



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung

Dominik Charousset

Gestenbasierte Steuerung auf Basis von Motion
Tracking am Beispiel einer einfachen Simulation
für Bogenschießen

Inhaltsverzeichnis

1	Einleitung	1
2	Motion Tracking als Eingabemedium	2
2.1	Start-/Enderkennung	2
2.2	Motivation für den Einsatz in Videospielen	3
3	Implementierung eines Prototyps	4
3.1	Verwendete Technologien	4
3.2	Funktionsprinzip, Modellierung und Geste	5
3.3	Abschluss von Pfeilen (Ende der Geste)	7
3.3.1	Überspannen	7
3.3.2	Spannen und explizites Abschließen per Tastendruck	7
3.4	Diskussion	7
3.5	Fazit	8
4	Ausblick	9
4.1	Forcierung eines mentalen Modells	9
4.2	Bewegung im dreidimensionalen Raum	9
4.3	Enderkennung der Geste	10
4.4	Usability-Studien	10
	Literatur	11
	Abbildungsverzeichnis	12
A	Anhang	13
A.1	Prototyp in Labor 1080	13
A.1.1	Powerwall	13
A.1.2	Erster Aufbau (vgl. Kap. 3.3.1)	14
A.1.3	Zweiter Aufbau (vgl. Kap. 3.3.2)	15
A.2	Physikbeispiel	16

1 Einleitung

Vor gut 40 Jahren, am 9. Dezember 1968 (Stanford (2008)), wurde der Weltöffentlichkeit die Computermaus vorgestellt - und sie hat den Umgang mit Computersystemen revolutioniert. Erstmals reagierte ein Computer auf Bewegung, wenn auch nur indirekt durch die Maus.

Doch so revolutionär die Computermaus zu ihrer Zeit auch gewesen ist, sie ist kein 3D-Eingabemedium. Doch dem Grundgedanken (Interaktion durch Bewegung) folgend setzten sich in den letzten Jahren neuartige Eingabegeräte durch, die die Bewegungen des Spielers direkt zur Manipulation der fiktionalen Spielwelt nutzen. Ein Beispiel für gestenbasierte Interaktion im 2D-Raum ist Sonys *EyeToy* (Sony (2008)), das Webcambasiert arbeitet.

Die erste Gestensteuerung im 3D-Raum für den Massenmarkt erschien 2006. Mit seiner Spielekonsole *Wii* führte Nintendo die Wiimote (Abb. 1) ein, die nicht nur ein Zeigemedium ist, sondern auch eine Gestenerkennung durch Beschleunigungssensoren ermöglicht. Die neue Art der Interaktion ermöglichte nicht nur neue Spielkonzepte, wie z.B. Bowling, Tennis und Golf, sondern machte Spielen für ein völlig neues Publikum interessant und zugänglich.



Abbildung 1: Wiimote [Nintendo (2008b)]

Zugänglich vor allem durch die schnell und einfach erlernbare Interaktion. Um eine Bowlingkugel zu werfen reicht es, die Geste des Werfens mit der Wiimote in der Hand auszuführen. Der Spieler kann sein aus der Realität geformtes mentales Modell beim Spielen wiederfinden und muss keine andere Art der Interaktion dafür erlernen - wie dies z.B. bei Gamepads der Fall ist.

Doch die Beschleunigungssensoren der Wiimote bieten bei der Erkennung dreidimensionaler Gesten weniger Genauigkeit und vor allem weniger Möglichkeiten als Motion Tracking. Die machbar- und nutzbarkeit von Motion Tracking zur Erkennung von 3D-Gesten belegen frühere Arbeiten an der HAW, wie z.B. die Arbeiten von Boetzer (2008) und Heitsch (2008).

Diese Arbeit beschäftigt sich mit der Fragestellung, welche Möglichkeiten sich durch den Einsatz von Motion Tracking zur Steuerung von Videospiele ergeben und welche Probleme es dabei zu lösen gilt. Hierzu soll das Szenario eines virtuellen Bogenschießens realisiert werden, mit besonderem Blick auf intuitive, schnell und einfach erlernbare Steuerung.

2 Motion Tracking als Eingabemedium

Motion Tracking ist eine Technologie, mit der man Bewegungen der handelnden Person zur Interaktion mit Computersystemen nutzen kann.

“Motion Tracking is the process of recording a live motion event and translating it into usable mathematical terms by tracking a number of key points in space over time.” Menache (2000).

Bei den Gesten unterscheidet man zwischen *Bewegungsverfolgung*, *kontinuierlichen Gesten* und *symbolisch-manipulativen Gesten* (Heitsch, 2008, S. 31).

Im vorliegenden Szenario ist nur die *Bewegungsverfolgung* relevant, bei der der Benutzer durch Bewegung festgelegte Eigenschaften eines Objektes (in diesem Fall: des Bogens), wie z.B. Position oder Orientierung, manipuliert.

Motiviert wird die Nutzung von Motion Tracking als Eingabemedium durch das Ziel ein möglichst natürliches, intuitives Bedienerlebnis für den Nutzer zu schaffen. Im Idealfall bedarf es keiner erklärenden Worte mehr, wie ein System zu bedienen ist. Die gewählten Metaphern und aus der Realität entnommene Bewegungsabläufe sollen es dem Benutzer ermöglichen, seine Erfahrungen im täglichen Leben direkt auf das System anzuwenden.

