

Bitte Platzhalter löschen  
und durch eigenes Bild  
ersetzen



# Bringing Your Hands Into Virtual Reality

Pflichtenheft

Studiengang: Informatik  
Autoren: Simon Meer  
Betreuer: Prof. Urs Künzler  
Experten: Yves Petitpierre  
Datum: 01.02.2014

# Versionen

Version	Datum	Status	Bemerkungen
0.9	09.03.2015	Entwurf	Entwurf erstellt

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Vision “IMVR”</b>	<b>2</b>
2.1	Problemstellung . . . . .	2
2.2	Technologien . . . . .	2
2.3	Anwendung . . . . .	5
<b>3</b>	<b>Anforderungen</b>	<b>8</b>
3.1	Funktionale Anforderungen . . . . .	8
3.2	Sonstige Spezifikationen . . . . .	9
3.3	Use-Cases . . . . .	10
3.4	Designskizzen . . . . .	10
<b>4</b>	<b>Prototyp</b>	<b>11</b>
4.1	Implementierung . . . . .	11
4.2	Probleme . . . . .	11
<b>5</b>	<b>Organisation</b>	<b>12</b>
5.1	Projektbeteiligte . . . . .	12
5.2	Projektmanagement . . . . .	12
5.3	Zeitplan . . . . .	12
<b>6</b>	<b>Tests</b>	<b>13</b>
	<b>Glossar</b>	<b>14</b>

# 1 Einleitung

Der Markt der stereoskopischen Brillen befindet sich in einem starken Wachstum. Oculus Rift, SteamVR, Project Morpheus, Nvidia VR - die Liste der teilnehmenden Parteien wächst und wächst, doch neue Ausgabegeräte benötigen auch neue Eingabegeräte. Die Leap Motion kann als ein solches gezählt werden und bietet eine freie Handerkennung ohne irgendwelche Handschuhe tragen zu müssen.

In diesem Projekt geht es darum, die neuartige Eingabemöglichkeiten der Leap Motion in Verbindung mit der Oculus Rift einzusetzen und damit eine interaktive Virtual Reality (VR)-Applikation zu schreiben. Im Rahmen eines vorhergehenden Projekts wurde die Materie bereits bearbeitet und die Möglichkeiten erforscht. Es geht nun also hauptsächlich um die erfolgreiche Anwendung der erarbeiteten Informationen.

## 2 Vision “IMVR”

Die Vision soll dazu dienen, einen Überblick über die Applikation zu geben, die im Rahmen der Bachelorarbeit erstellt wird.

### 2.1 Problemstellung

Es soll eine Applikation namens “IMVR” entwickelt werden, welche Gebrauch von der Oculus Rift macht, um die Bilder- und Musiksammlung des Anwenders ansprechend darzustellen, z.B. in Form eines 3D-Karussells. Die zusätzliche “Tiefe”, die durch den Einsatz eines Head-Mounted Display (HMD) entsteht, soll dem Anwender helfen, sich in seiner Medienbibliothek schneller zurechtzufinden.

Zusätzlich dazu soll die Leap Motion dazu verwendet werden, um vollständige Handfreiheit zu gewähren: Der Anwender soll komplett ohne Maus und Tastatur imstande sein, sich durch seine Bilder zu navigieren.

Kurz zusammengefasst muss die Applikation:

- Die Bild- und Musikbibliothek des Benutzers in stereoskopischem 3D darstellen.
- Diese freihändig durchsuchbar machen mit Sortier- und Gruppierfunktion.
- Die Bilder betrachtbar und die Musik abspielbar machen.
- Metainformationen darstellen (z.B. in Form von Diagrammen).

### 2.2 Technologien

Das Projekt verwendet spezielle Hardware und Software. Es folgt eine kurze Erklärung zu diesen Technologien.

### 2.2.1 Oculus Rift

Die Oculus Rift ist ein stereoskopisches HMD, welches durch Oculus VR entwickelt wird. Die aktuelle Version ist das Developer Kit 2 (DK2). Ein Termin für die finale Version steht noch nicht fest.

