

**„Entwicklung einer Gestenerkennung auf Basis  
des Beschleunigungssensors in der Wii Remote“**

**Diplomarbeit**

zur Erlangung des Diplomgrades  
Diplom-Informatiker (FH)

**Fachhochschule Köln - Campus Gummersbach**  
Fakultät für Informatik und Ingenieurwissenschaften

Studiengang Allgemeine Informatik

ausgearbeitet von:

**Marco Polak**

Matrikel-Nr.: 11045897

Erster Prüfer: Prof. Dr. Wolfgang Konen  
Zweiter Prüfer: Dipl.-Inform. (FH) Stefan Zingel

Gummersbach, im August 2008

# Inhaltsverzeichnis

<b>1 Einleitung und Aufgabenstellung</b> .....	<b>6</b>
1.1 Gliederung der Arbeit.....	6
1.2 Motivation.....	6
<b>2 Problemanalyse und Anforderungsdefinition</b> .....	<b>8</b>
2.1 Ist-Situation.....	8
2.1.1 Project Wiigee.....	9
2.2 Soll-Situation.....	9
<b>3 Darstellung verwendeter physikalischer Größen und Arbeitsmittel</b> .....	<b>11</b>
3.1 Die Wii Remote.....	11
3.2 Eingesetzte Software.....	12
3.2.1 Bluetooth Umgebung „Microsoft Bluetooth Umgebung“.....	12
3.2.2 Hilfsprogramm „Glove PIE“.....	13
3.3 Entwicklungsumgebung.....	13
3.3.1 Schnittstellenbibliothek zur Wii Remote.....	13
3.4 Beschleunigung.....	13
3.5 Signale.....	14
<b>4 Analyse der Gesten</b> .....	<b>15</b>
4.1 Testpersonen.....	15
4.2 Kreis.....	16
4.3 Frisbee.....	19
4.4 Schlag.....	22
4.5 Bowling.....	26
4.6 Überkopfwurf.....	29
4.7 Fazit der Analyse.....	32
<b>5 Gestenerkennung</b> .....	<b>33</b>
5.1 Schritte der Gestenerkennung.....	33
5.1.1 Lernprozess.....	33
5.1.2 Vorverarbeitung.....	34
5.1.3 Klassifikation.....	35
5.2 Automatische Erkennung der Gestenausführung.....	38
<b>6 Die Programmkomponente „Wiinput“</b> .....	<b>40</b>
6.1 Interface.....	40

---

6.2 Programmablauf.....	42
6.3 Klassendiagramm.....	43
6.4 Strategien.....	44
6.4.1 Unnormalized.....	44
6.4.2 Normalized.....	44
6.4.3 CompareWithAll.....	44
6.5 Speicherung der Daten.....	45
<b>7 Evaluation der Gestenerkennung.....</b>	<b>46</b>
<b>8 Schluss.....</b>	<b>50</b>
8.1 Ansatzpunkte für Folgeuntersuchungen.....	51
8.2 Ausblick.....	53
<b>Literaturverzeichnis.....</b>	<b>54</b>
<b>Ehrenwörtliche Erklärung.....</b>	<b>55</b>

# Abbildungsverzeichnis

Abbildung 1: Wii Remote.....	11
Abbildung 2: Achsenverlauf der Wii Remote.....	12
Abbildung 3: Signal einer Geste.....	14
Abbildung 4: Kreisbewegung.....	16
Abbildung 5: Kreis der Testperson 1.....	16
Abbildung 6: Kreis der Testperson 10.....	17
Abbildung 7: Frisbeewurf.....	19
Abbildung 8: Frisbeewurf der Testperson 1.....	19
Abbildung 9: Frisbeewurf der Testperson 10.....	20
Abbildung 10: Frisbeewurf der Testperson 12.....	21
Abbildung 11: Schlagbewegung.....	22
Abbildung 12: Schlag der Testperson 1.....	23
Abbildung 13: Schlag der Testperson 10.....	24
Abbildung 14: Schlag der Testperson 12.....	25
Abbildung 15: Bowlingwurf.....	26
Abbildung 16: Bowlingwurf der Testperson 1.....	26
Abbildung 17: Bowlingwurf der Testperson 10.....	27
Abbildung 18: Bowlingwurf der Testperson 12.....	28
Abbildung 19: Überkopfwurf.....	29
Abbildung 20: Überkopfwurf der Testperson 1.....	29
Abbildung 21: Überkopfwurf der Testperson 10.....	30
Abbildung 22: Überkopfwurf der Testperson 12.....	31
Abbildung 23: Zeitnormierung.....	34
Abbildung 24: Klassifikation Schritt 1.....	36
Abbildung 25: Klassifikation Schritt 2.....	37
Abbildung 26: Klassifikation Schritt 3.....	38
Abbildung 27: Interface der Klasse „Wiinput“.....	40
Abbildung 28: Programmablauf von „Wiinput“.....	42
Abbildung 29: Klassendiagramm von „Wiinput“.....	43
Abbildung 30: Speicherung der Gesten.....	45

## **Tabellenverzeichnis**

<b>Tabelle 1: Testpersonen.....</b>	<b>15</b>
<b>Tabelle 2: Testergebnisse des ersten Testlaufs.....</b>	<b>47</b>
<b>Tabelle 3: Testergebnisse des zweiten Testlaufs.....</b>	<b>47</b>
<b>Tabelle 4: Testergebnisse des dritten Testlaufs.....</b>	<b>48</b>
<b>Tabelle 5: Testergebnisse des vierten Testlaufs.....</b>	<b>48</b>

# 1 Einleitung und Aufgabenstellung

Die Wii Remote ist ein interaktives Eingabemedium der Spielkonsole Wii von Nintendo. Die Firma Nurogames hat es sich, neben der Entwicklung von anderen Spielen, ebenfalls zur Aufgabe gemacht, Spiele für die Nintendo Wii zu entwickeln. Die Steuerung der Wii, durch eigene Gesten der Spieler, bietet eine innovative Möglichkeit, das Spiel selbstständig durch Bewegung zu steuern und ist daher sowohl bei jüngeren als auch bei älteren Benutzern sehr beliebt.

Diese intuitive Steuerung soll auch in den Spielen der Firma Nurogames verwendet werden. Somit wird eine Programmkomponente benötigt, die möglichst gut jede vorstellbare Geste erkennt. Außerdem soll der Programmieraufwand der Entwickler, eine neue Geste einzupflegen, möglichst gering bleiben. Dementsprechend soll im Rahmen dieser Arbeit eine Programmkomponente entwickelt werden, mit der man sowohl neue Gesten problemlos aufnehmen, als auch ausgeführte Gesten unter den Aufgenommenen wieder erkennen kann.

## 1.1 *Gliederung der Arbeit*

Zunächst wird das Problem der Gestenerkennung auf Basis des Beschleunigungssensors in der Wii Remote analysiert und die Anforderungen an die Programmkomponente definiert. Außerdem wird erörtert, welche Lösungen bereits auf dem Markt vorhanden sind und wie diese aussehen.

Anschließend werden verwendete physikalische Größen und Arbeitsmittel, die für die Entwicklung der Programmkomponente zur Gestenerkennung eingesetzt wurden, so wie die angewendeten Arbeitsmethoden erläutert.

Im Hauptteil der Arbeit werden zunächst ausgeführte Gesten von Testpersonen graphisch dargestellt und analysiert. Bevor die entwickelte Programmkomponente beschrieben wird, werden die einzelnen Schritte der zur Gestenerkennung eingesetzten Verfahren, die bei der Entwicklung verwendet wurden, näher erläutert. Diese Verfahren werden dann mit den Gesten der Testpersonen evaluiert.

Abschließend werden noch offen stehende Punkte diskutiert und Ausblicke für die zukünftige Fortführung der Forschungsarbeit gegeben.

## 1.2 *Motivation*

In den Spielen der Firma Nurogames, die für die Wii Konsole entwickelt werden, wird der Einsatz einer Gestenerkennung angestrebt. Hierbei spielt die Aufnahme von neuen Gesten eine große Rolle, da diese regelmäßig neuen Spielen hinzugefügt werden müssen. Dabei

muss ein einfacher Arbeitsvorgang zum Einfügen neuer Gesten garantiert sein, damit die Erweiterung der Spiele problemlos erfolgen kann. Des Weiteren wird eine hohe Erkennungsrate der Gesten in den Spielen, trotz der Vielfältigkeit der verschiedenen Gesten und der Vielfältigkeit der verschiedenen Personen, die mit der Wii spielen, vorausgesetzt, damit die Programmkomponente auch in der Praxis eingesetzt werden kann.

## 2 Problemanalyse und Anforderungsdefinition

Das Ziel dieser Diplomarbeit ist es, eine Gestenerkennung für die Wii Remote zu entwickeln, die auf dem Beschleunigungssensor der Wii Remote basiert. Diese Gestenerkennung soll in Spielen für die Nintendo Wii eingesetzt werden.

Eine weitere Anforderung, der zu entwickelnden Gestenerkennung, ist die Vereinfachung beim Hinzufügen weiterer Gesten in neue Spiele. Dies soll erreicht werden, indem die Gesten der Programmkomponente durch einen Lernprozess beigebracht werden, bei dem die neu zu erlernende Geste mehrmals hintereinander ausgeführt werden muss. Folglich müssen Programmierer vorher nicht jede neue Geste manuell analysieren und anschließend programmieren. Da die Programmierer durch den Einsatz der Gestenerkennung unnötige Arbeitsschritte und somit auch Zeit einsparen können, kann die Dauer der Entwicklung verkürzt werden, was sich wiederum positiv auf die Kosten der Entwicklung des Spieles auswirkt.

Die Gestenerkennung soll nach dem erfolgreichen Lernprozess nicht nur die Geste von dem Programmierer, sondern auch von jeder anderen beliebigen Person, die das Spiel spielen möchte, erkennen. Deswegen wird bei der Entwicklung dieser Komponenten untersucht, ob es ausreicht, eine Geste nur von einer Person aufzunehmen oder ob hierfür eine Aufnahme der Gesten von verschiedenen Personen erforderlich ist.

### 2.1 Ist-Situation

Die Firma Nurogames ist bisher mit einem Spiel für die DS Konsole von Nintendo auf dem Markt vertreten. Zukünftig sollen auch Spiele für die Nintendo Wii entwickelt werden. Da die Wii ein komplett anderes Steuerungssystem, als die bisher entwickelten Spiele, verwendet, ist eine neue Engine notwendig, die das Steuerungssystem der Wii Remote behandelt. Ein Teil der Steuerung beruht auf der Erkennung von Gesten, die mit der Wii Remote ausgeführt werden. Die Gesten müssen analysiert und der richtigen Geste aus einem Referenzbestand zugewiesen werden.

Es sind bereits zwei Projekte in diesem Bereich vorhanden, die die Gestenerkennung thematisieren. Zum einen das kommerzielle Programm Allive und das Open Source Projekt Wiigee, das von Benjamin Poppinga in seiner Projektarbeit der Universität Oldenburg entwickelt wurde [1].

Das kommerzielle Produkt Allive findet aus Kostengründen bei der Firma Nurogames keine Verwendung. Außerdem kann der Quellcode der Software nicht eingesehen und somit auch nicht den Bedürfnissen der Firma Nurogames angepasst werden.



### **2.1.1 Project Wiigee**

Das Open Source Projekt Wiigee von Poppinga [1] kann ebenfalls nicht in der Spieleentwicklung der Firma Nurogames eingesetzt werden, da dieses in Java programmiert wurde und diese Sprache nicht zur Entwicklung der Spiele eingesetzt wird. Des Weiteren untersucht Poppinga in seiner Arbeit nicht die Anwendbarkeit der Gestenerkennung auf unterschiedliche Personen, was einen großen Nachteil für den Einsatz dieses Programms in der Spieleentwicklung wäre, da untersucht werden muss, ob verschiedene Personen eine Geste unterschiedlich ausführen. Nur wenn dies nicht der Fall wäre, kann man davon ausgehen, dass das Programm von Poppinga auch personenübergreifend eine hohe Erkennungsrate aufweist.

Poppinga selbst geht bei seiner Gestenerkennung in mehreren Schritten vor. Zunächst führt Poppinga eine Clusterisierung der Daten mit Hilfe eines k-means Algorithmus durch. Anschließend kommt ein Hidden-Markov-Modell zum Einsatz, welches für die aufgenommenen Gesten die Modellwahrscheinlichkeiten ermittelt. Durch diese Wahrscheinlichkeiten kann ein Bayes-Klassifikator die zu klassifizierenden Gesten den Referenzgesten zuordnen.

Dieser Ansatz ist sehr komplex und daher soll untersucht werden, ob eine Gestenerkennung auch mit einfachen Ansätzen möglich ist.

## **2.2 Soll-Situation**

Im Rahmen dieser Diplomarbeit soll eine Programmkomponente entwickelt werden, die mit der Wii Remote ausgeführte Gesten erkennen soll.

Dabei soll eine personenübergreifende Gestenerkennung stattfinden. Das heißt, die gleichen Gesten in einem Spiel müssen von jeder Person erkannt werden. Dabei muss sowohl zwischen Rechts- und Linkshänder als auch zwischen Personen unterschiedlichen Geschlechts und Alters differenziert werden. Zunächst muss also analysiert werden, ob unterschiedliche Personen gleiche Gesten ähnlich ausführen, so dass eine automatisierte Gestenerkennung möglich ist. Diese Diplomarbeit soll die erste sein, die eine Evaluation zu der Wii-Gestenerkennung über mehrere Testpersonen hinweg macht.

Um diese Bibliothek später erfolgreich in Spielen einsetzen zu können, muss eine Erkennungsrate von mindestens 95% der Gesten erreicht werden. Im Rahmen dieser Forschungsarbeit wurden fünf Gesten, die zukünftig in den Spielen der Firma Nurogames eingesetzt werden sollen, ausgewählt:

- ein Kreis, der in die Luft gezeichnet wird
- ein Frisbeewurf, bei dem die Wii Remote eine Frisbee darstellen soll

- ein Schlag von Rechts nach Links, der mit der Hand ausgeführt wird
- ein Bowlingwurf, bei dem die Wii Remote eine Bowlingkugel darstellen soll
- ein Überkopfwurf, bei dem die Wurfbewegung wie beim Handball ausgeführt werden soll

Diese Gesten, ausgeführt von mehreren Testpersonen, dienen sowohl der Analyse als auch der Evaluation der Gestenerkennung. Anhand dieser Testläufe soll überprüft werden, in wieweit die Forschungs- und Entwicklungsziele erreicht wurden.

Außerdem muss zum Einsatz in der Spieleentwicklung eine automatische Erkennung erfolgen, die den Anfang und das Ende einer Geste erkennt, damit der Benutzer diese Ereignisse nicht selbst auslösen muss und die Knöpfe für zusätzliche Aktionen eingesetzt werden können.

Anschließend soll dieses Programm der Gestenerkennung in der Spieleentwicklung ihren praktischen Einsatz finden und mühsame Arbeitsschritte der Programmierer erheblich vereinfachen.

## 3 Darstellung verwendeter physikalischer Größen und Arbeitsmittel

Für ein besseres Verständnis werden zunächst die, zur Entwicklung einer Gestenerkennung auf Basis des Beschleunigungssensors benötigten Arbeitsmittel und physikalischen Größen näher beschrieben. Das hauptsächliche Arbeitsmittel ist hierbei die Wii Remote. Doch auch bestimmte Software und Schnittstellen werden zur Entwicklung der Gestenerkennung benötigt. Des Weiteren werden begriffliche Grundlagen zu Beschleunigung, Signalen und Mustererkennungen erläutert und erklärt.

### 3.1 Die Wii Remote



Abbildung 1: Wii Remote

Die Wii Remote ist ein Eingabemedium der Spielekonsole Wii von Nintendo. Sie hat die Form einer Fernbedienung und hat deswegen von Nintendo den Namen „Wii Remote“ oder auch kurz „Wiimote“ erhalten. Neben den üblichen Steuerelementen wie die Knöpfe und das Steuerkreuz, enthält die Wii Remote zwei neue Bedienelemente, die es ermöglichen Spiele auf der Nintendo Wii Konsole auf völlig neue Weise zu steuern.

Zum einen enthält die Wii Remote eine Infrarotkamera, die die relative Position zu der ebenfalls zur Wii Konsole gehörenden Sensorleiste messen kann. Mit dieser Infrarotkamera wird der Cursor auf dem Bildschirm gesteuert, indem die Position der Wii Remote auf den Bildschirm übertragen wird.

Außerdem enthält die Wii Remote einen Beschleunigungssensor, der die aktuell wirkende Beschleunigung auf die Wii Remote in die X-, Y- und Z-Richtung messen kann. Durch diesen Beschleunigungssensor ist es möglich, ausgeführte Bewegungen der Benutzer aufzunehmen und auszuwerten.