Je konkreter die gewählte Metapher und je besser die implementierte Steuerung zum mentalen Modell des Nutzers passt, desto näher kann man dem Ideal einer handbuchfreien, intuitiven Bedienung kommen. Mit dem für diese Arbeit gewählten Beispiel eines Bogens, den der Nutzer per Bewegung steuert, soll untersucht werden, wie nahe man mit Motion Tracking diesem Ideal in Videospielen kommen kann.

2.1 Start-/Enderkennung

Nicht jede Bewegung des Nutzers ist bewusst, bzw. gewollt. Zudem bewegt sich der Mensch kontinuierlich, auch wenn er gerade nicht interagieren möchte. Darum ist ein definierter Start und Endpunkt erforderlich.

Tatsächlich ist eine für den Nutzer intuitive Start-End-Erkennung eines der Hauptprobleme, wie wir in den Kapiteln 3.4, und 4.3 sehen werden.

2.2 Motivation für den Einsatz in Videospielen

Der Einsatz von dreidimensionalen Interaktionsmedien zur Manipulation fiktionaler Welten bietet neue Wege und Möglichkeiten bei der Gestaltung. Bisherige Systeme erforderten vom Nutzer meist das Erlernen abstrakter Eingabemedien (z.B. Gamepads bei Konsolen), die weder in ihrer Form noch in ihrem Verhalten dem Nutzer Aufschluss über ihre Funktionsweise liefern. Ebenso wenig gibt es ein Gegenstück aus der realen Welt, so dass der Nutzer nicht auf Erfahrungen zurückgreifen kann.

Als erstes Unternehmen aus der Videospielebranche hat Nintendo mit ihrer Konsole *Wii* ein Produkt veröffentlicht, mit dem Gesten im dreidimensionalen Raum in Videospielen eingesetzt werden (bei Produkten wie *EyeToy* (Sony (2008)) werden zwar auch Gesten eingesetzt, allerdings nur im 2D-Bereich). Allerdings werden hierfür Beschleunigungssensoren eingesetzt, was die Zahl möglicher Anwendungsszenarien einschränkt und im Vergleich zu Motion Tracking ungenau ist und weniger Anwendungsszenarien abdeckt.

Ziel einer intuitiven Steuerung ist es, dass der Spieler möglichst viel seiner Alltagserfahrung einbringen kann und sich die Umgebung so verhält, wie er es erwartet. Im optimalen Fall sollte der gleiche Bewegungsablauf, den der Benutzer im realen Umfeld ausführt, zum gleichen Ergebnis führen. Menschen sind es gewohnt, physikbasiert zu interagieren. Also ist ein korrektes physikalisches Verhalten der fiktionalen Spielwelt ein Grundkriterium zur Immersion (Eintauchen in die Spielwelt, vgl. (Brown und Cairns, 2004, S. 1)).

Bildet man realitätsnahe Szenarien in Videospielen ab, so hat man aus der Wirklichkeit vorgegebene Anwendungsfälle, im Gegensatz zu generischen, gestengesteuerten Systemen. In der vorliegenden Arbeit die Bedienung eines Bogens. Dies hat den großen Vorteil, dass es eine konkrete Metapher und ein Gegenstück in der realen Welt gibt, so dass bei Benutzern keine unterschiedlichen mentalen Modelle auftreten können.

3 Implementierung eines Prototyps

Um die Eignung von gestenbasierter Steuerung auf Basis von Motion Tracking fundierter analysieren zu können, wurde ein einfacher Prototyp für ein Spiel entwickelt, in dem der Spieler einen virtuellen Bogen durch Gesten steuern soll.

3.1 Verwendete Technologien

Für den in C++ geschriebenen Prototyp wurde das in Raum 1080 eingerichtete Motion Tracking System verwendet (siehe Anhang A.1). Die Software wurde nach dem MVC-Pattern (Gamma u. a. (1995)) strukturiert (siehe Abb. 2).

Um eine, wie in Kap. 2.2 angesprochene, erwartungs- und erfahrungsgerechte Umgebung mit physikbasierter Interaktion zu schaffen, wurde für die Implementierung die *PhysX* Physik-Engine (Nvidia (2008)) verwendet. Die Physik beeinflusst direkt das Verhalten aller Objekte in der fiktionalen Spielwelt, manipuliert also das Model (siehe Anhang A.2).

Die Grafikausgabe - die View - wurde mit Hilfe der *Irrlicht* Grafik-Engine (Gebhardt (2008)) realisiert.

Das Motion Tracking wird von der Software *DTrack* (A.R.T. (2008)) übernommen. Die Bewegungsdaten werden vom Controller interpretiert, der die Interaktion des Spieler mit der fiktionalen Spielwelt realisiert; wobei der Spieler nur den Bogen direkt manipulieren kann.

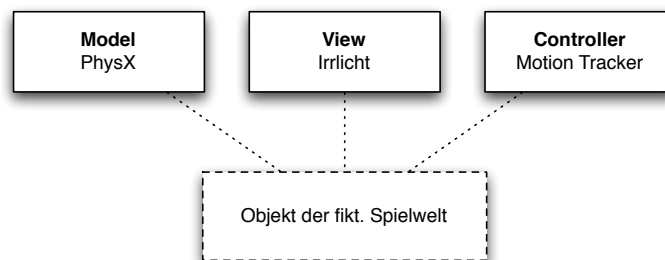


Abbildung 2: Konzeption der Software

3.2 Funktionsprinzip, Modellierung und Geste

Der Spieler bedient das System mit zwei in der Hand gehaltenen Markern, die von den Trackerkameras erfasst werden und deren genaue Position aus den Kamerabildern berechnet werden können. Zur genauen Funktionsweise siehe (Boetzer, 2008, S. 35). Dabei steht ein Marker für den Griff des Bogens (Abb. 3) und ein Marker für die Bogensehne (Abb. 4).

