es ausschließlich zur Wiedergabe der Grafik gedacht, andererseits benötigt es zur Darstellung der Grafik selbst seine spezifische Position um Raum, damit das passende Bild für den Träger des HUD errechnet werden kann. Das Headup-Display benötigt somit auch eine eigene Sensorenkomponente, die es mit den benötigten Werten versorgt. Die momentane Umsetzung erfolgt noch unter Ausschluss des ActiveMQ. Zur grafischen Darstellung wird Java 7 und das darauf aufsetzende, 3D-Engine JMonkey<sup>10</sup> Framework verwendet.

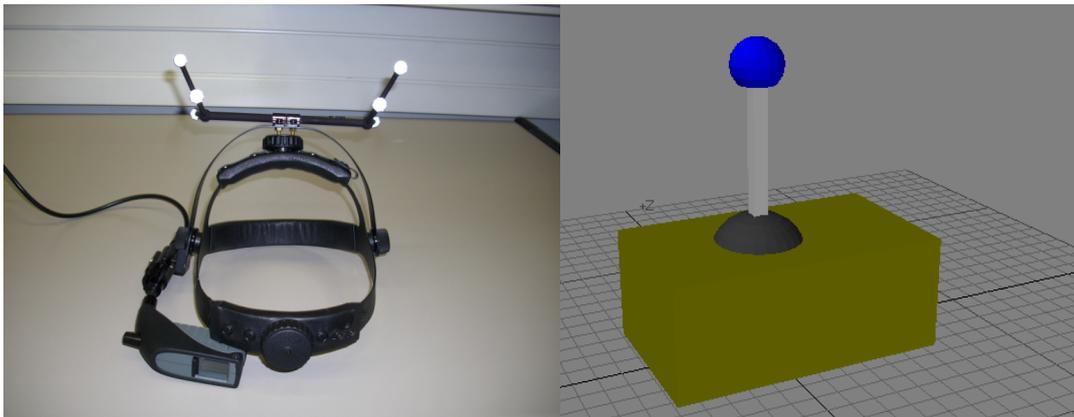


Abbildung 2.4: : Links: HUD mit ART-Artefakt

Rechts: Grafische Darstellung des Joysticks aus der Physiksimulation

### 2.2.5 Powerwall

Die hier in Abbildung 2.5 zu sehende Powerwall ist mittlerweile mit 2 „ATI Radeon™ HD 5870 Eyefinity 6“<sup>11</sup> Grafikkarten ausgestattet und ermöglicht so eine maximale Auflösung

<sup>10</sup><http://jmonkeyengine.org/>

<sup>11</sup><http://www.amd.com/de/products/desktop/graphics/ati-radeon-hd-5000/hd-5870-eyefinity-6-edition/Pages/overview.aspx>

von 5760 \* 2160 Pixeln. Für die anstehenden Versuche wird die Powerwall ausschließlich als Anzeigegerät für die zu steuernde Endanwendung benutzt. Die Physiksimulation selbst läuft auf einem anderen Rechner, der wiederum nur die entsprechenden Steuerungssignale an die Powerwall sendet. Mittlerweile wird diese Kommunikation mittels ActiveMQ gehandhabt, die Powerwall abonniert also die entsprechenden Queues und erhält so die für sie relevanten Daten.



Abbildung 2.5: : Powerwall Stand 27.02.2013

# Kapitel 3

## Erste Versuche

Die ersten Versuche basieren auf den Vorarbeiten die bereits in Potratz (2012a) beschrieben wurden. Der Physikraum ist wie in Abschnitt 2.2.1 aufgeführt implementiert. Das verwendete Physikobjekt entspricht dem Joystick aus Abbildung 2.2. Für alle folgenden Versuche gilt das zur Verbindung von Aktor und Physikobjekt das in Abbildung 2.3b beschriebene Verfahren mittels Distanzunterschreitung zum Einsatz kommt.