Hardwaremässig besteht die Rift aus einem Headset für das Bild und einer Kamera für das Head-Tracking. Das Headset wird per USB und HDMI an den Computer angeschlossen und über ein Stromkabel mit Strom versorgt. Die Kamera wird auf dem Computerbildschirm platziert, und per USB an den Computer und per Sync-Kabel an die Rift angeschlossen.

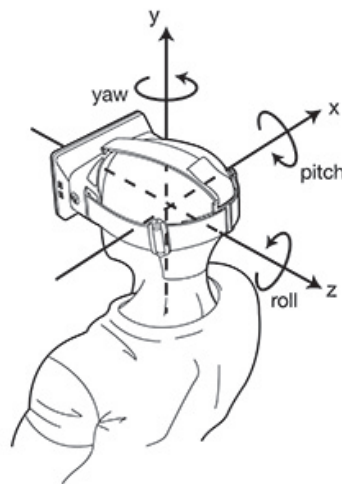


Abbildung 2.1: Illustration, welche die Anwendung und das Koordinatensystem der Oculus Rift verdeutlicht.

Quelle: <https://www.oculus.com/blog/building-a-sensor-for-low-latency-vr/>

Seit DK2 ist das Headset mit 40 Infrarot-LEDs bestückt, welche mit einer bestimmten Frequenz aufleuchten und von der Kamera für das Head-Tracking benutzt werden.

Auf der Software-Seite wird ein Treiber auf dem Computer installiert, der nach einer Registrierung als Entwickler auf der offiziellen Seite erhältlich ist. Die Software erlaubt die Erstellung von Benutzerprofilen, um die Inter-Pupillary Distance (IPD) korrekt einzustellen. Es ist ausserdem möglich, zwischen zwei Betriebsmodi auszuwählen: dem traditionellen Modus, wo die Rift als zweiter Bildschirm angesprochen wird, und dem neuen "Direct Mode", wo die Rift direkt angesprochen wird.

### 2.2.2 Leap Motion

Die Leap Motion besteht aus einem rechteckigen Gerät mit zwei Infrarotsensoren, welche deren Daten per USB auf den angeschlossenen Computer überträgt. Nach der Installation der Leap Motion Runtime läuft auf dem Computer ein Service, der diese Daten empfängt, verarbeitet und per API mit einer variablen Framerate verschiedenen Applikationen zur Verfügung stellt.

Die Daten, welche diese API liefert, sind in sogenannte "Frames" gruppiert. Ein Frame ist sozusagen eine Momentaufnahme der Realität, welche sich aus erkannten Händen zusammensetzt. Durchschnittlich werden pro Sekunde etwa um die 100 dieser Frames berechnet.

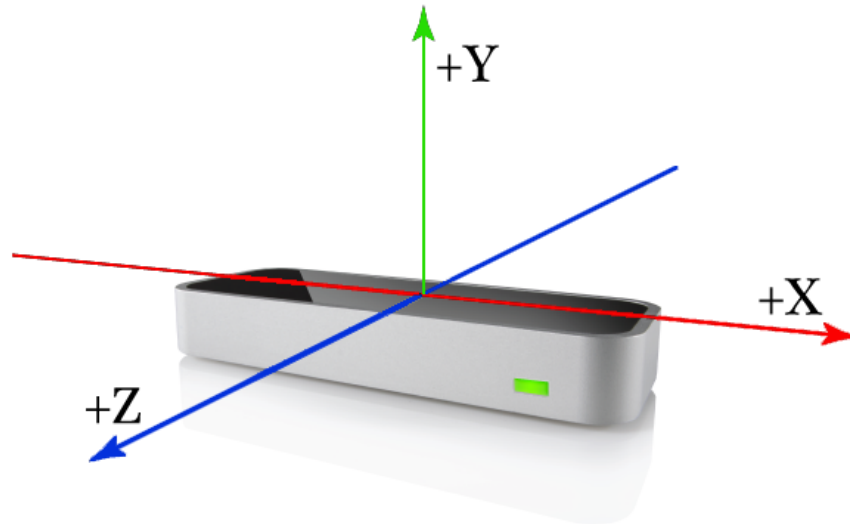


Abbildung 2.2: Verdeutlichung des Koordinatensystems.

