

# Gesteninterpretation zur Transformation von 3D-Primitiven auf einem Touchscreen in Kombination mit der Leap-Motion

## Bachelorarbeit

zur Erlangung des Grades einer Bachelor of Science (B.Sc.)  
im Studiengang Computervisualistik

vorgelegt von  
Eva Kreckel

Erstgutachter: Prof. Dr.-Ing. Stefan Müller  
Universität Koblenz-Landau, Arbeitsgruppe Computergrafik  
Zweitgutachter: MSc. Gerrit Lochmann  
Universität Koblenz-Landau, AG Computergrafik

Koblenz, im März 2015

## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja    Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.       

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.       

.....  
(Ort, Datum)

.....  
(Unterschrift)



## Aufgabenstellung für die Bachelorarbeit

Eva Kreckel

(Mat. Nr. 211 200 445)

**Thema:       Gesteninterpretation zur Transformation von 3D-Primitiven auf einem Touchscreen in Kombination mit der Leap-Motion**

Steuerung von virtuellen 3D-Objekten wird in der Regel durch zweidimensionale Eingabegeräte wie Touchscreens oder Maus ermöglicht und eignet sich weniger für Steuerung, die sich im dreidimensionalen Raum auswirken sollen. Gesten im Raum können wiederum durch Spezial-Hardware wie die Leap-Motion interpretiert werden. Die Benutzerfreundlichkeit dieser berührungslosen Steuerung wird allerdings durch fehlende Haptik beeinträchtigt.

Ziel dieser Arbeit ist es, eine Gestensteuerung zu entwickeln, die eine einfachere Bedienung im dreidimensionalen Raum ermöglicht. Die Handbewegung vor dem Bildschirm werden mit Hilfe der Leap-Motion aufgenommen und durch die Haptik des Touchscreens ergänzt. Schwerpunkt der Arbeit ist die Interpretation der Gesten zur Bewegung (Rotation, Translation, Skalierung) von 3D-Primitive, wie Quader, Pyramide, Kugel und Zylinder und der Navigation im 3D-Raum.

Die inhaltlichen Schwerpunkte der Arbeit sind:

1. Kenntnisse über Verwendung der Leap-Motion und des Touchscreens erwerben
2. Realisierung der Auswahl und Positionierung der Primitiven
3. Konzept über die Gestensteuerung erstellen
4. Implementierung der Gestensteuerung
5. Implementierung der Transformationen durch die Gesten
6. Evaluation und Dokumentation der Steuerung

Koblenz, 02.10.2014

– Eva Kreckel –

– Prof. Dr. Stefan Müller –

## Zusammenfassung

Diese Arbeit befasst sich mit einer 3D-Interaktion von primitiven Objekten, welche von Gesten gesteuert wird. Dabei werden Funktionen, welche auf eine absolute Positionierung angewiesen sind, mit Hilfe eines Touchscreens implementiert. Als zweites Eingabegerät dient die Leap-Motion, welche Gesten anhand von zwei CCD-Kameras und drei Infrarot-LEDs aufnimmt. Die Gesten bestehen dabei aus der Translation, Rotation und Skalierung von Objekten, sowie aus einer Steuerung, die die Bewegung im Raum ermöglicht. Zur visuellen Umsetzung wurde die Arbeit in Blender mit der Blender Game Engine unter Python geschrieben. Nur das Auswählen der Objekte wurde mit Hilfe des Touchscreens realisiert. Diese Steuerung wurde anschließend mit einer reinen Maussteuerung evaluiert. Diese beiden Eingabemöglichkeiten unterscheiden sich darin, dass die Gestensteuerung nicht etabliert ist. Sie besitzt allerdings den Vorteil, dass sie im dreidimensionalen Raum ausgeführt werden kann. Die Maus hingegen ist geläufig, beschreibt aber nur einen zweidimensionalen Raum als Eingabe. Es stellt sich also die Frage, ob bei der dreidimensionalen Interaktion von Objekten eine Maussteuerung oder eine Gestensteuerung bevorzugt wird. Das Ergebnis beläuft sich daran, dass die Maus favorisiert wird. Jedoch liegt die Gestensteuerung in manchen Bereichen nah an dem Ergebnis der Maus.

## Abstract

This thesis shows an interaction of primitives in a three-dimensional space which is done by gestures. Functions which are difficult to do by gestures without any absolute feeling of the position are implemented with a touchscreen. Besides the touchscreen a second input device, a Leap-Motion, is used to obtain data of the motion of the hand. To get its data the Leap-Motion uses two CCD-cameras and three infrared LEDs. The interactions that can be done without any feedback of the absolute position are the translation, rotation and scale. These three and the movement through space are implemented as gestures in this thesis. This is done in Blender with the Blender Game Engine and Python. The only function which has been implemented for the touchscreen is to select an object. Later on, a comparative control of the mouse was invented to contrast it with the control of the gestures. There are two big differences between these two controls. On the one hand, the gesture controls can be done in a three-dimensional space but most people aren't used to it yet. On the other hand, there is just a two-dimensional input possibility with the mouse control. Otherwise it is familiar to most persons. The evaluation should reveal if people prefer interaction by mouse control or by gestures. The result shows that the preferred control is done by the mouse. However in some categories of the tests the gestures are quite close to the result of the mouse.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Stand der Technik</b>	<b>2</b>
<b>3</b>	<b>Programmierung der Gestensteuerung</b>	<b>4</b>
3.1	Einbindung in Blender . . . . .	4
3.2	Funktion des Touchscreens . . . . .	6
3.3	Hauptfunktion . . . . .	9
3.4	Translation der Objekte . . . . .	12
3.5	Rotation der Objekte . . . . .	14
3.6	Skalierung der Objekte . . . . .	16
3.7	Bewegung im Raum . . . . .	18
3.8	Weitere Verbesserungen des Programms . . . . .	22
3.9	Vorevaluation . . . . .	25
<b>4</b>	<b>Evaluation</b>	<b>26</b>
4.1	Evaluationsthesen . . . . .	28
4.2	Aufgaben der Evaluation . . . . .	28
4.3	Steuerung der Gesten und der Maus . . . . .	31
4.4	Evaluationsablauf . . . . .	32
4.5	Ergebnisse . . . . .	34
4.5.1	Bewegen im Raum . . . . .	34
4.5.2	Translation der Objekte . . . . .	37
4.5.3	Rotation der Objekte . . . . .	38
4.5.4	Skalierung . . . . .	39
4.5.5	Weitere Erkenntnisse . . . . .	41
4.5.6	Zusammenfassung der Ergebnisse . . . . .	42
<b>5</b>	<b>Fazit</b>	<b>45</b>
<b>6</b>	<b>Anhang</b>	<b>46</b>

# 1 Einleitung

Im Gegensatz zu den gewöhnlichen Eingabegeräten wie Maus, Touchpad oder Tastatur hat die Gestenerkennung einen großen Nachteil. Die Gestenerkennung findet berührungslos statt, was gute Algorithmen zur Erkennung erfordert. Bei anderen Eingabegeräten wie der Maus oder dem Touchpad hingegen, ist die Position immer genau bestimmbar. Das berührungslose Bedienen kann der Gestensteuerung aber auch zum Vorteil werden. Es erfordert nur eine schnelle Armbewegung und schon hat man im Browser gescrollt. Wie beim Touchscreen ist dabei immer ein Finger nötig, der den Bildschirm berührt. Der Aufwand mit der Maus ist dabei sogar noch größer.

Da die Erkennung der Hand und der Finger oft noch nicht ausgereift ist, können präzise Ausführungen noch nicht mit Gesten umgesetzt werden. Das Platzieren der Hand über einem Objekt, welches weiter in der Tiefe liegt, ist ohne Haptik schwer auszuführen. Ein weiteres Beispiel dafür ist das Auswählen eines Objektes. Da das Auswählen mit Gesten ohne präzise Steuerung rein visuell schon eine Herausforderung darstellt, wurde hierbei zusätzlich ein Touchscreen gebraucht. Dadurch konnte die fehlende Haptik in diesem Bereich umgangen werden. Die Gesten, welche sich auf das Verschieben, Rotieren und Skalieren von Objektes befassen, können mit visuellem Feedback verständlich gemacht werden. Die Bewegung hierbei ist relativ zur sich bewegendem Hand. Hier kann die Geste irgendwo im Raum beginnen, ohne dass sie mit der Position der Objekte im virtuellen Raum übereinstimmen muss. Das Auswählen der Objekte wäre hingegen absolut zur Hand, welches wie vorhin beschrieben ohne haptisches Feedback schwer handhabbar ist.

Des Weiteren bietet die Anwendung der Steuerung durch den Raum Grund zur Diskussion. Diese Bewegung wird in vielen Spielen gebraucht. Dabei ist die Frage, wie man sich am Besten durch den Raum bewegt. Auch in dieser Arbeit sollen dazu zwei unterschiedliche Methoden zur Bewegung im Raum vorgestellt und getestet werden.

Um dies zu realisieren, wurde als Hardware die Leap-Motion benutzt. Diese ist seit Juli des Jahres 2013 auf dem Markt [12], [5]. Zusätzlich ist diese Software auf Hand-, Finger- und Toolerkennung spezialisiert. Sie arbeitet ähnlich wie die Kinect von XBoX und besitzt dazu zwei CCD-Kameras, die das Licht der drei Infrarot-LEDs aufzeichnen [3]. Die Kinect wird heutzutage in vielen Spielen und auch bei manchen Robotersystemen verwendet und hat sich damit schon etabliert. Da die Leap-Motion jedoch auf eine Erkennung der Hand und der Finger spezialisiert ist, wurde für diese Arbeit das neuere Produkt gewählt. Denn die Gesten dieser Arbeit werden nur mit der Hand ausgeführt.

## 2 Stand der Technik

Gestenerkennung wird bereits oft in der Unterhaltsindustrie eingesetzt. Zunehmend findet sich diese auch im Alltag oder in der Arbeitswelt wieder. Der bekannteste Controller der Spielbranche ist dabei die Kinect von der Firma XBoX. Weitere Projekte sollen nun den Alltag und die Arbeitswelt erleichtern.

Ein Beispiel für den Alltag ist das Smart Home [17]. In diesem Zuhause soll mit Gesten das Licht an und ausgeschaltet, die Rollos runter oder hochgefahren oder der Fernseher ohne Fernbedienung bedient werden können. Diese Gesten bewirken eine Interaktion mit realen Objekten und nicht wie in Spielen üblich, mit Virtuellen. Das Ersetzen der Fernbedienung ist bereits mit SmartTVs [11] möglich. Dabei wird das Wechseln des Kanals oder das Einstellen der Lautstärke durch einfache Arm- oder Handbewegungen getätigt.

Innerhalb der Automobilindustrie wird ebenso an Verfahren zur Gestensteuerung und Interaktion mit Systemem gearbeitet. Zum Einen sieht die Firma IAV die Gestensteuerung als weitere Eingabemöglichkeit [3] während des Fahrens. Als Eingabegerät soll die Leap-Motion dienen. Viele Knöpfe und Funktionen können den Fahrer während der Autofahrt ablenken. Durch die Gesten sollen Funktionen wie das Wechseln des Radiokanals oder das Betätigen des elektronischen Fensterhebers vereinfacht werden. Des Weiteren gibt es Entwicklungen um bei der Qualitätskontrolle von Karosserieteilen Gesten einzusetzen [14]. Werden Schäden an einem Bauteil vom Prüfer erkannt, reicht ein Zeigen auf diese Stelle um den Schaden im System zu registrieren. Darauf aufbauend kann der Prüfer den Schaden weiter spezifizieren ohne dabei, wie zuvor, den Arbeitsplatz zu verlassen.

Allerdings bringen diese Gestensteuerungen im Gegensatz zu dieser Arbeit keine 3D-Interaktion im virtuellen Raum mit sich. Dennoch sind dies weitere Systeme, in denen die Gestensteuerung im Alltag neben den Spielen Einzug findet.

Des Weiteren bietet die Medizin ein weiteres Anwendungsgebiet. Innerhalb dieser Branche gibt es bereits erste Versuche die Gestensteuerung in den Operationssaal zu bringen [2]. Mit Hilfe der Gesten ist es möglich einen Bildschirm steuern, um zwischen Bildaufnahmen zu wechseln. Das Wechseln mit der Geste ist dabei einfacher und schneller. Allein auf Grund der sterilen Umgebung bietet sich das berührungslose Steuern an.

Anwendungen, in denen man mit der virtuellen Umwelt interagieren muss, gibt es auch im Bereich der Medizin. Der Air-Space, der App-Store der Leap-Motion, bietet dabei bereits eine Anwendung zur Sezierung von Tieren [4]. Dadurch können praktische Tätigkeiten zunächst am Computer erprobt werden. Das Lernen kann so in den Anfangsstadien ersetzt werden ohne dafür echte Tiere verwenden zu müssen.

Ansonsten bietet die Spielbranche ein weites Feld in der Gesten- und Be-

wegungssteuerung. Die Anwendung Kinect Training von Nike+ oder Dance Central von der Xbox beobachtet dabei den kompletten Körper. Die Spiele überprüfen dabei die Korrektheit der Bewegungen. Weichen die auszuführenden Bewegungen ab, wird dies mit visuellen und auch sprachlichen Hinweisen angedeutet.

### 3 Programmierung der Gestensteuerung

Zur Visualisierung der Gesten sollen Primitive bewegt, rotiert und skaliert werden können. Des Weiteren soll auch eine Navigation durch den Raum möglich sein. Um eine Umgebung zu schaffen, die bereits die Physik für die Primitiven und auch Eigenschaften zur Schwerkraft und Kollision besitzt, wurde Blender ausgewählt. Ebenso bietet dieses Programm die Formen der Primitiven, die Funktion des Lichtes und der Kamera. Die Leap-Motion stellt mehrere SDKs für unterschiedliche Programmiersprachen zur Verfügung, unter anderem Java, C++, C# , Python und weitere. Da Blender eine Python-Schnittstelle besitzt, wurde das SDK für Python von der Leap-Motion ausgesucht. Die verwendete Version von Blender ist hierbei 2.71. Bei der Leap-Motion wurde die Version 2.1.0 genutzt.

#### 3.1 Einbindung in Blender

Das Einbinden des SDKs von der Leap-Motion in Blender stellte zunächst ein Problem dar. Das Python, welches in der Blender-Version 2.71 genutzt wird, ist eine ältere Version, als die, die bei der Leap-Motion benutzt wird. Die Leap-Motion benutzt dabei die Python-Version 3.3.0, unterstützt aber auch die Version 2.7, siehe [9]. Um dieses Problem zu lösen, musste das Leap-Motion SDK auf eine vorherige Python-Version übertragen werden. Die Lösung des Problems wurde unter [15] vorgestellt. Da die Programmierung auf einem 64Bit Windows-Rechner stattfand, wird hier im Weiteren nur die Variante für diese Bit-Version von Windows besprochen.

Dieses Hindernis wurde letztendlich durch das Erstellen eines Python 3.3.0 Wrapper mit Hilfe von SWIG 2.0.9 überwunden. Der erste Schritt ist das Installieren von Visual Studio. Dabei kann sowohl die 2008-, 2010- oder auch 2012-Version benutzt werden. Weitere Programme, die installiert werden müssen, sind Python 3.3.0 und swig-2.0.9. Dazu sollte das SDK von der Webseite heruntergeladen worden sein. Anschließend muss mit Visual Studio ein leeres C++-Projekt erstellt werden. Die Dateien LeapMath.h, Leap.i und Leap.lib und Leap.h aus dem SDK von der Leap-Motion müssen nun dem leeren Projektordner hinzugefügt werden. Als nächstes wird die Konsole benötigt. Hierzu muss zunächst in den leeren Projektordner navigiert werden. Ist dies geschehen, muss swig mit folgendem Befehl ausgeführt werden.

```
"C : \ProgramFiles\swigwin - 2.0.9\swig.exe"  
- c + + - python - oLeapPython.cpp - interfaceLeapPythonLeap.i
```

Der Pfad gibt an, wo die exe-Datei von swig liegt. Die weiteren Parameter können dagegen ohne Änderungen übernommen werden.

Anschließend müssen die Projekteigenschaften geändert werden. Dazu wird erst auf Release umgeschaltet. Folgend darauf muss in den Reiter "General"gegangen werden. Hier muss nun "TargetName"mit LeapPython ersetzt und die Ausgabedatei muss auf eine Dynamic Library (.dll) gesetzt

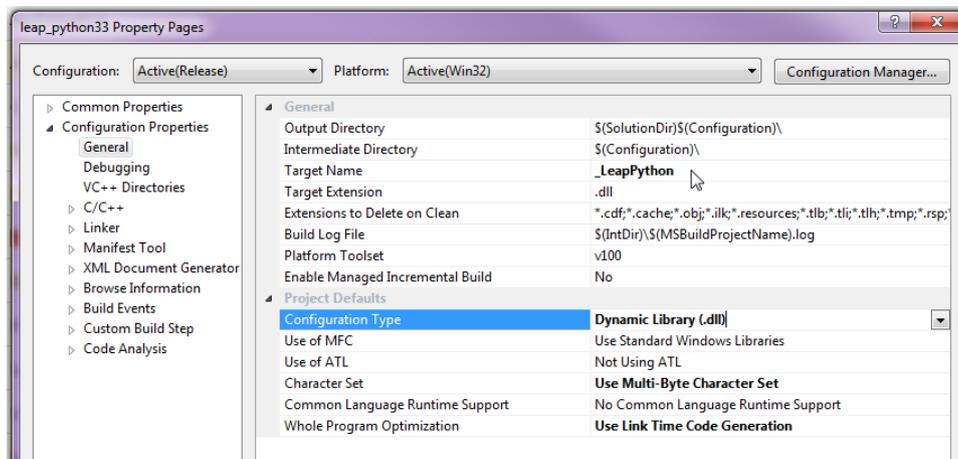


Abbildung 1: Änderungen der Eigenschaften im Reiter General

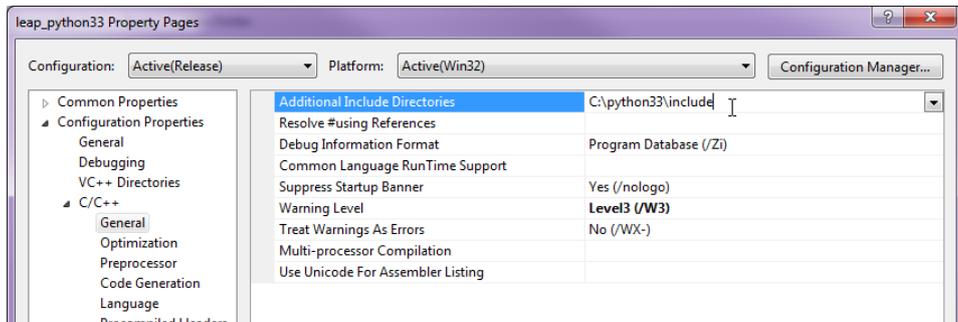


Abbildung 2: Änderungen der Eigenschaften im Reiter C/C++

werden. Eine grafische Hilfestellung ist unter der Abbildung 1 zu sehen.