### 3.1 Kinect

Im ersten Versuch sollte ein SpacInvader-Clone mithilfe der Kinect gesteuert werden. Die Auswertung der Sensordaten beschränkte sich in diesem Versuch auf die rechte Hand. Die Kommunikation zum Zeitpunkt dieses Versuchs war noch mittels UDP-Paketen und Broadcast realisiert. Die so mit Positionsdaten versorgte Physiks simulation war bereits in C# verfasst und verwendete XNA 4.0 und PhysikX als dazugehörige Physik-Engine. Die Übermittlung der ermittelten Gesten und somit hervorgehenden Steuerkommandos an das Computerspiel fand ebenfalls mit UDP-Paketen statt. Wie auf Abbildung 3.1 zu erkennen ist, wurde eine Joystick-Attrappe aus Holz verwendet. Diese Attrappe verfügt weder über Anschlüsse, noch über Marker und wird von den Sensoren nicht wahrgenommen. Zu Anfang wurde die Position der Attrappe ermittelt, indem der Benutzer den Joystick wie üblich am oberen Ende anfasst. Es wurden somit die Handposition bestimmt und somit ermittelt, wo sich die Attrappe im Raum befindet. Anschließend positionierte man den virtuellen Joystick im Physikraum auf die entsprechende Position, sodass beide Positionen in ihrem jeweiligen Raum identisch waren. Zur grafischen Ausgabe des SpacInvader-Clones kam ein Beamer zum Einsatz, da zu diesem Zeitpunkt die Powerwall aus technischen Gründen nicht zur Verfügung stand. Als Ergebnis aus diesem Versuch stellte sich heraus, dass der Aufbau an sich funktionierte, nur die zum Teil sehr langen Reaktionszeiten des Kinect-Sensors fielen negativ auf.

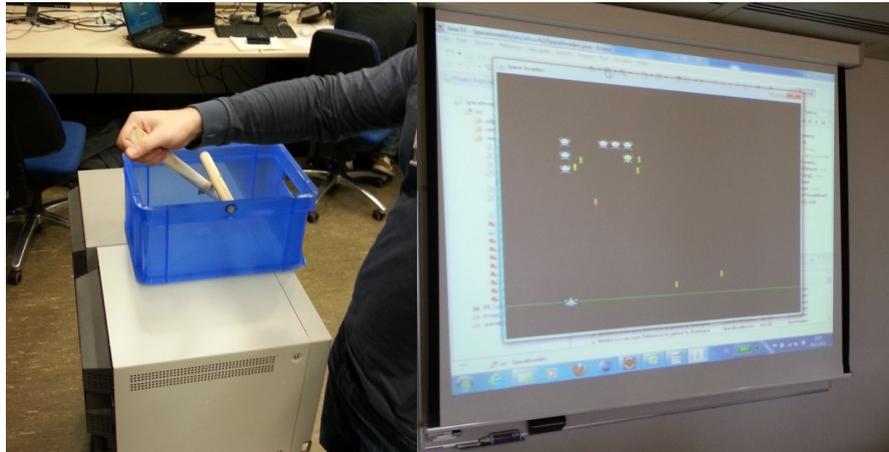


Abbildung 3.1: : Fotos vom Versuch mit der Kinect

## 3.2 ART-Tracker

Im zweiten Versuch geht es auch darum den SpacInvader-Clone zu steuern. Wieder sind die Bewegungen links und rechts zum Steuern, das nach vorne drücken des Joysticks zum Feuern gedacht. Zum Einsatz kommt als Sensor das ART-Tracking System. Hierfür ist es notwendig, dass der Benutzer einen Handschuh (wie in Abbildung 3.2 zu sehen) trägt, da ausschließlich die reflektierende Kugeln an den Fingerspitzen vom ART-Tracking System erkannt werden. Die Ausgabe des Spiels erfolgt über die in Abbildung 2.5 abgebildete Powerwall.

### 3.2.1 Mit Joystick-Attrappe

Im ersten Durchlauf kommt wieder die Joystick-Attrappe zum Einsatz, hauptsächlich damit der Benutzer ein Gefühl für die Bewegungen bekommt. Die Abmessungen für das Physikobjekt sind die gleichen wie auch beim ersten Aufbau mit der Kinect, nur die Position im Physikraum selbst musste angepasst werden, da der Benutzer nun sitzt. Das Anpassen der Position wurde unter Verwendung eines weiteren ART-Tracker Artefakts vorgenommen, das zu diesem System mitgelieferte Artefakt ist eingemessen und liefert somit eine genaue Position im Raum. Das Artefakt wurde zusammen mit der Joystick-Attrappe aufgestellt, wodurch die Position der Attrappe im Raum festgestellt wurde und deren Position für den Versuch in der Physikwelt entsprechend angepasst wurde.