Auf Basis dieser Beschleunigungsdaten soll die Gestenerkennung aufbauen. Bei der Messung der Beschleunigungswerte spielt die Position der Wii Remote im Raum keine Rolle. Der Beschleunigungssensor ist innerhalb der Fernbedienung unter dem „A“-Knopf

angebracht und somit etwas nach vorne verlagert.

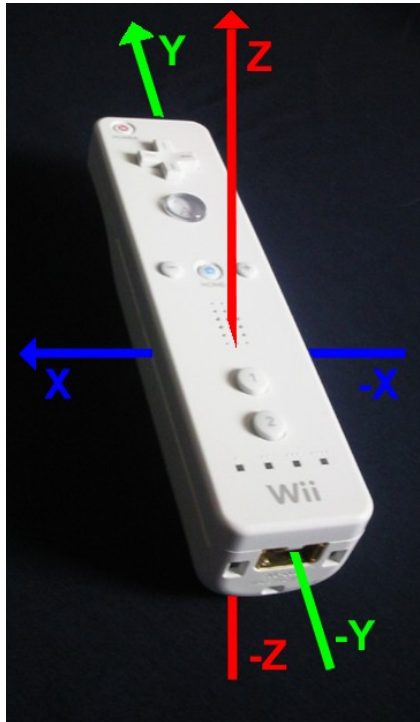


Abbildung 2: Achsenverlauf der  
Wii Remote

In Abbildung 2 ist zu sehen, wie die Achsen durch die Wii Remote verlaufen. Die X-Achse verläuft von rechts nach links quer durch die Wii Remote. Die Y-Achse verläuft von hinten nach vorne längs und die Z-Achse von unten nach oben durch die Wii Remote.

Die Kommunikation zwischen der Wii Remote und der Konsole erfolgt über die Bluetoothschnittstelle. Zur Entwicklung der Gestenerkennung wird die Wii Remote über die Bluetoothschnittstelle an einen PC angeschlossen.

## 3.2 **Eingesetzte Software**

Zur Entwicklung der Gestenerkennung wurde eine Bluetoothverbindung mit einem PC hergestellt, der über eine Bluetoothschnittstelle und die dazugehörigen Treiber verfügen muss. Im Folgenden wird die Software erläutert, die zur Erstellung der Gestenerkennung eingesetzt wurde.

### 3.2.1 **Bluetooth Umgebung „Microsoft Bluetooth Umgebung“**

Die zur Entwicklung der Gestenerkennung verwendete „Microsoft Bluetooth Umgebung“ ist bereits in Windows XP integriert. Diese verwaltet sowohl alle Bluetooth Geräte, die in Reichweite der Bluetoothschnittstelle sind, als auch die Dienste, die auf diesen Geräten zur

Verfügung stehen. Die Wii Remote verfügt nur über einen Dienst, der sämtliche Elemente der Wii Remote steuert.

### **3.2.2 Hilfsprogramm „Glove PIE“**

„Glove PIE“ ist ein freies Programm von Carl Kenner [4], mit dem man jede Anwendung auf dem PC mit beliebigen Eingabegeräten, wie beispielsweise der Tastatur oder einem Joystick, steuern kann. Es wurde auch ein Script für Glove PIE geschrieben, mit dessen Hilfe man die Wii Remote am PC steuern kann.

Glove PIE spielte hauptsächlich am Anfang der Entwicklung eine große Rolle. Dort wurde es ausschließlich eingesetzt, um erste Untersuchungen bezüglich dem Umgang mit der Wii Remote und dessen übermittelten Daten durchzuführen, da mit Glove PIE auch die aktuell gemessene Beschleunigung ausgegeben werden kann.

## **3.3 Entwicklungsumgebung**

Damit die Gestenerkennung in den Spielen der Firma Nurogames eingesetzt werden kann, wird eine Programmierumgebung vorausgesetzt, die den Anforderungen der Firma Nurogames gerecht wird. Hierbei wird ausschließlich die Programmiersprache C++ verwendet, die besonders wegen ihrer Geschwindigkeit und der Möglichkeit, hardwarenah zu programmieren, wohl in den meisten Spielengines eingesetzt wird. Außerdem bietet diese Programmiersprache eine ausgereifte Unterstützung für die objektorientierte Programmierung.

### **3.3.1 Schnittstellenbibliothek zur Wii Remote**

Für die Entwicklung der Gestenerkennung am PC wird eine C++ Bibliothek benötigt, die die Schnittstelle zur Wii Remote verwaltet und auf deren Dienste zugreift. Anders als Poppinga [1], der in seiner Arbeit selbst das Auslesen der Daten von der Wii Remote beschreibt, wurde hier die freie Bibliothek „wiuse“ von Michael Laforest eingesetzt [3], da diese Aufgabe später in den Spielen der Firma Nurogames eine lizenzierte Game-Engine übernehmen wird.

## **3.4 Beschleunigung**

Die Gestenerkennung wird auf Basis des Beschleunigungssensors in der Wii Remote entwickelt. Dieser misst die Beschleunigung in die X-, Y- und Z-Richtung.

Unter Beschleunigung versteht man die Änderung der Geschwindigkeit pro Zeitintervall und wird standardmäßig in Meter pro Sekunde zum Quadrat angegeben. Die Schnittstellenbibliothek „wiuse“ liefert umgerechnete Beschleunigungswerte in G-Kräften [3]. Diese stellen die erfahrene Beschleunigung auf die Wii Remote dar. Liegt die Wii Remote in

waagerechter Position auf einem Tisch, so sind die Werte der Beschleunigung für die X- und Y-Achse 0.0G. Die Werte für die Z-Achse 1.0G. Bewegt man nun beispielsweise die Wii Remote nach oben, so wird die Beschleunigung der Z-Achse größer als eins. Sie wird kleiner, wenn man die Wii Remote wieder abbremst. Sobald man wieder in der Ruheposition angekommen ist, wird sie wieder 1.0G.

### 3.5 Signale

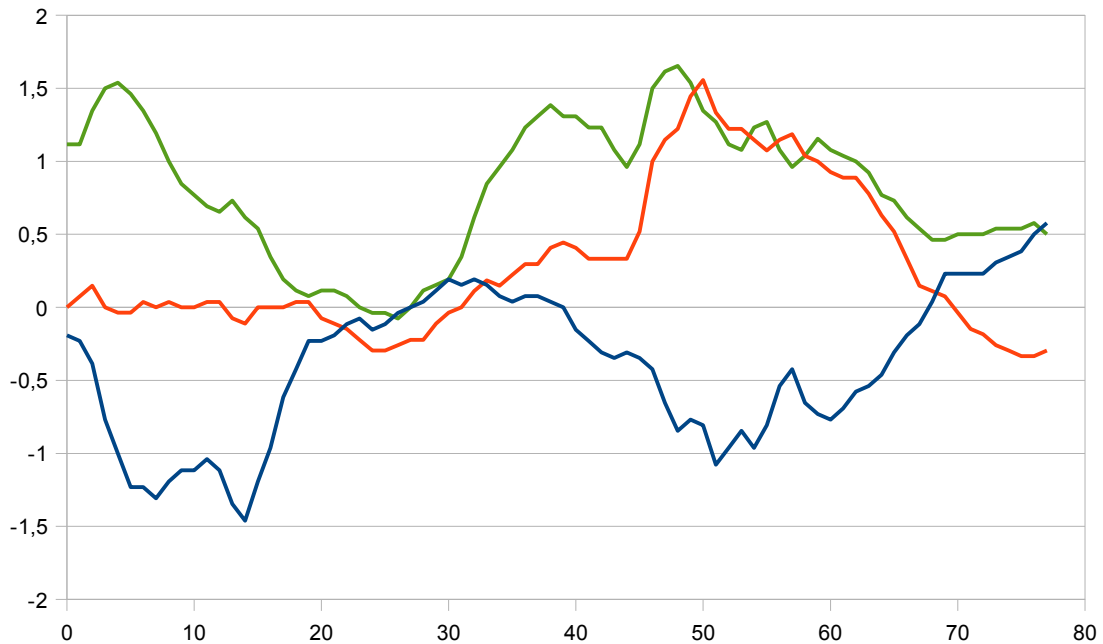


Abbildung 3: Signal einer Geste

Unter einem Signal versteht man einen Zeitverlauf, der zu jedem Zeitschritt Informationen abbildet. Die ausgelesenen Werte des Beschleunigungssensors über die Zeit bilden drei Signale für die X-, Y- und Z-Achse, wie in Abbildung 3 dargestellt ist. Da die Werte des Beschleunigungssensors nicht kontinuierlich, sondern nur mit einer bestimmten Abtastrate von dem Programm abgefragt werden, erhält man ein zeitdiskretes Signal. Die Beschleunigungswerte können jedoch jeden beliebigen Wert in dem Wertebereich des Beschleunigungssensors annehmen, wodurch ein wertkontinuierliches Signal entsteht.

## 4 Analyse der Gesten

Im folgenden Kapitel werden die aus der Anforderungsdefinition ersichtlichen Gesten grafisch ausgewertet und analysiert. Im Vergleich zu Poppinga [1] wird bei der zu entwickelnden Programmkomponente für die Firma Nurogames eine personenübergreifende Gestenerkennung angestrebt. Im Folgenden wird untersucht, in wieweit gleiche Gesten von verschiedenen Personen Ähnlichkeiten aufweisen und sich verschiedene Gesten unterscheiden. Hierbei soll festgestellt werden, ob eine automatisierte personenübergreifende Gestenerkennung überhaupt möglich ist.

### 4.1 Testpersonen

Zum analysieren der Gesten und zur späteren Evaluierung der Gestenerkennung wurden dreizehn Testpersonen ausgewählt, die ein möglichst breites Zielgruppenspektrum abdecken sollen, da die Gestenerkennung in den unterschiedlichsten Spielen für unterschiedliche Zielgruppen eingesetzt werden soll.

#	Alter	Geschlecht	Rechts-/Linkshänder	Wii
1	20	w	rechts	Ja
2	21	m	rechts	Ja
3	24	m	rechts	Ja
4	24	m	rechts	Nein
5	26	m	rechts	Nein
6	27	m	rechts	Nein
7	43	m	rechts	Nein
8	46	w	rechts	Nein
9	48	w	rechts	Nein
10	50	m	rechts	Nein
11	55	m	rechts	Nein
12	25	w	links	Nein
13	43	m	links	Nein

*Tabelle 1: Testpersonen*

In Tabelle 1 ist eine Auflistung der ausgewählten Testpersonen aufgeführt. Zwei der dreizehn ausgewählten Personen sind Linkshänder. Von den elf Rechtshändern besitzen drei eine eigene Wii und sind daher mit der Steuerung der Wii Remote bereits vertraut. Von den zwei Linkshändern besitzt keiner eine eigene Wii. Vier der dreizehn Testpersonen sind weiblich, von denen drei Rechtshänderinnen sind.

## 4.2 Kreis

Die erste Geste, die von den Testpersonen ausgeführt werden sollte, war die Kreisbewegung. Der Kreis sollte dabei im Uhrzeigersinn von unten beginnend ausgeführt werden.



Abbildung 4:  
Kreisbewegung

Abbildung 4 veranschaulicht die kreisförmige Bewegung im 2dimensionalen Raum. Beim Kreis sollte keine Bewegung in Y-Richtung, also nach vorne bzw. hinten, ausgeführt werden.

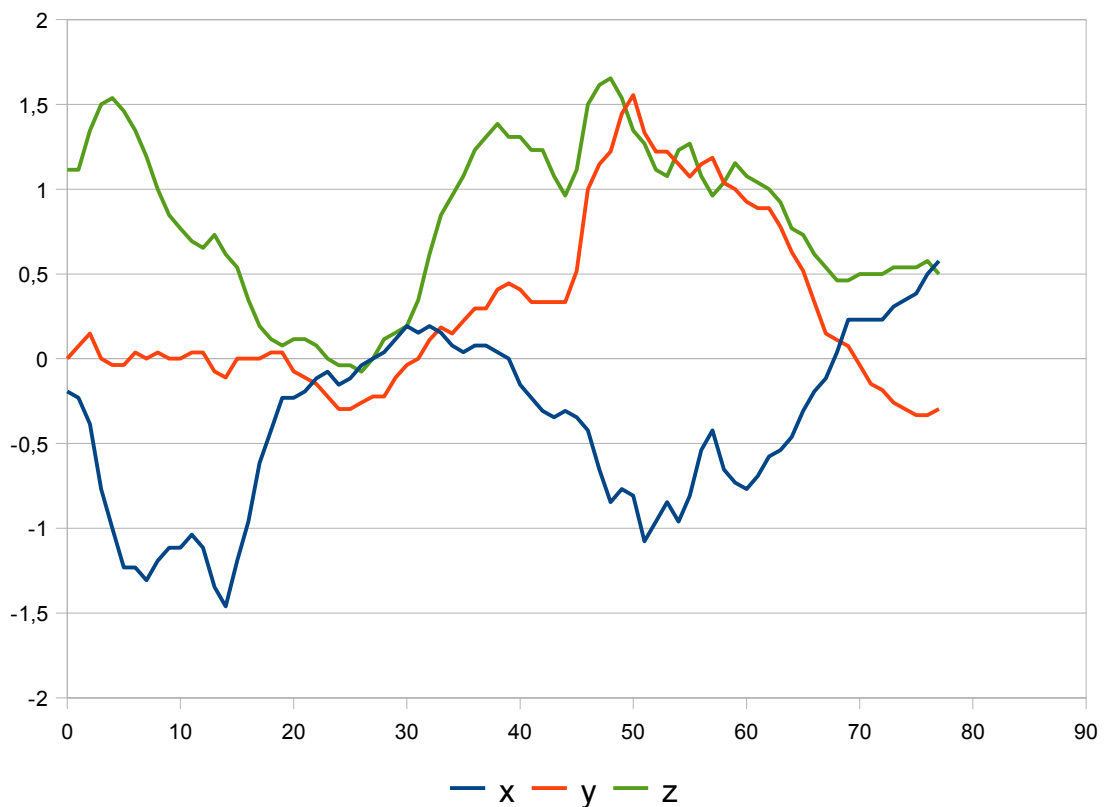


Abbildung 5: Kreis der Testperson 1



Abbildung 5 zeigt das Signal der Kreisbewegung der ersten Testperson, einer 20 jährigen Rechtshänderin mit eigener Wii. Sie führt die Geste in ungefähr 80 Zeitschritten aus und erreicht Beschleunigungen zwischen -1.5G und 1.5G.

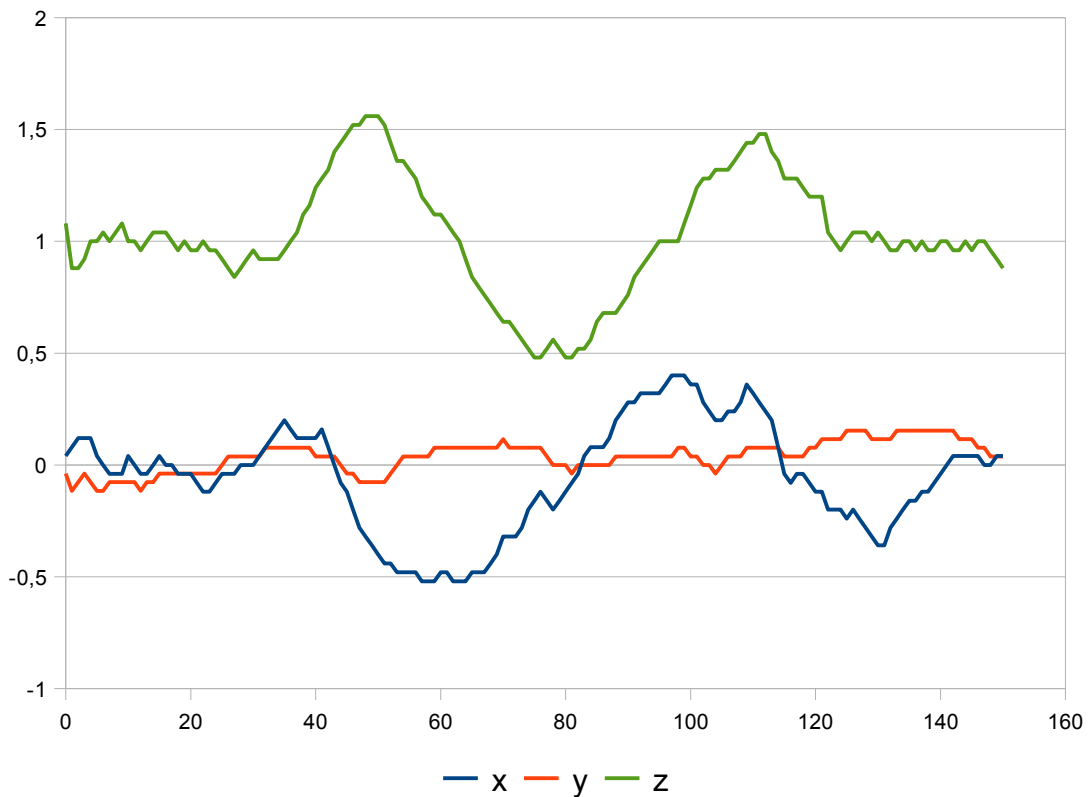


Abbildung 6: Kreis der Testperson 10

Abbildung 6 zeigt das Signal der Kreisbewegung der zehnten Testperson, einem 50 jährigen Rechtshänder ohne eigener Wii. Er benötigt zum Ausführen der Geste ungefähr 150 Zeitschritte und erreicht Beschleunigungen zwischen -0.5G und 1.5G.