Quelle: [https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Coordinate\\_Mapping.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Coordinate_Mapping.html)

Durch diese Frames kann man auf die Daten der Hände zugreifen. Jede Hand erhält eine ID, womit man gleiche Hände Frame-übergreifend identifizieren kann, sowie die dazugehörenden Koordinaten und Drehungen. Auf Abbildung 2.2 wird ersichtlich, dass sich die Koordinaten in einem rechtshändigen Koordinatensystem befinden, wobei die Y-Achse nach oben zeigt und die Masse in Millimeter angegeben sind. Die Koordinaten der Finger sind innerhalb von Finger-Instanzen gruppiert, welche wiederum ihre Gelenke als Instanzen einer Joint-Klasse zur Verfügung stellen.

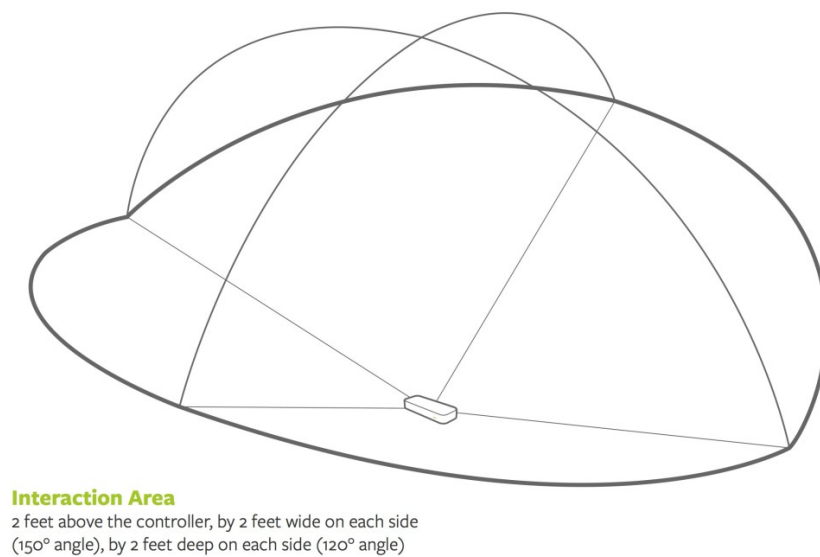


Abbildung 2.3: Das Sichtfeld der Leap Motion.

Quelle: <https://community.leapmotion.com/t/accurately-measuring-distances/842/3>

Die Leap Motion hat ungefähr eine Reichweite von einem Meter, wobei das Interaktionsfeld einer Form, wie in Abbildung 2.3 ersichtlich ist, entspricht. Das Sichtfeld liegt bei 150° in vertikaler und 120° in horizontaler Richtung.

### 2.2.3 Unity 5

Unity ist eine Entwicklungsumgebung und eine Spiel-Engine, die momentan aufgrund ihrer Bedienungsfreundlichkeit und einer frei erhältlichen Version sehr beliebt in der Szene der Indie-Developer ist. Gleichzeitig dient sie auch als gutes Prototyping-Tool, um schnell Ideen umzusetzen.

Am 3. März 2015 machte Unity einen Versionssprung und ist nun als Unity 5 erhältlich. Neben anderen Neuerungen sind jetzt alle Funktionen, für die früher eine kostenpflichtige Lizenz erworben werden musste, auch in der freien Version erhältlich, was auch für dieses Projekt einen grösseren Spielraum bedeutet.

In diesem Projekt wurde Unity wegen der angenehmen Lernkurve und der guten Integrierung mit der Oculus Rift und der Leap Motion für den Einsatz gewählt. Wie bereits erwähnt ist es leicht, schnell zu Ergebnissen zu kommen und die Entwicklungsumgebung verwendet die zwei Programmiersprachen für die Entwicklung, die der Verfasser dieses Dokuments am besten beherrscht.