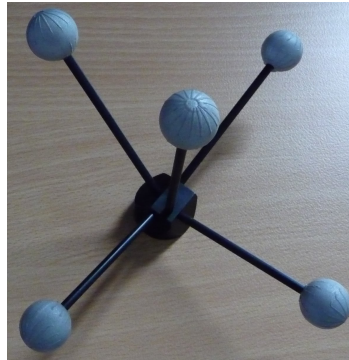


Abbildung 3: Marker (Bogen)



Abbildung 4: Marker (Sehne)

Aus dem Abstand zwischen den beiden Markern (*Bogen* und *Sehne*) wird Zielrichtung und Spannkraft (als Vektor) berechnet, wie Abb. 5 zeigt.

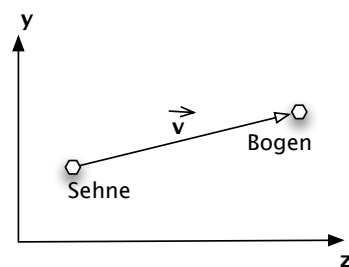


Abbildung 5: Errechneter Vektor (vereinfacht; ohne x-Achse)

Die Richtung des Vektors beschreibt, in welche Richtung der Spieler zielt und die Länge des

Vektors gibt an, wie weit der Bogen derzeit gespannt wurde. Dabei gibt es zwei Grenzwerte, die zu beachten sind: die Mindestspannung h_{min} , die für einen Schuss erreicht werden muss und die Maximalspannung h_{max} , wie Abb. 6 zeigt.

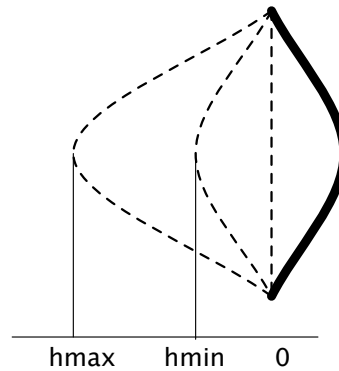


Abbildung 6: Grenzwerte

Die Länge des Vektors beschreibt zudem den Zustand, in dem der Bogen sich befindet (siehe Tabelle 1).

$0 \leq \vec{v} \leq h_{min}$	Bogen befindet sich in Grundposition
$h_{min} < \vec{v} < h_{max}$	Bogen gespannt
$ \vec{v} \geq h_{max}$	Bogen maximal gespannt

Tabelle 1: Zustände des Bogens

Die Geste für den Spieler besteht darin, den Bogen zu spannen, d.h. den Marker mit der Sehne nach hinten zu ziehen, während er den anderen Marker ruhig hält.

Dabei muss der Spieler vor jedem Schuss in die Grundposition zurückkehren, wodurch eine eindeutige Starterkennung gewährleistet ist. Die dahinter stehende Metapher ist das Auflegen eines neuen Pfeils, bevor er den Bogen erneut spannen kann.

Um keine höhere Spannung des Bogens als h_{max} zuzulassen, ist beim Abschuss des Pfeils der Vektor für die Startgeschwindigkeit (engl. velocity, kurz *vel*):

$$vel_0 = \frac{\vec{v}}{|\vec{v}|} * \min(h_{max}, |\vec{v}|).$$

3.3 Abschluss von Pfeilen (Ende der Geste)

In diesem Prototyp wurden zwei mögliche Steuerungsvarianten implementiert, zwischen denen umgeschaltet werden kann und die im Folgenden vorgestellt und diskutiert werden.

3.3.1 Überspannen

In der ersten und einfachsten Implementierung wird der Pfeil automatisch abgeschossen, sobald $|\vec{v}| \geq h_{max}$. Dies hat den Vorteil, dass keine weitere Modalität benötigt wird, jedoch lässt sich die Abschussgeschwindigkeit nicht regulieren und ein unbeabsichtigter Schuss ist nicht auszuschließen. Insbesondere beim ersten Kontakt mit dem Spiel kann dies zu einem unerwarteten Verhalten des Systems führen.

3.3.2 Spannen und explizites Abschießen per Tastendruck

Für die zweite Implementierung wurde als zusätzliche Modalität die Wiimote eingeführt. Der Benutzer spannt den Bogen, bis $|\vec{v}| \geq h_{min}$ und bekommt durch optisches Feedback (Anzeige, wie weit der Bogen derzeit gespannt ist) signalisiert, dass der Bogen die minimal erforderliche Spannkraft erreicht hat. Da der Benutzer selbst entscheidet, wann geschossen werden soll, kann er die Startgeschwindigkeit des Pfeils über $|\vec{v}|$ steuern. Die Geste endet, sobald der Nutzer einen festgelegten Knopf auf der Wiimote drückt (unter der Voraussetzung, dass $|\vec{v}| \geq h_{min}$ erfüllt ist).