Wie man sehen kann, unterscheiden sich beide deutlich in der Länge der Ausführungszeit, da die zehnte Testperson fast das Doppelte der Zeit benötigt, um den Kreis auszuführen. Im Wertebereich unterscheiden sich beide Testperson ebenfalls in der negativen Beschleunigung, da die erste Testperson den Kreis heftiger ausführte.

Im Verlauf der Beschleunigung weisen die beiden Gesten jedoch einige Übereinstimmungen auf. Bei beiden Gesten wird die Wii Remote zweimal negativ in X-Richtung beschleunigt. Dies macht Sinn, da bei der Kreisbewegung einmal zu Anfang die Wii Remote nach links bewegt wird und einmal am Ende, da der Start- und Endpunkt unten in der Mitte lag.

In der Y-Richtung unterscheiden sich die beiden Gesten. Bei der ersten Testperson erfährt die Wii Remote eine positive Beschleunigung in Y-Richtung. Bei der zehnten Testperson gar

keine. Normalerweise würde bei einem 2 Dimensionalen Kreis die Wii Remote nicht in Y-Richtung beschleunigt werden. Die positive Beschleunigung in Y-Richtung könnte sich dadurch erklären, dass die Testperson 1 den Kreis zum Ende hin nach vorne verschoben hat. Auch könnte die Wii Remote schräg gehalten worden sein, wodurch die Wii Remote ebenfalls in der Y-Richtung beschleunigt worden wäre.

In der Z-Richtung sieht man wieder eine Gemeinsamkeit. Beide Testpersonen beschleunigen die Wii Remote zunächst in eine positive Z-Richtung und anschließend in negative Z-Richtung. Da der Kreis unten begonnen werden sollte, stellt die positive Beschleunigung zunächst die Aufwärtsbewegung der ersten Kreishälfte dar und die negative Beschleunigung die Abwärtsbewegung der zweiten Kreishälfte. Die positive Beschleunigung am Ende stellt die Abbremsbewegung dar, bei der in entgegengesetzter Richtung beschleunigt wird oder der Kreis wurde weiter als  $360^\circ$  ausgeführt.

Bei der Kreisbewegung sollte es keine Rolle spielen, ob die Person Rechts- oder Linkshänder ist, da die Vorgaben, den Kreis im Uhrzeigersinn und unten beginnend auszuführen, für alle Personen gleich ist.

### 4.3 Frisbee

Die zweite Geste, die von den Testpersonen ausgeführt werden sollte, war der Frisbeewurf. Hierbei wird die Wii Remote vor dem Körper gehalten, anschließend in einer Vorwärtsbewegung vom Körper wegbewegt und am Ende eine Drehung nach rechts vollzogen.



Abbildung 7:  
Frisbeewurf

Abbildung 7 veranschaulicht den Frisbeewurf von oben gesehen. Hierbei könnte die Wii Remote, je nach Ansicht der Testperson, eventuell auch etwas nach oben beschleunigt werden. Dies ist wegen der schwierigen Darstellbarkeit hier nicht abgebildet.

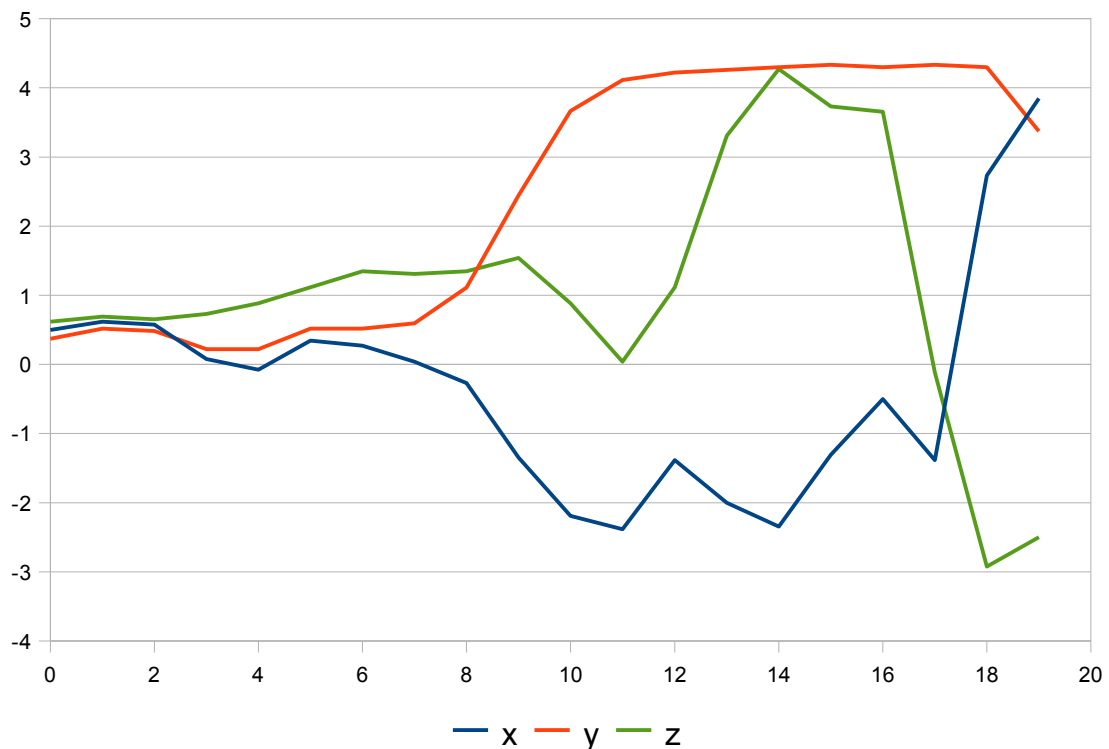


Abbildung 8: Frisbeewurf der Testperson 1

Abbildung 8 zeigt das Signal des Frisbeewurfs der ersten Testperson. Sie führt die Geste in ungefähr 20 Zeitschritten aus und erreicht Beschleunigungen zwischen  $-3.0G$  und  $4.5G$ .



Abbildung 9: Frisbeewurf der Testperson 10

Abbildung 9 zeigt das Signal des Frisbeewurfs der zehnten Testperson. Sie benötigte zum Ausführen der Geste ungefähr 80 Zeitschritte, wobei zu beachten ist, dass in den ersten 40 Zeitschritten keine Beschleunigungen zu registrieren sind und die Testperson somit wahrscheinlich die eigentliche Geste erst bei Zeitschritt 40 begonnen hat. Sie erreichte beim Ausführen der Geste Beschleunigungen zwischen  $-3.5G$  und  $4.5G$ .

Auch hier benötigt die zehnte Testperson ungefähr doppelt so lang wie die erste Person um die Geste auszuführen. Der Wertebereich ist diesmal bei beiden Testpersonen ungefähr gleich.

Auch im Verlauf der Signale kann man in jeder Achse Übereinstimmungen feststellen. Die X-Richtung wird negativ beschleunigt, da die Wii Remote beim Frisbeewurf durch Drehung des Handgelenks quer zum Körper mit der Spitze nach links gehalten wird. Beim Ausführen der Geste nach vorne wird dadurch die Wii Remote zur Seite und somit in X-Richtung bewegt.

Die Beschleunigung in Y-Richtung ist positiv, da die Wii Remote beim Frisbeewurf nach vorne gedreht wird.

Die Beschleunigung in Z-Richtung zeigt, dass die Personen den Frisbeewurf zuerst etwas

nach unten und anschließend nach oben hin ausgeführt haben und nicht auf einer Ebene gerade nach vorne.

Bei einem Linkshänder würde der Frisbeewurf spiegelverkehrt aussehen, da der Arm nach links weggedreht werden muss um den Frisbeewurf auszuführen. Das würde auch Änderungen bei den Beschleunigungen zur Folge haben, da die Wii Remote nun mit der Spitze nach rechts gehalten würde und somit eine positive Beschleunigung in X-Richtung erfolgen müsste. In den anderen beiden Richtungen müssten die Beschleunigungen ähnlich aussehen.

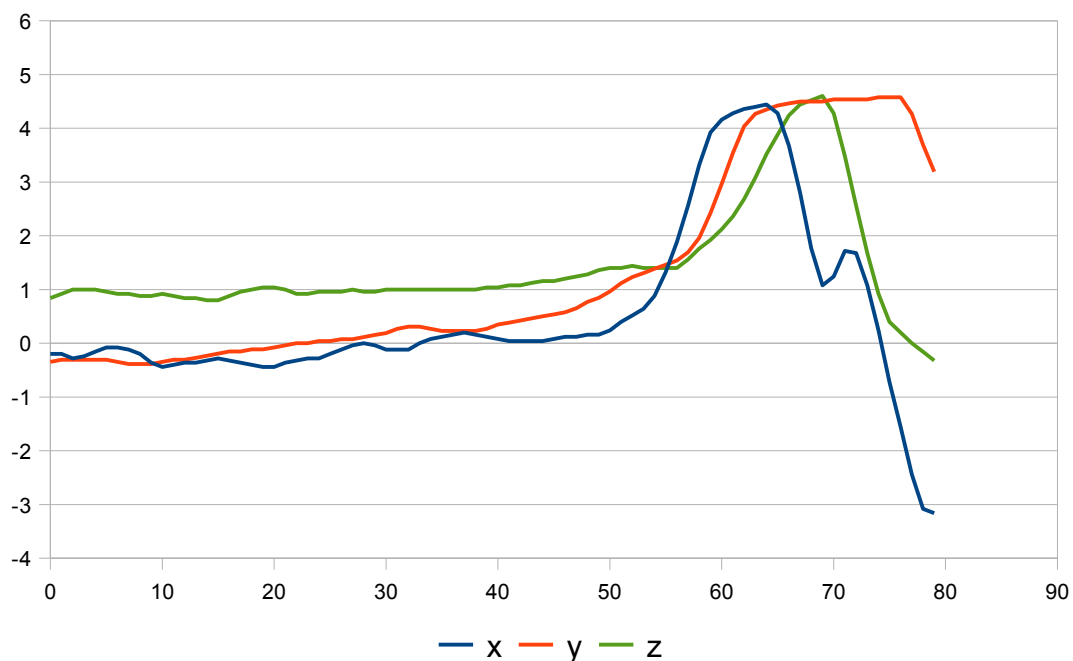


Abbildung 10: Frisbeewurf der Testperson 12

Abbildung 10 veranschaulicht den Frisbeewurf der zwölften Testperson, einer 25-jährigen Linkshänderin ohne eigener Wii und bestätigt die vorher genannten Vermutungen, da die X-Achse hier in positiver Richtung beschleunigt wird. Zur Ausführungszeit benötigt sie 80 Zeitschritte, wobei die ersten 40 wieder zu vernachlässigen sind. Auch in den Wertebereichen erreicht sie Beschleunigungen zwischen -3.0G und 4.5G, welche mit denen der Rechtshänder übereinstimmen.

Aufgrund der starken Verschiedenheit zwischen Rechts- und Linkshändern bei dieser Geste, müssten zwei unterschiedliche Gesten aufgenommen werden, die den Frisbeewurf sowohl für Rechts- wie auch für Linkshänder abbilden.

## 4.4 Schlag

Bei der dritten Geste handelt es sich um einen Schlag, den die Testpersonen von rechts nach links ausführen sollten. Dabei wird die Wii Remote zunächst rechts neben dem Körper gehalten. Anschließend wird eine Schlagbewegung nach vorne vor dem Körper ausgeführt.



*Abbildung 11:  
Schlagbewegung*

Abbildung 11 soll den Schlag von oben gesehen darstellen. Bei der Schlagbewegung wird die Wii Remote eventuell auch nach unten beschleunigt, was wegen der schwierigen Darstellbarkeit hier nicht abgebildet ist. Wie man sehen kann, hat diese Geste sehr viel Ähnlichkeit mit dem Frisbeewurf der Linkshänder, bei der ebenfalls eine Drehung mit der Wii Remote nach links vollzogen wird. Dementsprechend weist der Schlag der Linkshänder Ähnlichkeiten mit dem Frisbeewurf der Rechtshänder auf.

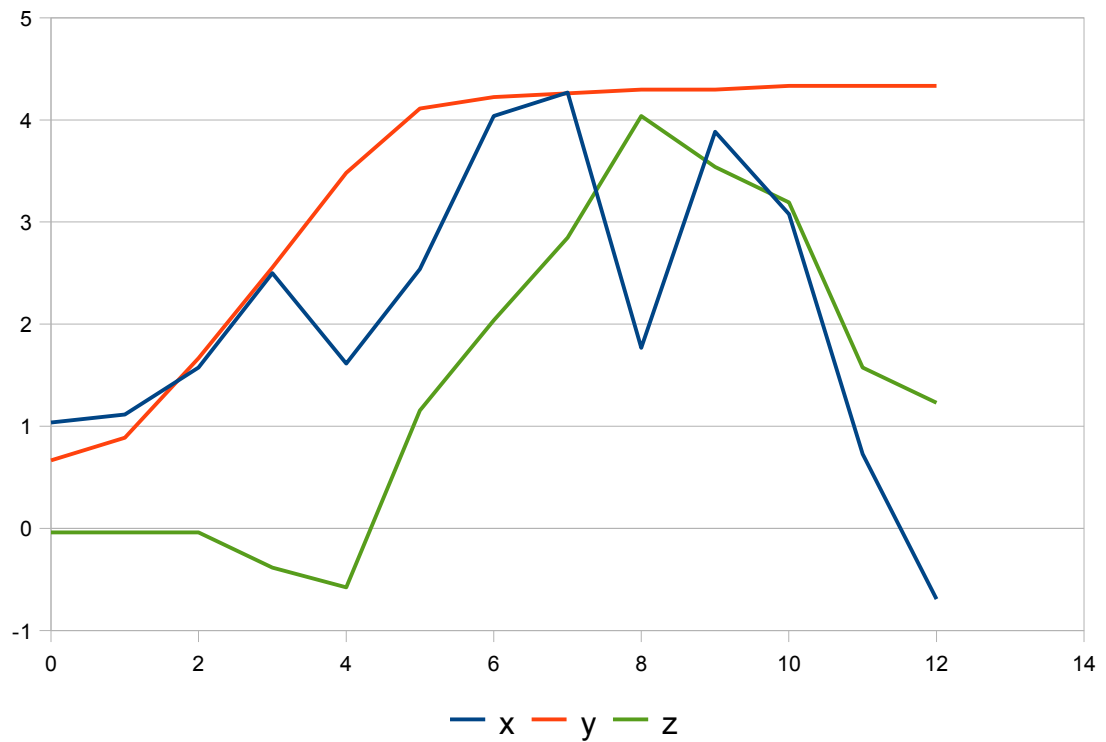


Abbildung 12: Schlag der Testperson 1

In Abbildung 12 sieht man den Schlag der ersten Testperson. Zum Ausführen der Geste benötigt sie nur 12 Zeitschritte und erreicht Beschleunigungen zwischen  $-1.0G$  und  $4.5G$ . In Abbildung 12 bestätigt sich, dass der Schlag der Rechtshänder mit dem Frisbeewurf der Linkshänder wie aus Abbildung 11 starke Ähnlichkeiten aufweist, da beide Signale eine positive Beschleunigung in allen drei Achsen aufweisen.