Unter dem Reiter C/C++ -> General muss unter zusätzliche Bibliotheken der Pfad der Python.h angegeben werden. Meistens liegt diese Datei unter  $C : \backslash Python33 \backslash include$ . Das Beispiel für diesen Reiter ist unter der Abbildung 2 zu sehen.

Anschließend werden im Reiter Linker -> Input als zusätzliche Abhängigkeiten Leap.lib und der Pfad zu python33.lib gesetzt. Der Pfad hierzu ist bei keinen Änderungen während der Installation unter  $C : \backslash Python33 \backslash libs \backslash python33.lib$  zu finden. Die Änderungen der Einstellungen zum Reiter Linker sind in der Abbildung 3 zu erkennen.

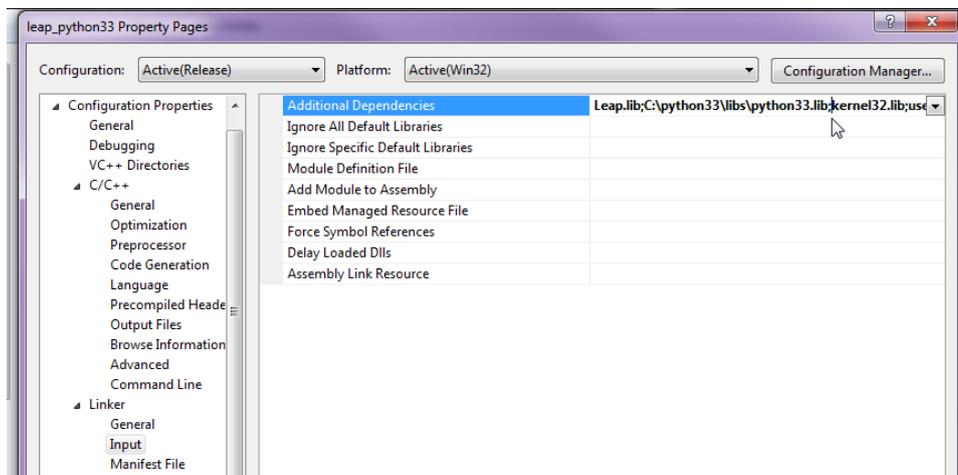


Abbildung 3: Änderungen der Eigenschaften im Reiter Linker

Nun muss das Projekt gebaut werden. Hierzu wird die Taste F7 gedrückt. Sollten Fehler auftreten, stammen diese von Bibliotheken, die vergessen wurden einzubinden oder daher, dass nicht alle Komponenten der 64-Bit Version zu Grunde liegen. Dazu gilt auch, dass in Visual Studio eine Win64 Konfiguration ausgewählt ist.

Zu guter Letzt muss die erzeugte LeapPython.dll in eine LeapPython.pyd-Datei umgenannt werden. Diese pyd- und die LeapPython.py-Datei müssen nun in den Ordner *python/lib/site-packages* im Blender-Verzeichnis hinzugefügt werden. Nun kann die Leap-Motion in Blender beim Importieren gefunden werden.

Die Anleitung zu diesem Wrapper und die Bilder aus Abbildung 1, 2 und 3 sind der Webseite [15] zu entnehmen.

### 3.2 Funktion des Touchscreens

Das Auswählen eines Objektes mit Hilfe einer Geste fällt bis heute schwer. Das genaue Positionieren der Hand beziehungsweise des Fingers, ohne ein optisches und fühlbares Feedback zu erhalten, ist nicht ganz leicht. Daher wurde in dieser Arbeit zum Selektieren von Objekten ein Touchscreen verwendet. Bei der Translation, Rotation und Skalierung hingegen, muss die Positionierung nicht absolut sein, was zu guten Ergebnissen einer Gestensteuerung führen kann.

Um nun das Auswählen zu realisieren, wird zunächst ein Listener für die Mausbewegung und zum Erkennen, ob sich die Maus über einem kollidierbaren Objekt befinden, gebraucht. Am Anfang der Arbeit wurde abgefragt, ob ein Links-Klick betätigt wurde. Da nun ein Touchscreen verwendet wird, werden beim Berühren keine Daten in Form von einem Links-Klick geliefert. Um dies zu umgehen, wird die Mausbewegung abgefangen. Dabei werden

beide Maus-Listener an das gleiche Skript gegangen. Da es möglich ist, dass die Maus, die sich über einem Objekt befindet, das Skript aufruft, bevor sich die Mausposition verändert hat, muss zuerst abgefragt werden, ob die Mausposition noch die Gleiche ist. Dazu wurde die jeweils letzte Mausposition im globalen Verzeichnis, `globalDict`, von Blender abgespeichert. Haben sich die Positionen nun geändert, gibt es drei mögliche Fälle, die eintreten können.

Im erste Fall ändert sich zwar die Mausposition, aber es wird immer noch das gleiche Objekt getroffen. Der zweite Fall hingegen, beschreibt das Auswählen eines anderen Objektes. Beim letzten Fall wird gar kein Objekt getroffen, was die Steuerung zur Bewegung durch den Raum auslöst. Hierbei wird also die Kamera selektiert.

Zunächst wird überprüft, ob sich die Mausposition geändert hat. Ist dies der Fall, wird die Farbe der Selektierung aufgehoben. Diese wurde durch ein zweites Objekt, welches als Kindobjekt mit einer etwas stärkeren Farbtexur belegt wurde, realisiert. Um nun die Selektierung aufzuheben, wird die Sichtbarkeit des Kindobjekts auf falsch gesetzt, um ein Rendern zu vermeiden. Da jedoch auch zuvor die Kamera hätte ausgewählt sein können, muss zunächst eine Abfrage stattfinden, ob das als selektiert deklarierte Objekt überhaupt zu den manipulierbaren Objekten, Würfel, Kegel oder Zylinder gehört. Denn wenn die Kamera zuvor das manipulierbare Objekt ist, sind bereits alle existierenden Kindobjekte unsichtbar und beim Versuch das Kindobjekt der Kamera, welches nicht existiert, aufzurufen, würden Fehler entstehen. Anschließend stellt sich die Frage, ob das aktuell getroffene Objekt ein Manipulierbares, ist. Sollte dies der Fall sein, wird das entsprechende Kindobjekt auf sichtbar gesetzt, um die Selektierung zu symbolisieren. Danach wird das getroffene Objekt als Variable im `globalDict` abgespeichert. Ist dies jedoch nicht der Fall, wird in diese Variable des selektierten Objektes die Kamera gespeichert. Die Umsetzung der Steuerung, wie sie hier beschrieben wurde, ist unter der Abbildung 4 als Diagramm dargestellt.

Verbesserungen in diesem Bereich können durch die Abfrage der auswählbaren Objekte geschehen. Beim Starten des Programms wird zunächst eine Liste der selektierbaren Objekte, Würfel, Zylinder und Kegel angelegt. Dadurch beschränkt sich die Abfrage allein darauf, ob das getroffene Objekt in dieser Liste vorhanden ist. Dies kann die Effizienz verbessern, wenn die Anzahl der Objekte ausreichend groß genug wird, siehe dazu [8]. Dies wurde ebenfalls in dieser Arbeit umgesetzt.

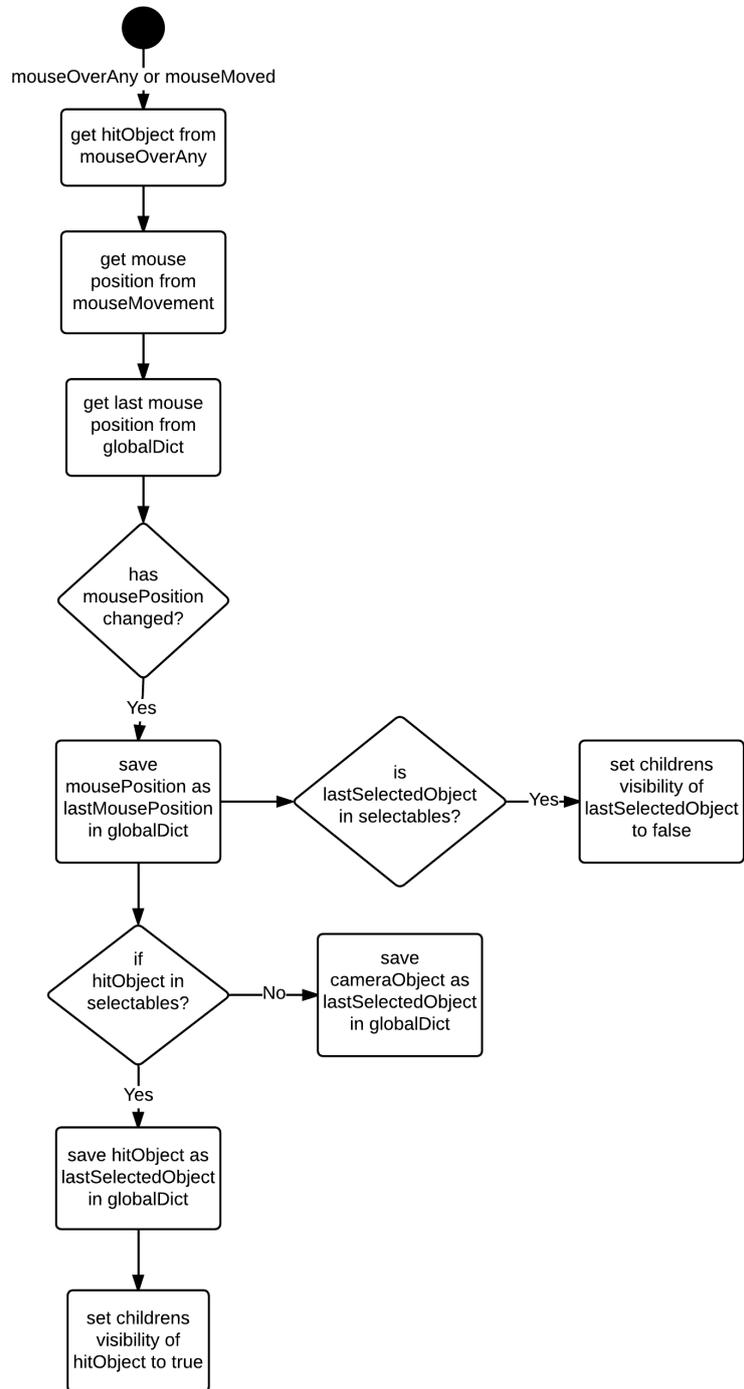


Abbildung 4: Umsetzung der Steuerung am Touchscreen als Diagramm

### 3.3 Hauptfunktion

Die Hauptfunktion des Programmes wurde durch einen Always-Trigger ausgelöst. Hierbei wurde ein Python-Skript aufgerufen, welches das Fallenlassen der Objekte nach einer Sekunde und das Aufrufen zur Manipulation der Objekte realisiert. Das Fallenlassen der Objekte ist dabei so umgesetzt, dass wenn ein Objekt manipulierbar war, also die Hand geschlossen ist, ein Zeitstempel in globalDict eingetragen wird. Beim Wiederaufrufen der main-Funktion, wird die aktuelle Zeit von der Vorherigen abgezogen. Ist die Zeitspanne dabei über einer Sekunde, wird die Dynamik von allen Objekten wiederhergestellt. Dabei muss die Dynamik von allen Objekten wiederhergestellt werden, da es möglich ist, ein anderes Objekt innerhalb dieser Sekunde auszuwählen. Dies würde sonst dazu führen, dass das zuvor ausgewählte Objekt ständig in der Luft schweben bliebe.

Die zweite Aufgabe, die diese main-Funktion hat, ist das Aufrufen der applyTransformation()-Methode im LeapController-Listener, der zuvor bei der Initialisierung erstellt wurde. Die Hauptaufgabe dieser Methode ist es festzustellen, ob nun eine Skalierung - bei zwei Händen - oder eine Translation beziehungsweise Rotation - bei nur einer Hand - ausgeführt wird. Ebenso überprüft diese Methode die Geschwindigkeit der Hand und wie stark sie geschlossen ist.

Die erste Abfrage beschäftigt sich damit, ob die Hand beziehungsweise die Hände aus dem letzten Frame geschlossen waren. Sollte dies nicht so sein, wird jede weitere Berechnung überflüssig. Hier wird zunächst nur überprüft, ob dies für den letzten Frame gilt. Der Grund dafür ist, dass die Hand, auch wenn sie geöffnet ist, weiterhin aufgezeichnet wird. Tritt der Fall ein, dass im aktuellen Frame die Hand geschlossen wird, führt dies zu verfrüht wahrgenommenen Bewegungen. Dazu kann das Objekt springen, wenn die Geschwindigkeit der Hand sehr groß ist. Um springende Objekte und zu frühe Bewegungen zu unterdrücken, wird an dieser Stelle nur der letzte Frame betrachtet.

Ist mindestens eine Hand aus dem letzten Frame geschlossen, muss nun die Geschwindigkeit abgefragt werden. Durch das Setzen einer Mindestgeschwindigkeit werden Sprünge beim Öffnen der Hand unterdrückt. Ebenso werden Bewegungen beim fast Stillhalten der Hand nicht wahrgenommen. Ohne die Mindestgeschwindigkeit kann nicht garantiert werden, dass das Objekt an seiner Stelle bleibt, auch wenn keine Handbewegung vom Anwender wahrgenommen wird. Dies liegt daran, dass das Gerät minimale Bewegungen aufzeichnet [12], wobei der Anwender diese Bewegungen nicht bewusst wahrnimmt. Dazu kann es daran liegen, dass die Daten des Controllers nicht immer zu 100% korrekt sind und jederzeit Schwankungen auftreten können.

Daraufhin rückt die Anzahl der Hände in den Fokus. Ist sowohl im Aktuellen, als auch im letzten Frame nur eine Hand vorhanden, so kann dies ein Zeichen für die Kamerabewegung oder für die Translation und Rotation eines

Objektes sein. Sind jedoch genau zwei Hände detektiert worden, muss der Anwender eine Skalierung des Objekt im Sinn gehabt haben. Sollte jedoch keine Hand erkannt worden sein, wird der Schwellwert, welcher eine Greifgeste erkennt, wieder auf den Ursprungswert gesetzt. Denn bei keiner erkannten Hand kann niemals eine Greifgeste im sichtbaren Bereich der Leap-Motion ausgeführt werden.

Ist nun nur eine Hand entdeckt worden, wird untersucht, zu wie viel Prozent die Hand geschlossen ist. Zu Beginn muss ein Schwellwert von 90% überschritten werden, um als Greifgeste zu gelten. Sind die 90% überschritten, wird dieser Schwellwert auf 70% herabgesetzt. Dies hat den Grund, dass wegen leichter Schwankungen die Geste zwischendurch nicht mehr erkannt wird und so die Transformationen nicht mehr umgesetzt werden, wenn der Schwellwert zu hoch bleibt. Aus der Perspektive des Anwenders hat das Gerät die Hand dann verloren. Dazu wird die Handstellung nicht mehr als Greifgeste gewertet, wenn sich die Geste nur leicht gelockert. Deshalb wird der Wert auf 70% herabgesetzt. Zu Beginn hingegen einen Wert von 70 zu wählen, führt dazu, dass die Greifgeste oft zu früh erkannt wird.

Ist die Greifgeste erkannt, wird die Position der Hand überprüft. Ist diese zu weit in negativer Z-Richtung, also zu nach am Touchscreen, findet ebenso keine Transformation statt. Dies ist notwendig, da beim Auswählen am Touchscreen die Hand zum darauf Klicken geschlossen wird. Um dadurch ungewollte Verschiebungen oder Rotationen zu vermeiden, wurde der Sichtbereich der Leap-Motion in negativer Z-Richtung eingeschränkt.

Anschließend folgt die Entscheidung, ob ein Objekt oder die Kamera momentan ausgewählt ist. Ist es die Kamera, wird die Verschiebung der Hand vom letzten Frame berechnet und mit Hilfe der Klasse Kamera weiterverarbeitet. Die weitere Verarbeitung der Daten ist dabei im Abschnitt 3.7 zu finden. Im Gegensatz dazu wird bei einem ausgewählten Objekt die Eigenschaften der Dynamik aufgehoben. Danach wird sowohl die Verschiebung als auch die Drehung der Hand seit dem letzten Frame berechnet. Ebenso wird die Kameraeinstellung benötigt. Sind diese Daten vorhanden, werden diese an die Funktionen der Translation und der Rotation weitergegeben. Auch hier ist die genaue Weiterverarbeitung unter dem Abschnitt 4.5.2 für die Verschiebung und unter Abschnitt 4.5.3 für die Rotation ausführlicher dokumentiert.

