Leact: Fusion der KinectV2 und Leap Motion zur erweiterten Menüsteuerung in VR

Patrick Höling, Tobias Michel Latta

Zusammenfassung

In dieser Ausarbeitung wird eine Möglichkeit zur Fusion von KinectV2- und Leap Motion-Sensordaten erklärt um eine Menüsteuerung in der Virtuellen Realität (VR) zu implementieren. Das erstellte Framework nennt sich *Leact* und ist motiviert durch die Kombination aus hoher Auflösung der Fingererkennung der Leap Motion (LM) sowie dem großen Bewegungsradius der Kinect V2. Validiert wird das Konzept über die Implementierung einer kaskadierenden Menüführung und Manipulation von 3D-Objekten. Die Menüführung wird über den großen Bewegungsradius der Kinect V2 realisiert und die Selektion sowie Manipulation der Objekte findet mit der LM statt. Aufgrund der verbesserungswürdigen Verwertung der Kinect V2-Sensordaten, die zu fehlerhaften Interaktionen in rund zehn Prozent der Fälle führt, wird eine Optimierung durch eine Tiefpassfilterung der Sensordaten empfohlen. So werden nur eindeutig und zeitlich über einen gewissen Zeitraum bestehende Gesten erkannt.

# Einleitung

Diese Ausarbeitung ist im Zusammenhang mit der Vorlesung *Human Computer Interaction* (HCI), gehalten von Prof. Dr. Birgit Wendholt im Wintersemester 2017/2018 an der Hochschule für Angewandte Wissenschaft Hamburg, entstanden. Während der Vorlesung werden unterschiedliche Implementierungen im Bereich HCI behandelt und in insgesamt drei Laborterminen getestet. Unter anderem handelt es sich dabei um das VR-Headset *HTC Vive*, sowie das Handtracking-System *Leap Motion*. Beides in Verbindung mit der Entwicklungsumgebung Unity. Ebenfalls evaluiert wurde das Ganzkörper-Tracking-System von Microsoft, genannt *Kinect V2*.

Das Ziel dieser Ausarbeitung ist die Kombination von großem Bewegungsraum der Kinect mit dem genauen Handtracking der Leap Motion um eine intuitive Bedienung mit den Händen vor dem Körper hochauflösend abzubilden sowie außerhalb des Leap Motion-FOV Gestensteuerung zu ermöglichen. Durch die Bedienung der VR mit dem eigenen Körper wird die Immersion nicht zuletzt durch die visuelle Repräsentation der eigenen Hände gefördert. Auch muss kein externes Eingabegerät verwendet werden, welches Körperaktionen nur durch Tastendruck aufnehmen kann. Der Nutzer kann intuitiv die Geste aus der Realität vollziehen und sie wird durch die Leap Motion aufgenommen sowie in der VR verarbeitet.

Diese Intuivität ist wichtig um VR als Medium zu etablieren. An mehreren technischen Produkten lässt sich die Relevanz von einfacher Bedienung erkennen. Beispielsweise ist der Erfolg des Apple iPhone der ersten Generation auch darüber zu erklären, dass es eine intuitive Bedienung geschaffen hat, welche es auch weniger technisch-versierten Nutzern ermöglichte komplexe Funktionen abzurufen und zu nutzen (Laugesen, 2010). So wird die Hemmschwelle zur Nutzung von Technik herabgesenkt, was wiederum bei den Entwicklern und Unternehmern Interesse weckt, in die gleiche Richtung zu investieren/produzieren. Ohne größere Erklärungen oder Schnittstellen (Werkzeuge) sollen Abläufe mit dem eigenen Körper durchgeführt werden, so wie es auch in der Realität hauptsächlich der Fall ist. Die Fehlerhäufigkeit bei einem Funktionsaufruf soll dabei auf maximal einen Fehlaufruf unter fünf Versuchen beschränkt werden.