## 2.3 Anwendung

Innerhalb von IMVR wird die Leap Motion mit einer Halterung an der Oculus Rift befestigt, welche auf der offiziellen Website von Leap Motion für rund \$15 erhältlich ist. Wie in Abbildung 2.4 ersichtlich ist, sind beide Geräte unter Anderem per USB am PC befestigt und senden darüber



ihre Daten an die entsprechenden Services. Diese leiten wiederum die ausgewerteten Daten durch öffentliche APIs an Unity bzw. IMVR weiter.

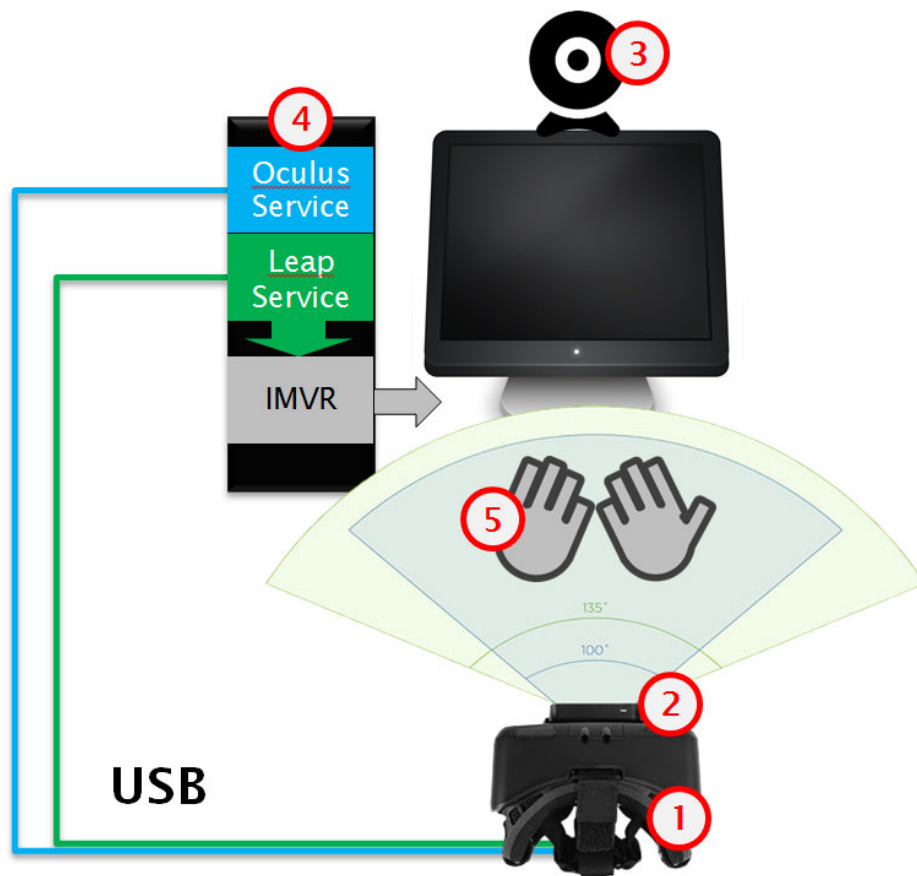


Abbildung 2.4: Eine Übersicht der Technologien und wie sie verbunden sind.

Nr.	Komponente	Beschreibung
1.	Oculus Rift DK2	HMD für den grafischen Output.
2.	Leap Motion	Gerät, welches Hände erkennt und ihre Koordinaten an den Computer sendet.
3.	Oculus Rift Kamera	Kamera, welche seit dem DK2 für das örtliche Tracking zuständig ist.
4.	Computer	Host-System für IMVR.
5.	Benutzer	Benutzer, der die Oculus Rift trägt und mit seinen Händen das Programm steuert.

# 3 Anforderungen

In diesem Kapitel sollen die Anforderungen, die in der Vision grob geschildert wurden, konkretisiert und aufgelistet werden.

## 3.1 Funktionale Anforderungen

In der folgenden Tabelle sind alle Anforderungen aufgelistet mit einer Priorität von 1 bis 5, wobei eine tiefere Zahl für eine höhere Priorität steht.