Sollte jedoch genau zwei Hände erkannt werden, wird zunächst kontrolliert, ob beide Hände aus dem aktuellen und aus dem vorherigen Frame geschlossen sind. Tritt dieser Fall ein, wird der Schwellwert wie vorhin beschrieben herabgesetzt. Ebenso wird die Dynamik des Objektes ausgeschaltet. Nun wird das Objekt, die beiden Hände aus beiden Frames und die Kameraorientierung an die Methode der Skalierung übergeben. Die Verarbeitung ist dabei unter dem Abschnitt 4.5.4 näher erläutert.

Der Ablauf der `applyTransformation()`-Methode ist im Diagramm in der Abbildung 5 dargestellt.

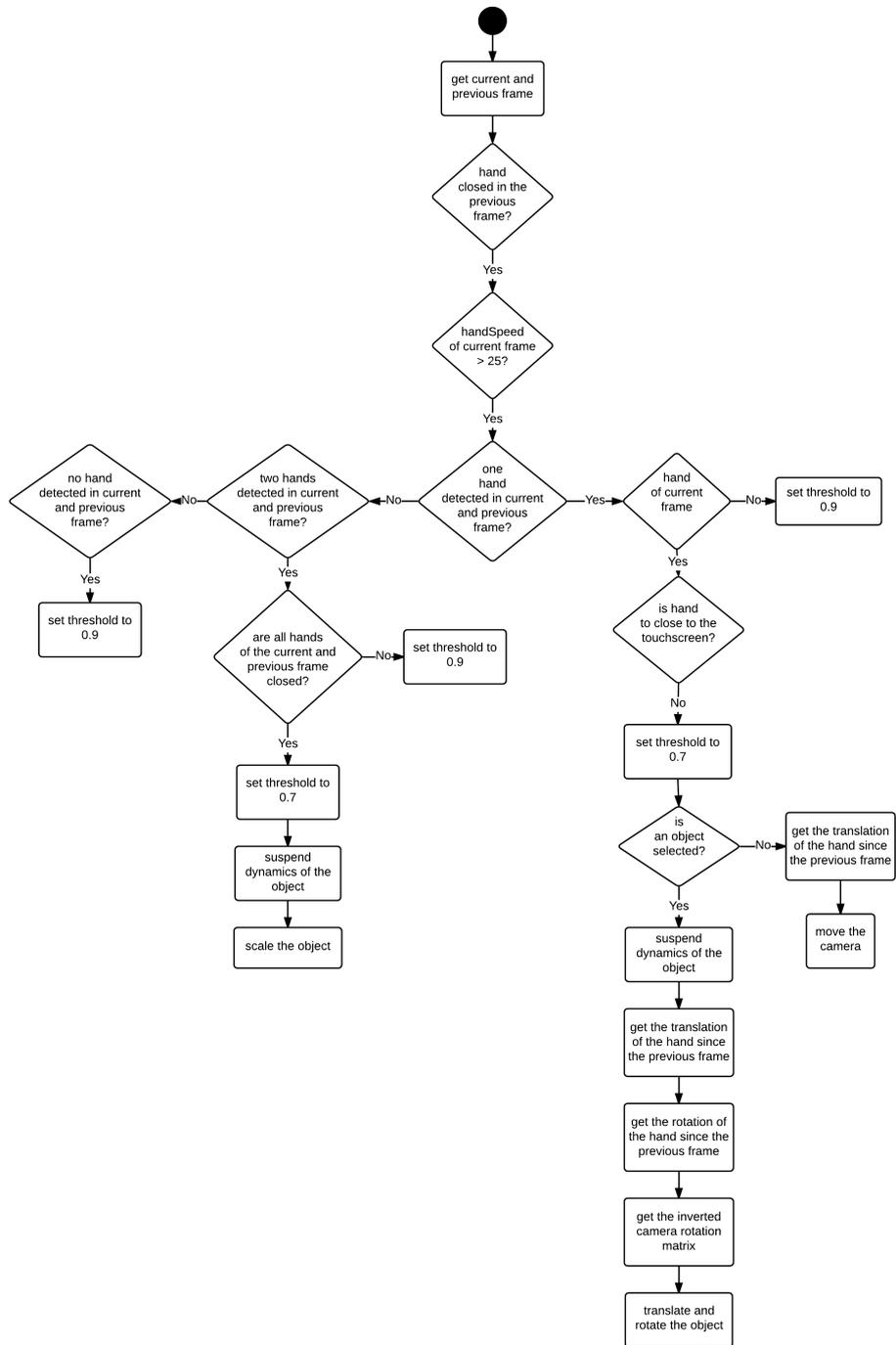


Abbildung 5: Umsetzung der applyTransformation()-Methode als Diagramm

### 3.4 Translation der Objekte

Um ein Objekt zu bewegen, muss die Hand geschlossen werden. Nun ist es wie als wäre das Objekt in der greifenden Hand. Dementsprechend wird die Hand dorthin bewegt, wo man das Objekt platzieren möchte, siehe Abbildung 8. Hier bietet die Leap-Motion bereits eine Funktion zur Wahrnehmung der Verschiebung der Hand vom Aktuellen bis zu einem andere Frame. Diese Funktion wurde genutzt um die Verschiebung der Hand in allen drei Raumrichtungen zu erhalten. Das Prinzip besteht darin, dass der Vektor der aktuellen Position vom Vektor der Position des vorherigen Frames abgezogen wird. Dies ist auch unter [8] nachzulesen. Da sich in dieser Arbeit die Kameraorientierung ändern kann, muss die Orientierung noch in Betracht gezogen werden. Die Verschiebung der Hand wird daher mit der invertierten Kameraorientierung transformiert. Dies soll das Ausüben der gleichen Geste ermöglichen, auch wenn sich die Orientierung der Kamera geändert haben mag. Danach wird die neue Position gesetzt. Der transformierte Vektor wird mit einem Faktor verringert, um die Geschwindigkeit der Objekte nicht zu groß geraten zu lassen. Um gleichzeitig die aktuelle Position des Objektes mit zu berücksichtigen, wird der neu errechnete Wert zur aktuellen Position addiert, siehe [8].

Bei kleineren Evaluationen gaben Probanden an, dass es sie stört, wenn gleichzeitig die Translation und Rotation möglich ist. Um dies in geringem Maße einzudämmen, wird die Winkelgeschwindigkeit des selektierten Objektes betrachtet. Wenn diese groß wird, dreht sich das Objekt. Die Translation ist dann nicht mehr durchführbar. Ist die Veränderung im Winkel des Objektes jedoch gering, ist die Verschiebung des Objekt gleichzeitig mit der Rotation möglich. Dies ist kein vollkommenes Trennen der beiden Transformationen. Die Rotation ist immer möglich, wobei die Translation von der Rotation abhängt. Da sich die Meinungen in der Frage spalten, ob eine Trennung der Transformationen gut sei, wurde dieser Mittelweg gewählt, um beides gleichzeitig zu realisieren. Die Translation wurde dabei an die Rotation gekoppelt, da die Wahrscheinlichkeit höher ist, gleichzeitig eine Verschiebung bei der Rotation auszuführen als bei der Translation eine Rotation. Dies ergibt sich aus der Ausführung der jeweiligen Gesten. Beim Rotieren wird die Hand automatisch ein bisschen verschoben. Hingegen ist es leichter eine Translation durchzuführen ohne dabei die Hand zu drehen. In Abbildung7 kann die Translation als vereinfachter Ablauf betrachtet werden.

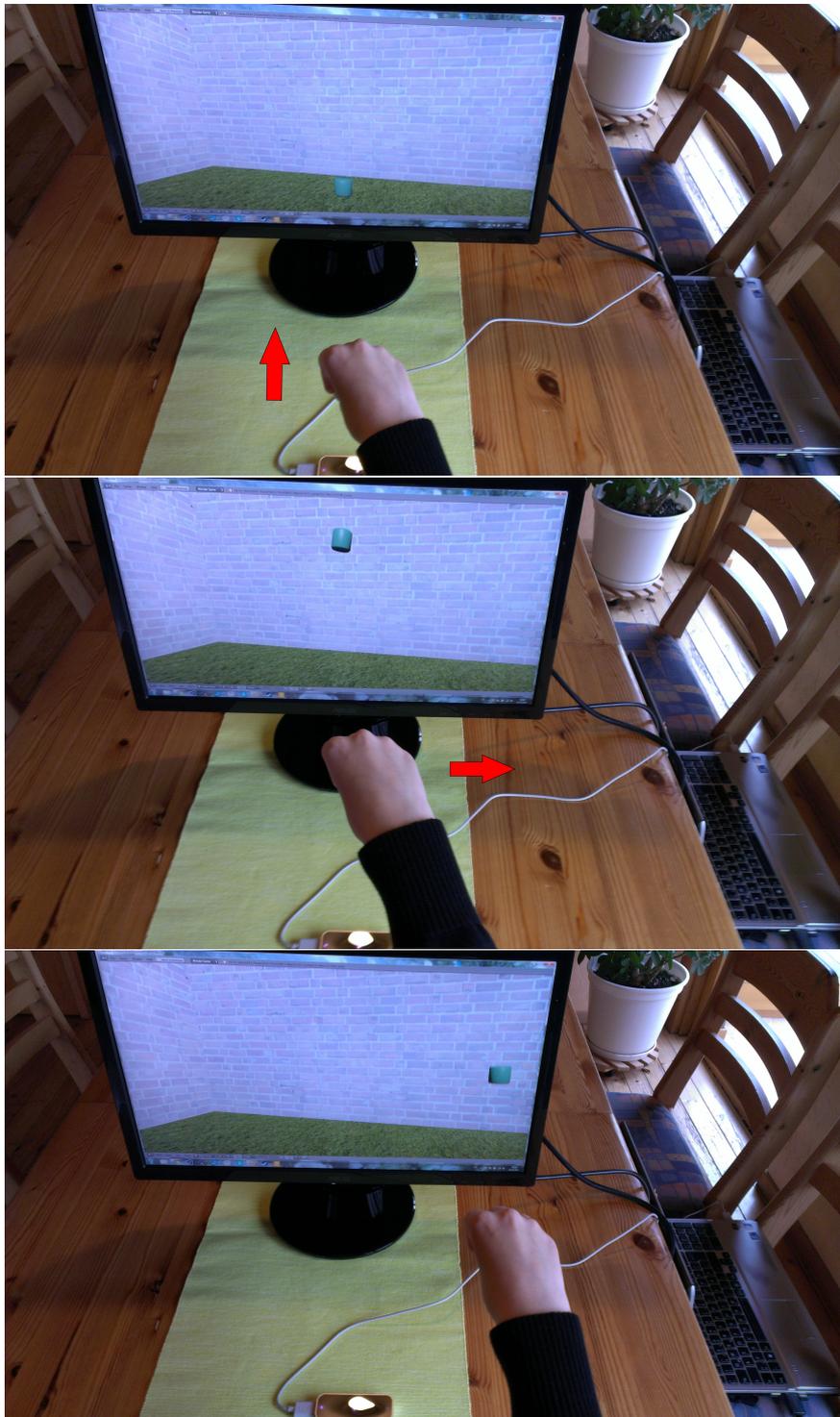


Abbildung 6: Geste der Translation

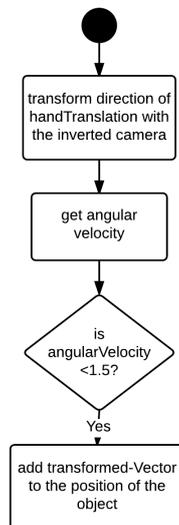


Abbildung 7: Umsetzung der Translation als Diagramm

### 3.5 Rotation der Objekte

Bei der Geste zur Rotation wird die Hand ebenfalls zunächst geschlossen. Anschließend wird die Hand gedreht, siehe Abbildung 8. Dabei wirkt es, als habe man das ausgewählte Objekt in der Hand und wollte es von einer anderen Seite aus betrachten. Beim Rotieren wird berechnet, um wie viel Grad sich die Hand um die drei Achsen jeweils dreht. Hier wird ebenfalls eine Implementierung von Seiten der Leap-Motion geliefert. Dennoch wäre eine andere Geste zum Rotieren hilfreich, da das Drehen der Hand nur bis zu einem bestimmten Punkt möglich ist. Dazu mehr im Abschnitt 4.5.3 der Evaluation.

Die Eingabe zur Rotation ist die Verdrehung der Hand zum vorherigen Frame. Daraufhin wird die Verdrehung mit der momentanen Kameraeinstellung transformiert. Durch diese Transformation ist es später möglich, die Geste gleich auszuführen, auch wenn die Kamera einer neuen Rotationsmatrix unterliegen sollte. Das Weiteren wird für jede Achse eine Rotationsmatrix mit Hilfe des transformierten Vektors erstellt. Um das Objekt zu rotieren, werden diese drei neu erstellten Matrizen zunächst miteinander multipliziert, um dadurch die Rotationsmatrix für die Drehung der Hand zu erhalten. Da das Objekt zuvor schon rotiert sein kann, wird die berechnete Rotationsmatrix mit der Vorherigen multipliziert. Das Ergebnis wird dem Objekt schlussendlich als neue Rotationsmatrix übergeben. Abbildung 9 stellt dabei den zeitlichen Ablauf dar.

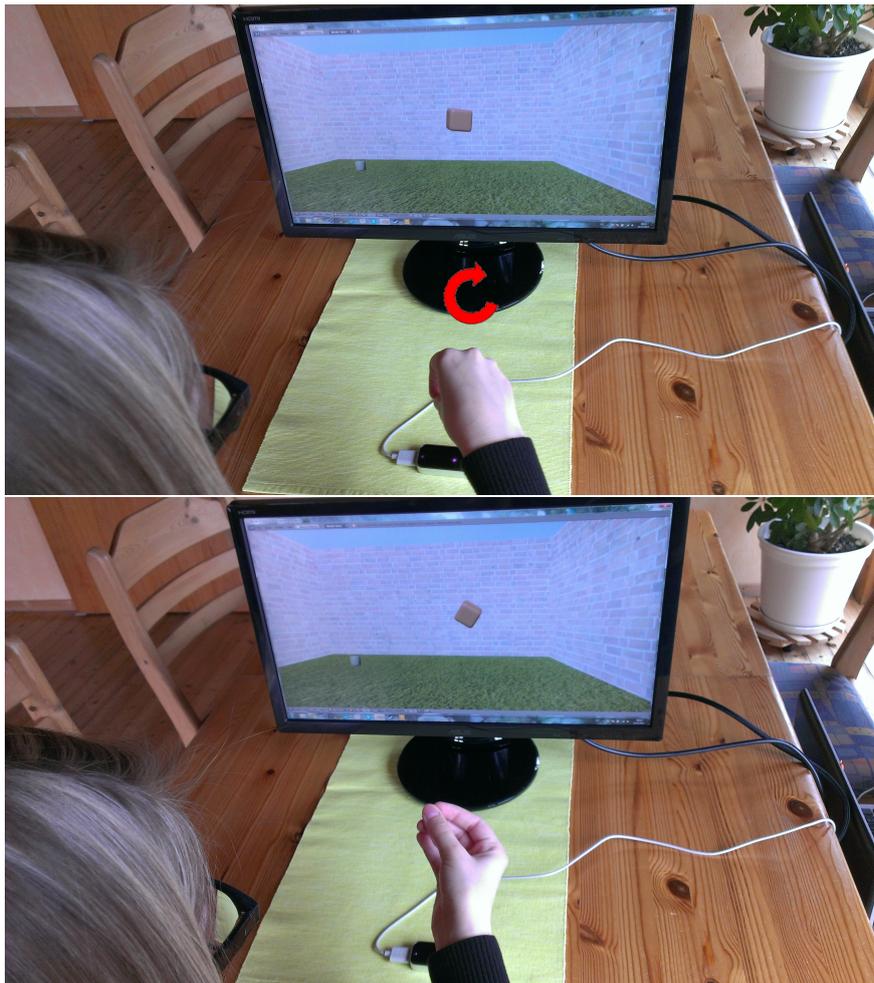


Abbildung 8: Geste der Rotation

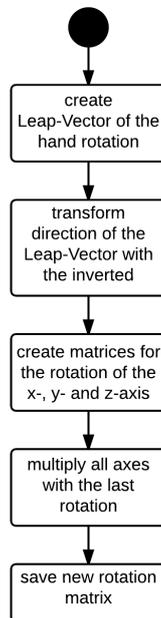


Abbildung 9: Umsetzung der Rotation als Diagramm

### 3.6 Skalierung der Objekte

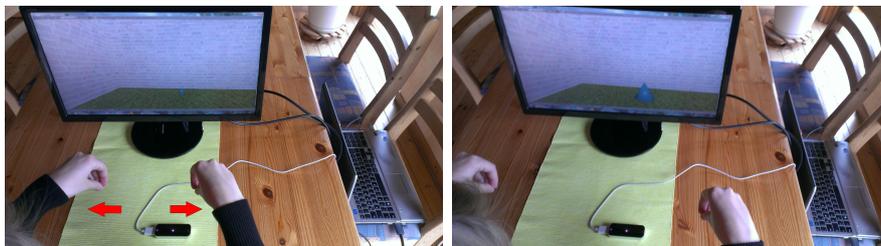


Abbildung 10: Geste der Vergrößerung

Die Skalierung der Objekt ist nur mit zwei Händen gleichzeitig umsetzbar. Hierbei ist die Geste wie folgt: Bewegen sich die Hände auseinander, vergrößert sich das selektierte Objekt. Bewegen sich die Hände aufeinander zu, wird das Objekt kleiner. Die Gesten sind dazu in der Abbildung 10 und der Abbildung 11 dargestellt.

Damit das Programm bei der Ausführung auf keine Ausnahmen, wie das Fehlen einer zweiten Hand stößt, wird die Methode zur Skalierung nur betreten, wenn im Aktuellen und im vorherigen Frame genau zwei Hände erkannt wurden. Nun gibt es mehrere Möglichkeiten zur Umsetzung der Erkennung dieser Geste.