Um den Lösungsansatz verständlich zu dokumentieren, wird dieser Aufsatz wie folgt gegliedert: Zuerst findet eine wissenschaftliche Einordnung statt, indem die Inhalte mit bestehenden Arbeiten verglichen werden. Darüber hinaus werden verwendete Frameworks und Beispiele zur jeweiligen Implementierung von Kinect V2 und Leap Motion näher erläutert. Anschließend gibt es einen kurzen Rundumschlag über den derzeitigen Stand der Technik, der einen kritischen Blick auf die genutzten Produkte wirft, bevor sie in den Zusammenhang zum Vorlesungsinhalt von *HCI WS17/18* gebracht werden. Diese Einordnung ist sinnvoll um die Sensorik tiefergehend zu erläutern. Der Hauptteil beschäftigt sich mit der Konzeptionierung und Umsetzung der Idee.

# Realisierung

## Wissenschaftliche Einordnung

In dem Paper *Multi-sensor data fusion for hand tracking using Kinect and Leap Motion* (Penelle, 2014) wird eine Verbindung zwischen dem hochauflösenden Handtracking der Leap Motion sowie dem großen Bewegungsbereich der Kinect V1 beschrieben. Auf mathematischer Ebene wird beschrieben, wie die Handgelenke bei beiden Sensoren pixel-genau übereinander zu legen sind, wie also eine Kalibrierung und Registrierung beider Komponenten erfolgt. Anwendungsfall ist dabei die Schmerztherapie bei medizinischen Patienten, welche einen Arm verloren haben. Setzt sich der Betroffene ein VR-Headset auf, sollen die Bewegungen des vorhandenen Arms auf das letzte vorhandene Gelenk auf der amputierten Seite gespiegelt werden, um den Eindruck zu erwecken, dass beide Arme vorhanden sind. So können einfache Tätigkeiten ausgeführt werden und die Illusion des rekonstruierten Armes reduziert den Phantomschmerz. Überschneidungen zu diesem Aufsatz finden sich bei der komplementären Nutzung von Leap Motion und einer Kinect-Kamera. Zwar wird hier eine neuere Kinect-Version verwendet, jedoch ist der grundsätzliche Hintergrund einen größeren Bewegungsraum im Vergleich zur LM zu schaffen ebenfalls gegeben. Ein weiterer Unterschied ist die Herangehensweise an die Thematik. Durch die mathematische Beschreibung der Kalibrierungs- und Registrierungs-Algorithmen kann die Methodik auch auf andere Realisierung des Sensortyps angewendet werden. Dabei rückt jedoch die konkrete Implementierung in einer Entwicklungsumgebung in den Hintergrund, was die Ausführbarkeit für den durchschnittlichen Benutzer beschränkt. In diesem Aufsatz wird eine andere Zielsetzung verfolgt. Es soll ein Framework geschaffen werden, welches Sensorfusion von Kinect V2 und Leap Motion betreibt um eine interaktive und intuitive Menüführung zu bewerkstelligen, welche leicht für eigene Projekte adaptiert und genutzt werden kann.

*Interaktive Systeme (Band II)* (Preim & Dachselt, 2015) liefert in Kapitel 8 einen Überblick über die Interaktion mit 3D-Objekten und empfiehlt beispielsweise eine Kreismenü-Platzierung an festen Weltkoordinaten. Kontextabhängige Positionierung ist ebenfalls beschrieben und wird in dieser Ausarbeitung verwendet. Das Kreismenü ist an das Sichtfeld gekoppelt um stets verfügbar zu sein und eine klare Trennung zwischen Menüführung und Objektmanipulation zu realisieren. Des Weiteren sind im dritten Teil des Buches Natural User Interfaces beschrieben, genauer genommen auch Gesten die mit den Händen oder Armen ausgeführt werden. Die *Zeigegesten* (Preim & Dachselt, S.491, 2017) referenzieren (abstrakte) Objekte in der Umgebung und werden ruhig ausgeführt. Kinder nutzen diese nonverbale Referenzierung aufgrund ihrer intuitiven Deutung durch die Erwachsenen. Um die Menüauswahl möglichst einfach und selbsterklärend zu gestalten, wird daher diese Art von Gesten verwendet. Die verwendeten Körperteile definieren darüber hinaus die Gesten als Hand- und Körpergesten (Preim & Dachselt, S.500-503, 2017).

