



Bringing Your Hands Into Virtual Reality

Pflichtenheft

Studiengang: Informatik
Autoren: Simon Meer
Betreuer: Prof. Urs Künzler
Experten: Yves Petitpierre
Datum: 13.03.2015

Versionen

Version	Datum	Status	Bemerkungen
0.9	11.03.2015	Entwurf	Entwurf erstellt
1.0	13.03.2015	Definitiv	Finale Version

Inhaltsverzeichnis

1	Einleitung	1
2	Vision “IMVR”	2
2.1	Problemstellung	2
2.2	Technologien	2
2.3	Anwendung	5
3	Spezifikationen	7
3.1	Funktionale Anforderungen	7
3.2	Sonstige Spezifikationen	9
3.3	Use-Cases	9
3.4	Designskizzen	12
3.5	Gesten	14
4	Prototyp	15
4.1	Implementierung	15
4.2	Probleme	15
5	Organisation	16
5.1	Projektbeteiligte	16
5.2	Projektmanagement	16
5.3	Ordnerstruktur	17
5.4	Zeitplan	18
6	Testszenarien	19
6.1	Indexer	19
6.2	Bilder	19
6.3	Musik	20
	Glossar	21

1 Einleitung

Der Markt der stereoskopischen Brillen (siehe "Stereoskopie") befindet sich in einem starken Wachstum. Oculus Rift, SteamVR, Project Morpheus, Nvidia VR - die Liste der teilnehmenden Firmen 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 aufgearbeitet 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 stereoskopischen 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 evtl. Gruppierfunktion.
- Die Bilder betrachtbar und die Musik abspielbar machen.
- Metainformationen darstellen (z.B. in Form von Diagrammen).

Zusätzlich zur Applikation selbst soll noch ein zusätzliches Tool entwickelt werden, welches im Voraus die Dateien auf dem Host-System indexiert und für die visuelle Applikation bereitstellt.

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.

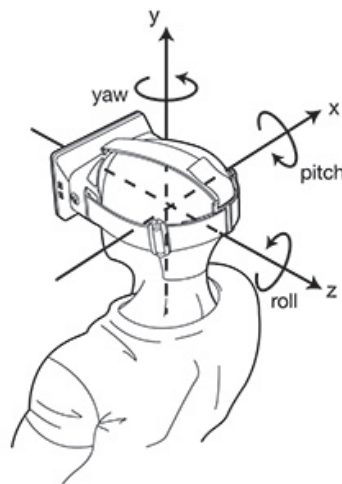


Abbildung 2.1: Illustration, welche die Anwendung und das Koordinatensystem der Oculus Rift verdeutlicht.

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.

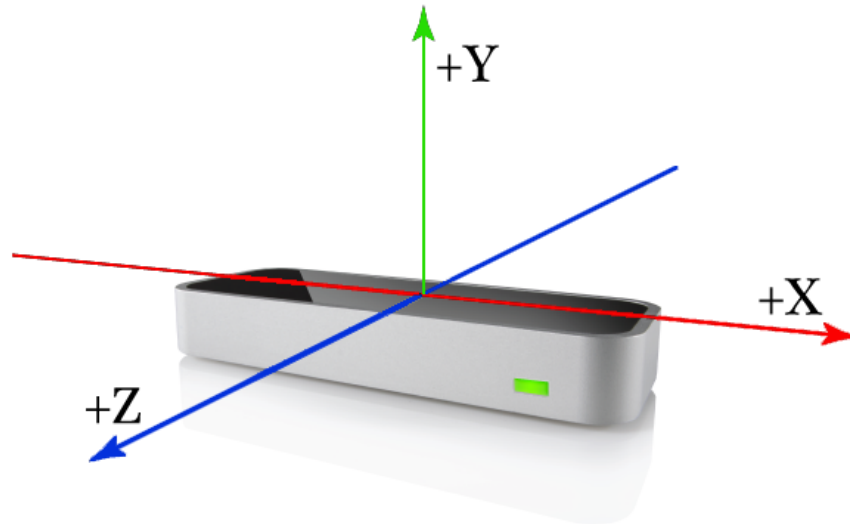


Abbildung 2.2: Verdeutlichung des Koordinatensystems.

Quelle: https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Coordinate_Mapping.html

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.

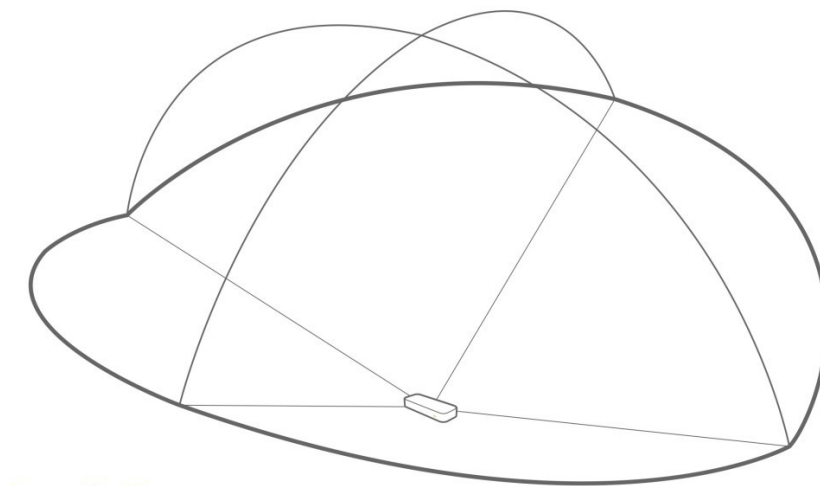


Abbildung 2.3: Das Sichtfeld der Leap Motion.