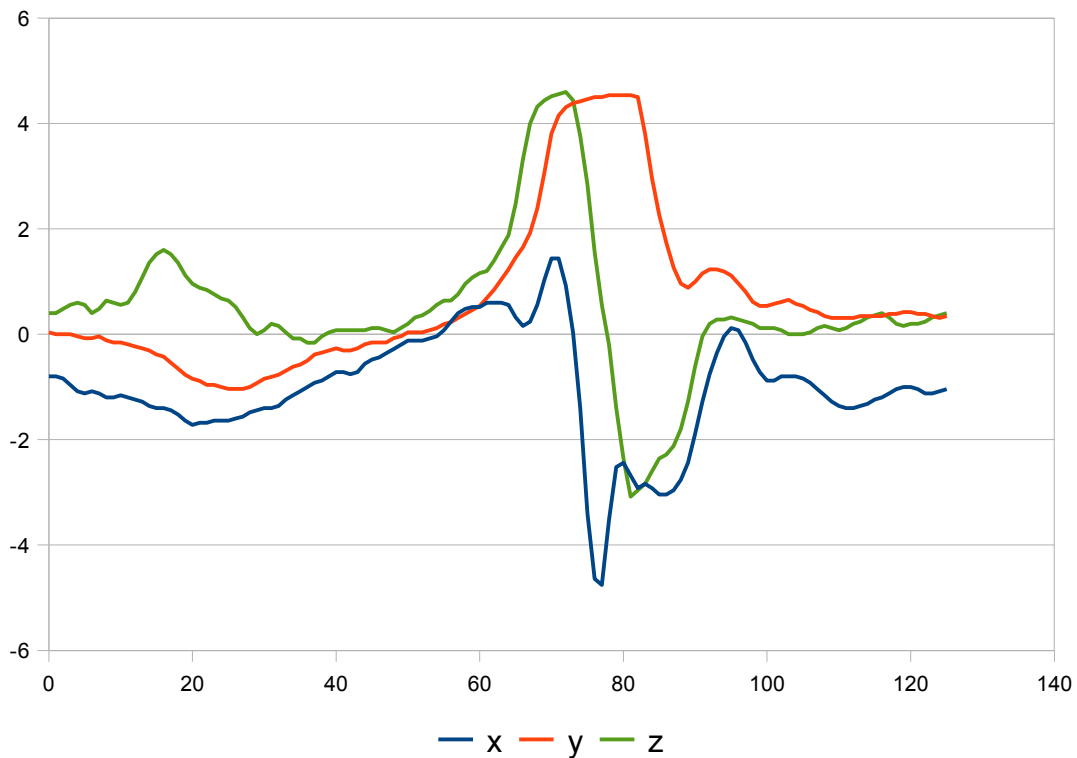


Abbildung 13: Schlag der Testperson 10

Abbildung 13 zeigt das Signal des Schlages von der zehnten Testperson. Er benötigt zum Ausführen der Geste ungefähr 120 Zeitschritte, wobei die ersten und letzten Zeitschritte wieder nicht zur eigentlichen Geste gehören. Beim Ausführen der Geste werden Beschleunigungen zwischen -5.0G und 4.5G erreicht.

Auch die Geste aus Abbildung 13 beschleunigt alle drei Achsen zunächst positiv. Die negativen Beschleunigungen am Ende fehlen bei der anderen Testperson in Abbildung 12 teilweise. Das könnte daran liegen, dass die Geste nicht bis zum Ende aufgenommen wurde, sondern die Aufnahme schon vorher abgebrochen wurde, was auch die kurze Ausführungszeit erklären würde.



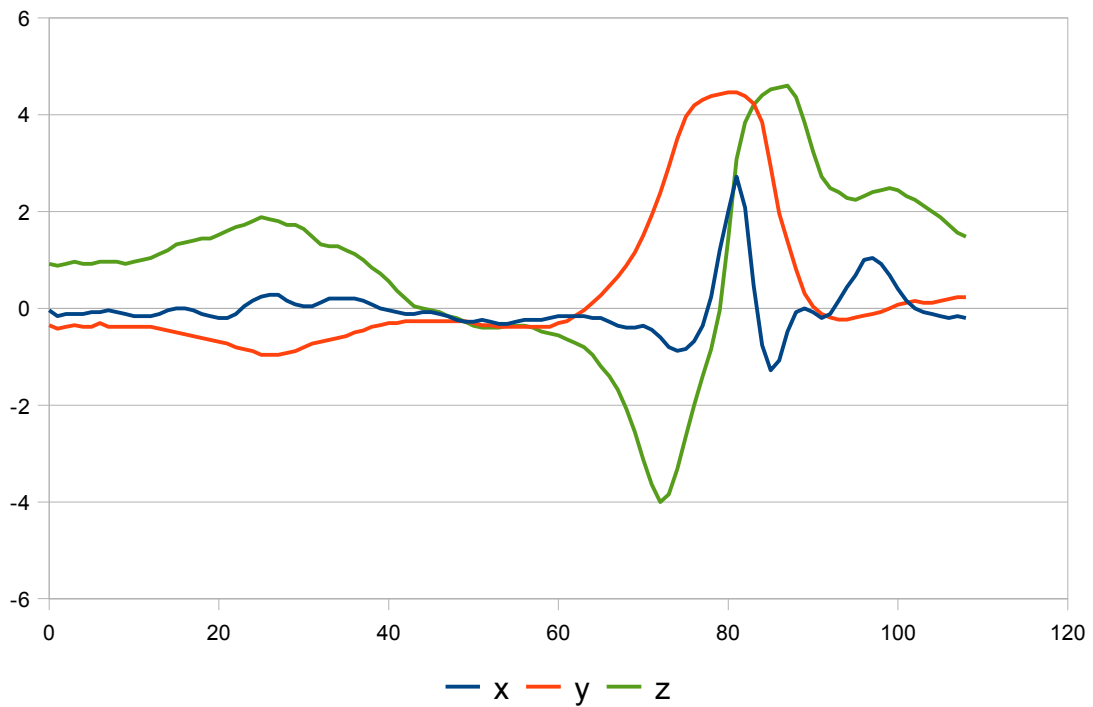


Abbildung 14: Schlag der Testperson 12

Abbildung 14 zeigt den Schlag der zwölften Testperson. Wie bereits angedeutet, wird hier ein ähnliches Signal wie beim Frisbeewurf der Rechtshänder, wie in Abbildung 8 und Abbildung 9 zu sehen, erzeugt.

## 4.5 Bowling

Bei der Geste Bowling soll von den Testpersonen eine Wurfbewegung ausgeführt werden, die beim Bowlingsport Anwendung findet. Dabei soll die Geste von hinten hinter dem Körper nach vorne in einer Aufwärtsbewegung ausgeführt werden.



Abbildung 15:  
Bowlingwurf

In Abbildung 15 ist der Bowlingwurf von der Seite dargestellt. Die X-Richtung ist hier nicht dargestellt. Dabei sollte die X-Richtung auch nur dann eine Beschleunigung erfahren, wenn die Person, die die Geste ausführt, die Wii Remote am Ende dreht oder den Arm nach links wegbewegt.

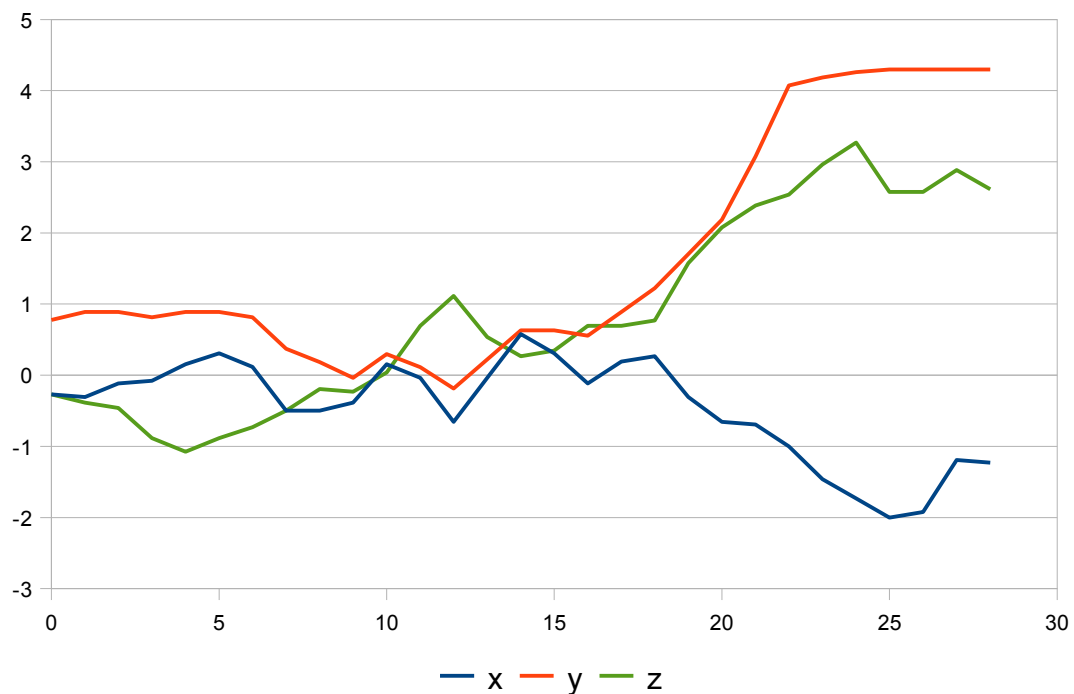


Abbildung 16: Bowlingwurf der Testperson 1

In Abbildung 16 ist der Bowlingwurf der ersten Testperson dargestellt. Zur Ausführung

wurden ungefähr 30 Zeitschritte benötigt und es wurden Beschleunigungen zwischen  $-2.0G$  und  $4.5G$  erreicht. Man sieht, dass diese Testperson die Wii Remote am Ende noch in die negative X-Richtung beschleunigt und somit die Wii Remote angedreht hat.

Man kann bei Abbildung 16 eine deutliche Ähnlichkeit zum Frisbeewurf derselben Person, der in Abbildung 8 zu sehen war, erkennen. Diese Ähnlichkeit kann man so erklären, dass der Frisbeewurf nicht gerade nach vorne ausgeführt wurde, sondern etwas schräg nach oben. Der Bowlingwurf hingegen wurde nicht gerade nach vorne ausgeführt, sondern etwas nach rechts vorne oder die Wii Remote wurde am Ende nach rechts gedreht. Hieraus kann man schließen, dass nicht perfekt ausgeführte Gesten leicht mit anderen Gesten verwechselt werden können.

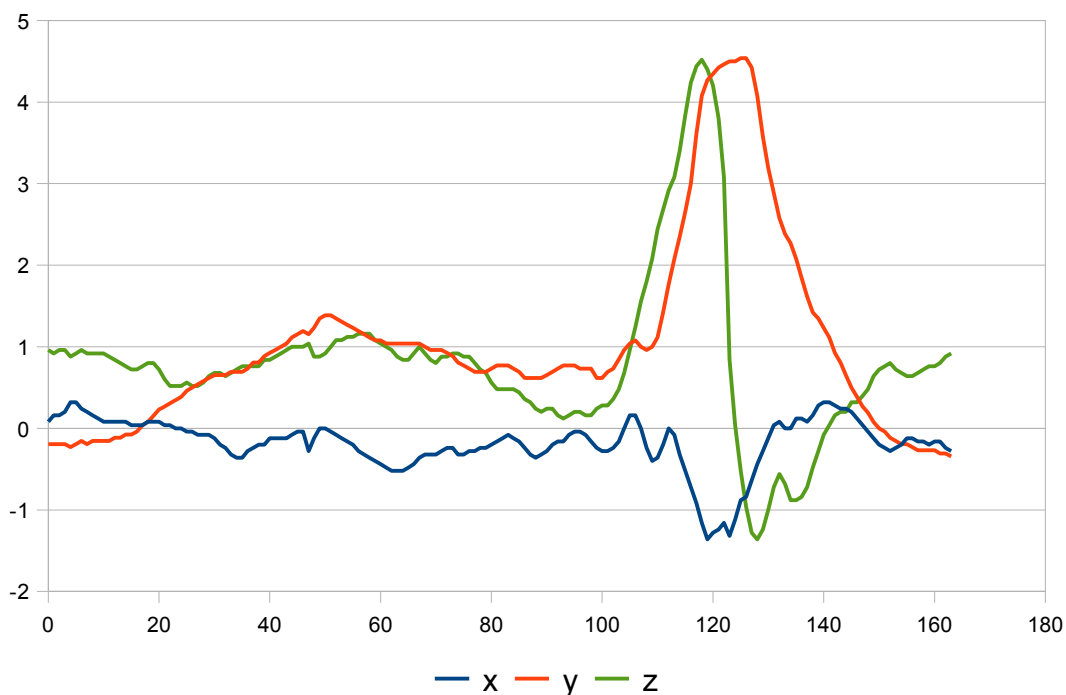


Abbildung 17: Bowlingwurf der Testperson 10

Abbildung 17 zeigt den Bowlingwurf der zehnten Testperson. Die beinahe ersten 100 Zeitschritte sind wieder nicht zu beachten, wodurch die Ausführungszeit ungefähr 70 Zeitschritte beträgt. Es werden Beschleunigungen zwischen  $-1.5G$  und  $4.5G$  erreicht.

Bei Linkshändern sollte sich der Bowlingwurf nicht von den Bowlingwürfen der Rechtshänder unterscheiden, bis auf die Drehung am Ende, die sowohl bei Rechts- wie auch bei Linkshändern sowohl nach rechts wie auch nach links ausfallen könnte.

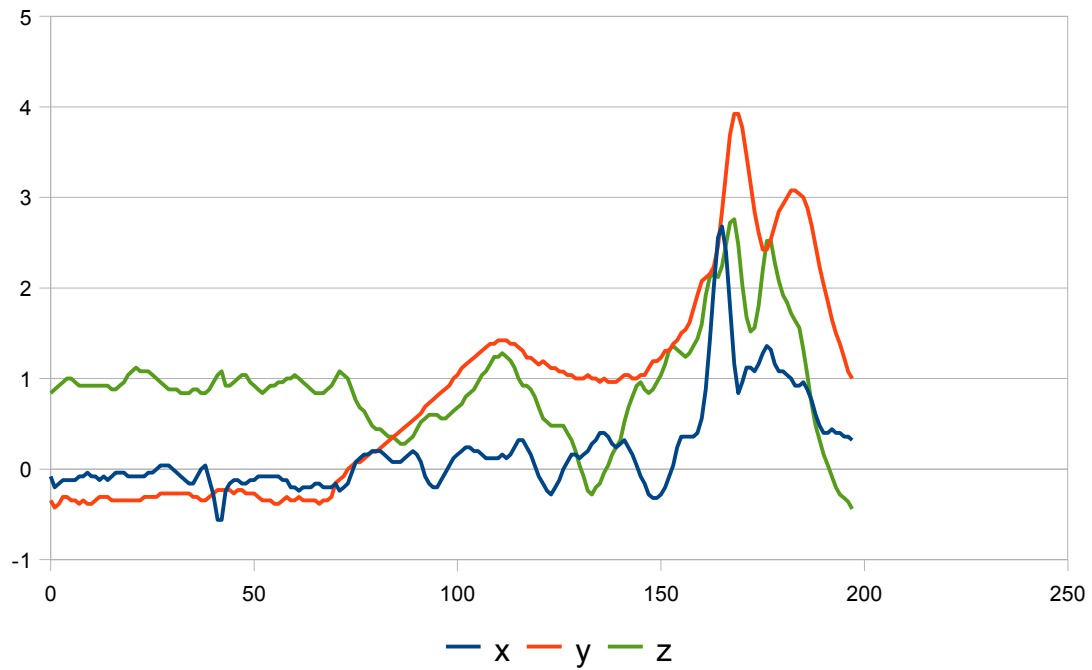


Abbildung 18: Bowlingwurf der Testperson 12

In Abbildung 18 sieht man den Bowlingwurf der zwölften Testperson. Die Beschleunigungen in Y- und Z-Richtung sind positiv wie bei den Rechtshändern. Der Unterschied liegt in der X-Achse, wo hier die Wii Remote nach links angedreht wurde, wohingegen die Rechtshänder die Wii Remote nach rechts gedreht haben.

Auch hier ist eine Ähnlichkeit zu einer anderen Geste derselben Person festzustellen. Der Bowlingwurf hat denselben Verlauf wie der Schlag aus Abbildung 14.

## 4.6 Überkopfwurf

Beim Überkopfwurf soll die Wii Remote zunächst rechts hinter dem Kopf gehalten werden. Anschließend soll eine Wurfbewegung nach vorne ausgeführt werden, wie zum Beispiel beim Handball.



Abbildung 19:  
Überkopfwurf

In Abbildung 19 ist der Überkopfwurf von der Seite dargestellt. Der Überkopfwurf ist die komplexeste Geste, da hierbei die Wii Remote auch noch durch eine Handgelenks- oder Unterarmdrehung ebenfalls um sich selbst gedreht werden kann und somit eine Vielzahl von Möglichkeiten der Ausführung der Geste gegeben ist.