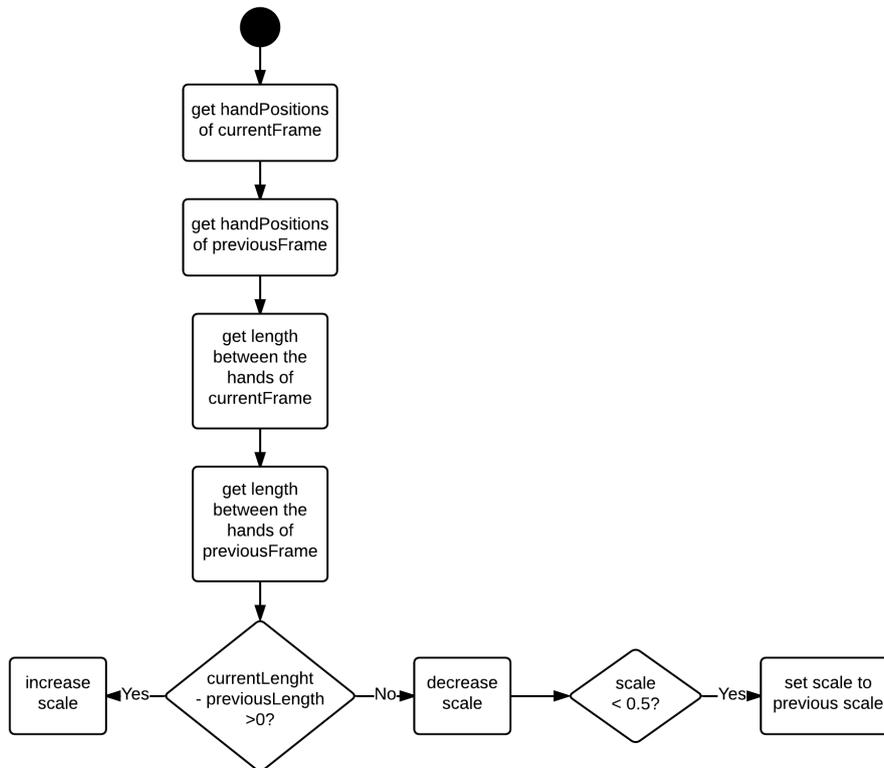


**Abbildung 11:** Geste der Verkleinerung

Zum Einen ist es denkbar, die Veränderung der Distanz der Hände mit gleichen IDs auszurechnen und anschließend aus diesen Werten eine Entscheidung zu treffen. Diese Option birgt jedoch einige Fehlerquellen. Bei den Daten, die von der Leap-Motion geliefert werden, kann es sein, dass die IDs der Hände wegen falscher Erkennung wechseln beziehungsweise sich generell ändern. Die Auswertung führe damit zu sehr großen Distanzänderungen. Dies bedeute laut Daten eine sehr schnelle Bewegung der Hände seitens des Anwenders, was in eine große Skalierung münden würde. Außerdem sind die zwei Hände sozusagen unabhängig voneinander. Die Hände vom Aktuellen und vom vorherigen Frame werden zwar miteinander in Bezug gebracht, aber während die eine Hand eine Verkleinerungsgeste ausführt, könnte die andere Hand eine Vergrößerung bewirken. Die Hände würden sich dabei jeweils in die gleiche Richtung bewegen. Dies kann ebenso zu unerwünschten und nicht erwartungsgemäßen Reaktionen führen. Diese Version wurde in den ersten Stufen dieser Arbeit umgesetzt.

Um diese zwei Fehlerfälle zu umgehen, wurde das Konzept überdacht und zu folgendem weiter entwickelt: Um das Verwechseln der IDs seitens des Controllers zu verhindern, wird zum Einen die Distanz der Hände vom aktuellen Frame und die Distanz der Hände vom vorherigen Frame berechnet. Selbst wenn sich die IDs der Hände nun unterscheiden sollten, bleibt das Ergebnis gleich, da bei der Distanzberechnung quadriert wird. Dadurch ist auch das Problem der Unabhängigkeit gelöst. Das gleichzeitige Ausführen der zwei Gesten ist nicht mehr möglich, denn nun werden die Hände pro Frame miteinander in Verbindung gesetzt. Sollten sich die Hände in die gleiche Richtung bewegen, bleibt die Distanz der Hände vom Aktuellen zum vorherigen Frame ungefähr gleich. Die Änderungen halten sich dabei im Rahmen. Durch einen Multiplikationsfaktor wird der Wert so gering, dass er kaum sichtbare Änderungen bewirkt.

Die Auswertung der Distanzen findet in dieser Arbeit wie folgt statt: Die Distanz der Hände vom aktuellen Frame wird von der Distanz des vorherigen Frames abgezogen. Wenn die Distanz sich vergrößert, also die Hände auseinander gehen, ist das Ergebnis kleiner null. In diesem Fall wird auf die aktuelle Skalierung ein Faktor addiert. Sollten sich die Hände jedoch aufeinander zu bewegen, verringert sich die Distanz und das Ergebnis der Berechnung wird



**Abbildung 12:** Umsetzung der Skalierung als Diagramm

positiv. In diesem Fall wird von der aktuellen Skalierung der gleiche Faktor abgezogen. Allerdings muss bei der Verkleinerung beachtet werden, dass die Skalierung auf null gehen kann. Sollte dies passieren, ist das Objekt nicht mehr sichtbar und auch nicht mehr selektierbar. Deshalb ist eine Abfrage der aktuellen Skaliergröße von nöten.

Die Skalierung, wie sie in dieser Arbeit vorliegt, wird in alle drei Raumrichtungen gleich gehandhabt. Es wäre ebenso denkbar, dies so zu erweitern, dass die Geste auf die verschiedenen Raumrichtungen eingeht.

### 3.7 Bewegung im Raum

Hierbei gibt es jeweils zwei Gesten. Eine, um sich im Raum vor und zurück zu bewegen und eine, um sich im Selben zu Drehen. Diese Gesten können nur ausgeführt werden, wenn kein Objekt ausgewählt ist und somit die Kamera das bewegende Objekt ist. Die Steuerung der Kamera wird dabei nur durch eine Hand realisiert. Wird von der Leap-Motion mehr als eine Hand erkannt, ist die Steuerung nicht möglich. Die Gesten funktionieren wie bei der Transformation der Objekte, um das Erlernen weiterer Gesten zu ver-

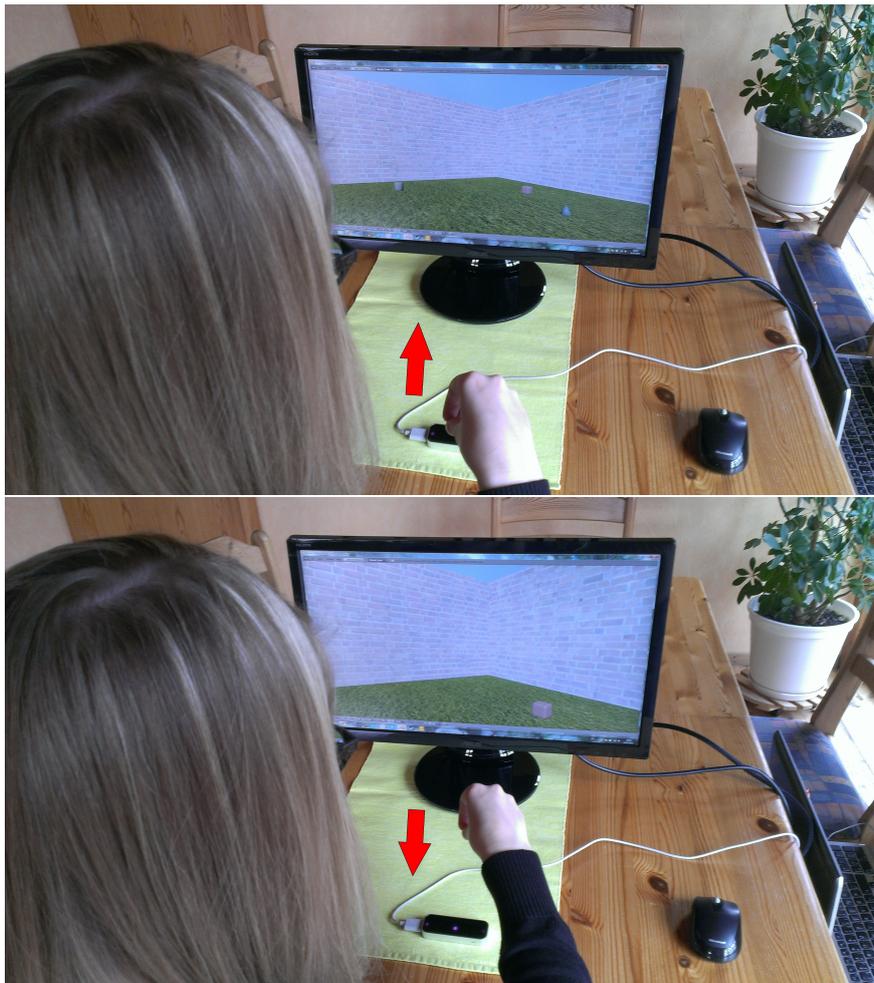


Abbildung 13: Geste der Bewegung durch den Raum

hindern. Denn das gleichzeitige Erlernen von Gesten beschränkt sich auf die ungefähre Anzahl von sieben Stück, [8]. Dies hat Hintergründe bei der Verarbeitung im Arbeitsgedächtnis.

Hierbei muss auch die Hand geschlossen werden, um ein Signal zur Ausübung einer Geste zu geben. Dabei ist die Steuerung wie folgt: Zum Bewegen der Kamera, wird die Hand nach vorne oder hinten bewegt, siehe 13. Beim Bewegen der Hand nach rechts oder links, dreht sich die Kamera auf der Stelle. Da es hierbei zwei verschiedene Versionen gab, ist die genaue Steuerung im Abschnitt 4 nachzulesen.

Diese Steuerung betrachtet nur die Verschiebung in X- und Z-Richtung. Die Unterscheidung, ob nun rotiert oder verschoben werden soll, wird anhand eines Schwellwertes in der X-Richtung getroffen. Ist also die Veränderung der Position der Hand in X-Richtung groß genug, wird rotiert, ansonsten wird die Kamera verschoben. Dies bedeutet auch, dass die Rotation je nach Schwellwert bevorzugt oder benachteiligt behandelt werden kann. Durch die Einschränkungen können ungewollte Bewegungen verhindert werden. Es wird bewusst die Y-Achse nicht berücksichtigt, um größere Orientierungslosigkeit zu vermeiden [8].

Beim Drehen im Raum wird zuerst auf die Verschiebung entlang der X-Achse ein Faktor hinzugerechnet, um den Wert möglichst klein zu halten. Dieser Wert wird anschließend als Winkel betrachtet. Aus diesem Winkel wird nun eine Rotationmatrix für die Z-Achse erstellt. Das Rotieren der Kamera steht dann für das Multiplizieren mit der aktuellen Rotationmatrix der Kamera. Da das Licht statisch ist, wird die errechnete neue Rotationmatrix ebenfalls mit der Rotationmatrix des Lichtes multipliziert. Würde das Licht nicht transformiert werden, könnte man bei einer  $180^\circ$ -Drehung die Objekte nur noch als schwarz erkennen, da gegen das Licht geschaut wird.

Das Gleiche gilt auch beim Verschieben der Kamera. Auch hier muss das Licht mitbewegt werden. Dabei wird die Veränderung der Position der Hand in Z-Richtung, also die Verschiebung nach vorne und hinten, betrachtet. Zusätzlich muss die Orientierung der Kamera auch mitberechnet werden. Wäre dies nicht der Fall, müsste nach einer  $180^\circ$ -Drehung die Fortbewegungs-Geste spiegelverkehrt ausgeführt werden. Da sich die Kamera jedoch nur um die Z-Achse dreht, sind in der Rotationsmatrix der Kamera nur die ersten zwei Werte aus der ersten Zeile notwendig. Dabei gibt der erste Wert den Cosinus-Wert der Drehung und der zweite Wert den Sinus-Wert an. Diese Werte werden jeweils mit der Verschiebung der Hand und einem Translationsfaktor multipliziert. Dabei wird der Cosinus-Wert für die Y-Achse der Kamera und der Sinus-Wert für die X-Achse der Kamera benötigt. Die jeweiligen Werte, die sich daraus ergeben, werden abschließend von den aktuellen Positionen der Kamera und des Lichts abgezogen.

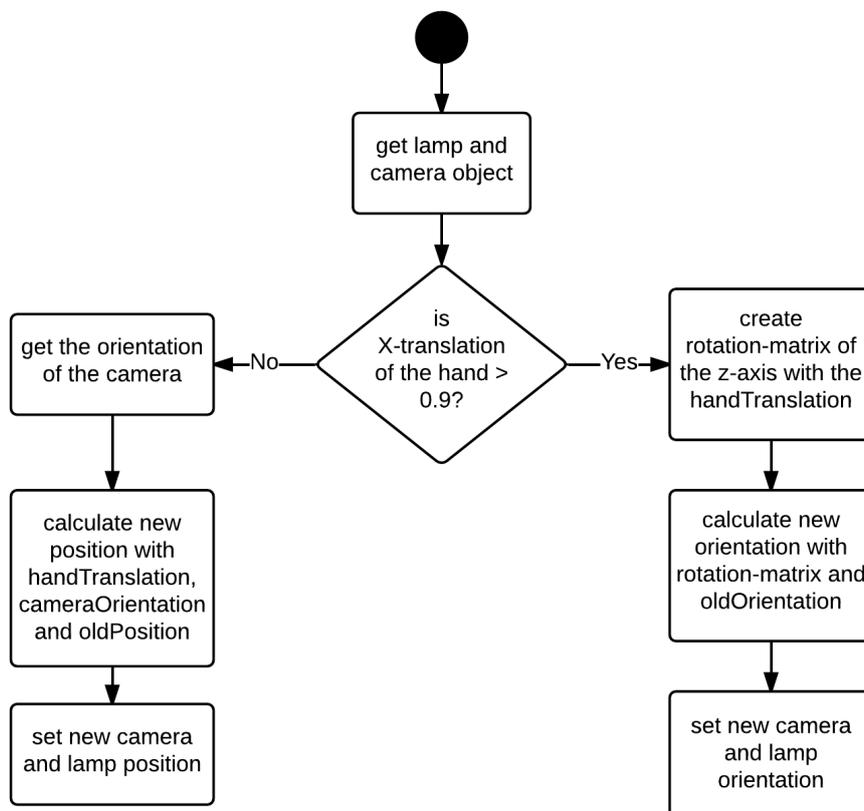


Abbildung 14: Umsetzung der Kamera als Diagramm

### 3.8 Weitere Verbesserungen des Programms

Während der Entwicklungsphase wurden immer wieder kleinere Evaluationen gestartet, um die Benutzerfreundlichkeit des bisherigen Programms festzustellen. Zu Beginn war das Umfeld eine weite Ebene. Es gab keine abgrenzenden Mauern und die Objekte konnten in alle Richtungen verstreut bewegt werden. Da zu diesem Zeitpunkt die Steuerung mit der Kamera noch nicht entworfen war, war ein Neustart des ganzen Programmes notwendig. Die Objekte waren nicht mehr auswählbar, weil sie außerhalb des Sichtfeldes gerieten. Eine kleinere Evaluation zu Beginn ergab, siehe Abschnitt 3.9, dass sich die Probanden eine Begrenzung für die Objekte wünschen, um das Verschwinden der Objekte zu vermeiden.

Um den verfügbaren Raum anzuzeigen, wurden fünf Ebenen aufgespannt. Eine für den Boden und jeweils vier für die Wände. Anschließend wird in einem Python-Script, das mit einem Always-Trigger verbunden ist, abgefragt, ob alle Objekte innerhalb dieser Beschränkungen liegen. Um die Abfrage zu vereinfachen, wurden die fünf Ebenen unter einem Empty-Objekt zusammengefasst. Dieses Empty-Objekt wurde auf die gewünschte Größe des Raumes skaliert. Daraufhin musste nur die Orientierung der fünf Ebenen entsprechend angepasst werden. Da alle unter einem Objekt zusammengefasst wurden und sonst keine Verschiebung seitens der Ebenen stattfand, betrifft die Abfrage nur das Empty-Objekt. Jede Achse wird darauf kontrolliert, ob die Position aller Objekte jeweils kleiner oder größer der Skalierung des Empty-Objektes ist. Ist also die Position eines Objektes größer als die positive Skalierung, befindet sich das Objekt außerhalb des definierten Raumes. In diesem Fall wird die Position des Objektes auf dieser Achse auf die positive Skalierung des Empty-Objektes gesetzt. Gleiches gilt für die Abfrage entlang der negativen Achsen. Somit gibt das Empty-Objekt die Ausdehnung des Raumes an. Die Abfrage ist dabei auf alle Objekte bezogen, sodass weder die manipulierbaren Objekte, als auch die Kamera außerhalb der Wände gebracht werden können. Der Ablauf ist in der Abbildung 15 dargestellt. Die Texturen für die Wände, des Bodens und der Kiste sind dabei aus den Quellen [1], [7] und [13] entnommen.

Eine weitere Veränderung bezog sich in erster Linie auf die Struktur des Programmes. Diese Strukturveränderungen sollten dazu dienen das Programm flüssiger zu machen. Die Gesten wurden zwar meist richtig erkannt, aber sie waren mit einem gewissen Zeitverzug wahrzunehmen. Die Objekte gingen nicht flüssig von Punkt A nach B über, sondern wirkten unterbrochen. Die Darstellung war ruckartig und das Programm war nicht robust genug. Der Rechner wurde zunehmend von den zahlreich ankommenden Daten der Leap-Motion überlastet, bis das Programm Blender beim Ausführen regelmäßig abstürzte. Da diese Problem nicht durch das Aufräumen des Codes behoben werden konnte, musste eine andere Lösung bedacht werden. Schlussendlich wurde ein Projekt zu einem ähnlichen Fall bei [6] gefunden.