Als Resumee aus dem Versuch lässt sich sagen, dass die ART-Tracker Sensoren wesentlich zeitnaher reagieren als es die Kinect tut, auch ist die erfasste Position der Hand wesentlich genauer.



Abbildung 3.2: : Foto vom Versuch mit dem ART-Tracker

### 3.2.2 Headup-Display

Der zweite Durchlauf des Versuchs erfolgte nun zusätzlich mit dem Headup-Display. Wie in Abschnitt 2.2.4 beschrieben ermöglicht das Headup-Display eine Darstellung des Physik-Objekts im realen Raum und sorgt so für eine "mixed reality" Song u. a. (2008). Dank dem Headup-Display ist man nun in der Lage auf die Joystick-Attrappe zu verzichten, da der Benutzer das zu manipulierende Objekt auch so sehen kann.

Die Interaktion ohne etwas real greifen zu können ist ungewohnt und bedarf einiger Übung, die Funktionalität selbst ist davon aber nicht betroffen.

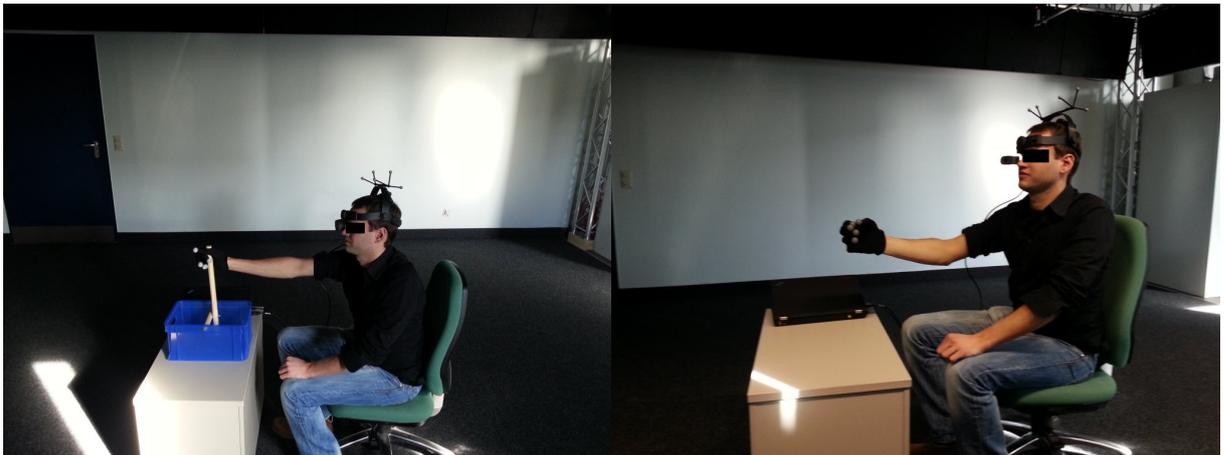


Abbildung 3.3: : Links: HUD und Joystick-Attrappe Rechts: ausschließlich HUD Nutzung

# Kapitel 4

## Zusammenfassung und Ausblick

Das letzte Kapitel dieser Arbeit beinhaltet eine Zusammenfassung und noch einen Ausblick auf die anstehenden Folgearbeiten.

### 4.1 Zusammenfassung

Diese Ausarbeitung befasste sich mit dem aktuellen Stand der Entwicklung aus der Veranstaltung „Projekt 2“. Zu Beginn der Arbeit wurde der Aufbau der Arbeit erläutert und eine kurze Motivation umrissen. Im zweiten Kapitel ist die Arbeit auf den Entwicklungsstand der einzelnen Komponenten eingegangen und das vorhandene Framework beleuchtet. Im dritten Kapitel wurden erste Versuche beschrieben. In allen Versuchen sollte ein SpaceInvader-Clone gesteuert werden. Die Steuerung umfasste zur Bewegung links und rechts steuern und als Befehl zum Feuern das nach vorne drücken des Joysticks. Es kamen hierbei jeweils die Kinect und der ART-Tracker als Sensoren zum Einsatz. Zur grafischen Ausgabe wurden ein Beamer, die Powerwall und das Headup-Display verwendet. Das vierte und letzte Kapitel umfasst diese Zusammenfassung und beschreibt im Anschluss den Ausblick mit den anstehenden Änderungen und Erweiterungen.