3.4 Diskussion

Das Hauptproblem jeder Gestensteuerung ist die Start- und Enderkennung. Beide wurden in der prototypischen Implementierung gelöst:

Die Starterkennung wurde in beiden Implementierung über den definierten Grundzustand $0 \leq |\vec{v}| \leq h_{min}$ erreicht.

Die Enderkennung wurde unterschiedlich gelöst. Im ersten Fall war die Geste beendet, sobald $|\vec{v}| \geq h_{max}$. Dies hat den Nachteil, dass der Spieler die Geste unbeabsichtigt beenden kann.

Im zweiten Fall wurde die Enderkennung über eine zusätzliche Modalität gelöst (Knopfdruck auf der Wiimote), wobei die Bedingung $|\vec{v}| \geq h_{min}$ erfüllt sein muss.

Wünschenswert wäre ein bewusstes Beenden der Geste durch den Nutzer, jedoch ohne zusätzliche Modalität. Das direkte Vorbild aus der Realität wäre das Öffnen der Hand, also das Loslassen des auf der gespannten Sehne befindlichen Pfeils. Da der im Labor zur Verfügung stehende Tracker keine Möglichkeit hat die Hand, bzw. die einzelnen Finger direkt zu erfassen, muss hier eine sinnvolle, intuitive Geste gefunden und auf ihre Umsetzbarkeit hin überprüft werden.

3.5 Fazit

Der Prototyp stellt eine Basis für weitergehende Experimente zur Verfügung. Die Starterkennung funktioniert ohne weitere Modalität und hat eine direkte Metapher aus der realen Welt.

Bei der Enderkennung gibt es jedoch noch viele offene Fragen, insbesondere fehlt jegliche Usability-Untersuchung. Das Verwenden der Wiimote löst zwar das Problem der ungewollten Enderkennung, allerdings wäre eine Lösung ohne zusätzliche Modalität wünschenswert. Gegenstand weiterer Untersuchungen sollte insbesondere die Implementierung einer möglichst intuitiven und realitätsnahen Geste für den Abschuss des Pfeils sein.

4 Ausblick

Wie im Fazit (Kap. 3.5) angedeutet, lässt der bestehende Prototyp noch viele Fragen offen, bildet aber eine gute Grundlage zur Vertiefung des Themas. Welche Fragen vorrangig gestellt werden sollten und welche Ansätze es gibt, soll im folgenden abschließend diskutiert werden.

4.1 Forcierung eines mentalen Modells

Auch wenn das Verhalten der beiden Marker denen eines Bogens entspricht, so verraten sie in ihrer Form noch nichts über ihre Funktionsweise. Überlegenswert wäre es, durch die äußere Form ein mentales Modell beim Nutzer zu forcieren, wie dies z.B. durch Aufsätze für die Wiimote getan wird (siehe Abb. 7).



Abbildung 7: Tennisaufsatz für Wiimote [Logic3 (2008)]

4.2 Bewegung im dreidimensionalen Raum

Um dem Spieler wirklich das Gefühl einer dreidimensionalen Umgebung zu geben, fehlt derzeit noch jegliche Möglichkeit sich in diesem zu bewegen. Je nach gewünschter Bewegungsfreiheit gibt es grundsätzlich zwei Möglichkeiten: ohne zusätzliche Modalität oder mit.

Die Position des Spielers im realen Raum direkt auf seine Position in der fiktionalen Spielwelt

zu übertragen hat den Vorteil, dass es eine sehr direkte und intuitive Art der Interaktion ist. Jedoch begrenzt sich dadurch die Bewegungsfreiheit im virtuellen Raum durch die real vorhandenen Platzverhältnisse (außer in Virtual Reality Umgebungen, die jedoch nicht Teil dieser Arbeit sind). Für große virtuelle Welten ist dies daher ungeeignet.

Nimmt man zusätzliche Medien hinzu, wie z.B. das Wii Balance Board (Abb. 8), ermöglicht man dem Spieler eine größere Freiheit beim Bewegen durch den virtuellen Raum, setzt aber wieder einen Lernaufwand beim Spieler voraus, da dieser nicht mehr seine natürliche Bewegung zur Interaktion nutzen kann.



Abbildung 8: Wii Balance Board [Nintendo (2008a)]

4.3 Enderkennung der Geste

Wie bereits in Kap. 3.4 und 3.5 erwähnt, sollte der Schwerpunkt weiterer Untersuchungen auf der Enderkennung liegen, da die im Prototyp implementierten Verfahren entweder zu unerwartetem Systemverhalten führen können oder zusätzliche Modalitäten voraussetzen.

4.4 Usability-Studien

Um die bisher getroffenen Aussagen zu überprüfen und vor allem zukünftig implementierte Erweiterungen des Prototyps untersuchen zu können, ist eine Usability-Studie unabdingbar. Kernpunkt der Arbeit ist ein intuitiv und einfach bedienbares Bedienkonzept für Videospiele. Inwiefern und ob sich die zum Teil implementierten Lösungen und aufgezeigten Wege in der Praxis bewähren, kann nur eine abschließende Usability-Studie zeigen.