## Einordnung *HCI Wintersemester 2017/2018*

Nachdem nun Vergleiche zu anderen wissenschaftlichen Arbeiten gezogen wurden, findet eine Verbindung zu den Vorlesungsinhalten statt.

Die Leap Motion, welche 2013 auf dem Markt erschienen ist, kann prinzipiell in jeder Position die Finger vor sich tracken. Häufig wird sie daher auf dem Schreibtisch vor die Tastatur gelegt um die Bedienung mit herkömmlichen Peripheriegeräten durch Handtracking zu ergänzen (Abbildung XXX). Im VR-Kontext wird sie hauptsächlich an dem Head Mounted Display (HMD) als Inside OutVariante fürs Handtracking verwendet (Abbildung XXX).

Abbildung 1:Leap Motion in Desktop-Verwendung



Abbildung 2: Leap Motion an HTC Vive befestigt

Sie arbeitet auf Basis der Infrarot-Laufzeitmessung und gleicht diese Sensordaten mit einem Handmodell ab um nicht plausible Fingerstellungen direkt raus zu filtern. Der genaue Ablauf ist wie folgt: Ein Raster aus Infrarot-Lichtpunkten wird in das FOV gesendet und anschließend wieder aufgenommen. In Abhängigkeit der Entfernung verändert sich die Zeit, welche das Licht benötigt um wieder einfangen zu werden. Mit allen Punkten kann somit das Raster in ein dreidimensionales Bild umgewandelt werden, welches wiederum auf charakteristische Gelenke untersucht wird. Gibt es nun mehrere Übereinstimmungen mit den Gelenken eines beispielhaften Handmodells, wird es auf die detektierten Gelenkpunkte gelegt und eine digitale Repräsentation findet statt. Liegt nun die Hand außerhalb des Infrarot-Rasters, kann dementsprechend keine Detektion stattfinden. Die 27 Freiheitsgrade einer Hand werden nur selten (1 von 100) falsch digitalisiert. Hauptsächlich treten Fehler auf, wenn sich die Hand an den Rand des *Sichtbereichs*/*Field of View* (FOV) begibt oder in einer unvorhergesehenen Position steht, da dann keine plausiblen Übereinstimmungen zu dem Beispiel-Handmodell bestehen.

Die Kinect V2 ist eine Infrarot-Tiefenkamera, welche von Microsoft für das Ganzkörper-Tracking entwickelt wurde. Sie arbeitet nach dem Outside-In Verfahren, benötigt also einen festen Standpunkt um den Nutzer zu tracken. Mit 6 Freiheitsgraden wird die Position wie folgt ermittelt: Eine großflächige Infrarot-Lichtquelle wird von der Kinect ausgesendet. Der zu detektierende Nutzer reflektiert in Abhängigkeit seines Standpunktes und seiner Haltung gewisse Bereiche der Infrarotlichtwolke, die von der Infrarot-Kamera aufgenommen werden. Dabei ist es ebenfalls möglich, Infrarotlicht aufzunehmen, welches von dem detektierten Objekt selbst ausgesendet wird. Beispielsweise kann so parallel zur Entfernung des Nutzers von der Kinect auch die Temperatur des Nutzers gemessen werden. In der Praxis wird eine Entfernung von 0,5 m bis 2,5 m zur Kinect empfohlen. Außerdem liegt der Fokus auf einer Ganzkörpererkennung, welche bis zu 25 Gelenke des Körpers trackt. Im Bereich der Hand sind nur der Daumen, der Zeigefinger sowie das Handgelenk erfassbar. Es treten bei Nutzung in mehr als zehn Prozent der Fälle falsche Erkennungen der Handstellung auf, was vermutlich an der Sensorverarbeitung liegt, welche ohne weiterführende Logik, die Position der Gelenke zu Gesten interpretiert. Nach der Entwicklung von Kinect V1 und V2 wird in Zukunft keine Weiterentwicklung dieses Produktes erfolgen (Warren, 2017).