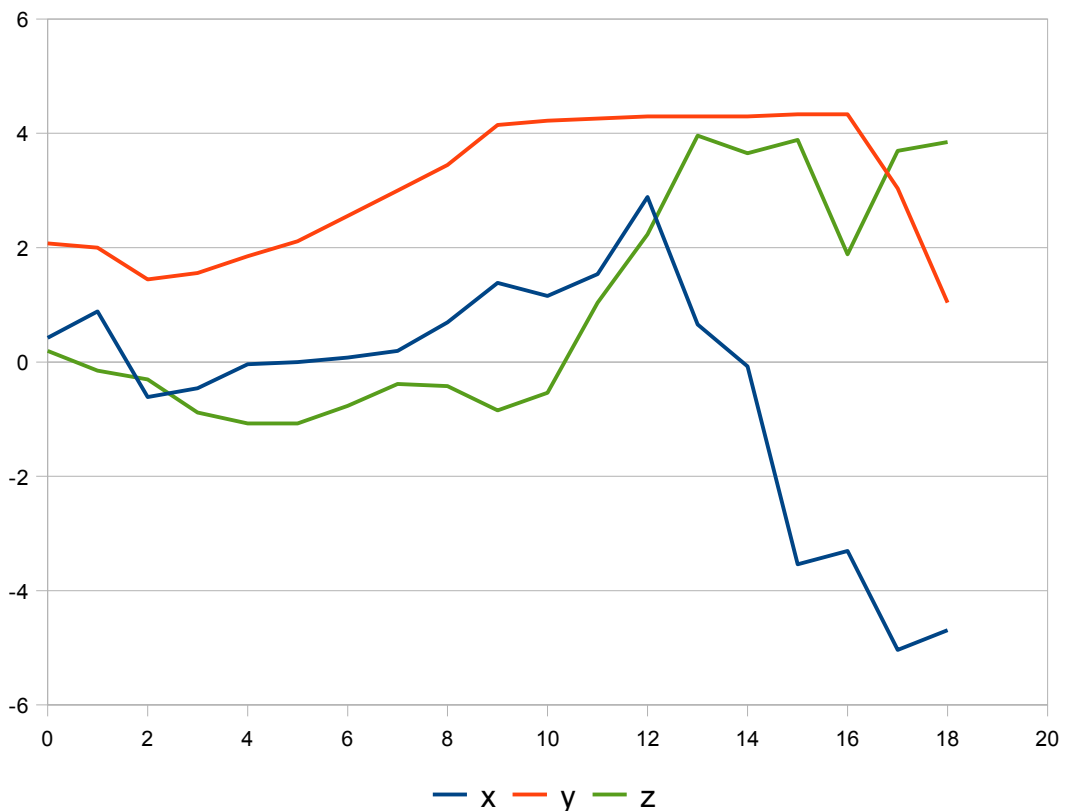


Abbildung 20: Überkopfwurf der Testperson 1

In der Abbildung 20 ist der Überkopfwurf der ersten Testperson dargestellt. Zum Ausführen der Geste wurden ungefähr 20 Zeitschritte benötigt. Es werden Beschleunigungen zwischen -5.0G und 4.5G erreicht.

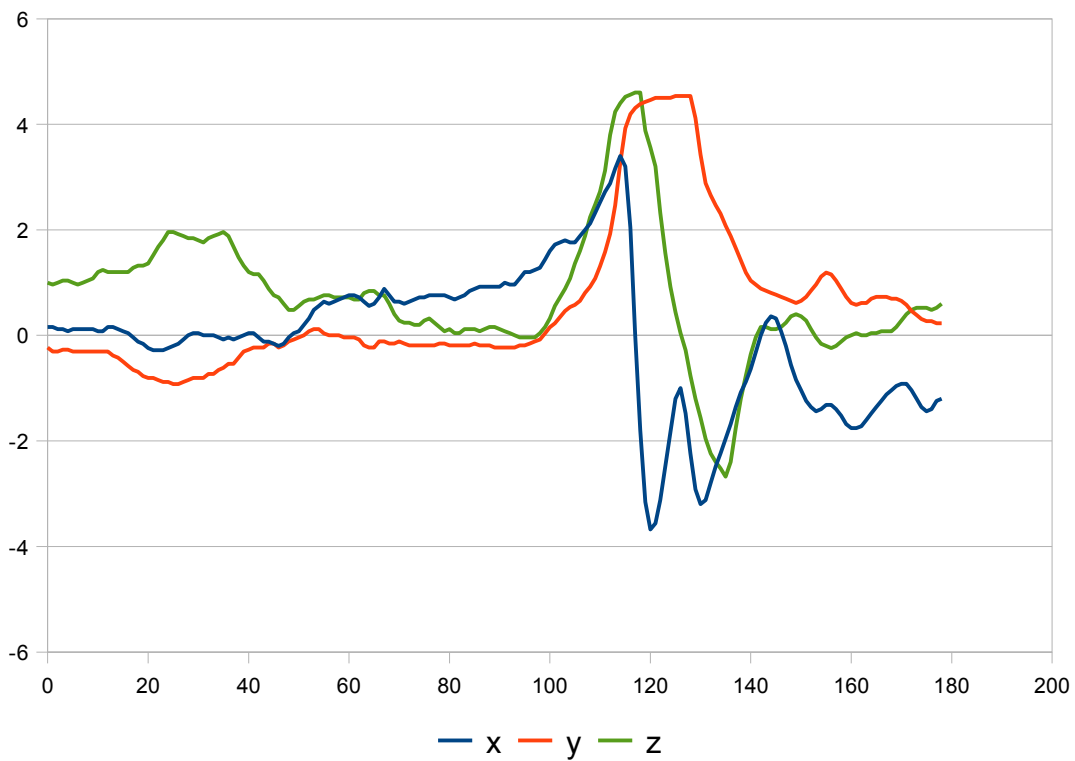


Abbildung 21: Überkopfwurf der Testperson 10

In Abbildung 21 ist der Überkopfwurf der zehnten Testperson dargestellt. Zum Ausführen der Geste wurden ungefähr 80 Zeitschritte benötigt. Es wird eine Beschleunigung zwischen -4.0G und 4.5G erreicht.

Wie man sehen kann, wird bei beiden Gesten eine starke Beschleunigung in X-Richtung erreicht, zunächst in positive Richtung, anschließend in negative, was auf eine starke Drehung der Wii Remote bei der Ausführung des Überkopfwurfes zurückzuführen ist.

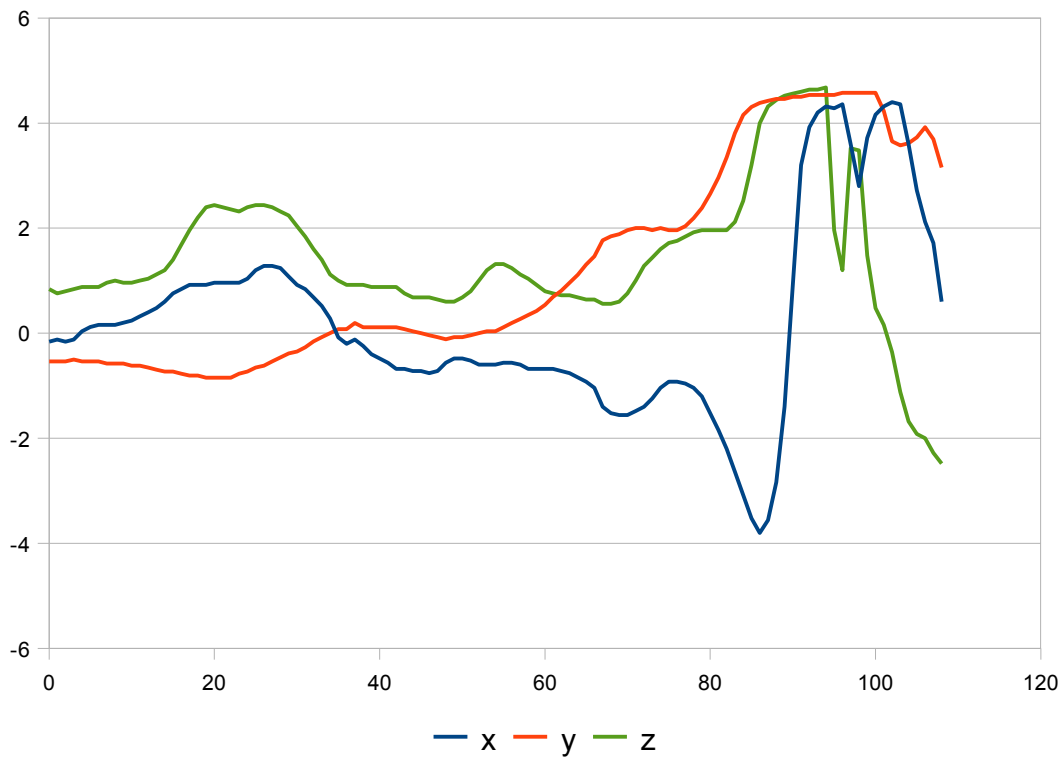


Abbildung 22: Überkopfwurf der Testperson 12

Abbildung 22 zeigt den Überkopfwurf der zwölften Testperson. Auch hier werden sowohl die X- wie auch die Z-Richtung positiv beschleunigt. Der Unterschied liegt in der X-Richtung, da hier zunächst eine negative und anschließend eine positive Beschleunigung erreicht wird, wohingegen es bei beiden Rechtshändern andersherum war.

## **4.7 Fazit der Analyse**

Durch die Analyse der Gesten ist deutlich geworden, dass gleiche Gesten von unterschiedlichen Personen durchaus Übereinstimmungen im Verlauf der Signale besitzen.

Sie unterscheiden sich allerdings stark in der Ausführungszeit und in den erreichten Wertebereichen, wodurch eine Normalisierung der Gesten erforderlich wird.

Es wurde auch festgestellt, dass manche Gesten Ähnlichkeiten mit anderen Gesten aufweisen, was eine eindeutige Gestenerkennung schwierig macht.

Außerdem werden einige Gesten von Linkshändern anders ausgeführt, wodurch es nötig ist, die gleiche Geste sowohl für Rechts- wie auch für Linkshänder getrennt zu erstellen. Dabei tritt das weitere Problem auf, dass manche Gesten der Linkshänder so aussehen, wie andere Gesten der Rechtshänder. Das hat zur Folge, dass vor der Gestenerkennung bestimmt werden muss, ob der Ausführende der Gesten Rechts- bzw. Linkshänder ist.

Abschließend kann man aus den Analysen den Wertebereich ablesen, der zwischen -5.0G und 5.0G liegt. Zeitmessungen bei den Ausführungen der Testpersonen haben ergeben, dass bei den Gesten ungefähr 33 Werte pro Sekunde eingelesen werden und die Frequenz der Signale somit 33 Hz beträgt.

Des Weiteren kann man aus der Analyse der Gesten erkennen, dass die Testpersonen für die Ausführung des Kreises sehr viel mehr Zeit benötigten und nicht so hohe Beschleunigungen erreicht haben, als bei den anderen Gesten. Daraus kann man herleiten, dass alle Personen zum Beispiel beim Frisbeewurf die Wii Remote stark beschleunigen um diese weit zu werfen. Dieses Verhalten liegt am Vorwissen der Personen, dass eine Frisbee beim Werfen stark beschleunigt werden muss, damit diese überhaupt in die Luft fliegt. Genauso ist es bei den anderen Würfeln. Dahin gegen wurde der Kreis langsam in die Luft gezeichnet, da hierbei nichts geworfen wird und somit keine starke Beschleunigung notwendig ist.



## 5 Gestenerkennung

Die Gestenerkennung ist ein Mustererkennungsverfahren. Bei der Mustererkennung sollen in den gewonnenen Signalen Ähnlichkeiten und Muster festgestellt werden um diese so automatisch in Kategorien einzuteilen. Man unterscheidet heutzutage zwischen drei Ansätzen der Mustererkennung: syntaktische, statistische und strukturelle Mustererkennung.

Bei der **syntaktischen Mustererkennung** werden die zu vergleichenden Objekte durch Folgen von Symbolen beschrieben, so dass Objekte der gleichen Kategorie die gleichen Beschreibungen aufweisen.

Bei der **statistischen Mustererkennung** werden für die Objekte Wahrscheinlichkeiten bestimmt. Die Wahrscheinlichkeit gibt an, ob das Objekt eher der einen oder der anderen Kategorie angehört. Das Objekt wird dann in die Kategorie mit der höchsten Wahrscheinlichkeit eingeordnet.

Die **strukturelle Mustererkennung** fasst die syntaktische und die statistische Mustererkennung in einem Verfahren zusammen, indem zunächst Merkmale für die Objekte festgelegt werden. Anschließend führen diese dann durch statistische Verfahren zu einer Zuordnung des Objektes zu einer Kategorie.

Bei der im Rahmen dieser Diplomarbeit entwickelten Gestenerkennung wird die statistische Mustererkennung eingesetzt. Poppinga benutzt in seiner Arbeit die strukturelle Mustererkennung, da er zunächst Merkmale festlegt und die Gesten dann durch ein statistisches Verfahren den Referenzgesten zugeordnet werden [1].

### 5.1 Schritte der Gestenerkennung

Der Prozess der Gestenerkennung lässt sich in mehrere Teilschritte zerlegen. Bevor Gesten erkannt werden können, müssen Signale aufgenommen und gespeichert werden. Diese stellen die Referenzsignale der Gesten dar. Anschließend findet eine Vorverarbeitung statt, bei der die Referenzsignale normiert werden, wobei eine Zeit- und eine Wertnormierung stattfindet. Wird nun im Erkennungsprozess eine Geste ausgeführt, wird diese zunächst ebenfalls normiert und anschließend durch ein Klassifikationsverfahren einem der Referenzsignale zugeordnet.

#### 5.1.1 Lernprozess

Als erstes müssen Signale aufgenommen und gespeichert werden. Diese bilden später die Referenzen für die Gestenerkennung, denen die zu erkennenden Gesten zugeordnet werden sollen. Wird eine gleiche Geste mehrmals aufgenommen, so wird daraus eine einzige

Referenzgeste erstellt, indem für jeden Zeitschritt und für jede Achse der Mittelwert gebildet wird. Die Mittelwerte ergeben dann das Referenzsignal.

### 5.1.2 Vorverarbeitung

Da die Gesten, wie in Kapitel 4 untersucht wurde, von unterschiedlichen Personen unterschiedlich ausgeführt werden, zum Beispiel mal schneller und mal langsamer oder mal etwas heftiger und mal etwas schwächer, müssen die Gesten normiert werden. Es muss sowohl eine Zeitnormierung, für schnellere und langsamere Ausführung, sowie eine Wertnormierung für unterschiedliche Heftigkeiten der Gesten erfolgen. Die zu erkennende Geste wird, genau wie die Referenzgesten, ebenfalls normiert. Die Normierung der Zeit wird so durchgeführt, dass immer nur eine bestimmte Anzahl an Werten aus dem Signal entnommen wird.

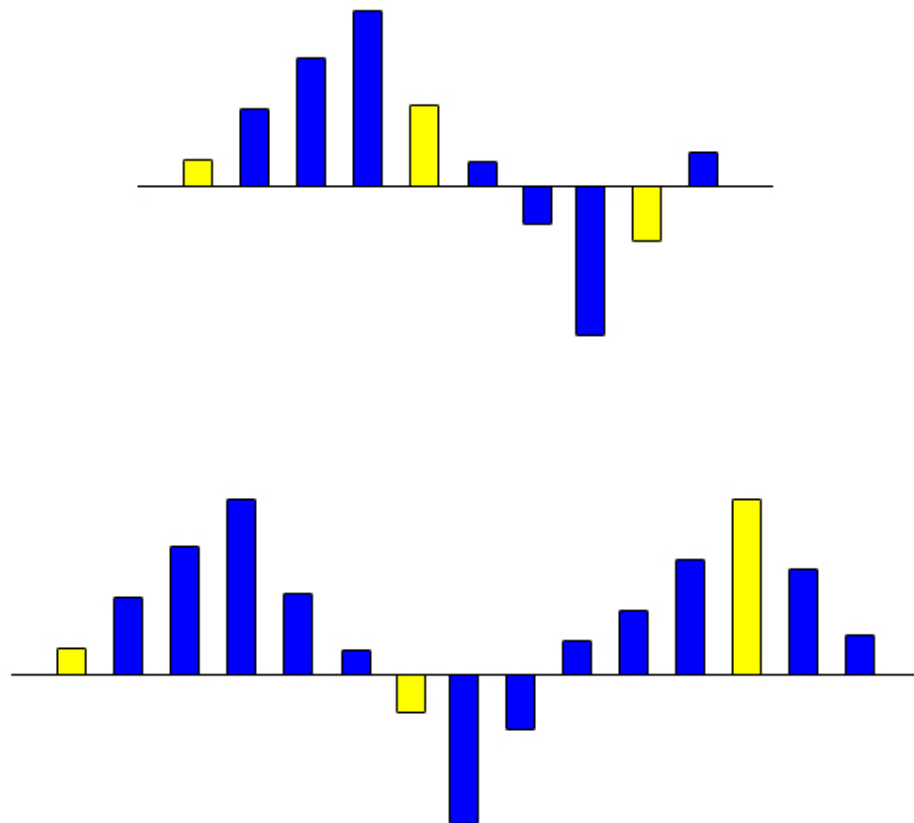


Abbildung 23: Zeitnormierung

In Abbildung 23 ist ein Beispiel dargestellt, wie aus zwei unterschiedlich langen Gesten, jeweils nur drei Werte abgetastet werden. Aus der ersten zehn Zeitschritt langen Geste wird jeder vierte Wert abgetastet und aus der zweiten fünfzehn Zeitschritt langen Geste, wird

jeder sechste Wert abgetastet. Diese Abtastrate berechnet sich, indem man die Gesamtzahl der Zeitschritte der Geste durch die Zahl der Werte, die man entnehmen möchte, teilt und das Ergebnis rundet. Wie wir in Kapitel 4 gesehen haben, gibt es Ausführungen von Gesten, die nur zwölf Zeitschritte lang sind. Da die Anzahl der Werte, die wir der Geste entnehmen möchten kleiner als die Geste sein muss, wählen wir für unsere Normierung eine Schrittzahl von zehn.