Literatur

- [A.R.T. 2008] A.R.T.: *DTrack*. Webseite. 2008. – URL http://www.ar-tracking.de/DTrack.23+B6Jkw9MA__.0.html. – Aufgerufen 2008-11-12
- [Boetzer 2008] BOETZER, Joachim: *Bewegungs- und gestenbasierte Applikationssteuerung auf Basis eines Motion Trackers*, HAW Hamburg, Bachelorarbeit, August 2008
- [Brown und Cairns 2004] BROWN, Emily ; CAIRNS, Paul: A grounded investigation of game immersion. In: *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2004, S. 1297–1300. – ISBN 1-58113-703-6
- [Gamma u. a. 1995] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLIS-SIDES, John: *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Professional, 1995
- [Gebhardt 2008] GEBHARDT, Nikolaus: *Irrlicht Engine*. Webseite. 2008. – URL <http://irrlicht.sourceforge.net>. – Aufgerufen 2008-12-10
- [Heitsch 2008] HEITSCH, Johann: *Ein Framework zur Erkennung von dreidimensionalen Gesten*, HAW Hamburg, Bachelorarbeit, August 2008
- [Logic3 2008] LOGIC3: *Tennisaufsatz für Wii Remote*. Grafik. 2008. – URL <http://www.logic3.com/details/?prod=211>. – Aufgerufen 2008-12-10
- [Menache 2000] MENACHE, Alberto: *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann Publishers, 2000
- [Nintendo 2008a] NINTENDO: *Balance Board*. Grafik. 2008. – URL <http://www.de-bug.de/games/wp-content/uploads/2008/04/balance-board-2.jpg>. – Aufgerufen 2008-12-10
- [Nintendo 2008b] NINTENDO: *Wii Remote*. Grafik. 2008. – URL http://www.ixbt.com/consoles/images/info_wii/wii_mote.jpg. – Aufgerufen 2008-12-10
- [Nvidia 2008] NVIDIA: *NVIDIA PhysX*. Webseite. 2008. – URL http://www.nvidia.com/object/nvidia_physx.html. – Aufgerufen 2008-12-10
- [Sony 2008] SONY: *EyeToy*. Webseite. 2008. – URL <http://www.us.playstation.com/PS2/Accessories/97036>. – Aufgerufen 2008-12-15
- [Stanford 2008] STANFORD: *Doug Engelbart 1968 Demo*. Webseite. 2008. – URL <http://sloan.stanford.edu/MouseSite/1968Demo.html>. – Aufgerufen 2008-11-12

Abbildungsverzeichnis

1	Wiimote [Nintendo (2008b)]	1
2	Konzeption der Software	4
3	Marker (Bogen)	5
4	Marker (Sehne)	5
5	Errechneter Vektor (vereinfacht; ohne x-Achse)	5
6	Grenzwerte	6
7	Tennisaufsatz für Wiimote [Logic3 (2008)]	9
8	Wii Balance Board [Nintendo (2008a)]	10
9	Powerwall	13
10	1. Steuerungsart - Überspannen	14
11	2. Steuerungsart - Abschuss per Wiimote	15
12	Grundaufbau	16
13	Getroffene Wand	17
14	Eingestürzte Wand	18

A Anhang

A.1 Prototyp in Labor 1080

A.1.1 Powerwall

Der Prototyp läuft auf der Powerwall (Abb. 9). An den Ecken der Powerwall befinden sich vier Trackerkameras, die den Spieler (bzw. die Marker) von vorne erfassen. Hinter dem Spieler befinden sich zwei weitere Kameras (auf dem Bild nicht zu sehen), so dass insgesamt sechs Kameras zur Verfügung stehen, von denen mindestens zwei freie Sicht auf einen Marker haben müssen, um ihn korrekt zu erfassen.



Abbildung 9: Powerwall

A.1.2 Erster Aufbau (vgl. Kap. 3.3.1)

Beim ersten Aufbau sehen wir den Aufbau der Steuerungsart *Überspannen* aus Kap. 3.3.1). Der grüne Balken rechts unten auf der Powerwall zeigt dem Spieler, wie weit der Bogen gespannt ist. Sobald der Balken am Anschlag ist, löst sich der Pfeil von der Sehne.