Nr.	Beschreibung	Priorität
<b>1. Elementares</b>		
1.1	Die Applikation nutzt die Oculus Rift im direkten Modus für die Ausgabe.	1
1.2	Die Applikation ist vollständig mit den Händen bedienbar.	1
1.3	Die Applikation erlaubt den Zugriff auf alle gefundenen, validen Dateien.	1
1.4	Bilder können angeschaut werden, Musik kann angehört werden.	1
1.5	Die Hintergrundmusik kann von der Musikbibliothek ausgewählt werden.	2
1.6	Die Applikation kann jederzeit beendet werden.	1
1.7	Hände werden abstrakt dargestellt.	2
1.8	Die Applikation hat eine stetige Framerate von mindestens 60fps.	1
<b>2. Übersicht (Bilder)</b>		
2.1	Elemente können in einem 3D-Karussell dargestellt werden.	1
2.2	Elemente können selektiert werden.	2
2.3	Elemente können getaggt werden.	3
2.4	Es kann auf ein bestimmtes Element fokussiert werden (siehe Detailansicht).	1
2.5	Es ist eine Mehrfachselektierung möglich.	3
2.6	Elemente können in anderen Anordnungen dargestellt werden (Tunnel, Fluss, Fläche, Würfel, etc.)	3
2.7	Bilder können nach Farbton, Sättigung und Helligkeit sortiert werden.	1
2.8	Bilder können alphabetisch und chronologisch sortiert werden.	2
2.9	Bilder können nach Dateiordner gruppiert werden.	3
<b>3. Detailansicht (Bilder)</b>		
3.1	Bild kann vergrößert und verkleinert werden.	1
3.2	Bild kann gelöscht werden	2

Nr.	Beschreibung	Priorität
3.3	Ein (3D)-Histogramm des Bildes wird angezeigt.	3
3.4	Die Metadaten von Fotos werden dargestellt	4
3.5	Der Hintergrund passt sich den Metadaten entsprechend an	5
3.6	Es können zusätzliche Versionen des Bildes (inkl. Histogramm) angezeigt werden und mit Punktoperationen verändert werden.	4
3.7	Die zusätzlichen Bilder können auch mit lokalen und globalen Operationen gefiltert werden.	
<b>4. Übersicht (Musik)</b>		
4.1	Unterstützt MP3 und WAVE.	1
4.2	Unterstützt weitere Musikformate (Ogg Vorbis, FLAC, M4A, APE, TAK, etc.)	4
4.3	Dateien werden alphabetisch gruppiert nach Artisten dargestellt	1
4.4	Dateien können nach Album, Genre und Jahr gruppiert werden	3
4.5	Dateien können ungruppiert dargestellt werden.	5
4.6	Alben können anhand ihres Covers wie Bilder dargestellt werden.	3
<b>5. Detailansicht (Musik)</b>		
5.1	Album mit einer Liste von Liedern wird dargestellt.	1
5.2	Eine Datei kann zur Wiedergabe ausgewählt werden.	1
5.3	Die Wiedergabe wird visuell untermalt (Spektrogramm)	2
5.4	Informationen zum Artisten werden dargestellt (Ort, Beschreibung, Gründungsjahr)	3
<b>6. Indexierung</b>		
6.1	Der Benutzer kann einen oder mehrere Ordner angeben, die nach Bildern und Musik gescannt werden.	1
6.2	Die gefundenen Dateien werden mit diversen Kennwerten indexiert.	1
6.3	Die Indexierung kann innerhalb der Applikation durchgeführt werden.	4
<b>7. Sonstige Funktionen</b>		
7.1	Es können Filter über die Musik gelegt werden (Low-Pass, High-Pass, Compressor, etc.)	4
7.2	Es ist eine Netzwerkfunktion vorhanden, die es erlaubt, die Fotosammlung zu zweit anzuschauen.	5
7.3	Bildergruppen können als "Diashow" angeschaut werden.	4
7.4	Gesamtstatistiken können angezeigt werden (Bildalter / Anzahl, Auflösung / Anzahl, Auflösung / Anzahl, Kartenchart mit Anzahl Fotos)	3

Tabelle 3.1: Pakete

## 3.2 Sonstige Spezifikationen

- Die Applikation wird aufgeteilt in eine Unity-Applikation und einen Indexer.
- Die Unity-Applikation wird erstellt mit Unity 5 Personal.