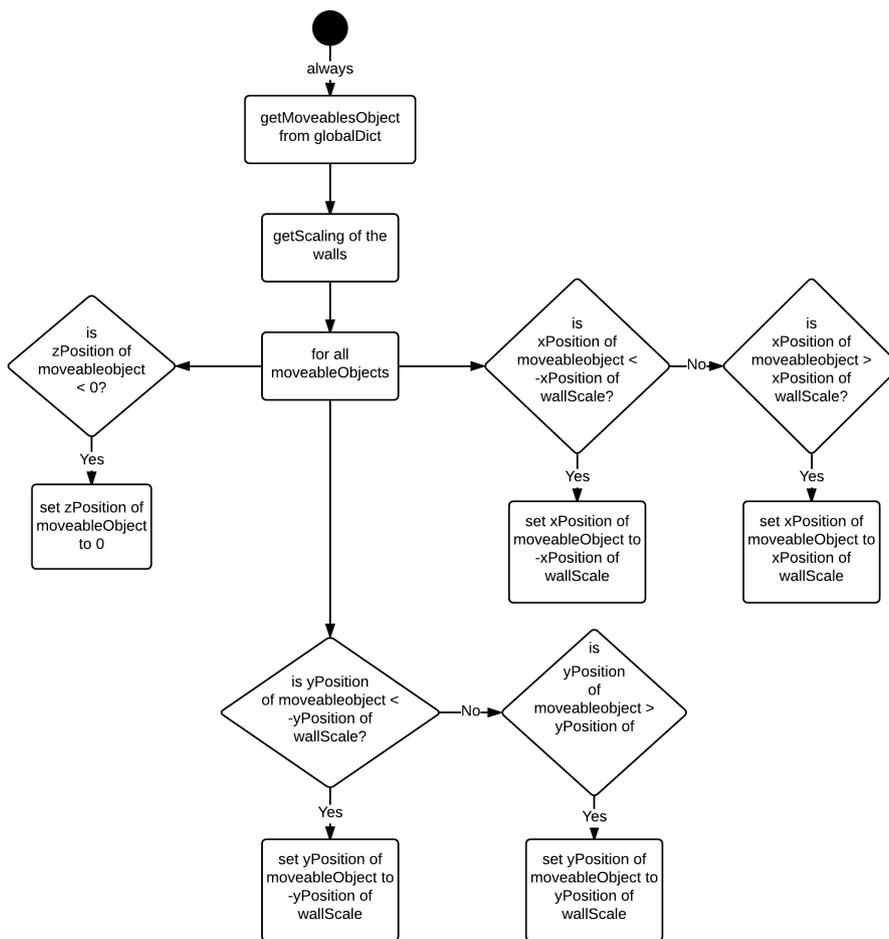


Abbildung 15: Umsetzung der Raumbeschränkung als Diagramm

In diesem Fall ging es auch um die Benutzung der Daten der Leap-Motion in der Blender-Game-Engine, wie sie auch hier benutzt wird. Allerdings wurden in diesem Projekt die Daten der Hände nur visuell dargestellt. Das Problem bestand letztendlich darin, dass die Verarbeitung der Gesten zuvor bei jedem ankommenden Frame von der Leap-Motion stattfand. Diese Rate der ankommenden Frames ist im Gegensatz zu der Rate, in der Blender rendert, um einiges größer. Leap-Motion liefert dabei 200 Frames pro Sekunde [10], während Blender mit einer Rate von 30 Frames auskommt. Daher ist die Lösung, die Verarbeitung unabhängig vom Eingang der Frames zu gestalten. Dazu ist die Verarbeitung der Daten pro Sekunde nur in dem Maße notwendig, wie das menschliche Auge Bilder pro Sekunde als flüssig wahrnehmen kann. Deshalb sollte die Verarbeitung nicht unter 20 Frames pro Sekunde fallen [16]. Da die Verarbeitungsrate von Blender größer 20 ist und es keine weiteren Vorteile bei der Wahrnehmung bei 200 Frames pro Sekunde hätte, sollte die Rate von Blender verwendet werden. Dies ist folgendermaßen implementiert worden.

Bei der Initialisierung wird ein Leap-Listener-Objekt erstellt, sofern für diese Szene noch keiner vorhanden ist. Innerhalb dieser erstellten Klasse des Listeners befinden sich die zwei wichtigsten Methoden. Zum Einen ist es die `applyTransformation()`-Methode, die zum Verändern der Kamera und für die Transformationen auf die Objekte zuständig ist. Eine weitere wichtige Methode ist die `on_frame()`-Methode, welche bei jedem neu ankommenden Frame aufgerufen wird. Da die Methode `applyTransformation()` durch die `main`-Funktion aus Abschnitt 3.3 aufgerufen wird, wird diese durch Blender aufgerufen. Zuvor war `applyTransformation()` in der Methode `on_frame()` enthalten. Kommt nun ein neuer Frame hinzu, wird die `on_frame()`-Methode aufgerufen. Um die Berechnung so gering wie möglich zu halten, wird hier nur der Aktuelle und der vorherige Frame aus den Daten abgespeichert. Zusätzlich wird der letzte Frame auf `None` gesetzt, falls eine Hand detektiert wurde, diese aber keine Greifgeste ausführt. Die letzte Abfrage soll dabei verhindern, dass die Hand weiterhin aufgezeichnet wird, selbst wenn keine Greifgeste besteht. Ohne diese Abfrage könnte der Anwender seine Hand öffnen, diese in eine Richtung bewegen und währenddessen wieder schließen. Das Resultat dieser Bewegung wäre zunächst das Stehen bleiben des Objektes, wenn die Hand sich öffnet. Anschließend würde das Objekt beim erneuten Schließen einen weiten Sprung machen und erst dann der Hand erwartungsgemäß wieder folgen.

### 3.9 Vorevaluation

Bei den kleineren Evaluationen wurde die Anzahl der Neustarts und auch die Zeit zum Lösen der Aufgabe aufgezeichnet. Die Aufgabe bestand darin einen Turm mit den drei Objekten zu bauen. Dazu gab es keine Zeitbeschränkung. Bei der aller ersten Evaluation, waren noch keine Wände implementiert. Dabei wurden im Durchschnitt um die zehn Neustarts gebraucht. Dazu hatte in diesem Test keine Person die Aufgabe gelöst. Nach dem die Wände hinzugefügt waren, ergab ein neuer Test, dass sich die Anzahl der Neustarts reduzierten. Sie beliefen sich schließlich im Durchschnitt nur noch auf zwei. Dazu gab es Versuchspersonen, die die Aufgabe von nun an Lösen konnten. Der Grund der restlichen Neustart lag daran, dass das Objekt noch unter die Kamera geraten konnte und dieses Objekt wieder nicht mehr auswählbar war. Deshalb wurde nach dem zweiten Test die Steuerung durch den Raum entwickelt. Nach dem dieser Stand nochmals getestet wurde, war kein Neustart mehr nötig. Dazu konnten fast alle Probanden den Test absolvieren. Dieses spiegelt sich auch in der Abfrage der Bedienbarkeit insgesamt wieder. Zu Beginn wurde diese von Personen als fünf, also sehr schlecht, bewertet. Dieser Eindruck der nach dem ersten Test im Durchschnitt bei 4.5 lag, hatte sich im Laufe der weiteren Tests zu einer 3 verbessert.

## 4 Evaluation

Im Rahmen dieser Evaluation soll herausgefunden werden, ob eine Bedienung mit Gesten in Kombination mit einem Touchscreen genauso leicht auszuführen ist, wie eine gewohnte Maussteuerung in Bezug auf eine 3D-Interaktion. Dabei sind folgende Bereiche in der Bedienung untersucht worden.

- Bewegung im dreidimensionalen Raum
- Bewegen der Objekte im dreidimensionalen Raum
- Rotieren von Objekten im dreidimensionalen Raum
- Skalieren von Objekten im dreidimensionalen Raum

Dazu wurde beim Bewegen im dreidimensionalen Raum eine weitere Unterscheidung von zwei Metaphern vorgenommen und untersucht. Diese soll zeigen, welche Art von Bewegung hier bevorzugt wird und damit intuitiver für Personen erscheint. Dabei wurde die Bewegung einmal mit Hilfe einer Seil-Metapher und einmal mit Hilfe einer Spieler-Metapher untersucht. Im Weiteren sind beide Metaphern genauer beschrieben. Die Seil-Metapher wird zudem in der Literatur [8] beschrieben. Hier ist es unter dem Begriff *grabbing the air* aufgeführt. Um den Probanden eine leichteres Verständnis zu ermöglichen, wurde hier der Begriff der Seil-Metapher gewählt. Dazu wird dadurch auch der Bezug zur Spieler-Metapher deutlich.

Bei der Seil-Metapher bewegt man sich vorwärts, in dem man sich vorstellt, man habe ein Seil an dem man sich nach vorne ziehen kann, siehe Abbildung 16. Dementsprechend bewegt man die Maus beziehungsweise die Hand zu sich selbst. Will man sich in entgegengesetzter Richtung bewegen, führt man die Maus oder die Hand von sich weg. Beim Drehen im Raum muss man sich ein um sich selbst gespanntes Seil vorstellen, an dem man sich beim Ziehen daran auf der Stelle drehen kann. Wird die Maus beziehungsweise die Hand nun nach rechts bewegt, kann der Anwender nach links schauen. Im Gegensatz dazu dreht man sich nach links, wenn die Maus oder die Hand nach rechts geht. Dies ist in der Abbildung 17 zu erkennen.

Genau das Gegenteil gilt für die Spieler-Metapher. Bei dieser Steuerung bewegt der Anwender die Maus oder seine Hand dorthin, wohin er schauen oder gehen möchte. Beim Fortbewegen geht die Maus oder die Hand also nach vorne und beim Rückwärtsgehen nach hinten. Beim Drehen geht die Maus oder die Hand nach rechts, wenn der Anwender nach rechts schauen will und er schaut nach links, wenn die Bedienelemente entsprechend nach links gehen.



**Abbildung 16:** Heranziehen mit der Seil-Metapher



**Abbildung 17:** Drehen mit der Seil-Metapher

## 4.1 Evaluationsthese

Im Allgemeinen wurden durch die im folgenden Abschnitt 4.2 beschriebenen Aufgaben und durch die entwickelte Steuerung, siehe 4.3, folgende Thesen aufgestellt.

1. Die Bewegung im Raum ist mit Gesten intuitiver als mit einer Maus
2. Die Bewegung im Raum ist mit Hilfe der Spieler-Metapher intuitiver
3. Die Geste der Translation ist intuitiver und schneller als bei einer Maus
4. Die Translation mit der Maus ist präziser als bei einer Geste
5. Die Geste der Rotation ist intuitiver als bei einer Maus
6. Die Rotation mit der Maus ist präziser als bei einer Geste
7. Die Geste der Skalierung ist intuitiver als bei einer Maus
8. Die Skalierung mit der Maus ist schneller als bei einer Geste

Im Weiteren werden die zu lösenden Aufgaben der Evaluation beschrieben, auf die im späteren Abschnitt 4.4 zurückgegriffen wird.

## 4.2 Aufgaben der Evaluation

Insgesamt gab es fünf Aufgaben zu lösen. Diese sind einmal mittels Maus und einmal mit Hilfe von Gesten durchzuführen. Jede Aufgabe hat jeweils ein Zeitlimit von fünf Minuten. Die Zeit läuft, wenn der Proband zum ersten Mal ein manipulierbares Objekt anklickt.

Der erste Test bezieht sich auf die Bewegung im Raum mit unterschiedlichen Metaphern. Aufgabe ist es, die drei manipulierbaren Objekte, einen Zylinder, einen Würfel und einen Kegel mindestens einmal in eine offene Kiste zu legen. Dabei ist der Würfel, der Zylinder und die Kiste von der Ausgangsstellung der Kamera aus sichtbar, siehe Abbildung 18. Des Weiteren wurde der Kegel bewusst hinter der Kamera positioniert, siehe Abbildung 19, um ihn erst durch die Bewegung durch den Raum sehen und damit auswählen zu können. Dabei symbolisiert das rote Kreuz den Standpunkt der Kamera. Hierbei sollen die Evaluationsthese 1 und 2 untersucht werden. Diese besagen, dass die Bewegung im Raum mit Gesten intuitiver ist und dass die Spieler-Metapher von den Probanden bevorzugt wird.

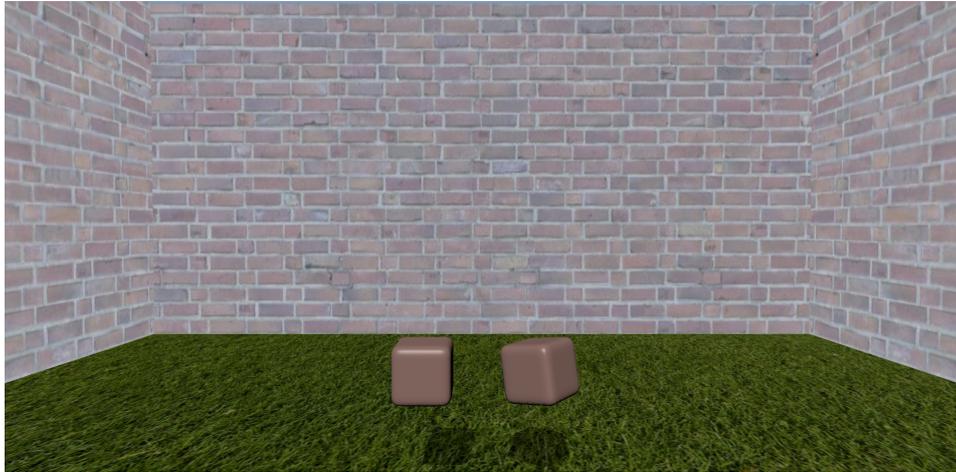
Die zweite Aufgabe beinhaltet das Bauen eines Turmes aus den drei manipulierbaren Objekten. Dabei gibt es keine Einschränkungen zu deren Realisierung. Die Aufgabe gilt als gelöst, wenn der Turm nicht mehr umfällt. Damit sollen die Evaluationsthese 3 und 4 untersucht werden, die beschreiben, dass die Geste der Translation intuitiver und schneller ist, aber die Maussteuerung dafür zu einer präziseren Positionierung führt.



**Abbildung 18:** Startposition der ersten Aufgabe



**Abbildung 19:** Startposition der ersten Aufgabe mit Sicht auf alle Objekte



**Abbildung 20:** Startposition der Aufgabe zur Rotierbarkeit

Anschließend wurde mit der dritten Aufgabe die Rotierbarkeit der unterschiedlichen Steuerungen getestet. Hierbei wird auf der rechten Seite ein Objekt mit einer zuvor bestimmten Rotationsmatrix vorgegeben. Auf der linken Seite wird das gleiche Objekt abgebildet, jedoch mit einer Einheitsmatrix als Rotationsmatrix, zur Veranschaulichung siehe Abbildung 20. Die Aufgabe für ein Objekt galt als gelöst, wenn sich die Objekte mit einer kleinen Toleranz gleich aussahen. Anschließend wird der Test wiederholt, bis die Aufgabe mit allen drei manipulierbaren Objekten erfüllt ist. Dabei hat jedes Objekt eine andere Rotationsmatrix. Die Rotationsmatrizen sind pro Objekt für jeden Probanden identisch, sodass für alle die gleichen Bedienungen geschaffen sind. Diese Aufgabe wurde ausgewählt um die Thesen 5 und 6 genauer zu betrachten. Also soll mit dieser Aufgabe herausgefunden werden, ob die Geste bei der Rotation intuitiver ist, dafür aber die Rotation mit der Maus präziser.

In der letzten Aufgabe wird die Skalierbarkeit der beiden Steuerungen gemessen, in dem auf der rechten Seite wieder ein Objekt mit einer vorher bestimmten Skalierung angezeigt wird und das linke Objekt auf die gleiche Größe skaliert werden muss. Auch hier mussten nacheinander alle drei Objekte mit unterschiedlich vorherbestimmten Skalierungen auf die gleiche Größe gebracht werden. Dabei sollen die letzten beiden Thesen, 7 und 8, untersucht werden. Diese besagen, dass die Geste der Skalierung intuitiver ist, aber die Skalierung mit der Maus schneller ausführbar ist.

Im nächsten Abschnitt wird die komplette Steuerung zur Manipulation der Objekte mit Hilfe der Maus und der Gesten beschrieben.

### 4.3 Steuerung der Gesten und der Maus

Das Auswählen eines Objektes mit der Maus erfolgt durch das Anklicken und gedrückt Haltens des Selbigen. Dabei erscheint das Objekt, solange es ausgewählt ist, dunkler als zuvor. Beim Auswählen kann der Anwender zwischen der rechten und der linken Maustaste wählen, die jeweils unterschiedliche Funktionen beinhalten. Wird eine der beiden Maustasten losgelassen, gilt das Objekt als nicht mehr selektiert und wechselt wieder in die Ursprungsfarbe.

Das Aktivieren der Bewegung im Raum geschieht, durch das Klicken und gedrückt Halten der jeweiligen Taste an einer beliebigen Stelle des Raums, außer auf einem manipulierbaren Objekt. Dabei bewirkt die linke Maustaste die Vorwärts- und Rückwärtsbewegung, wohingegen die rechte Maustaste die Drehung verursacht. Die genaue Steuerung mit Hilfe der unterschiedlichen Metaphern ist im Abschnitt 4 beschrieben.

Wird die linke Maustaste auf einem Objekt betätigt, kann das Objekt entsprechend der sichtbaren zweidimensionalen Ebene verschoben werden. Dabei geht das Objekt in die Richtung, in die sich der Mauszeiger bewegt. Das Objekt kann demnach nur immer nach rechts und links oder nach oben und unten bewegt werden.