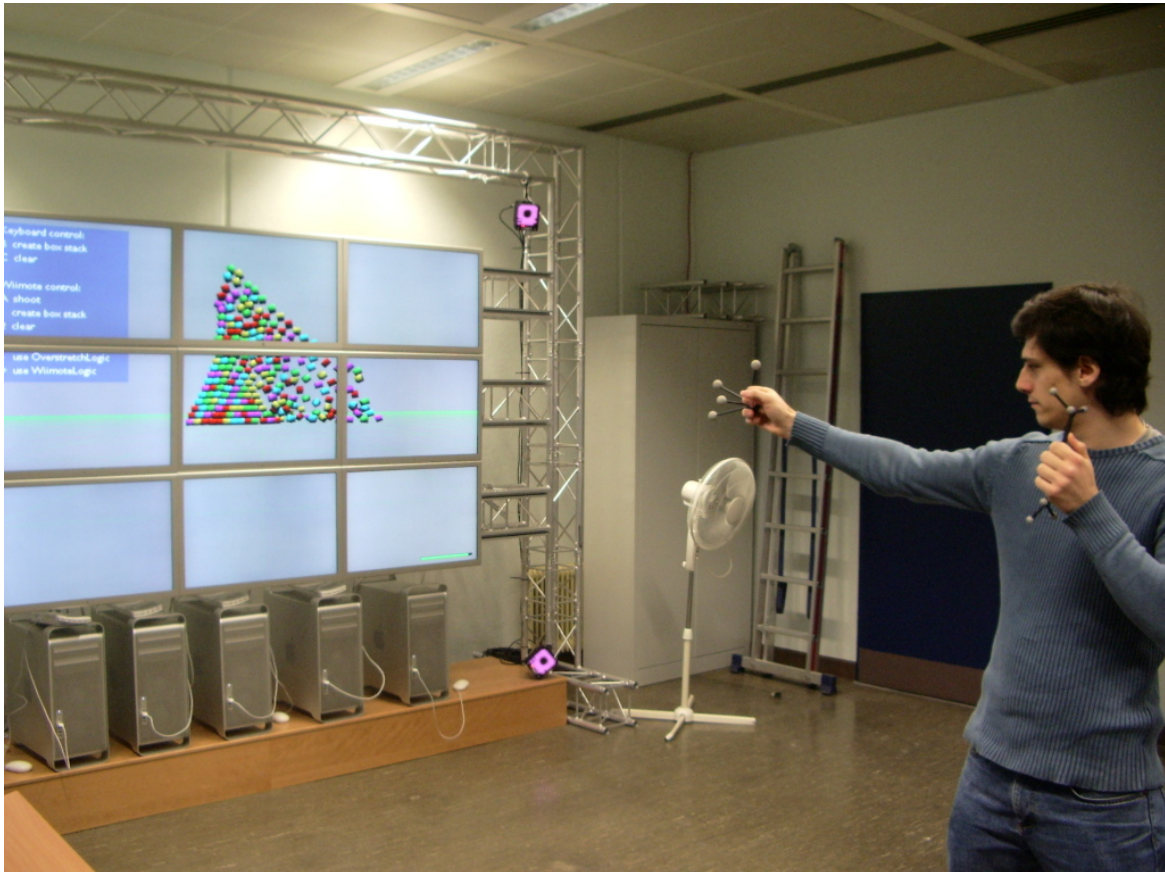


Abbildung 10: 1. Steuerungsart - Überspannen

A.1.3 Zweiter Aufbau (vgl. Kap. 3.3.2)

Beim zweiten Aufbau sehen wir den Aufbau aus Kap. 3.3.2. Der Spieler hält zusätzlich zur Sehne die Wiimote in der Hand und schießt per Tastendruck. Der rote Balken unten rechts signalisiert, dass der Spieler in die Grundposition zurück kehren muss (einen neuen Pfeil 'auflegen').

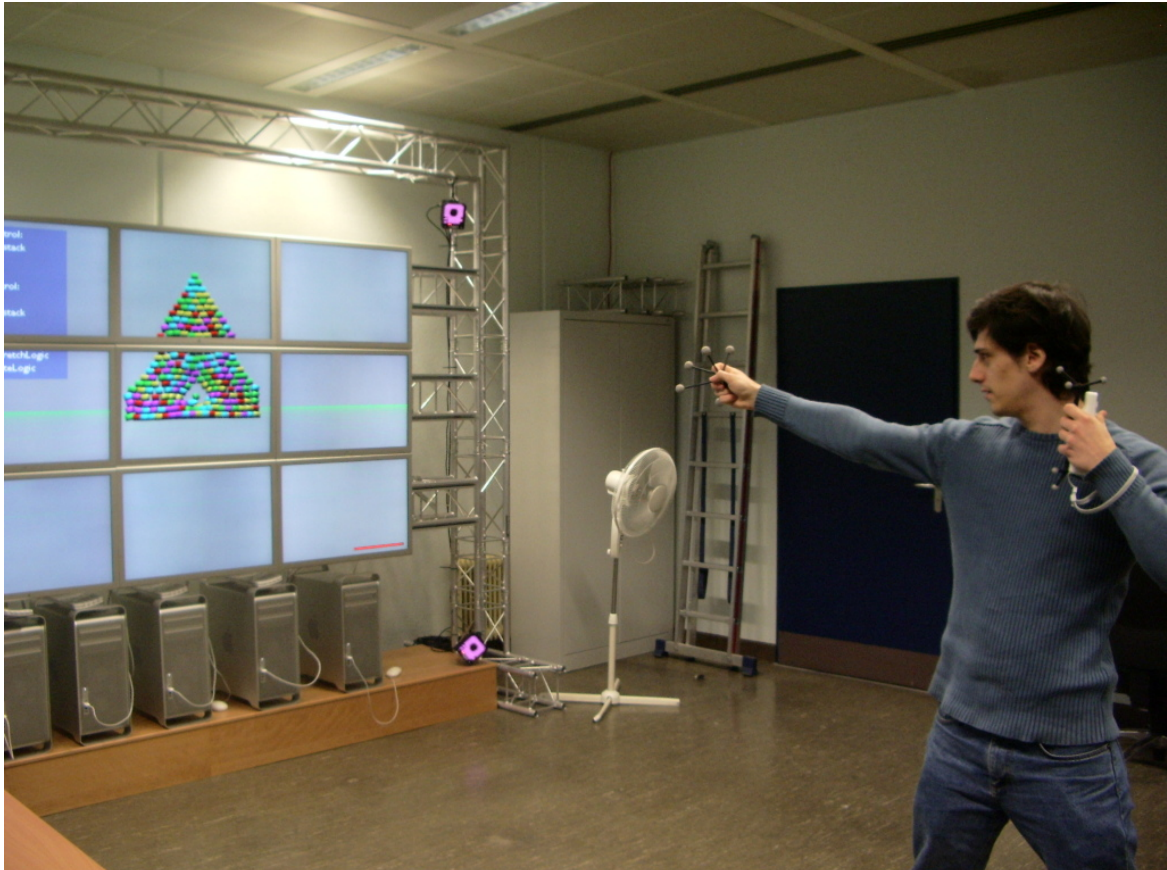


Abbildung 11: 2. Steuerungsart - Abschuss per Wiimote

A.2 Physikbeispiel

Der Grundaufbau der Simulation ist eine dreieckige Wand aus Würfeln (Abb. 12). Jeder Würfel ist ein eigenes physikalisches Objekt.