- Der Indexer wird erstellt mit C# für das .NET Framework 4.
- Für die Speicherung der Daten wird eine SQLite-Datenbank verwendet.
- Als Entwicklungsumgebung wird Visual Studio 2012 benutzt.

### **3.3 Use-Cases**

(Use-Cases mit Gesten)

### **3.4 Designskizzen**

(Skizzen und Abläufe)

## 4 Prototyp

In der Inception-Phase wurde ein Prototyp geschrieben, um zu testen, ob Unity in der Lage ist, Bilder und Musik so zu manipulieren, wie es die Anforderungen verlangen.

### 4.1 Implementierung

Der Prototyp besteht aus zwei Teilen: dem Unity-Programm und dem Indexer, der im Hintergrund Infos über Dateien sammelt. Zwischen den beiden Prozessen gibt es eine dateibasierte SQLite Datenbank, auf die mit normalem SQL oder ORM-Modellen zugegriffen werden kann.

Um Bilddateien zu analysieren, wird ImageMagick, welches bereits eine Vielzahl von Kennwerten zurückgibt, verwendet, da dieses über ein praktisches C#-Interface verfügt.

Musikdateien werden noch nicht analysiert. Dies kann jedoch mithilfe des jMIR-Projekts, der VAMP-Plugins und der Echo Nest API erreicht werden. Die Visualisierung der Musik wurde mit der NAudio Library realisiert - es ist allerdings möglich, die nötigen Werte direkt über die Unity-API zu ermitteln.

Alles in allem zeigt das Unity-Programm, dass es ohne Ruckeln möglich ist, 100 Bilder zu laden und zu sortieren. Die anfänglichen +100 Draw-Calls konnten auf 2 reduziert werden.

### 4.2 Probleme

In Sachen Bilder gibt es ein Problem, das es zu überwinden gilt: Unity erlaubt es nicht, Texturen asynchron zu laden. Texturen der Grösse 256x256 brauchen nach einer Optimierung etwa 1ms zum Laden, was für eine Framerate von 60fps noch verkraftbar ist. Grössere Texturen (512, 1024) erreichen jedoch schnell eine Ladezeit von 18-50ms, was sich langsam aber sicher bemerkbar macht.

Eine Möglichkeit dieses Problem zu lösen, findet sich in den nativen Plugins: Es ist in Unity möglich, DirectX (oder OpenGL) Code in C++ zu schreiben. Mit dieser Methode könnte man evtl. die Texturen besser allozieren.

Einfacher wäre es allerdings, alle nötigen Bilder einfach vorzuladen.

# 5 Organisation

Dieses Kapitel beschreibt kurz die Stakeholder des Projekts sowie den allgemeinen Aufbau.

## 5.1 Projektbeteiligte

Name	Funktion	E-Mail Adresse
Simon Meer	Durchführer	simon.meer@students.bfh.ch
Prof. Urs Künzler	Betreuer	urs.kuenzler@bfh.ch
Yves Petitpierre	Experte	?

## 5.2 Projektmanagement

Meetings zwischen dem Studenten und dem Betreuer finden jeden zweiten Mittwoch in Biel statt.

Das Projekt wird innerhalb eines Git-Projektes geführt und im Laufe des Projekts entweder auf GitHub.com oder auf das interne Projektmanagement-Tool migriert.

Gearbeitet wird durch den Studenten Montags bis Donnerstags an einem Arbeitsplatz im CPVR Labor in Biel.

## 5.3 Zeitplan

## 6 Tests



# Glossar

**HMD** Head-Mounted Display.

**IMVR** Kurz für "Images & Music in VR". Die Applikation, die im Rahmen dieses Projekts mit Unity 5 entwickelt wird.

**Inter-Pupillary Distance** Beschreibt den Augenabstand und stellt ein wichtiges Mass für die stereoskopische Darstellung von Bildern dar.

**IPD** Inter-Pupillary Distance.

**VR** Virtual Reality.