Da der Fokus dieser Ausarbeitung auf den eben genannten Tracking-Systemen liegt, wird nachfolgend nur ein kleiner Überblick über den verwendeten VR-Aufbau gegeben.  
Als Anzeigemedium wird die HTC Vive in Verbindung mit der Entwicklungsumgebung Unity verwendet. Derzeitig verfügt das HMD über eine Auflösung von 2160 x 1200 Pixel und ein Blickfeld von 110°[[1]](#footnote-2). Unity hat sich als Entwicklungsumgebung für VR-Inhalte durchgesetzt. Es besitzt eine hohe Kompabilität zu unterschiedlichen VR-Headsets und ermöglicht eine unkomplizierte Implementierung von gewünschten Inhalten.

## Implementierung

### Verwendete Beispiele und Frameworks

* VR-Package von Rumen Filkov 🡪 Erweiterung durch Leap
* Dynamic UI von LeapMotion 🡪 Erweiterung durch Kinect
* RadialMenuFramework von Brett Gregory 🡪 Visualisierung in VR und Aufruf über Kinect/Leap

## Usability und Bewertung

Da der Fokus dieser Ausarbeitung auf der technischen Umsetzung liegt, lassen sich Verbesserungen hauptsächlich im Bereich der Usability finden. Bespielsweise ist derzeitig eine einhändige Menüführung realisiert, welche jedoch aufgrund der einseitigen Armbelastung auf Dauer als anstrengend empfunden werden kann. Würde eine zweihändige Gestensteuerung implementiert werden, welche die Menüaufrufende Hand als dominant definiert, könnten sich die Körperhälften abwechseln. Dadurch würde nicht nur jeder Nutzer seine dominante Körperhälfte nutzen können, sondern auch die subdominante Hälfte für weitere, steuernde Gesten. Derzeitig ist beispielsweise die Geste zum Schließen des Menüs durch ein Hereinführen der linken Hand in den Bereich der LM realisiert. Würde man das Schließen auf eine Wischgeste der rechten Hand legen, könnte die Bedienung erweitert werden. Generell lässt sich diese Anwendung auf weitere Konzepte zur alternativen Bedienung prüfen.

Im technischen Bereich ist ein Hauptaugenmerk auf das Refactoring zu legen. Derzeitig bedingt die Umsetzung mehrere Abänderungen der APIs von LM und Kinect V2. Es wäre jedoch sehr gut, eine Implementierung alleine durch das Einfügen des selbst geschriebenen Packages zu realisieren. Dafür müssen, wie in Abbildung XXX zu sehen, die Fassaden-Methoden des SHL und SLL mit in den Leact-Manager gezogen werden um ein autonom funktionierendes Package zu ermöglichen. Auch kann man das Hinzufügen und Abändern der Menühierarchien vereinfachen, wenn durch eine Add-Methode des Leact-Managers die Vorgänge erledigt werden, welche die notwendigen Einträge und Verknüpfungen erstellen um eine Einbindung des neuen Menüs zu bewerkstelligen.

Generell ist eine Erweiterung für mehrere, gleichzeitige Nutzer wünschenswert um die Einsatzmöglichkeiten des Frameworks noch umfangreicher zu machen. Die Verwertung von Sensorsignalen innerhalb der vorhandenen Frameworks kann ebenfalls überarbeitet werden um falsche Erkennung von Gesten und Handstellungen zu vermeiden.