Die Normierung der Werte wird mit

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \cdot (norm_{max} - norm_{min}) + norm_{min}$$

berechnet, wobei

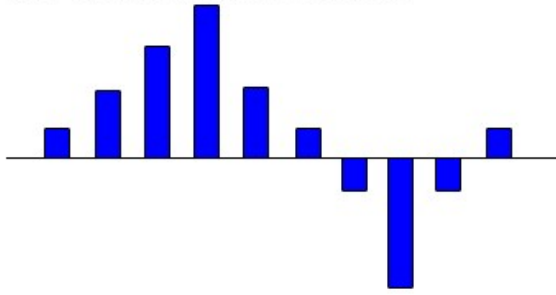
$$\begin{aligned} norm_{max} &= 1 \\ norm_{min} &= -1 \end{aligned}$$

ist, da die Werte zwischen 1 und -1 normiert werden sollen.

### 5.1.3 Klassifikation

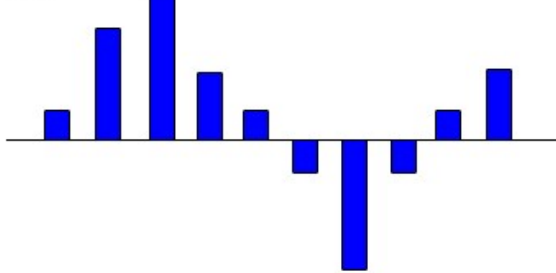
In der nachfolgenden Abbildung 24 ist ein Beispiel für eine Geste zu sehen, die erkannt werden soll. Zur beispielhaften einfachen Darstellung wurde nur eine Achse betrachtet. Die Referenzen bestehen dabei aus den beiden Gesten (A) und (B). Alle drei Gesten sind bereits auf zehn Zeitschritte normiert und besitzen aus Gründen der beispielhaften Berechnung einen Wertintervall zwischen -4 und 4. Wie man erkennen kann, besteht eine große Ähnlichkeit zwischen der zu klassifizierenden Geste und der Referenzgeste (A).

### Zu erkennende Geste



### Referenzgesten

(A)



(B)

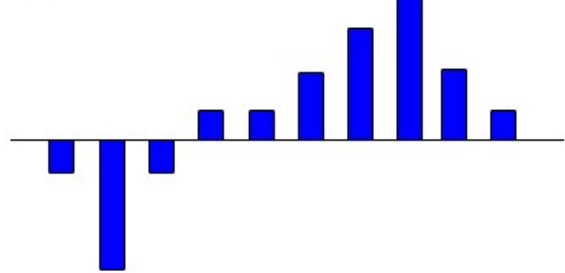
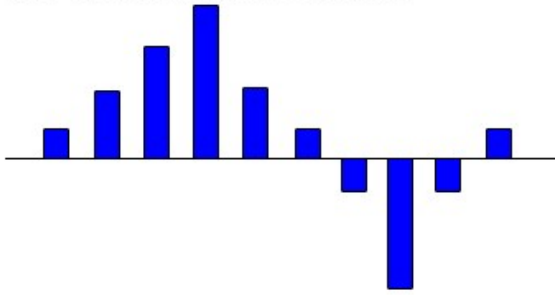


Abbildung 24: Klassifikation Schritt 1

In der nachfolgenden Abbildung 25 wurden nun die Differenzen für jeden einzelnen Zeitschritt zwischen der zu klassifizierenden Geste und den beiden Referenzgesten gebildet. Summiert man diese Werte auf, so bekommt man bereits das Ergebnis, dass die zu klassifizierende Geste zu der Referenzgeste (A) eine bessere Übereinstimmung hat.

## Zu erkennende Geste



## Referenzgesten

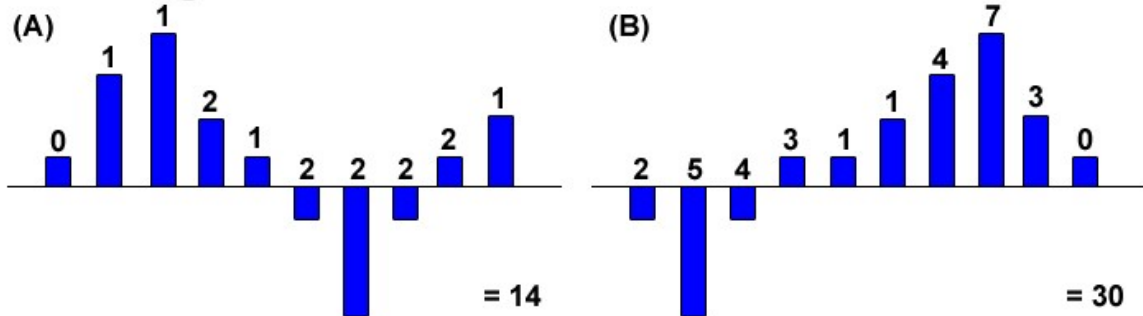
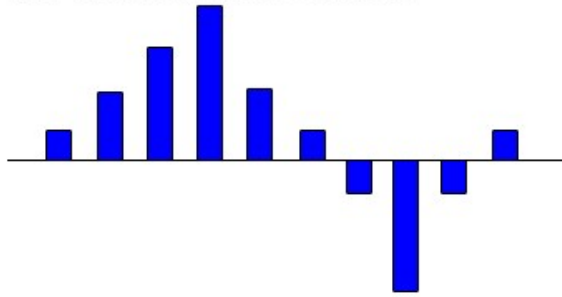


Abbildung 25: Klassifikation Schritt 2

Die Summen aus Abbildung 25 liegen aber noch relativ dicht beieinander. Um die großen Abstände mehr ins Gewicht fallen zu lassen, wird nun zu jedem Wert das Quadrat gebildet und anschließend aufsummiert, wie in Abbildung 26 zu sehen ist. Nun ist eindeutig zu sehen, dass die zu klassifizierende Geste größere Übereinstimmungen zu der Referenzgeste (A) als zur Referenzgeste (B) besitzt.

### Zu erkennende Geste



### Referenzgesten

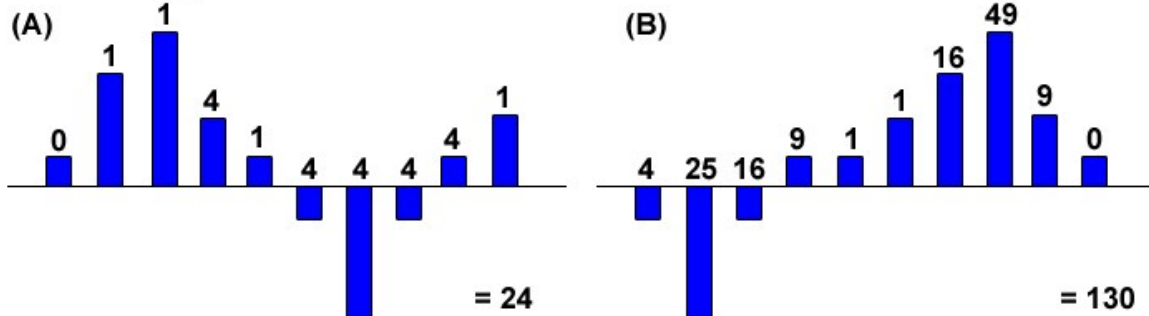


Abbildung 26: Klassifikation Schritt 3

Diese durchgeführte Art der Klassifikation nennt man Abstandsklassifikation, da der Abstand zwischen der ausgeführten Geste und allen Referenzgesten gebildet wird. Die niedrigste Summe der Abstände der gesamten Geste wird dann der ausgeführten Geste zugeordnet.

Da wir bei ausgeführten Gesten mit der Wii Remote drei Achsen betrachten wollen, wird der gesamte Abstand einer Geste mit dem euklidischen Abstand

$$d(x, y, z) = \sum_{t=0}^{t_{max}} \sqrt{(x_t - x_{t_{ref}})^2 + (y_t - y_{t_{ref}})^2 + (z_t - z_{t_{ref}})^2}$$

berechnet.

## 5.2 Automatische Erkennung der Gestenausführung

Aus der Anforderungsdefinition in Kapitel 2 wird ersichtlich, dass die Gestenerkennung automatisch erkennen muss, wann ein Benutzer eine Geste angefangen und wieder beendet hat. Bei der bisherigen Aufnahme der Gesten musste während der Ausführung ein Knopf an der Wii Remote vom Benutzer festgehalten werden.

Da die Beschleunigung der Wii Remote in der Ruheposition, wie in Kapitel 3.4 beschrieben,

0.0G in X- und Y-Richtung bzw. 1.0G in Z-Richtung beträgt, wird ein Schwellwert in positiver und ein Schwellwert in negativer Richtung festgelegt. Wird dieser Schwellwert von nur einer Achse in positiver Richtung überschritten bzw. in negativer Richtung unterschritten, so soll dies als Beginn der Geste definiert sein. Die Geste gilt als beendet, wenn die Beschleunigung aller drei Achsen für einen gewissen Zeitraum wieder zwischen diesen Schwellwerten liegt.

Aus den Abbildungen in Kapitel 4 lässt sich ein Schwellwert von 0.5G bzw. -0.5G für die X- und Y-Richtung sowie von 1.5G bzw. 0.5G für die Z-Richtung ablesen. Nach zehn Zeitschritten zwischen diesen Schwellwerten soll die Geste als beendet gelten. Diese letzten zehn Zeitschritte werden anschließend wieder aus der Geste entfernt, da hier keine großen Veränderungen in der Beschleunigung mehr stattgefunden haben.

## 6 Die Programmkomponente „Wiinput“

Die zu entwickelnde Programmkomponente „Wiinput“ soll in der Firma Nurogames zur Gestenerkennung eingesetzt werden. Um eine reibungslose Integration der Programmkomponente in die interne Spiele-Engine zu gewährleisten, wird zur Entwicklung der Programmkomponenten die Programmiersprache C++ verwendet. Diese hat sich bereits in der Firma Nurogames zur Spieleentwicklung bewährt und durchgesetzt. Zur Einbindung in die Spieleengine, muss ein Interface erstellt werden, dass die Funktionen zum Aufnehmen neuer Gesten wie auch zum Erkennen der bereits verfügbaren Gesten zur Verfügung stellt.

### 6.1 Interface

Wiinput
Wiinput()
~Wiinput()
bool: startRecording()
bool: stopRecording()
bool: startMatching()
bool: stopMatching()
int: iterate(int: wiimote, float: x, float: y, float: z, bool: button)
bool: saveGesture(int: wiimote, string: name)
string: matchedGesture(int: wiimote)
void: matchStrategy(MatchStrategy*: matchStrategy)
bool: saveGesturesPersistent()
bool: loadPersistentGestures()

Abbildung 27: Interface der Klasse „Wiinput“

In Abbildung 27 ist das Interface der Programmkomponente „Wiinput“ dargestellt. Um die Programmkomponente benutzen zu können, wird ein Objekt der Klasse „Wiinput“ benötigt.

Die beiden Methoden „startRecording“ bzw. „startMatching“ starten den Prozess des



Aufnehmens neuer Gesten bzw. das Erkennen der bereits verfügbaren Gesten. Mit „*stopRecording*“ bzw. „*stopMatching*“ werden diese Prozesse wieder beendet.

Die Methode „*iterate*“ muss in jeder Iteration der Hauptschleife aufgerufen werden. Diese erwartet neben den drei Werten für die Beschleunigung in X-, Y- und Z-Richtung, die Nummer der Wii Remote, zu der die Beschleunigung gehört, so wie den Wahrheitswert, ob der Knopf zum Aufnehmen der Geste auf der Wii Remote gedrückt ist.

Die Rückgabe der Methode gibt an, ob gerade das Ende einer Geste erkannt wurde. Ist dies der Fall, so muss beim Lernprozess die Methode „*saveGesture*“ mit dem Namen, der der Geste zugewiesen werden soll, und der Nummer der zugehörigen Wii Remote aufgerufen werden. Existiert der Name bereits, so wird die neue Geste der bekannten Geste hinzugefügt, andernfalls wird eine neue Geste angelegt.

Beim Erkennungsprozess muss die Methode „*matchedGesture*“, ebenfalls mit der Nummer der Wii Remote, aufgerufen werden. Diese gibt den Namen der gefundenen Geste zurück, die für die ausgeführte Geste erkannt wurde.

Durch die übergebenen Nummern der Wii Remote kann eine Geste einer entsprechenden Wii Remote zugeordnet werden. Dies ist für einen eventuellen Mehrspielermodus wichtig, bei dem die Gesten jeder Wii Remote getrennt voneinander erkannt werden müssen. Bei der Aufnahme der Gesten spielt es keine Rolle, mit welcher angeschlossenen Wii Remote die Gesten ausgeführt werden.

Über das Interface aus Abbildung 27 wurde im Rahmen dieser Diplomarbeit eine konsolenbasierte Benutzeroberfläche entworfen, die zu Demonstrations- und Testzwecken dient.

Die Methode „*matchStrategy*“, die einen Pointer auf ein Objekt der Klasse „*MatchStrategy*“ erwartet, wählt ein anderes Verfahren aus, dass zur Gestenerkennung eingesetzt werden soll [5].

Die beiden letzten Methoden „*saveGesturesPersistent*“ und „*loadPersistentGestures*“ dienen dazu, die aufgenommenen und angelernten Gesten persistent zu speichern und bei einem späteren Neustart des Programms wieder zu laden.

## 6.2 Programmablauf

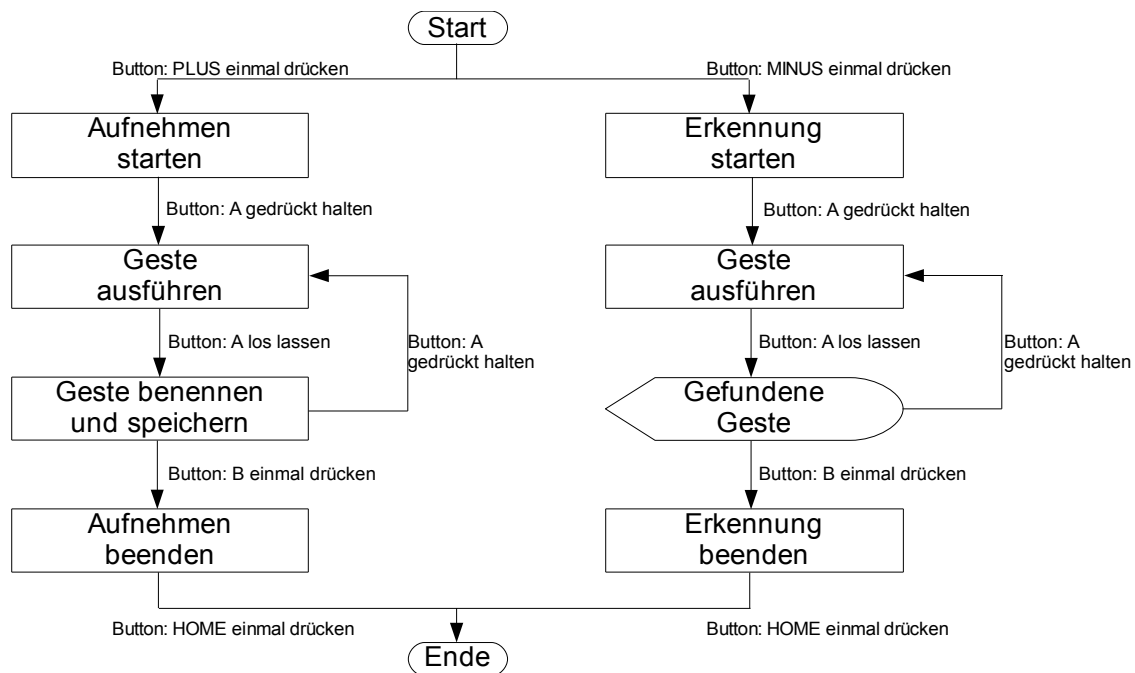


Abbildung 28: Programmablauf von „Wiinput“

In Abbildung 28 ist der Verlauf der Gestenerkennung einschließlich des Lern- und des Erkennungsprozesses graphisch dargestellt. Die Benutzereingaben beziehen sich hierbei auf die konsolenbasierte Benutzeroberfläche, die im Rahmen dieser Diplomarbeit entwickelt wurde.