### 4.2 Ausblick

Im Ausblick soll sich mit den bereits jetzt feststehenden Erweiterungen und Änderungen am vorhandenen System befasst werden.

#### **Anpassung des Frameworks**

Wie bereits in Kapitel 2 aufgeführt wurde im Laufe der Entwicklung klar, dass noch Änderungen am Framework notwendig sind. Der nun angestrebte Ausbau des Frameworks wird

im Seminarbericht Potratz (2013) detailliert aufgeführt und verdeutlicht. Bevor weitere Implementierungen fortgeführt werden, soll der vorhandene Stand der Implementierung dem nun angestrebtem Framework entsprechen. Dies verspricht auch für die noch folgenden Komponentenumsetzungen von Vorteil zu sein, da sie die Einbindung erleichtern werden.

### **Erkennen der Greifgeste**

Das in Abschnitt 2.2.1 beschriebene Verfahren zum „Greifen“ nach den Physikobjekten und in dem Fall dem Joystick gilt es umzusetzen. Als Grundlage für dieses Verfahren gelten Überlegungen die in der Bachelorarbeit Potratz (2011) in Kapitel 4 gemacht wurden. Wie bereits im ersten Abschnitt des Ausblick angekündigt, soll die Umsetzung dieser Komponente im Stil des erneuerten Frameworks erfolgen. Diese Komponente wird also nicht wie zuerst gedacht in der Physik selbst angesiedelt, sondern bereits Bestandteil der ART-Tracker Sensors werden. Dafür wird die Nachricht die die ART-Tracker Sensorkomponente an den ActiveMQ verschickt um eine Variable erweitert, nämlich ob die Hand geöffnet oder geschlossen ist.

### **Erweiterung des Physikmodells**

Nach dem Umbau des Frameworks und der Inbetriebnahme der Erkennung für Greifgesten gilt es weitere Interaktionsobjekte für die Physiksimulation zu erschaffen. Wie in der Seminararbeit Potratz (2013) beschrieben, gilt es eine Schalttafel zu konstruieren und hierfür geeignete Instrumente herzustellen. Da in Folge der Greifgeste die Interaktion mit den jeweiligen Objekte vereinfacht wurde und auf Haftreibung verzichtet werden kann, können die Objekte wesentlich einfacher designt werden. Die nun zu konstruierenden Objekte werden von denen, die als ersten Überlegungen aus der Veranstaltung Anwendungen 1 Potratz (2012b) entstanden sind abweichen und keinerlei Haltevorrichtungen für den Akteur zur Reibungsverbesserung besitzen.

# Literaturverzeichnis

- [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: *Medienfassaden*. 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master11-12-aw1/potratz/bericht.pdf>
- [Potratz 2013] POTRATZ, Olaf: *Physikbasierte Interaktion im virtuellen Raum*. 2013. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master12-13-seminar/potratz/bericht.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>

# Abbildungsverzeichnis

2.1	: Jetztiger Aufbau des Frameworks . . . . .	3
2.2	: Aufbau des Physik-Joysticks . . . . .	4
2.3	: Verfahren zur Interaktion zwischen Aktor und Physikobjekt . . . . .	5
2.4	: Links: HUD mit ART-Artefakt Rechts: Grafische Darstellung des Joysticks aus der Physiksimulation . . . . .	6
2.5	: Powerwall Stand 27.02.2013 . . . . .	7
3.1	: Fotos vom Versuch mit der Kinect . . . . .	9
3.2	: Foto vom Versuch mit dem ART-Tracker . . . . .	10
3.3	: Links: HUD und Joystick-Attrappe Rechts: ausschließlich HUD Nutzung . . . . .	10