Beim Benutzen der rechten Maustaste wird das Rotieren des Objektes aktiviert. Wird dabei die Maus nach rechts und links bewegt, dreht sich das Objekt entlang der Achse, die nach oben zeigt, der Z-Achse. Ist die Mausrichtung von vorne nach hinten wird je nach Kameraeinstellung das Objekt um die nach rechts oder um die am Anfang nach hinten zeigende Achse gedreht, also X- und Y-Achse. Dabei ist es auch möglich, je nach Kameraeinstellung das Objekt um die X- und Y-Achse gleichzeitig zu drehen. Die letzte Funktion ist das Skalieren, welches durch das Mousrad realisiert ist. Dabei ist es wichtig, dass das Objekt zuvor entweder mit der linken oder der rechten Maustaste ausgewählt wurde, dies ist je nach Anwender anders gewünscht. Wird das Mousrad zu sich selbst bewegt, vergrößert sich das Objekt. Das Gegenteil gilt somit für das Verkleinern des Objektes. Beim Skalieren wird jedoch immer in alle drei Raumrichtungen gleich hoch und runter skaliert.

Die Steuerung mit Hilfe der Gesten ist wie folgt: Zum Auswählen eines Objektes muss das gewünschte Objekt auf dem Touchscreen angeklickt werden. Dieses wechselt ebenfalls zu einer dunkleren Farbe, bis ein anderes Objekt ausgewählt oder irgendwo sonst im Raum hin geklickt wird.

Ist kein manipulierbares Objekt ausgewählt, wird die Bewegung durch den Raum ausgeführt. Dabei ist es wichtig, dass nur eine Hand von der Leap-Motion registriert wird. Wird diese Hand bis zu einem bestimmten Grad geschlossen, kann mit Hilfe der Seil- oder der Spieler-Metapher, siehe Abschnitt 4, die Kamera manipuliert werden. Ist jedoch ein manipulierbares Objekt ausgewählt, dann sind folgende Manipulationen am Objekt möglich.

Beim Verschieben dieses Objektes wird ebenfalls nur eine Hand benötigt. Dabei gilt das Gleiche wie beim Bewegen im Raum. Wenn mehr als eine Hand registriert worden ist, kann die Verschiebung nicht realisiert werden. Anschließend muss die Hand geschlossen werden, die Greifgeste ausgeführt und aufrechterhalten werden, bevor eine Translation stattfinden kann. Das Objekt bewegt sich dabei genau in die gleiche Richtung wie sich die Hand bewegt.

Die Rotation kann auch nur beim Erkennen einer Hand aktiviert werden. Hierbei kann die Rotation auch nur stattfinden, wenn die Greifgeste erkannt wurde. Daraufhin wird das Drehen der Hand um alle drei Achsen erkannt und entsprechend auf das Objekt umgesetzt.

Die einzige Geste, die genau zwei Hände erkennen muss, ist die Geste zur Skalierung des Objekts. Um das Objekt zu Vergrößern oder zu Verkleinern müssen jedoch beide Hände die Greifgeste ausführen. Beim Auseinanderbewegen der beiden Hände, egal in welche Richtung, vergrößert sich das Objekt. Bewegen sich die Hände im Gegensatz dazu auf einander zu, wird das Objekt verkleinert.

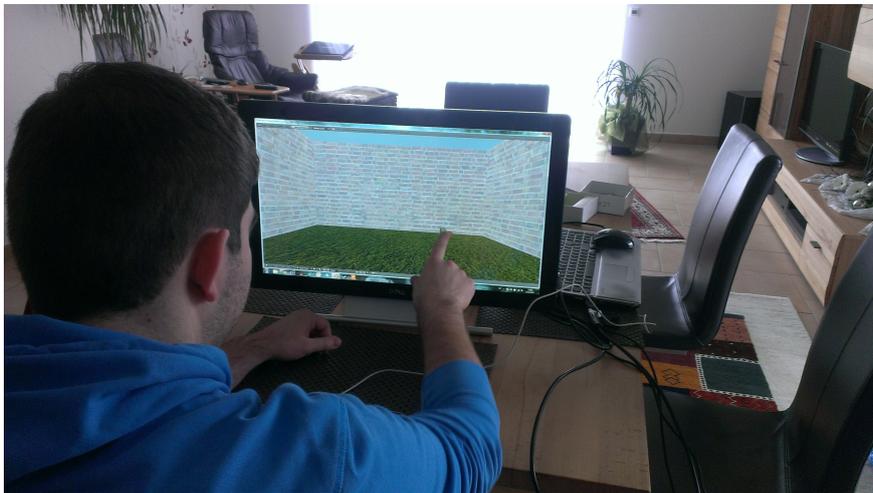
Abschließend ist zur Gestensteuerung zu sagen, dass wenn eine bestimmte Geste nicht mehr ausgeführt werden soll, mindestens eine Hand geöffnet werden muss.

#### **4.4 Evaluationsablauf**

Am Anfang wurde der Proband gebeten einen Fragebogen bezüglich seines Alters, Geschlechts und seiner Erfahrung mit Maus- und Gestensteuerung ausfüllen. Dabei sollte die Testperson sich selbst einschätzen, wie schwer oder leicht der Person die Maus- und die Gestensteuerung fallen wird. Danach konnte mit dem ersten Test begonnen werden. Bei jedem Test wurde immer zu erst die Maussteuerung getestet. Im Anschluss wurde der Test neu gestartet und es galt die Aufgabe mit der Gestensteuerung zu lösen.

Vor der ersten Aufgabe wurde dem Proband die Funktion der Maus mit der Seil-Metapher erklärt und eventuell auftretende Fragen diesbezüglich beantwortet. Anschließend wurde die erste zu erfüllende Aufgabe erläutert. Nachdem der Proband sowohl die Maussteuerung, als auch die Aufgabe verstanden hatte, wurde diese gestartet. Bevor der Test mit der Gestensteuerung absolviert wurde, wurde diese Steuerung ebenfalls erklärt.

Der erste Test beinhaltete das Bewegen durch den Raum, bei dem Objekte in eine Kiste gelegt werden mussten. Hierbei erfolgte die Steuerung nach der Seil-Metapher. Nachdem diese Aufgabe zuerst mit der Maus und danach mit Gesten absolviert wurde, wurde ein Fragebogen bezüglich der Steuerung mit der Seil-Metapher ausgefüllt. Danach konnte der Test wiederholt werden, mit der Änderung, dass die Bewegung durch den Raum mit der Spieler-Metapher erfolgte. Hierbei wurde auch zunächst mit der Maus, daraufhin mit den Gesten getestet und im Anschluss der gleiche Fragebogen



**Abbildung 21:** Proband beim Auswählen eines Objektes

ausgewertet.

Bevor der nächste Test starten konnte, gab es einen Fragebogen, in dem die Unterschiede zwischen Seil- und Spieler-Metapher analysiert wurden. Dabei konnten auch Präferenzen in Bezug auf eine Maus- oder Gestensteuerung genannt werden.

Danach durfte der Proband zwischen den zwei Metaphern für die nachfolgenden Aufgaben wählen. Dadurch konnten sich die Probanden auf die nachfolgenden Aufgaben konzentrieren und waren nicht durch eine gewöhnungsbedürftige Steuerung abgelenkt.

Die nächsten drei Tests bestanden jeweils aus den Aufgaben einen Turm zu bauen, die Objekte gleich zu rotieren und die Objekte gleich zu skalieren. Wiedermals gab es nach jedem Test einen Fragebogen, der sich nun nur auf das Verschieben, nur auf das Rotieren und nur auf das Skalieren der Objekte bezog.

Zuletzt musste ein Gesamteindruck abgegeben werden. Dieser bezog sich auf die vier zu testenden Bedienungen jeweils für Maus und Geste und auf die Frage, wie man die Maus- und Gestensteuerung insgesamt empfand. Dies soll im Vergleich zum ersten Teil des Fragebogens erörtern, ob die Einschätzungen und Erwartungen der Probanden bezüglich der unterschiedlichen Steuerungen übereinstimmen.



**Abbildung 22:** Proband beim Versuch einen Turm zu bauen

## 4.5 Ergebnisse

Im Folgenden werden die Ergebnisse der Evaluation präsentiert und auf die Thesen bezogen. Die Ergebnisse sind dabei jeweils mit einem Mittelwert angegeben. Insgesamt haben 11 Personen an der Evaluation teilgenommen, wobei sechs männlich und fünf weiblich waren. Die Probanden waren dabei zwischen 22 und 54 Jahre alt.

### 4.5.1 Bewegen im Raum

Die erste Aufgabe wurde im Durchschnitt mit der Spieler-Metapher eine Minute und dreizehn Sekunden schneller gelöst, als mit der Seil-Metapher, vergleiche Tabelle 2 mit Tabelle 3. Dazu wählte jeder Proband für die letzten drei Aufgaben die Spieler-Metapher als Präferenz aus. Der Unterschied zwischen den Zeiten ist jedoch nicht allein auf die unterschiedliche Steuerung zurückzuführen. Da der Test mit der Seil-Metapher begann, ist davon auszugehen, dass im zweiten Durchlauf, die Steuerung geläufiger war als zuvor. Dennoch kann man an den Zeiten und an der Präferenz der Probanden ablesen, dass die Spieler-Metapher für die Probanden intuitiver ist.

Die Versuchspersonen haben nicht nur mit den Zeiten gezeigt, dass die Spieler-Metapher intuitiver ist, sondern diese im Allgemeinen auch besser bewertet, wie es in Tabelle 1 zu erkennen ist. Dabei gab es eine Auswahlmöglichkeit von eins bis fünf, wobei eine eins eine sehr leichte Bedienung und eine fünf eine sehr schwere Bedienung repräsentierte.

Aus diesen Ergebnissen lässt sich die These, dass die Spieler-Metapher intuitiver ist, bestätigt.

Ebenfalls ist aus den Zeiten abzulesen, dass die Aufgabe mit der Gestensteuerung schneller gelöst wurde. Bei der Seil-Metapher ist zudem ein

	<b>Maus</b>	<b>Geste</b>	<b>Mittelwert</b>
<b>Seil-Metapher</b>	2.405	2.405	2.405
<b>Spieler-Metapher</b>	1.675	1.855	1.765

**Tabelle 1:** Bewertung der Metaphern in Bezug auf die Bedienungsmöglichkeit

	<b>Maus</b>	<b>Geste</b>	
	02:26	01:59	
	04:50	04:50	
	03:04	02:21	
	02:29	01:30	
	04:25	02:49	
	04:57	02:01	
	02:17	02:35	
	02:20	04:16	
	02:06	01:36	
	01:53	03:40	
	01:55	01:50	
<b>Insgesamt</b>	02:58	02:35	02:46

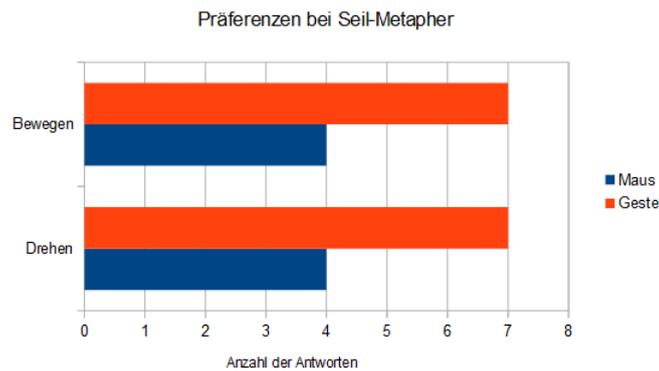
**Tabelle 2:** Zeiten zum Lösen der ersten Aufgabe mit der Seil-Metapher

	<b>Maus</b>	<b>Geste</b>	
	00:50	01:33	
	00:56	00:49	
	01:23	00:53	
	01:33	00:54	
	00:50	01:24	
	01:03	00:50	
	01:11	01:07	
	02:33	02:42	
	01:43	02:26	
	03:41	02:07	
	01:49	02:08	
<b>Insgesamt</b>	01:35	01:32	01:33

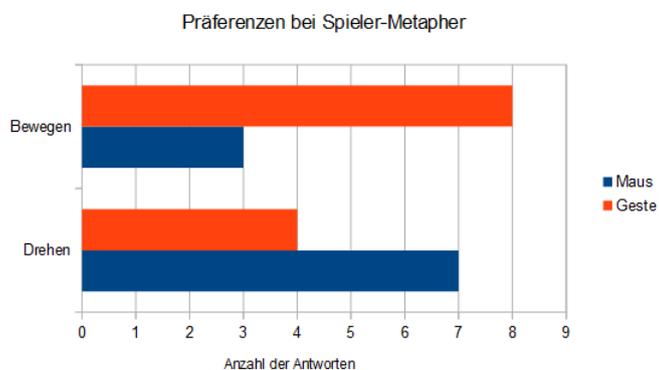
**Tabelle 3:** Zeiten zum Lösen der ersten Aufgabe mit der Spieler-Metapher

größerer Unterschied zwischen Maus- und Gestensteuerung wahrzunehmen. Dies kann darauf zurückzuführen sein, dass die Bedienung mit der Maus um die Objekte zu bewegen nur im zweidimensionalen Raum möglich war und das Prinzip der Maussteuerung erst nach ein paar Sekunden vollständig verstanden wurde. Dazu wurde immer zuerst mit der Maus die Aufgabe getestet. Was einen Nachteil bezüglich der Maus verursacht haben könnte. Da jedoch auch die Zeitunterschiede sehr gering sind, lässt sich daraus keine Rückschlüsse ziehen, ob die Gestensteuerung für die Probanden einfacher zu handhaben ist.

Des Weiteren wurden die Probanden aufgefordert anzugeben, ob sie bei einer Steuerung eine bestimmte Metapher bevorzugen. Daraus ist zu erkennen, dass die Probanden die Seil-Metapher bei der Gestensteuerung, sowohl beim Drehen als auch beim Bewegen im Raum, der Maussteuerung vorziehen würden, siehe 23. Hingegen wird bei der Spieler-Metapher beim Drehen die Maus bevorzugt, beim Bewegen im Raum wiederum die Gestensteuerung, siehe Abbildung 24.



**Abbildung 23:** Welche Steuerung wird bei der Seil-Metapher bevorzugt?



**Abbildung 24:** Welche Steuerung wird bei der Spieler-Metapher bevorzugt?

Abschließend ist zu sagen, dass die Probanden die Steuerung der beiden Eingabemöglichkeiten ungefähr gleich einfach zu benutzen empfanden. Es gibt jedoch eine leichte Tendenz zur Maussteuerung. Ob eine andere Geste zur Bewegung im Raum oder etwas Übung diesen Trend entgegenwirken würde, lässt sich aus diesen Daten nicht sagen.

#### 4.5.2 Translation der Objekte

Die Zeiten zur Lösung der zweiten Aufgabe aus Tabelle 4 deuten darauf, dass die Verschiebung der Objekte mit der Maus präziser zu bewerkstelligen ist. Da bei der Aufgabe ein Turm zu bauen war, der nicht mehr umfällt, ist eine genaue Positionierung erforderlich. Aus diesen Zeiten lässt sich jedoch nicht erkennen, ob die Steuerung mit den Gesten dabei schneller ist. Um diesen Aspekt verifizieren zu können, müsste man einen weiteren Test hinzuziehen, indem das genaue Positionieren keine Rolle spielt. Eine Zeit von fünf Minuten bedeutet, dass der Proband an der Aufgabe gescheitert ist und der Test danach abgebrochen wurde.

Außerdem lässt sich aus den Testergebnissen der Probanden schließen, dass die Bedienung mit der Maus geringfügig besser bewertet wurde. Die Ergebnisse können dahingehend negativ für die Gestensteuerung beeinflusst worden sein, dass die genaue Positionierung bei der Translation nicht funktioniert hat, da der Großteil der Probanden die Aufgabe mit Gesten erst gar nicht lösen konnte. Die Durchschnittswerte der Bewertungen sind in der Tabelle 7 angegeben. Dabei war wieder eine Skala von eins bis fünf gegeben, wobei eine eins sehr einfach und eine fünf sehr schwer dargestellt hat.

Ebenso wurde auch die Maussteuerung von den Probanden als intuitiver und einfacher erachtet, siehe Abbildung 25 und 26. Die Durchschnittswerte berufen sich hierbei bei der intuitiven Steuerung auf 1.72, was eine Tendenz zur Maus bedeutet. Dazu liegt der Durchschnittswert bei der einfachen Steuerung bei 2.72. Auch hier ist also die Tendenz zur Maussteuerung erkennbar. Wichtig ist hierbei einzukalkulieren, dass die Probanden an die Benutzung der Maus gewöhnt sind, weshalb die Maus oft besser bewertet wurde.

	<b>Maus</b>	<b>Geste</b>
	00:47	03:04
	04:47	04:34
	02:38	03:10
	04:19	01:37
	04:18	01:41
	03:55	05:00
	03:57	05:00
	03:43	05:00
	02:09	05:00
	05:00	05:00
	05:00	05:00
<b>Insgesamt</b>	03:41	04:00

**Tabelle 4:** Zeiten um einen Turm zu bauen

### 4.5.3 Rotation der Objekte

An dieser Stelle soll die These, dass die Geste der Rotation im Gegensatz zu einer Maussteuerung intuitiver ist, erörtert werden. Ihr ist laut der Evaluationsergebnisse eher nicht zuzustimmen. Die Geste schien intuitiv, da die Probanden, die Geste direkt verstanden. Dennoch ergab der Mittelwert aus den Ergebnisse der Intuition aus Abbildung 25 eine 1.54, was einen deutlichen Vorteil für die Maus bedeutet.