# Reflexion

* Einsetzen des SimpleLeapListeners sowie eigens geschriebenen Methoden aus SimpleHandListener in LeactManager.cs um Funktionalität zu ermöglichen ohne jegliche Änderungen an Leap/Kinect-Komponenten vorzunehmen
* SelectionFollower des RadialMenu fixen
* MultiUser Support mit zweiter Leap
* Menüunterscheidung für Linke/Rechte Hand machen, damit unterschiedliche Menüs mit unterschiedlichen Händen gespawnt werden können 🡪 sogar gleichzeitig jeweils mit links und rechts ein Menü spawnen lassen, welche unabhängig voneinander anwählbar sind (mit jeder Hand autonom Auswahl möglich)
* Geste zum Schließen des Radialmenüs von *Hände in Leap sichtbar* ändern, da es teilweise etwas buggy ist und nicht sofort ausgeblendet wird, wenn LeapHände zu sehen sind🡪 über Kinect könnte Wischgeste mit inaktiver Hand zum Schließen des Menüs führen
* Weiteren Menüpunkt/Geste um Objekte zu Löschen 🡪 wenn Objekt im Anchor ist wird es durch Wischgeste mit anderer Hand gelöscht / oder aber Ziehen des Objektes aus dem Sichtbereich des Leap-Controllers führt zum Löschen
* Hinzufügen von weiteren Menühierarchein wird jetzt bewerkstelligt durch
  + 1. Neues Prefab in MenuFabric
  + 2. Id für Menü vergeben im public RMF\_RadialMenu menu(…)
  + 3. Id im RMF\_RadialMenu-Attribut angeben
  + 4. Menü als Nachfolger eines anderen Menüs deklarieren
  + 5. Ggf. Nachfolger des eingefügten Menüs festlegen
  + Automatisieren dieses Vorgangs durch Add-Methode, mit Parametern zu neuer Id, Menünachfolgern und Menüvorgängern, welche vorherige Schritte selbstständig übernimmt
* Besseres Abfangen von falschen Kinect-Gesten durch Tiefpassfilterung von Sensordaten

# Zusammenfassung / Ausblick

Während der Bearbeitung dieses Projekts, beziehungsweise dieser Ausarbeitung, wird ein großer Einblick in die VR-Programmierung geliefert. Dabei liegt der Erkenntnisgewinn eher bei der Arbeit mit Unity als beim Verfassen der C#-Skripte. Durch die C#-Entwicklungsumgebung *MonoDevelop*, welche bei Unity standardmäßig für die Bearbeitung der Skripte eingestellt ist, wird durch Autokorrektur- und Vervollständigung das Verfassen stark vereinfacht. Daher ist es leicht, die entwickelten Konzepte für die Programmlogik in Code umzuwandeln. Sehr viel interessanter ist die Arbeit mit dem *Szenen-Konzept* von Unity. Das Einfügen und wichtige verschachteln der Projekt-Strukturen über die *Hierarchy* stellt sich als recht intuitiv heraus. Eine gewisse Einarbeitungszeit benötigt es jedoch, die Umsetzung von angefügten Komponenten an ein Projekt-Objekt nach zu vollziehen. Da benutzte Frameworks meist Skripte verwenden, welche für eigene Implementierungen noch abgeändert werden, ist es teilweise nicht so leicht, das richtige *GameObject* auszuwählen, welches die Skript-Komponente beinhaltet. Generell erfolgt das Einfügen von bereits vordefinierten Projekt-Bauteilen, den sogenannten *Assets*, sehr intuitiv. Durch den offiziellen und integrierten *Asset-Store* können vorprogrammierte Bausteine mit gewissen Funktionalitäten oder Optiken mit wenigen Klicks in das eigene Projekt eingebunden werden.

Generell ist es eine gute Erfahrung zwei Sensoren erfolgreich zu implementieren und durch die Kombination beider Signale eine Erweiterung der bestehenden Funktionalitäten zu erreichen. In dieser Ausarbeitung gelingt es eine Ganzkörper-Gestensteuerung mit hochauflösendem Handtracking zu erweitern. Die Beschränkungen bei der Nutzbarkeit dieses Produktes sind auf die Ungenauigkeiten der Sensorsignale zurückzuführen. Die Daten der Infrarot-Kameras lassen sich nur mit entsprechender Vor- und Nachfilterung durch die mitgelieferten Frameworks verwenden. Dadurch ist man auf die eingestellten Parameter angewiesen und muss bei ungenügender Genauigkeit eigene Algorithmen zur Verbesserung der Signale aufbauen. In dem Rahmen dieses Projektes ist es leider nicht möglich, kann jedoch für zukünftige Weiterentwicklungen in Aussicht gestellt werden.