Um in den Lernprozess zu gelangen muss der „Plus“-Knopf auf der Wii Remote betätigt werden. Mit dem „Minus“-Knopf gelangt man in den Erkennungsprozess. Um eine Geste ausführen zu können, muss während der Ausführung der Geste der „A“-Knopf gedrückt gehalten werden. Lässt man den „A“-Knopf wieder los, wurde die Ausführung der Geste beendet. Befindet man sich im Lernprozess muss nun ein Name für die Geste eingegeben werden. Befindet man sich im Erkennungsprozess, so gibt nun das Programm den Namen der Geste zurück, die der aufgenommenen Geste zugeordnet wurde. Um den jeweiligen Prozess zu beenden, muss der „B“-Knopf gedrückt werden.

### 6.3 Klassendiagramm

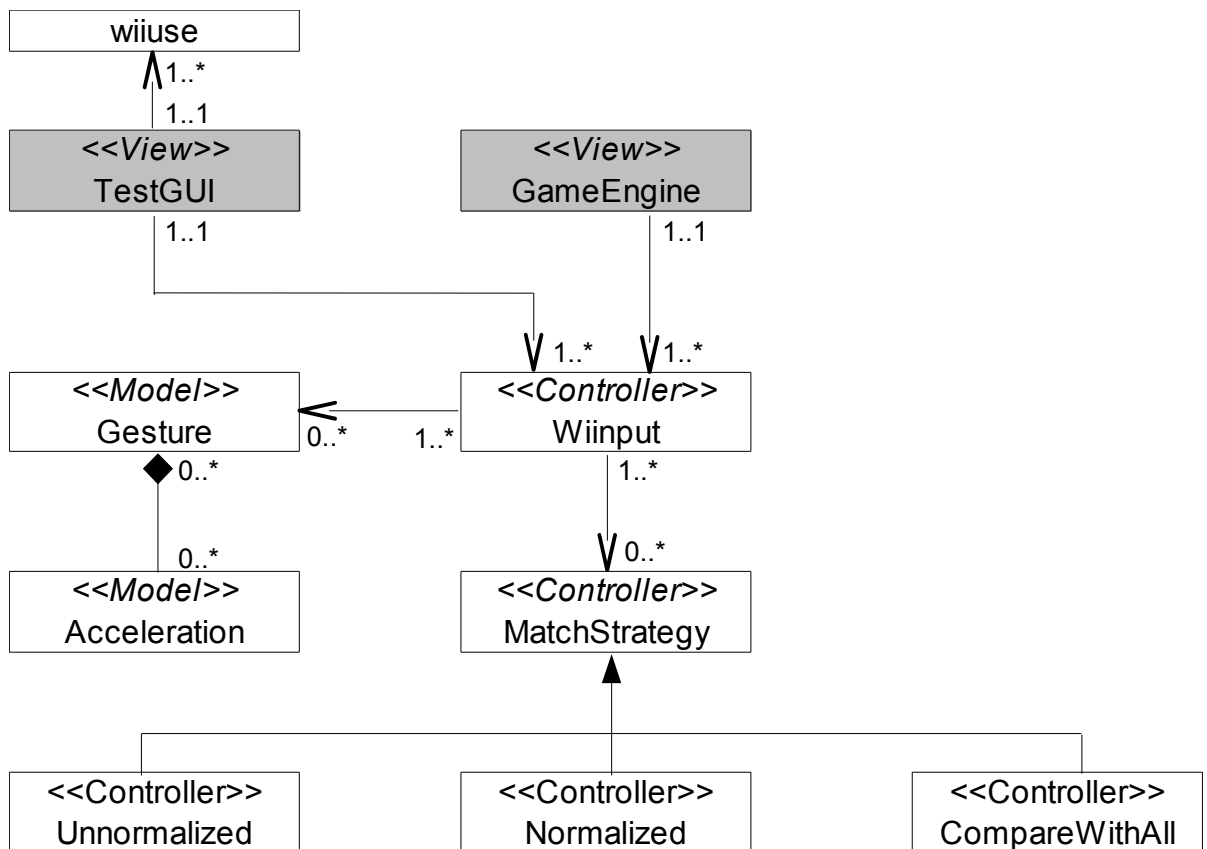


Abbildung 29: Klassendiagramm von „Wiinput“

Abbildung 29 zeigt das Klassendiagramm der Programmkomponente. Die Klasse „Wiinput“ stellt hierbei, wie oben bereits beschrieben, das Interface zu der Programmkomponente dar.

Für die Evaluation der Programmkomponente wurde das Strategy-Pattern eingefügt [5]. Dieses Entwurfsmuster findet sich in der abstrakten Klasse „MatchStragey“ und allen davon abgeleiteten Klassen wieder. Dieses ermöglicht es, neue Verfahren einfach hinzuzufügen und zu testen und bietet trotzdem noch die Möglichkeit, auf die alten Verfahren zurückzugreifen. So wurde beispielsweise bei einem Arbeitsschritt der Diplomarbeit die Normalisierung der Geste zu dem bereits implementierten Einfügen einer Geste hinzugefügt. Für die Evaluation dieses Arbeitsschrittes ist es jedoch wichtig, auch das Verfahren ohne die Normalisierung zu überprüfen, um zu zeigen, ob die Normalisierung ein besseres Ergebnis erzielt oder nicht.

Wie in Abbildung 29 ebenfalls zu sehen ist, gibt es eine Klasse „Gesture“. Jedes Objekt dieser Klasse stellt eine Geste dar. Eine einzelne Geste enthält dabei jede Geste, die für diese Geste ausgeführt wurde. Wird zum Beispiel der Kreis fünfmal aufgenommen, so enthält das Objekt der Geste mit dem Namen „Kreis“ fünf Gesten. Aus diesen fünf Gesten

wird dann eine einzige Referenzgeste erstellt. Es besteht aber auch die Möglichkeit, durch den Einsatz einer anderen „*MatchStrategy*“, die Geste gegen alle aufgenommenen Gesten einzeln zu überprüfen. So wird bei einem Testlauf getestet, ob die zusammengesetzten Referenzgesten bessere Resultate liefern, als alle Referenzgesten einzeln.

Die Klasse „*Acceleration*“ stellt jeden einzelnen Zeitschritt der Gesten mit den Werten für den Zeitschritt und für die X-, Y- und Z-Achse dar.

Die grau hinterlegten Klassen in Abbildung 29 stellen die Programmeinsprungspunkte dar, die auch die Hauptschleife des Programms bzw. Spiels beinhalten. Die konsolenbasierte Testumgebung für die Entwicklung der Gestenerkennung ist in der Klasse „*TestGUI*“ implementiert. Die Klasse „*GameEngine*“ steht für die Engine, die später in den Spielen der Firma Nurogames eingesetzt wird.

## **6.4 Strategien**

### **6.4.1 Unnormalized**

In der Klasse „Unnormalized“ werden die Gesten durch ihre Rohdaten verglichen. Aus den aufgenommenen Gesten des gleichen Typs wird eine einzige Referenzgeste erzeugt. Die zu klassifizierende Geste wird dann mit den Referenzgesten verglichen. Als Klassifikator wird der Abstandsklassifikator, wie in Kapitel 5.1.3 beschrieben, verwendet. Ist eine Geste länger, als die andere, wird der restliche Teil der längeren Geste nicht weiter betrachtet.

### **6.4.2 Normalized**

In der Klasse „Normalized“ werden die Gesten, bevor sie klassifiziert werden, wie in Kapitel 5.1.2 beschrieben, normalisiert. Anschließend erfolgt wieder die Klassifikation über den euklidischen Abstand. Auch bei dieser Strategie wird mit jeweils nur einer Referenzgeste pro Geste verglichen.

### **6.4.3 CompareWithAll**

In der Klasse „CompareWithAll“ werden die Gesten zunächst ebenfalls, wie in Kapitel 5.1.2 beschrieben, normalisiert. Anschließend wird die zu klassifizierende Geste mit allen einzelnen Referenzgesten verglichen und durch den euklidischen Abstand klassifiziert. Dabei wird die Geste der Referenzgeste zugeordnet, die das beste Ergebnis enthält.

## 6.5 Speicherung der Daten

Wie in Kapitel 6.1 beschrieben gibt es eine Methode, die dazu dient, Daten persistent abzuspeichern. Das ist wichtig, da die Referenzgesten aufgenommen und daraus ein Datenbestand erstellt werden soll, der bei jedem Neustart der Gestenerkennung und später bei jedem Neustart der Spiele zur Verfügung stehen muss. Da die Spiele auf keine komplexe Datenbank zugreifen können, kommt nur eine einfache Speicherung in Textdateien in Frage. Hierbei wurden CSV-Dateien verwendet. CSV steht für „character separated values“ und bedeutet, dass die Werte in den CSV-Dateien durch einfache Character-Zeichen, meist ein Komma oder Semikolon, getrennt werden. Neue Datensätze werden dabei in eine neue Zeile geschrieben.

<b>Gestures</b>	<b>Accelerations</b>
gestureId; name	gestureId; refId; t; x; y; z
0; Circle	0; 0; 0; 0.5; 0.3; 1.1
1; Frisbee	0; 0; 1; 0.6; 0.4; 1.2
2; Hit	0; 0; 2; 0.7; 0.5; 1.3
3; Bowling	0; 1; 0; 0.2; 0.5; 0.9
4; Throw	0; 1; 1; 0.1; 0.4; 0.8
	1; 0; 0; 0.3; 0.3; 1.2
	1; 0; 1; 0.2; 0.6; 1.3
	1; 1; 0; 0.0; 0.3; 0.8
	2; 0; 0; 0.5; 0.3; 0.9

Abbildung 30: Speicherung der Gesten

Abbildung 30 zeigt ein Beispiel für die Speicherung der Gesten. Es gibt zwei CSV-Dateien, wobei die erste nur die Namen der Gesten und eine Identifikationsnummer beinhaltet. Die zweite Datei enthält die Werte für jeden einzelnen Beschleunigungsschritt. Die erste Zahl steht für die Identifikationsnummer der Geste und die zweite Zahl für die Zahl der Ausführung. Anschließend folgt der Zeitschritt und die Werte für die einzelnen Beschleunigungsschritte in X-, Y- und Z-Richtung.

Für die Entwicklung der Gestenerkennung wird zunächst jeder Wert der Rohdaten, das heißt jede Aufnahme von jeder Geste gespeichert. Für den späteren Einsatz in den Spielen reicht es dann aus, nur noch die normalisierten und zusammengeführten Gesten zu speichern, da so Speicherplatz und vor allem Ladezeit gespart wird.

## 7 Evaluation der Gestenerkennung

Zur Überprüfung der erwarteten Ergebnisse und zur Bestätigung der Aussagen aus Kapitel 4 wurden eine Reihe von Tests mit den Testpersonen aus Kapitel 4.1 durchgeführt. Alle Tests wurden nur mit den Rechtshändern durchgeführt, da wir in Kapitel 4 gesehen haben, dass für Linkshänder andere Gesten aufgenommen werden müssten. Für die Berechnung der Erkennungsrate wurden jeweils zehn Tests pro Geste, also 50 Tests pro Testlauf durchgeführt.

Zunächst sollte die automatische Erkennung des Anfangs und des Endes einer Geste getestet werden. Hierzu wurden die Testpersonen aufgefordert ein paar Gesten auf Kommando auszuführen. Hierbei ergaben sich jedoch schon einige Schwierigkeiten. Es wurde bereits der Anfang einer Geste erkannt, wenn die Testperson die Wii Remote in der Ruheposition gehalten hat, da die Hand wohl mit einer höheren Beschleunigung als dem gewählten Schwellwert gezittert hat. Um dieses Problem in den Griff zu bekommen, wurde eine Mindestlänge für die Gesten eingeführt, die überschritten werden muss, damit die Geste gültig ist. Dies stellt jedoch ein Problem bei sehr kurzen bzw. schnell ausgeführten Gesten dar, da diese nun ebenfalls nicht mehr gültig sind. Das Problem konnte damit auch nicht ganz beseitigt werden. Des Weiteren ergibt sich durch das Zittern der Hände, dass so die Beschleunigungen zum Erkennen des Endes einer Geste nicht in der gewünschten Zeitspanne zwischen den Schwellwerten bleiben. Ein weiteres Problem ergab sich daraus, dass Gesten erkannt wurden, wenn die Testperson die Wii Remote in die Ausgangsposition der Geste gebracht hat, was verständlich ist, da hierzu ebenfalls Beschleunigungen notwendig sind. Ein weiteres Problem trat beim Ausführen der Geste auf, da bereits mitten in der Geste das Ende erkannt wurde, was auf eine zu kurz gewählte Zeitspanne, in der die Beschleunigungen zwischen den Schwellwerten liegen sollen, zurückzuführen ist. Wählt man diese jedoch zu groß, so dauert es zu lange, bis das Ergebnis der Gestenerkennung ausgegeben wird, da erst im Anschluss an diese Zeitspanne mit dem Erkennen der Geste begonnen werden kann. Diese Probleme ergaben Anlass, die weiteren Tests wieder mit der manuellen Funktion, durch gedrückt halten eines Knopfes an der Wii Remote, durchzuführen.

Als erstes Verfahren wurde die Gestenerkennung ohne Normalisierung, wie in Kapitel 6.4.1 beschrieben, getestet. Dabei wurde die Testperson 7 als Referenzperson ausgewählt. Zur Überprüfung, ob mehrere gleiche Aufnahmen einer Geste eine Verbesserung der Ergebnisse bewirkt, wurde der Testlauf zunächst mit nur einer Aufnahme pro Geste und anschließend mit zehn Aufnahmen pro Geste durchgeführt. Als Testpersonen wurden zum einen die selbe Person, die auch die Referenzen aufnimmt, ausgewählt und zum anderen die Testperson 1.

Referenzperson	Referenzen pro Person	Testperson	Kreis	Frisbee	Schlag	Bowling	Überkopfwurf
m 43J	1	m 43J	100%	100%	100%	0%	100%
m 43J	10	m 43J	100%	100%	70%	80%	70%
m 43J	1	w 20J	60%	20%	0%	10%	0%
m 43J	10	w 20J	100%	0%	0%	20%	0%

*Tabelle 2: Testergebnisse des ersten Testlaufs*

In Tabelle 2 werden die Ergebnisse des ersten Testlaufs dargestellt. Bei nur einer aufgenommenen Referenz pro Geste ist die Erkennung bei der gleichen Testperson wie die Referenzperson bis auf den Bowlingwurf bei 100%. Der Bowlingwurf der Testperson wurde zu 70% dem Überkopfwurf und zu 30% dem Frisbeewurf zugeordnet. Wie wir in Kapitel 4 gesehen haben, weisen diese Gesten, bei nicht perfekter Ausführung, durchaus starke Gemeinsamkeiten auf. Bei zehn aufgenommenen Referenzen pro Geste erhöht sich die Erfolgsrate beim Bowlingwurf enorm von 0% auf 80%. Im Gegensatz dazu geht die Trefferquote der Schlagbewegung und des Überkopfwurfes um jeweils 30% auf 70% zurück.

Bei der Testperson 1 wird, bis auf das Ergebnis beim Kreis, keine Geste annähernd gut erkannt. Auch bei zehn Referenzen pro Gesten ist, bis auf beim Kreis, der dann sogar zu 100% erkannt wird, keine Besserung zu erkennen.

Als nächstes wurde die Gestenerkennung mit Normalisierung, wie in Kapitel 6.4.2 beschrieben, getestet. Der Test wurde genau wie bei der Gestenerkennung ohne Normalisierung durchgeführt um einen direkten Vergleich beider Methoden zu gewährleisten. Die Testpersonen sind ebenfalls identisch.