Aus Abbildung 26 ist abzulesen, dass die Maussteuerung auch hier bevorzugt wurde, da der Durchschnitt hier bei 1.9 liegt. Dies hat mehrere Gründe. Zum Einen hatten einige Probanden ein Problem die Geste auszuführen, da ab einem bestimmten Grad der Verdrehung der Hand diese nicht mehr ausführbar war. Ein weiteres Problem, welches von den Probanden angemerkt wurde, ist, dass die Objekte sich beim Rotieren im Raum bewegten. Dies war im Gegensatz zur Maussteuerung unabhängig voneinander. Die Frage, ob nun Maus oder Geste einfacher auszuführen ist, sollte nochmals mit Hilfe einer anderen Geste untersucht werden, die ebenso unabhängig von der Translation ist. Des Weiteren sollte der Test so ausgelegt werden, dass die Präzision eine minder große Rolle spielt, um nur die Intuitivität und Einfachheit der Geste im Gegensatz zur Maus zu testen.

Ein weiteres Problem lag auch beim Öffnen der Hand. Dabei verdrehte sich das Objekt ungewollt. Dies ist das gleiche Problem, was bei der Translation in Abschnitt 4.5.2 beschrieben wurde. Das präzise Ausrichten ist damit schwer zu realisieren. Dadurch entstanden Verzögerungen beim Lösen der Aufgabe, welches sich in den Zeiten der Tabelle 5 widerspiegeln. Deshalb ist auch hier sagen, dass sich das Rotieren mit der Maus präziser gestalten lässt.

	<b>Maus</b>	<b>Geste</b>
	01:17	01:58
	01:14	01:21
	00:58	00:59
	01:26	01:03
	00:39	01:11
	01:04	00:51
	01:06	03:32
	01:06	02:26
	03:34	01:46
	02:42	01:45
	01:50	04:02
<b>Insgesamt</b>	01:32	01:54

**Tabelle 5:** Zeiten um Objekt gleich aussehen zu lassen

#### 4.5.4 Skalierung

Bei der Skalierung von Objekten ist eindeutig zu erkennen, dass die Maussteuerung um einiges schneller ist als die der Gestensteuerung, siehe Tabelle 6. Dies lässt sich dadurch erklären, dass obwohl die Probanden die Geste ausgeführt haben, die Reaktion nicht erwartungsgemäß oft genug eintrat. Dies führte zu einer Verzögerung, da die Geste mehrmals hintereinander ausgeführt werden musste.

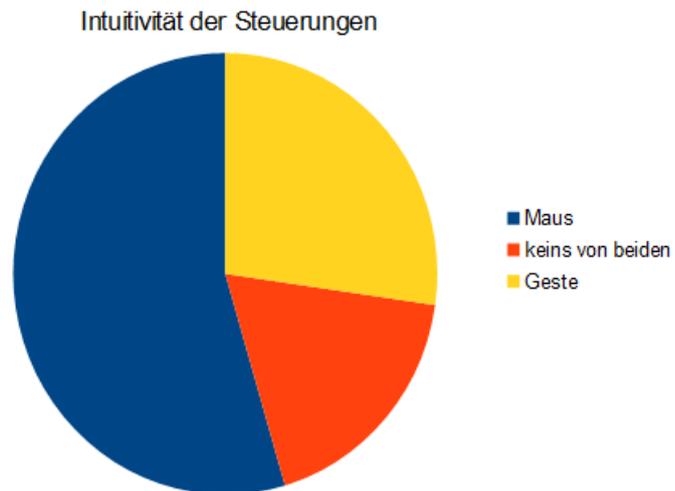
Da die Geste nicht oft genug erkannt wurde, sind die Bewertungen diesbezüglich im Gegensatz zur Maus schlechter ausgefallen, wie in Tabelle 7 zu sehen ist.

Außerdem empfanden die Probanden, wie aus Abbildung 26 zu sehen ist, die Maussteuerung einfacher zu bedienen. Hier ist der Durchschnittswert bei 1.81, was deutlich für die Maus spricht. Dies kann mit den bereits genannten Gründen zusammenhängen.

Was die Intuitivität betrifft, sprechen die Werte ebenso für die Maussteuerung. Dabei wurde in der Abbildung 25 ein Mittelwert von 1.54 erreicht.

	<b>Maus</b>	<b>Geste</b>
	00:12	01:02
	00:13	01:16
	00:19	00:54
	00:10	00:28
	00:10	00:30
	00:10	00:53
	00:06	01:07
	00:07	00:42
	00:05	00:20
	00:05	00:17
	00:08	02:22
<b>Insgesamt</b>	00:09	00:53

**Tabelle 6:** Zeiten um Objekt richtig zu skalieren



**Abbildung 25:** Welche Steuerung ist intuitiver?

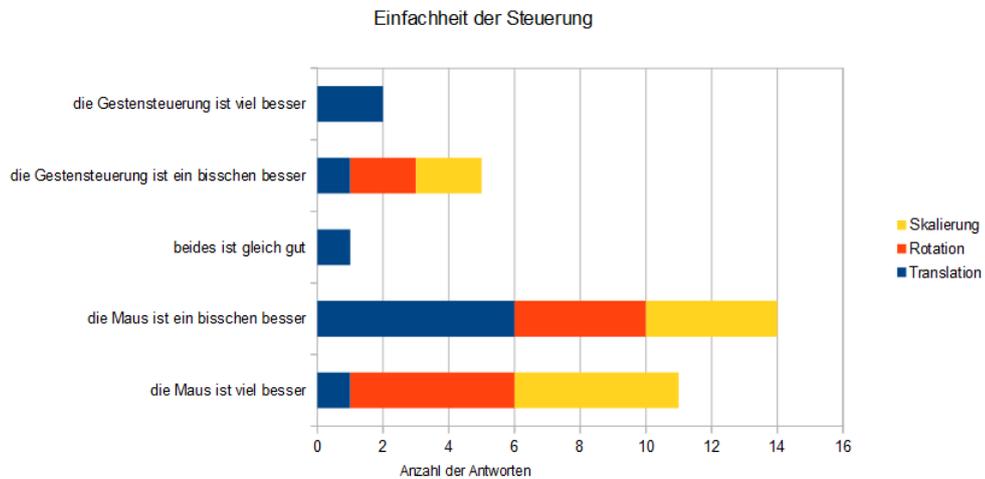


Abbildung 26: Welche Steuerung ist einfacher?

#### 4.5.5 Weitere Erkenntnisse

Zusätzlich zu den oben angesprochenen Problemen haben einige Probanden angemerkt, dass sie die Maus im Gegensatz zur Gestensteuerung gewohnt sind und diese daher sowieso bevorzugen. Dies ist auch deutlich in der Abbildung 26 zu sehen.

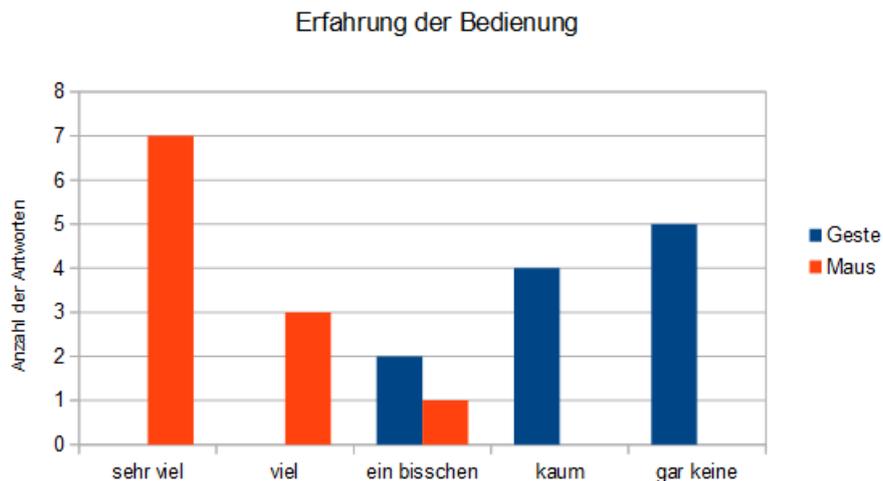
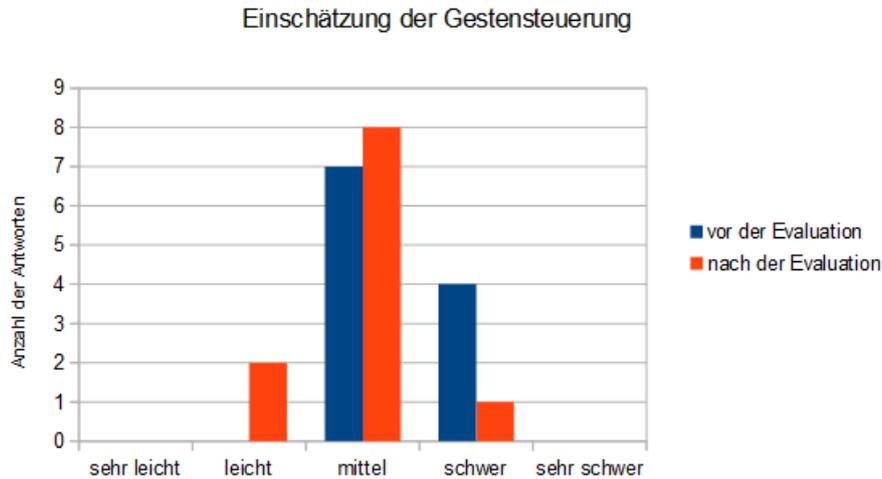


Abbildung 27: Erfahrung der Probanden mit den Steuerungen

Zu Beginn mussten die Probanden einschätzen, wie gut sie jeweils die Steuerungen handhaben können, wie in Abbildung 28 zu sehen ist. Dabei schätzten sich die Probanden bei der Gestensteuerung zu Beginn deutlich

schlechter ein.



**Abbildung 28:** Einschätzung der Leichtigkeit der Gestensteuerung

Im Gegensatz dazu ist bei der Maus ein leichter negativ Trend wahrzunehmen, wie die Abbildung 29 zeigt.

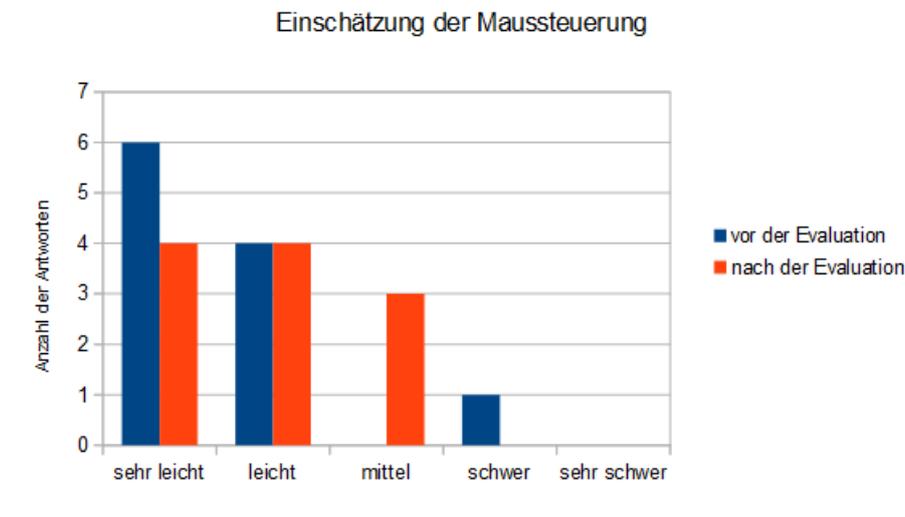
Auch wurden manchen Probanden während der Ausführung der Gesten der Arm schwer, was ebenfalls einen negativen Einfluss in Bezug auf diese Steuerung hat. Dies trat nicht nur in dieser Evaluation auf. Auch andere Quellen wie [12] und [5] bestätigen diese Tatsache.

#### 4.5.6 Zusammenfassung der Ergebnisse

	Maus	Geste
<b>Bewegung im Raum</b>	2.04	2.13
<b>Translation</b>	2.45	2.81
<b>Rotation</b>	2.18	3.45
<b>Skalierung</b>	1.18	2.36
<b>Insgesamt</b>	1.9625	2.6875

**Tabelle 7:** Bewertung der Bedienungsmöglichkeiten nach dem Test

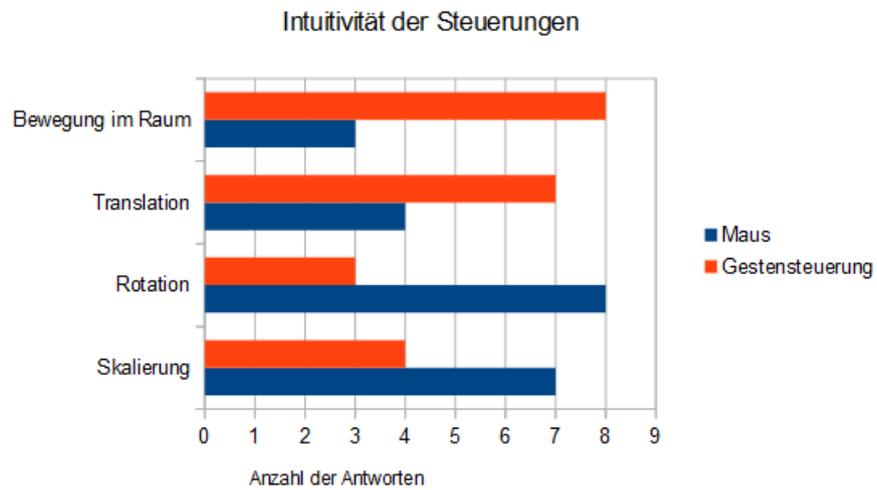
Allgemein würden die Probanden die Gestensteuerung bei der Bewegung im Raum von sich selbst und von Objekten bevorzugen. Bei Rotation und Skalierung von Objekten hingegen, präferieren sie die Maussteuerung. Letzteres kann anhand der Abbildungen 30, 31 und der Bewertungen in der Tabelle 7 gut erkannt werden. Dabei gab es für die Tabelle 7 eine Auswahlmöglichkeit von eins bis fünf, wobei eine eins für die Maus und eine fünf für die Geste stand.



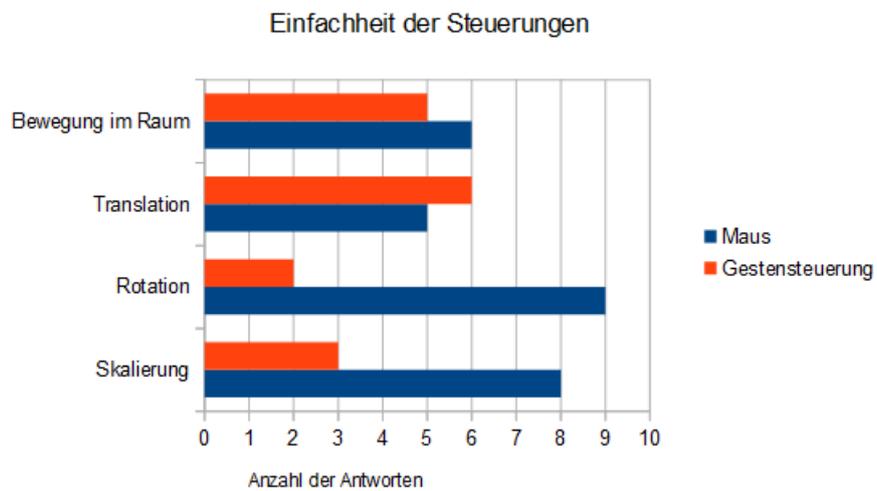
**Abbildung 29:** Einschätzung der Leichtigkeit der Maussteuerung

Bei der Bewegung der Objekte und der Kamera im Raum stimmen die Bewertungen, die direkt nach dem Test geprüft wurden und jener aus Abbildung 30 nicht exakt überein. Dafür sind die Werte aus der Tabelle 7 und die der Bewertung aus Abbildung 31 weitestgehend identisch. Da sich die Werte bei der Translation und bei der Bewegung im Raum nur geringfügig unterscheiden, könnte dies ein Zeichen für die schlechtere Bewertung der Gestensteuerung direkt nach den Tests sein, die auf Grund der zuvor beschriebenen Probleme aufgetreten sind. Damit lässt sich zeigen, dass die Gesten der Translation und der der Bewegung im Raum intuitiv erscheinen, aber genauso leicht umzusetzen sind, wie die der Maussteuerung.

Im Gesamten ist also zu sagen, dass sich einige Thesen als wahr erwiesen haben. Darunter fällt zum Einen, dass die Spieler-Metapher intuitiver eingeschätzt wird. Das Endergebnis stimmt sogar der ersten These zu, dass die Gesten beim Bewegen im Raum intuitiver sind. Da die Bewertung direkt nach dem Test allerdings abweicht, wäre diese These nochmals zu untersuchen. Die These drei kann ebenso nicht vollständig verifiziert werden, da das Ergebnis direkt nach dem Test ein anderes Bild aufweist wie in der Gesamtbewertung. Es ist also nicht zu sagen, ob die Geste der Translation intuitiver und auch schneller im Gegensatz zur Maussteuerung ist. Die Präzision mit der Maus hingegen ist sowohl bei der Translation als auch bei der Rotation besser als mit Gesten, was die Thesen vier und sechs bestätigt. These fünf kann widerlegt werden, denn kein Ergebnis zeigt, dass die Rotation als intuitiver erachtet wird. Das Gleiche gilt für die These sieben, welche eine intuitivere Steuerung der Geste für die Skalierung unterstellt. Die letzte These jedoch, kann allein an den Zeiten zum Lösen der Aufgaben verifiziert werden. An ihnen erkennt man deutlich, dass die Maus bei der Skalierung schneller



**Abbildung 30:** Gesamtergebnis zur Intuitivität der Steuerung



**Abbildung 31:** Gesamtergebnis zur Einfachheit der Steuerung

ist als bei der Geste.