Grundsätzlich sollte bei einer Weiterentwicklung in Betracht gezogen werden, dass eine offizielle Weiterentwicklung von Leap Motion und Kinect ausgesetzt ist. Daher ist es eventuell nicht sinnvoll neue Ausarbeitungen auf Basis dieser Produkte umzusetzen, welche keinen Support mehr erfahren. Trotz alledem sind diese Sensoren in ihrem Anwendungsfeld noch die besten Implementierungen für die VR. Der Fokus sollte dennoch auf neuen, frischen Produkten liegen, welche zukünftig auf dem Markt erscheinen oder auch nicht direkt zur VR gehören wie beispielsweise die Microsoft HoloLens mit ihrer eingebauten Handgesten-Erkennung.

Literaturverzeichnis

Im Literaturverzeichnis ordnet ihre die Angaben bitte alphabetisch nach Nachnamen des ersten Autors und dann nach Veröffentlichungsdatum (bei mehreren Titeln von einem Autor).

Es gibt eine Reihe sehr unterschiedlicher Regelungen und Normen für Literaturangaben. Bitte die folgende Syntax verwenden:

* Für Buchzitate: <Autor(en)> <(Jahr)>. <*Titel*>. <Erscheinungsort>: <Verlag>. Die Autoren erscheinen dabei in der Form: <Nachname>, <Abgekürzte Vornamen> – Zwei Autoren werden mit & getrennt. Bei mehr als zwei Autoren werden die letzten beiden ebenfalls mit &, alle weiteren durch Kommata getrennt. Bei Verlagen mit mehreren Verlagsorten wird nur der erste genannt.
* Für Zeitschriftenzitate: <Autor(en)> <(Jahr)>. <Titel>. <*Titel der Zeitschrift*> <*Jahrgangsnummer*(Band-)Nummer>, <Seitenangaben>.
* Für Zitate aus Tagungs- und Sammelbänden: <Einzelbeitrag wie bei Zeitschriften>. In <Herausgeber, analog zu Autor(en)> (Hrsg.): <*Titel*> [eventuell <*Auflage*>]. <Erscheinungsort>: <Verlag>. [S. <Seitenangaben>].

Für das Layout des Literaturverzeichnisses ist das Format <Literatur> zu verwenden. Angaben zur Formatierung der Literaturverweise finden Sie in Abschnitt 4.4.

Nachfolgend sind beispielhaft einige Literaturangaben aufgeführt:

Borghoff, U. M. & Schlichter, J. (1998). *Rechnergestützte Gruppenarbeit – Eine Einführung in Verteilte Anwendungen. 2. Auflage*. Heidelberg: Springer.

DIN EN ISO 9241-110 (2006). *Ergonomics of human-system interaction – Part 110: Dialogue principles*. Beuth, Berlin.

Friedrich, J., Herrmann, T., Peschke, M. & Rolf, A. (Hrsg.) (1995). *Informatik und Gesellschaft*. Heidelberg: Spektrum Akademischer Verlag.

Nake, F. (1993). *Die erträgliche Leichtigkeit der Zeichen*. Baden-Baden: Agis.

Oberquelle, H. (1991). MCI – Quo Vadis? Perspektiven für die Gestaltung und Entwicklung der Mensch-Computer-Interaktion. In Ackermann, D. & Ulich, E. (Hrsg.): *Software-Ergonomie '91*. Stuttgart: Teubner, S. 9-24.

Raskin, J. (1994). Intuitive equals Familiar. *Communications of the ACM, 37*(9), 17-18.