Referenzperson	Referenzen pro Person	Testperson	Kreis	Frisbee	Schlag	Bowling	Überkopfwurf
m 43J	1	m 43J	60%	100%	100%	40%	90%
m 43J	10	m 43J	100%	100%	80%	50%	100%
m 43J	1	w 20J	100%	20%	0%	40%	20%
m 43J	10	w 20J	100%	10%	0%	40%	50%

*Tabelle 3: Testergebnisse des zweiten Testlaufs*

In Tabelle 3 sind die Ergebnisse des zweiten Testlaufs dargestellt. Bei Testperson 7 und nur einer Referenz pro Geste ist das Ergebnis schlechter ausgefallen, als ohne Normalisierung. Nur der Frisbeewurf und der Schlag wurden zu 100% erkannt. Bei zehn Referenzen pro Geste ist bei den restlichen Gesten jedoch eine deutliche Steigerung zu einer Referenz pro Geste zu beobachten. Hier wird der Schlag zu 80% und alle anderen Gesten zu 100% erkannt. Nur der Bowlingwurf wird lediglich zu 50% richtig erkannt. Insgesamt ist dies jedoch keine Steigerung zu der Gestenerkennung ohne Normalisierung.

Bei Testperson 1 ist ebenfalls eine kleine Steigerung zu beobachten. Wurde vorher lediglich der Kreis erkannt, wird nun immerhin der Bowlingwurf zu 40% und der Überkopfwurf zu 50% richtig erkannt. Insgesamt ist dies jedoch noch kein zufriedenstellendes Ergebnis.

Im nächsten Testlauf wurde die Gestenerkennung mit mehreren Personen als Referenzpersonen getestet. Dabei wurden zunächst vier Referenzpersonen und anschließend zehn der elf Testpersonen aus Kapitel 4.1 ausgewählt. Als Testpersonen wurden Testperson 6, die sich ebenfalls unter den Referenzpersonen befindet, und Testperson 1, die sich nicht unter den Referenzpersonen befindet, ausgewählt. Getestet wurde die Gestenerkennung mit Normalisierung, da diese bei Unterschieden zwischen Referenz- und Testperson etwas bessere Ergebnisse erzielt hat.

Referenzpersonen	Referenzen pro Person	Testperson	Kreis	Frisbee	Schlag	Bowling	Überkopfwurf
4	3	m 24 J	90%	30%	0%	70%	0%
4	3	w 20 J	10%	10%	20%	40%	20%
10	3	m 24 J	100%	30%	0%	80%	30%
10	3	w 20 J	100%	10%	40%	60%	60%

*Tabelle 4: Testergebnisse des dritten Testlaufs*

In Tabelle 4 sind die Ergebnisse des dritten Testlaufs aufgeführt. Bei vier Referenzpersonen ist das Ergebnis schlechter ausgefallen als bei nur einer Referenzperson, da alle Gesten, außer der Bowlingwurf, insgesamt schlechter erkannt wurden. Bei zehn Referenzpersonen ist eine deutliche Steigerung im Gegensatz zu vier Referenzpersonen zu erkennen, da sämtliche Gesten, außer der Frisbeewurf, besser erkannt wurden. Außerdem ist bei zehn Referenzpersonen ebenfalls eine Steigerung zum Ergebnis mit nur einer Referenzperson zu sehen. Zumindest dann, wenn die Testperson nicht die Referenzperson ist. Bei zehn Referenzpersonen ist eine Steigerung bei den Gesten Schlag, Bowling und Überkopfwurf festzustellen. Insgesamt ist das Ergebnis jedoch noch nicht zufriedenstellend.

Ein letzter Testlauf soll zeigen, ob die Idee, aus allen Aufnahmen für jede Geste nur eine Referenzgeste zu erstellen, nur eine bessere Performance oder auch bessere Erkennungsraten liefert. Deshalb werden nun die Testgesten mit jeder einzelnen Geste, die aufgenommen wurde, verglichen. Als Datenbestand dienen die Referenzgesten der zehn Personen des vorherigen Testlaufs.

Referenzpersonen	Referenzen pro Person	Testperson	Kreis	Frisbee	Schlag	Bowling	Überkopfwurf
10	3	m 24 J	40%	100%	50%	90%	80%
10	3	w 20 J	40%	0%	10%	50%	60%

*Tabelle 5: Testergebnisse des vierten Testlaufs*



In Tabelle 5 sind die Ergebnisse des letzten Testlaufs aufgeführt. Wie man sehen kann, hat die Testperson 6, bis auf bei der Kreisbewegung, ein deutlich besseres Ergebnis erzielt, wohingegen Testperson 1 ein deutlich schlechteres Ergebnis erzielt hat. Die könnte daran liegen, dass Testperson 6 sich ebenfalls unter den Referenzpersonen befand im Gegensatz zu Testperson 1. Insgesamt hat das Erstellen einer Referenzgeste mehr Vorteile, da sich die späteren Benutzer der Spiele nicht unter den Referenzpersonen befinden werden.

## 8 Schluss

Im Rahmen dieser Diplomarbeit wurde eine Gestenerkennung entwickelt, die auf dem Beschleunigungssensor in der Wii Remote basiert. Hierzu wurde eine Programmkomponente entwickelt, die zum Testen der eingesetzten Verfahren herangezogen wurde und die später in den Spielen der Firma Nurogames eingesetzt werden soll.

Bei der Entwicklung der Gestenerkennung wurden auf den Grundlagen der Mustererkennung aufgebaut. Zum Einsatz kamen mathematische Verfahren zur Normalisierung der Gesten sowie zur späteren Klassifikation, um die Gesten den bereits vorhandenen Gesten zuzuordnen. Für die Einordnung der Gesten wurde ein einfacher Abstandsklassifikator verwendet.

Wichtig in den Untersuchungen war die personenübergreifende Gestenerkennung, bei der untersucht werden sollte, ob es möglich ist, die Gesten jeder beliebigen Person zu erkennen. Dazu wurden zunächst ein paar Gesten anschaulich gemacht und anschließend analysiert. Später wurden mit einigen Testpersonen, von unterschiedlichem Geschlecht und Alter, Tests durchgeführt um die entwickelte Gestenerkennung zu evaluieren.

Getestet wurde zunächst, ob die Vorverarbeitung, also die Normalisierung der Gesten, eine Verbesserung der Gestenerkennung erreicht. Hierzu wurde anfangs nur eine Referenzperson eingesetzt. Als Ergebnis lässt sich erkennen, dass die Gestenerkennung, bei einer anderen Testperson als der Referenzperson, bessere Ergebnisse lieferte, wenn sämtliche Gesten normalisiert wurden.

Als nächstes wurde der Einsatz von mehreren Referenzpersonen getestet. Zum Einsatz kamen bei dem einen Testdurchgang vier und bei dem anderen zehn Personen, von denen jeder jeweils dreimal eine Geste ausführen sollte. Somit sollte getestet werden, ob die Gestenerkennung bei einem Testlauf mit mehreren Personen bessere Ergebnisse liefert, da mehr Referenzdaten zur Verfügung stehen. Als Ergebnis war festzustellen, dass bei nur vier Referenzpersonen eine schlechtere Erkennung der Gesten im Gegensatz zu einer Person auftrat. Hingegen war bei zehn Referenzpersonen eine Steigerung der Gestenerkennung gegenüber einer Person zu beobachten.

Als letztes wurde ein Versuch durchgeführt, bei dem jede einzelne aufgenommene Geste mit der zu klassifizierenden Geste verglichen wurde. Im Gegensatz dazu, wurden bei den zuerst verwendeten Verfahren alle Gesten einer Kategorie zu einer Geste zusammengefasst. Das zuletzt verwendete Verfahren würde natürlich einige Performance Verluste bedeuten und von daher nicht relevant für den Einsatz in Spielen sein. Des Weiteren war auch das Ergebnis nicht überzeugend. Zwar wurden die Gesten der Testperson, die sich ebenfalls

unter den Referenzpersonen befand, besser erkannt, jedoch hat sich die Gestenerkennung bei einer Testperson, die sich nicht unter den Referenzpersonen befand, deutlich verschlechtert.

Insgesamt ist das Ergebnis noch nicht zufriedenstellend, da bei keinem der Versuche die Gestenerkennung eine konstant hohe durchschnittliche Erkennungsrate geliefert hat. Das beste Ergebnis wurde erzielt, wenn es nur eine Referenzperson gab und diese gleichzeitig auch die Testperson war. Dies wird in der Praxis jedoch nicht der Fall sein, da die Spieleentwickler die benötigten Gesten aufnehmen. Des Weiteren hat Poppinga mit seiner Forschungsarbeit bereits ähnliche Ergebnisse für eine Person erzielt [1].

## **8.1 Ansatzpunkte für Folgeuntersuchungen**

Bei der Entwicklung der Gestenerkennung sind einige Aspekte noch nicht untersucht worden, die in Folgeuntersuchungen weitergeführt werden können.

Da mit zehn Referenzpersonen bessere Ergebnisse erzielt wurden, als mit weniger Personen, könnten weitere Tests durchgeführt werden, die aufzeigen, ob mit mehr Referenzpersonen noch bessere Ergebnisse erzielt werden.

Des Weiteren wurden von jeder Person jede Geste nur dreimal aufgenommen. Somit könnte die Auswirkung von mehr aufgenommen Gesten der gleichen Person auf die Ergebnisse untersucht werden, da so eine größere Anzahl von Referenzdaten vorhanden wäre.

Die Abtastung in der Vorverarbeitung ist sehr störanfällig. Es könnten weitere Versuche durchgeführt werden, die untersuchen, ob eine Mittelung über einen Bereich, der je nach Länge der Geste verschieden groß gewählt werden kann, um die Abtastpunkte herum eine bessere Abbildung der Geste darstellt und dadurch bessere Ergebnisse erzielt werden.

Der Abstandsklassifikator könnte noch dahingehend erweitert werden, dass automatisch erkannt wird, welche Achsen bei den unterschiedlichen Gesten wichtiger und welche unwichtiger sind. Die wichtigeren Achsen würden dann eine größere Gewichtung bekommen, als die unwichtigen. Die automatische Ermittlung der wichtigen und unwichtigen Achsen könnte nur bei einer großen Anzahl von Referenzgesten erfolgen, da bei manchen Ausführungen auch Achsen stark beschleunigt werden, obwohl diese nicht so wichtig sind. In wie weit so eine automatische Erkennung überhaupt möglich ist, muss zunächst auch erst einmal untersucht werden.

Als Mustererkennungsverfahren wurde nur die statistische Mustererkennung eingesetzt. Die Gestenerkennung könnte noch dahingehend weiter entwickelt werden, dass die strukturelle Mustererkennung zum Einsatz kommt. Dies könnte erreicht werden, indem vorher bestimmte Merkmale der Gesten automatisch ermittelt und diese dann miteinander verglichen werden.

Hierzu ist eine andere Vorverarbeitung notwendig, bei der die Gesten nicht normalisiert werden, sondern bestimmte Merkmale aus den Gesten extrahiert werden. Bei der Klassifikation könnte dann, durch diese so genannten Merkmalsvektoren, die Wahrscheinlichkeit ermittelt werden, zu der die Geste zu der einen oder anderen Geste gehört.

Wie in Kapitel 4.7 erläutert wurde, werden für unterschiedliche Gesten unterschiedlich lange Ausführungszeiten und maximale Beschleunigungen erreicht. Hieraus könnte man weitere Abstandsmaße oder Merkmale für die Klassifikation einführen. Der Abstandsklassifikator könnte mit

$$d(T) = a * (T - T_{ref})^2$$

erweitert werden, wobei  $a$  ein geeigneter Wichtungsfaktor,  $T$  die Dauer der aktuell zu klassifizierenden Geste und  $T_{ref}$  die gemittelte Dauer der Referenzgesten ist. Als weitere Merkmale könnte man beispielsweise noch die maximale und minimale Beschleunigung in allen Richtungen, die für die Geste wichtig sind, einfügen.

Die oben beschriebenen Merkmalsvektoren können mit unterschiedlichen Verfahren ausgewertet werden, wodurch eine Klassifizierung erreicht werden kann. Im folgenden werden die drei Verfahren Neuronale Netze, Entscheidungsbäume und der Bayes-Klassifikator erläutert.

Bei den künstlichen neuronalen Netzen handelt es sich um Verfahren, bei denen das menschliche Gehirn am Computer simuliert wird. Hierbei wird zwischen Input Neuronen, die einen Eingabewert erwarten, und den Output Neuronen, die Ausgabewerte liefern, unterschieden. Bei der Gestenerkennung mittels künstlicher neuronaler Netze wird für jedes Merkmal ein Input Neuron gebildet. Außerdem erhält jede Geste ein Output Neuron, das angibt, wie die zu klassifizierende Geste mit den einzelnen Gesten übereinstimmt. Trainiert wird das Netz mit den Referenzgesten, bei denen sowohl die Eingabe-, wie auch die Ausgabewerte bekannt sind. Durch diese Referenzdaten werden die Gewichtungen der Kanten des neuronalen Netzes verändert, wodurch erreicht wird, dass bei bestimmten Eingabewerten bestimmte Ausgaben erzeugt werden. Gibt man nun eine zu klassifizierende Geste an das neuronale Netz, so wird durch die Kantengewichtung ermittelt, zu welcher Referenzgeste die zu klassifizierende Geste gehört.

Beim Entscheidungsbaum handelt es sich um einen Baum, dessen Blätter die Gesten, die zu erkennen sind, enthalten. Jeder Knoten des Baumes enthält ein Merkmal der Merkmalsvektoren. An den Kanten der Knoten sind Wahrscheinlichkeiten angegeben, durch

die entschieden werden muss, welcher Weg zu gehen ist. Ist man an einem Blatt angelangt, kann die zu klassifizierende Geste der Geste, die das Blatt enthält, zugeordnet werden. Der Entscheidungsbaum wird mit Hilfe der Referenzgesten angelegt.

Beim Bayes-Klassifikator wird anhand der Merkmalsvektoren ermittelt, mit welcher Wahrscheinlichkeit die zu klassifizierende Geste zu einer der Referenzgesten gehört. Auch hier werden die Referenzgesten benötigt, um die Wahrscheinlichkeiten der zu klassifizierenden Geste zu ermitteln.

Bisher kommt in der Gestenerkennung noch nicht die automatische Anfangs- und Enderkennung zum Einsatz, da diese, wie in Kapitel 7 beschrieben, Probleme bereitet. Zum Einsatz der Gestenerkennung ist es jedoch erforderlich, dass diese automatisch Erkennung funktioniert und der Benutzer nicht wie bisher einen Knopf gedrückt halten muss. Ein weiterer Ansatz, der hierfür untersucht werden könnte, ist der Leaky Integrator, bei dem die Integrale für jeden Zeitschritt bis zu einem bestimmten Zeitschritt zurück berechnet und aufsummiert werden. Dabei erhalten die Zeitschritte, je nach dem, wie lange sie zurück liegen, eine kleiner werdende Gewichtung, so dass der Zeitschritt, der am dichtesten ist, die größte Gewichtung erhält. Übersteigt die Summe einen bestimmten Schwellwert, wurde die Geste angefangen. Fällt die Summe wieder unter diesen Schwellwert zurück, wurde die Geste beendet.

## **8.2 Ausblick**

Mit dem momentanen Entwicklungsstand kann die Gestenerkennung noch nicht in den Spielen der Firma Nurogames eingesetzt werden. Da die Benutzer nach Ausführung einer Geste auch eine bestimmte Erwartung an die Reaktion des Spiels haben, ist eine sehr hohe Erkennungsrate notwendig. Außerdem ist die automatische Erkennung des Anfangs und des Endes der Geste essentiell, da der Benutzer nicht erst einen Knopf betätigen soll um eine Geste auszuführen.

In Folgeuntersuchungen im Anschluss an diese Diplomarbeit soll die Gestenerkennung weiter entwickelt werden, um letztendlich den Einsatz in die Spiele der Firma Nurogames zu ermöglichen.

## Literaturverzeichnis

- [1] Benjamin Poppinga: Beschleunigungsbasierte 3D-Gestenerkennung mit dem Wii-Controller, Carl von Ossietzky Universität Oldenburg, Oldenburg, August 2007
- [2] Rüdiger Hoffmann: Signalanalyse und -erkennung: Eine Einführung für Informationstechniker, Springer, Berlin Heidelberg, 1998
- [3] Michael Laforest: wiiuse – The Wiimote C Library, <http://wiiuse.net>, 2007, download am 02.08.2008
- [4] Carl Kenner: GlovePIE, <http://carl.kenner.googlepages.com/glovepie>, 2004, download am 02.08.2008
- [5] Erich Gamma et. al.: Design Patterns – Elements of Reusable Object-Oriented Software, Addison-Wesley, Amsterdam, 1995
- [6] Keinosuke Fukunaga: Introduction to Statistical Pattern Recognition, Academic Press, New York, 2. Auflage, 1991
- [7] Heinrich Niemann: Klassifikation von Mustern, Springer, Berlin, 1983

## **Ehrenwörtliche Erklärung**

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Gummersbach, den 11.08.2008

---

Marco Polak