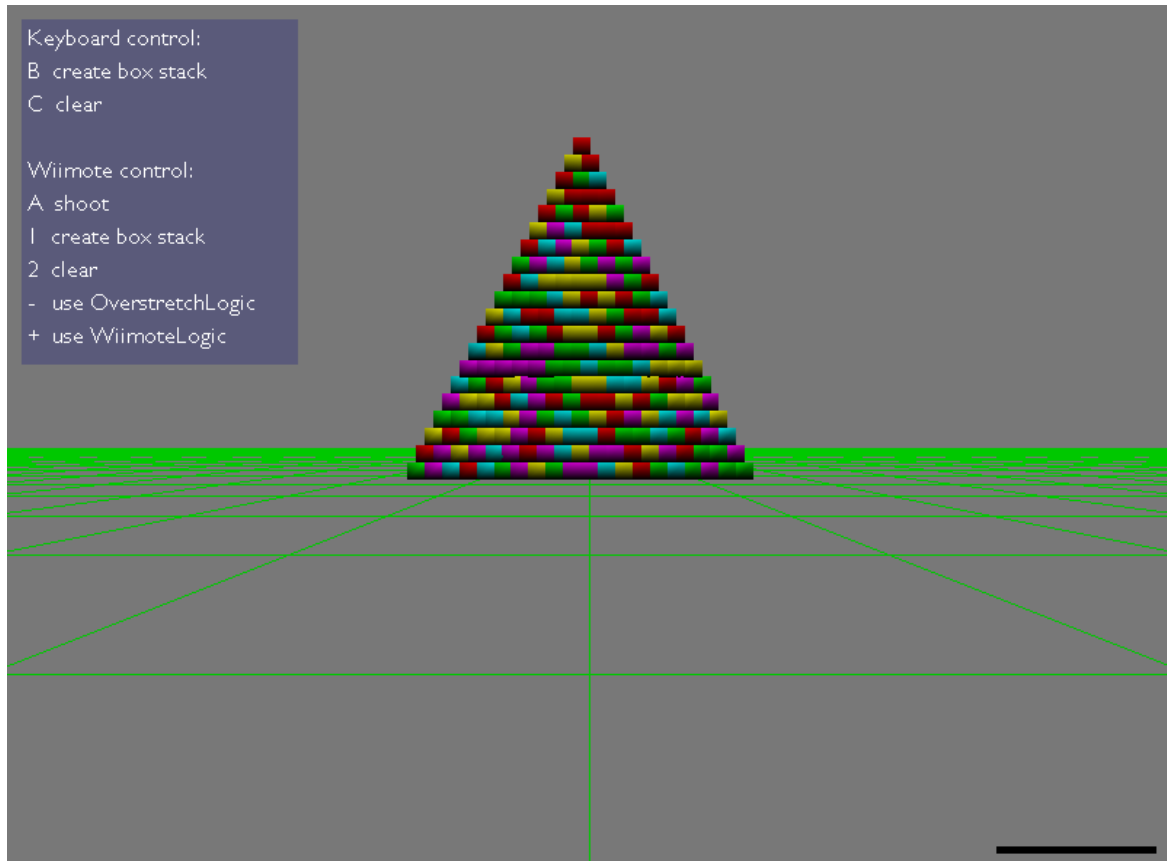


Abbildung 12: Grundaufbau

Abb. 13 zeigt, wie die Wand im unteren Teil getroffen wurde. Der Einschlag pflanzt sich bis in die Spitze fort.