Robertson, A (2016). *Leap Motion's revamped hand tracking is getting built straight into VR headset*. <https://www.theverge.com/2016/2/17/11021214/leap-motion-hand-tracker-virtual-reality-orion-mobile-vr>

Penelle, B. (2014). *Multi-Sensor Data Fusion for Hand Tracking using Kinect and Leap Motion*.Université Libre Brussels. Brüssel. <https://dl.acm.org/citation.cfm?id=2620710>

Laugesen, John & Yufei, Yuan (2010). *What factors contributed to the success of Apple’s iPhone?*. McMaster University, DeGroote School of Business, Ontario

Preim, Bernhard & Dachselt, Raimund (2015). *Interaktive Systeme, Band 2: User Interface Engineering, Natural User Interfaces*, Berlin, Springer Vieweg Verlag, S 343-396

Warren, Tom (2017). *Microsoft kills of Kinect, stops manufacturing it*. <https://www.theverge.com/2017/10/25/16542870/microsoft-kinect-dead-stop-manufacturing>

* (optional => VR als bedeutendes Medium in der dritten Entwicklungsphase)
* Während des Praktikums sind Grenzen der Handerkennung und großer Bewegungsradius von Kinect aufgefallen sowie der geringe Bewegungsradius der LeapMotion, welche dafür aber eine gute Handerkennung implementiert.
* Vorstellung positive Eigenschaften beider Sensoren zu kombinieren um eine gute Handerkennung im nahen Sichtbereich (120° FOV, max. 1 m Entfernung) des HMD-Trägers mit Gestenerkennung im Bereich um den HMD-Träger
* Beitrag zu Informatik: liefern Basis um die Eingabemethoden in VR durch bestehenden Sensoren zu verbessern, um das Interagieren intuitiver zu machen da keine Controller sondern nur der eigene Körper verwendet wird
* Es soll erreicht werden, dass man in einer Umgebung ohne große Erklärung Menüpunkte auswählen kann
* Ziel: hinreichend gute Bedienung in der VR mit minimaler Interaktion realer Gegenstände
* Anforderung: Fehlerhäufigkeit beschränken auf maximal „sehr häufig“ (1 von 5 Versuchen)
* Aufbau der Arbeit:
  + Vergleich:
    - Arbeit von Penelle2014 🡪 nicht so wissenschaftlicher Ansatz; leichter selbst anzuwenden; Visualisierung der ausgewählten Bereiche; konkreter Menüaufrauf
      * VR-Package von Rumen Filkov 🡪 Erweiterung durch Leap
    - Dynamic UI von LeapMotion 🡪 Erweiterung durch Kinect
    - RadialMenuFramework von Brett Gregory 🡪 Visualisierung in VR und Aufruf über Kinect/Leap
  + Stand der Technik:
    - LeapMotion wird gemischt aufgefasst -> zukünftig wird direkter Einbau in VR-Headsets angestrebt (Robertson, 2016)
    - Kinect wird nicht weiterentwickelt
    - Allgemeiner Trend zu Augmented Reality mit HoloLens (eingebaute Gestenerkennung mit möglicher Erweiterung zu Handtracking)
  + Einordnung in Vorlesungsinhalt:
    - LeapMotion am HeadMountedDisplay als InsideOut-Variante (Nutzer braucht keine Kameras aufstellen, die sich nicht mitbewegen) für Handtracking mit Infrarotlicht-Laufzeitmessung und Modellabbildung (insgesamt 27 DOF in der Hand)
    - KinectV2 als Outside-In Bewegungstracker, der unter anderem mit Infrarotlicht-Laufzeitmessung die Position mit 6DOF bestimmt
    - Unity als Entwicklungsumgebung für VR-Inhalte (unter anderem zu Nutze machen von *FindObjectByTag*.
    - HTC Vive als HeadMounted-Display
    - Umfang / Aufbau und Inhalt

1. https://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/ [↑](#footnote-ref-2)