## 5 Fazit

Die Gestensteuerung hat in manchen Bereichen wie dem Verschieben gut abgeschnitten. Die Geste scheint intuitiv genug zu sein, um sie direkt zu verstehen. Dazu ist die Ausführung leicht zu handhaben. Die Rotation hingegen erhielt keine gute Bewertung. Hier liegt das Problem darin, dass die Geste zu früh physikalischen Beschränkungen unterliegt. Des Weiteren gab es ebenso Probleme bei der Skalierung. Die Geste wurde nicht oft genug erkannt. Es konnte jedoch beobachtet werden, dass die Geste besser funktioniert, wenn sie langsamer ausgeführt wurde. Die Navigation durch den Raum hat dabei zum Einen gezeigt, dass die Spieler-Metapher als klarer Gewinner hervorgeht. Zum Anderen zeigt die Abbildung 25, dass die Geste intuitiver ist als die Maus, wobei sich die beiden Steuerungen was der Einfachheit betrifft in Abbildung 26 nicht viel unterscheiden.

Außerdem haben sich manche Probanden über Schmerzen im Arm beklagt. Die Ursachen liegen zum Einen darin, dass sie bei der Greifgeste die Hand zu stark verkrampften. Ansonsten wurde der Arm bei einigen auf Grund der ungewohnten Haltung schwer.

Im Gesamtergebnis bevorzugten die meisten Probanden weiterhin die Maus. Grund ist die Umstellung auf eine neue Bedienung. Zum Anderen ist eine Gestensteuerung noch nicht präzise genug und hatte noch einige Schwächen in der Hälfte der untersuchten Bereiche.

Abschließend ist zu sagen, dass die Gestensteuerung immer mehr Fortschritte macht. Die Erkennungsalgorithmen werden immer besser und sie wird dadurch zunehmend akzeptiert. Spiele, die bereits Bewegungen als Steuerungen nutzen, sind schon seit geraumer Zeit auf dem Markt und auch beliebt. Nur zum Einsatz in der Arbeitswelt sind die Verfahren noch nicht genügend ausgereift. Da demnächst die Leap-Motion bereits in erste Laptops eingebaut werden soll, ist davon auszugehen, dass sich diese immer weiter etablieren wird. Scrollen im Browser durch eine leichte Fingerbewegung ist bereits möglich und könnte in Zukunft als weitere Eingabemöglichkeit übernommen werden. Gesten, die eine genaue Positionierung voraussetzen, sind in den nächsten Jahren noch nicht zu erwarten. Hier ist die Maus noch unersetzbar.

## 6 Anhang

### Literatur

- [1] Stacy Burke. *Brick Wall Background Ninety Six*. [Online; Zugriff am 21.02.2015]. URL: <http://textures8.com/brick-wall-background-ninety-six/>.
- [2] MacGregor Campbell. *Kinect imaging lets surgeons keep their focus*. [Online; Zugriff am 24.03.2015]. 2012. URL: <http://www.newscientist.com/article/mg21428655.200-kinect-imaging-lets-surgeons-keep-their-focus.html>.
- [3] Automotion-Magazine IAV. *Ein Wink genügt*. [Online; Zugriff am 23.03.2015]. 2014. URL: <https://www.iav.com/automotion-magazine/automotion-03-2014/ein-wink-genuegt>.
- [4] Matthias Krempl. *Bewegungssteuerung Leap Motion: Die Fuchtel-Fernbedienung*. [Online; Zugriff am 24.03.2015]. 2013. URL: <http://www.spiegel.de/netzwelt/gadgets/angefasst-leap-motion-a-914252.html>.
- [5] Florian Kurzmaier. *Bewegungssteuerung Leap Motion im Test*. [Online; Zugriff am 24.03.2015]. 2013. URL: <http://www.macwelt.de/produkte/Bewegungssteuerung-Leap-Motion-im-Test-8102108.html>.
- [6] Dr. rer. nat. Magnus Benjes. *Wriggling fingers in Blender BGE*. [Online; Zugriff am 01.02.2015]. 2014. URL: [http://www.magben.de/?h1=3d&h2=leap\\_motion\\_hand](http://www.magben.de/?h1=3d&h2=leap_motion_hand).
- [7] Simon Murray. *Green Grass Texture 01*. [Online; Zugriff am 21.02.2015]. 2010. URL: <http://goodtextures.deviantart.com/art/Green-Grass-Texture-01-155704377>.
- [8] Miguel A. Otaduy, Takeo Igarashi und Joseph J. LaViola Jr. "Interaction: Interfaces, Algorithms, and Applications". In: *ACM SIGGRAPH 2009 Courses*. SIGGRAPH '09. New Orleans, Louisiana: ACM, 2009, 14:1–14:66. DOI: 10.1145/1667239.1667253. URL: <http://doi.acm.org/10.1145/1667239.1667253>.
- [9] Leap Motion Developer Portal. *SDK Release Hinweise*. [Online; Zugriff am 23.03.2015]. 2014. URL: [https://developer.leapmotion.com/documentation/csharp/supplements/SDK\\_Release\\_Notes.html](https://developer.leapmotion.com/documentation/csharp/supplements/SDK_Release_Notes.html).
- [10] Leap Motion Product. *Leap Motion Product*. [Online; Zugriff am 23.03.2015]. 2014. URL: <https://www.leapmotion.com/product>.
- [11] Samsung. *Smart TV Sprach- und Gestensteuerung - Gesichtserkennung und Touch Fernbedienung*. [Online; Zugriff am 25.03.2015]. 2013. URL: <http://www.samsung.com/de/article/smart-tv-facial-and-gesture-recognition>.

- [12] Marc Sauter. *Mehr Wedeln als Machtgriff*. [Online; Zugriff am 24.03.2015]. 2013. URL: <http://www.golem.de/news/leap-motion-im-test-mehr-wedeln-als-machtgriff-1308-101040.html>.
- [13] Uwe Schäfer. *Holzkiste mit Nägel*. [Online; Zugriff am 21.02.2015]. URL: <http://www.bildburg.de/texturen/holz/holzobjekte/holzkistemitnaegel.html>.
- [14] Dr.-Ing. Alexander Schick. *Qualitätskontrolle per Fingerzeig*. [Online; Zugriff am 23.03.2015]. 2013. URL: <http://www.iosb.fraunhofer.de/servlet/is/33978/>.
- [15] Leap Motion Developer Support. *Generierung eines Python 3.3.0 Wrappers mit SWIG 2.0.9*. [Online; Zugriff am 23.03.2015]. 2013. URL: <https://support.leapmotion.com/entries/39433657-Generating-a-Python-3-3-0-Wrapper-with-SWIG-2-0-9>.
- [16] Uni-Protokolle. *Bildwiederholfrequenz*. [Online; Zugriff am 23.03.2015]. 2015. URL: <http://www.uni-protokolle.de/Lexikon/Bildwiederholfrequenz.html>.
- [17] Matthias Wölfel. *3D Gestenerkennung als moderne Mensch-Maschine Schnittstelle*. [Online; Zugriff am 23.03.2015]. 2013. URL: <http://www.fhv.at/media/pdf/veranstaltungen-vortraege/uday/uday-11/tagungsbeitraege/wifel>.

## Abbildungsverzeichnis

1	Änderungen der Eigenschaften im Reiter General . . . . .	5
2	Änderungen der Eigenschaften im Reiter C/C++ . . . . .	5
3	Änderungen der Eigenschaften im Reiter Linker . . . . .	6
4	Umsetzung der Steuerung am Touchscreen als Diagramm . . .	8
5	Umsetzung der applyTransformation()-Methode als Diagramm	11
6	Geste der Translation . . . . .	13
7	Umsetzung der Translation als Diagramm . . . . .	14
8	Geste der Rotation . . . . .	15
9	Umsetzung der Rotation als Diagramm . . . . .	16
10	Geste der Vergrößerung . . . . .	16
11	Geste der Verkleinerung . . . . .	17
12	Umsetzung der Skalierung als Diagramm . . . . .	18
13	Geste der Bewegung durch den Raum . . . . .	19
14	Umsetzung der Kamera als Diagramm . . . . .	21
15	Umsetzung der Raumbeschränkung als Diagramm . . . . .	23
16	Heranziehen mit der Seil-Metapher . . . . .	27
17	Drehen mit der Seil-Metapher . . . . .	27
18	Startposition der ersten Aufgabe . . . . .	29
19	Startposition der ersten Aufgabe mit Sicht auf alle Objekte .	29
20	Startposition der Aufgabe zur Rotierbarkeit . . . . .	30
21	Proband beim Auswählen eines Objektes . . . . .	33
22	Proband beim Versuch einen Turm zu bauen . . . . .	34
23	Welche Steuerung wird bei der Seil-Metapher bevorzugt? . . .	36
24	Welche Steuerung wird bei der Spieler-Metapher bevorzugt? .	36
25	Welche Steuerung ist intuitiver? . . . . .	40
26	Welche Steuerung ist einfacher? . . . . .	41
27	Erfahrung der Probanden mit den Steuerungen . . . . .	41
28	Einschätzung der Leichtigkeit der Gestensteuerung . . . . .	42
29	Einschätzung der Leichtigkeit der Maussteuerung . . . . .	43
30	Gesamtresultat zur Intuitivität der Steuerung . . . . .	44
31	Gesamtresultat zur Einfachheit der Steuerung . . . . .	44

## Tabellenverzeichnis

1	Bewertung der Metaphern in Bezug auf die Bedienungs- möglichkeit . . . . .	35
2	Zeiten zum Lösen der ersten Aufgabe mit der Seil-Metapher .	35
3	Zeiten zum Lösen der ersten Aufgabe mit der Spieler-Metapher	35
4	Zeiten um einen Turm zu bauen . . . . .	38
5	Zeiten um Objekt gleich aussehen zu lassen . . . . .	39
6	Zeiten um Objekt richtig zu skalieren . . . . .	40
7	Bewertung der Bedienungsmöglichkeiten nach dem Test . . . .	42

# Fragebogen zur Evaluation

\* Erforderlich

Ich bin .. \*

- männlich  
 weiblich

Wie alt sind Sie? \*

Welcher Proband sind Sie? \*

Wie viel Erfahrung haben Sie bei der Bedienung einer Maus? \*

1 2 3 4 5

sehr viel      gar keine

Wie leicht schätzen Sie wird Ihnen die Bedienung der Maus fallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie viel Erfahrung haben Sie mit Gestensteuerung? \*

1 2 3 4 5

sehr viel      gar keine

Wie leicht schätzen Sie wird Ihnen die Bedienung mit Gesten fallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Weiter »

12 % abgeschlossen

# Fragebogen zur Evaluation

\* Erforderlich

## Bewegung im Raum mit der Seil-Metapher

Die Bewegung ähnelt dem Ziehen eines Seiles.

Wie leicht ist Ihnen die Bedienung der Maus beim Drehen im Raum mit der Seil-Metapher gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie leicht ist Ihnen die Bedienung der Maus beim Bewegen im Raum mit der Seil-Metapher gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie leicht ist Ihnen die Gestensteuerung beim Drehen im Raum mit der Seil-Metapher gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie leicht ist Ihnen die Gestensteuerung beim Bewegen im Raum mit der Seil-Metapher gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie viel Zeit haben Sie zur Lösung der Aufgabe mit der Maus bei der Seil-Metapher gebraucht? \*

Std. ▼ : Min. ▼ : Sek. ▼

Wie viel Zeit haben Sie zur Lösung der Aufgabe mit der Gestensteuerung bei der Seil-Metapher gebraucht? \*

Std. ▼ : Min. ▼ : Sek. ▼

« Zurück

Weiter »

 25 % abgeschlossen

# Fragebogen zur Evaluation

\* Erforderlich

## Bewegung im Raum mit der Spieler-Metapher

Ich bewege die Maus bzw. Hand in die Richtung, in die ich schauen will.

Wie leicht ist Ihnen die Bedienung der Maus beim Drehen im Raum mit der Spieler-Metapher gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie leicht ist Ihnen die Bedienung mit der Maus beim Bewegen im Raum mit der Spieler-Metapher gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie leicht ist Ihnen die Gestensteuerung beim Drehen im Raum mit der Spieler-Metapher gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie leicht ist Ihnen die Gestensteuerung beim Bewegen im Raum mit der Spieler-Metapher gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie viel Zeit haben Sie zur Lösung der Aufgabe mit der Maus bei der Spieler-Metapher gebraucht? \*

Std. ▼ : Min. ▼ : Sek. ▼

Wie viel Zeit haben Sie zur Lösung der Aufgabe mit der Gestensteuerung bei der Spieler-Metapher gebraucht? \*

Std. ▼ : Min. ▼ : Sek. ▼

« Zurück

Weiter »

 37 % abgeschlossen

# Fragebogen zur Evaluation

\* Erforderlich

## Bewegung im Raum generell

Welche Metapher bevorzugen Sie bei welcher Steuerung? \*

	Maus	Gesten
Drehen mit der Seil-Metapher	<input type="radio"/>	<input type="radio"/>
Drehen mit der Spieler-Metapher	<input type="radio"/>	<input type="radio"/>
Bewegung mit der Seil-Metapher	<input type="radio"/>	<input type="radio"/>
Bewegung mit der Spieler-Metapher	<input type="radio"/>	<input type="radio"/>

Welche Steuerung ist für Sie beim Drehen im Raum intuitiver? \*

1 2 3

Maus    Gestensteuerung

Welche Steuerung ist für Sie beim Bewegen im Raum intuitiver? \*

1 2 3

Maus    Gestensteuerung

Welche Steuerung war für Sie beim Drehen im Raum einfacher? \*

1 2 3 4 5

Maus      Gestensteuerung

Welche Steuerung war für Sie beim Bewegen im Raum einfacher? \*

1 2 3 4 5

Maus      Gestensteuerung

« Zurück

Weiter »



50 % abgeschlossen

# Fragebogen zur Evaluation

\* Erforderlich

## Turm bauen

Wie leicht ist Ihnen die Bedienung der Maus beim Bewegen der Objekte gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wenn 4 oder 5 angekreuzt: Warum ist Ihnen die Bedienung mit der Maus schwer gefallen?

Wie leicht ist Ihnen die Gestensteuerung beim Bewegen der Objekte gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wenn 4 oder 5 angekreuzt: Warum ist Ihnen die Bedienung mit der Gestensteuerung schwer gefallen?

Wie viel Zeit haben Sie gebraucht, um mit der Maus einen Turm zu bauen? \*

:  :

Wie viel Zeit haben Sie gebraucht, um mit der Gestensteuerung einen Turm zu bauen? \*

:  :

Welche Steuerung ist für Sie beim Bewegen der Objekte intuitiver? \*

1 2 3

Maus    Gestensteuerung

Welche Steuerung war für Sie beim Bewegen der Objekte einfacher? \*

1 2 3 4 5

Maus      Gestensteuerung

« Zurück

Weiter »



62 % abgeschlossen

# Fragebogen zur Evaluation

\* Erforderlich

## Objekte rotieren

Wie leicht ist Ihnen die Bedienung der Maus beim Rotieren der Objekte gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wenn 4 oder 5 angekreuzt: Warum ist Ihnen die Bedienung mit der Maus schwer gefallen?

Wie leicht ist Ihnen die Gestensteuerung beim Rotieren der Objekte gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wenn 4 oder 5 angekreuzt: Warum ist Ihnen die Bedienung mit der Gestensteuerung schwer gefallen?

Wie viel Zeit haben Sie gebraucht, um mit der Maus das Objekt richtig zu rotieren? \*

:  :

Wie viel Zeit haben Sie gebraucht, um mit der Gestensteuerung das Objekt richtig zu rotieren? \*

:  :

Welche Steuerung ist für Sie beim Rotieren der Objekte intuitiver? \*

1 2 3

Maus    Gestensteuerung

Welche Steuerung war für Sie beim Rotieren der Objekte einfacher? \*

1 2 3 4 5

Maus      Gestensteuerung

« Zurück

Weiter »



75 % abgeschlossen

# Fragebogen zur Evaluation

\* Erforderlich

## Objekte skalieren

Wie leicht ist Ihnen die Bedienung der Maus beim Vergrößern/Verkleinern der Objekte gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wenn 4 oder 5 angekreuzt: Warum ist Ihnen die Bedienung mit der Maus schwer gefallen?

Wie leicht ist Ihnen die Gestensteuerung beim Vergrößern/Verkleinern der Objekte gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wenn 4 oder 5 angekreuzt: Warum ist Ihnen die Bedienung mit der Gestensteuerung schwer gefallen?

Wie viel Zeit haben Sie gebraucht, um mit der Maus das Objekt zu skalieren? \*

:  :

Wie viel Zeit haben Sie gebraucht, um mit der Gestensteuerung das Objekt zu skalieren? \*

:  :

Welche Steuerung ist für Sie beim Skalieren der Objekte intuitiver? \*

1 2 3

Maus    Gestensteuerung

Welche Steuerung war für Sie beim Skalieren der Objekte einfacher? \*

1 2 3 4 5

Maus      Gestensteuerung

« Zurück

Weiter »



# Fragebogen zur Evaluation

\* Erforderlich

## Gesamtresultat

Welche Steuerung würden Sie generell als intuitiver erachten, um ... \*

	Maus	Gestensteuerung
sich im Raum zu bewegen	<input type="radio"/>	<input type="radio"/>
Objekte im Raum zu bewegen	<input type="radio"/>	<input type="radio"/>
Objekte im Raum zu rotieren	<input type="radio"/>	<input type="radio"/>
Objekte im Raum zu skalieren	<input type="radio"/>	<input type="radio"/>

Welche Steuerung würden Sie generell als einfacher erachten, um ... \*

	Maus	Gestensteuerung
sich im Raum zu bewegen	<input type="radio"/>	<input type="radio"/>
Objekte im Raum zu bewegen	<input type="radio"/>	<input type="radio"/>
Objekte im Raum zu rotieren	<input type="radio"/>	<input type="radio"/>
Objekte im Raum zu skalieren	<input type="radio"/>	<input type="radio"/>

Wie leicht ist Ihnen die Bedienung der Maus insgesamt gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer

Wie leicht ist Ihnen die Gestensteuerung insgesamt gefallen? \*

1 2 3 4 5

sehr leicht      sehr schwer