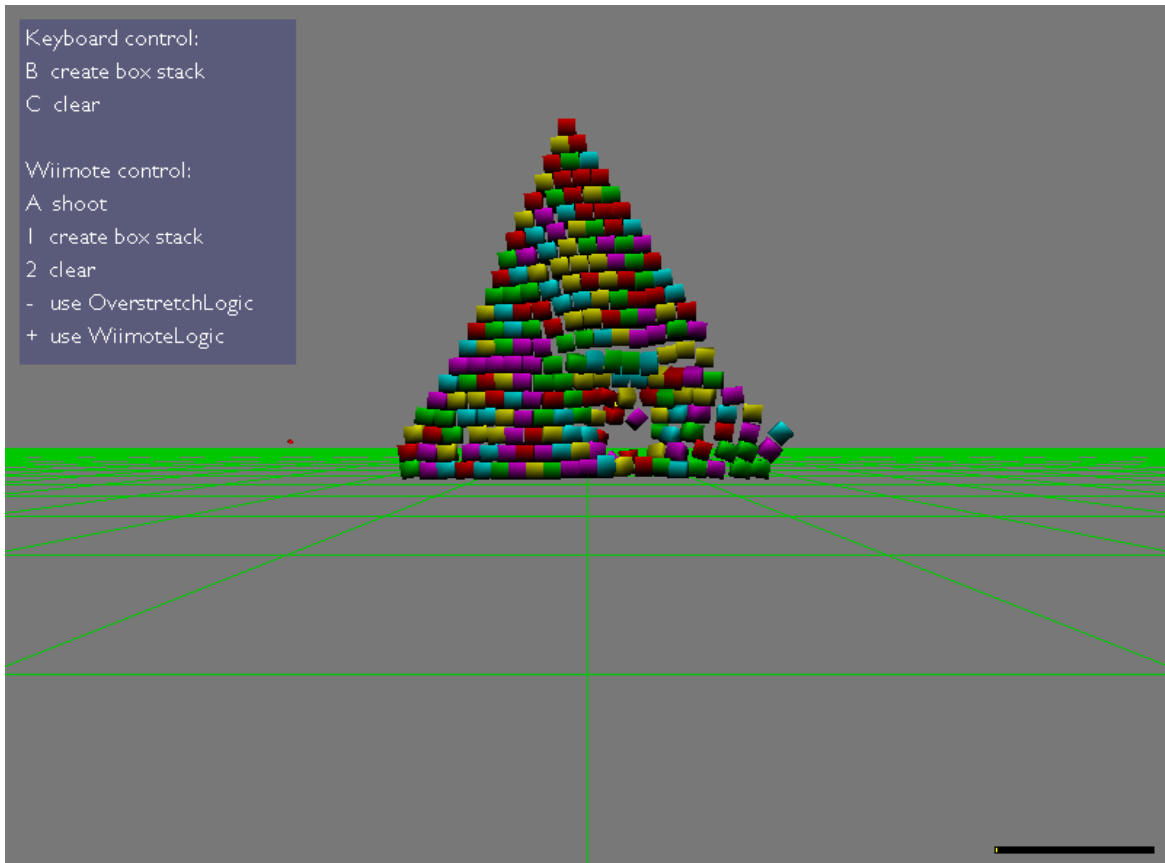


Abbildung 13: Getroffene Wand

Nachdem die Wand getroffen wurde und die Würfel z.T. keinen Halt mehr haben, stürzt die Wand in sich zusammen (Abb. 14).

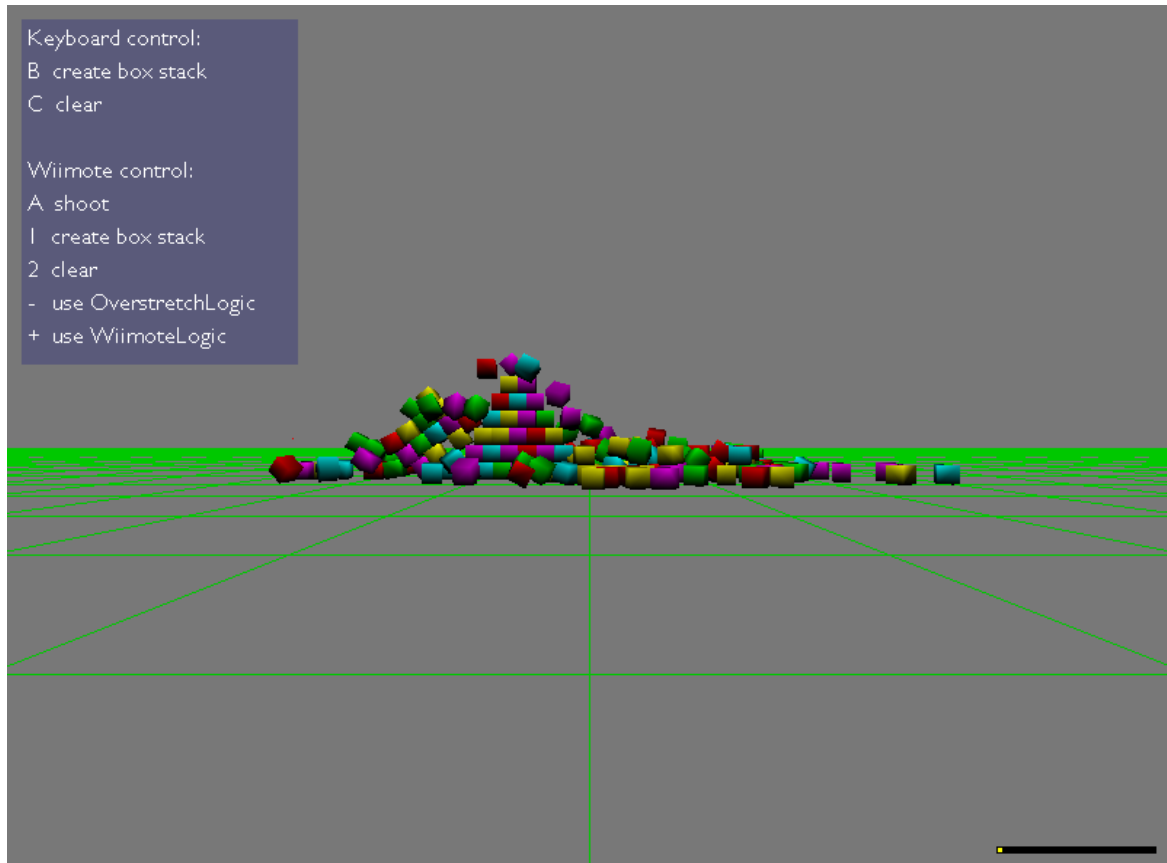


Abbildung 14: Eingestürzte Wand