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 unter dem Namen "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 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 wurde, ist es leicht, schnell zu Ergebnissen zu kommen, und die Entwicklungsumgebung unterstützt die zwei Programmiersprachen, 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 \$15¹ 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.

¹Siehe <https://www.leapmotion.com/product/vr>

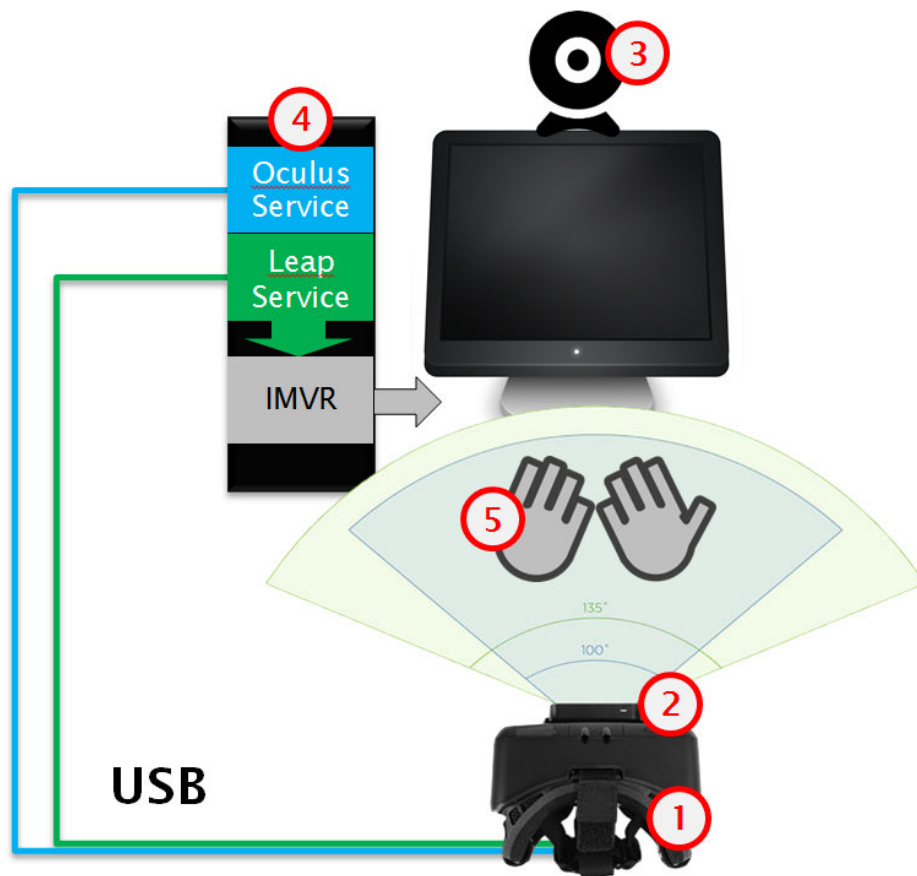


Abbildung 2.4: Eine Übersicht der Technologien und wie sie verbunden sind.

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 Spezifikationen

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. Grundsätzlich kann gesagt werden, dass alle Anforderungen, die mit einer 1 deklariert werden, implementiert werden müssen und alle Anforderungen, die mit einer 5 deklariert werden, nur Ideen sind, welche mit hoher Wahrscheinlichkeit nicht implementiert werden können.

Nr.	Beschreibung	Priorität
1. Elementares		
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 Applikation kann jederzeit beendet werden.	1
1.6	Die Applikation erreicht eine stetige Framerate von mindestens 60fps auf dem Referenzsystem ¹ .	1
1.7	Die Applikation kann min. 100'000 Dateien verarbeiten.	1
1.8	Hände werden abstrakt dargestellt.	2
1.9	Die Hintergrundmusik kann von der Musikbibliothek ausgewählt werden.	2
2. Übersicht (Bilder)		
2.1	Elemente können in einem 3D-Karussell dargestellt werden.	1
2.2	Bilder können nach Farbton, Sättigung und Helligkeit sortiert werden.	1
2.3	Elemente können selektiert werden.	2
2.4	Es kann auf ein bestimmtes Element fokussiert werden (siehe Detailansicht).	1
2.5	Bilder können alphabetisch und chronologisch sortiert werden.	2
2.6	Elemente können getaggt werden.	3
2.7	Es ist eine Mehrfachselektierung möglich.	3

¹Intel(R) Xeon CPU E5-1650 v3 @ 2.50Ghz mit 32GB RAM und einer NVIDIA GeForce GTX 980 Grafikkarte.

Nr.	Beschreibung	Priorität
2.8	Elemente können in anderen Anordnungen dargestellt werden (Tunnel, Fluss, Fläche, Würfel, etc.)	3
2.9	Bilder können nach Dateiordner gruppiert werden.	3
2.10	Bilder können nach Kennwerten (Varianz, Entropie) sortiert werden	3
3. Detailansicht (Bilder)		
3.1	Bild kann vergrößert und verkleinert werden.	1
3.2	Bild kann gelöscht werden	2
3.3	Ein (3D)-Histogramm des Bildes wird angezeigt.	3
3.4	Die Metadaten von Fotos werden dargestellt	4
3.5	Es können zusätzliche Versionen des Bildes (inkl. Histogramm) angezeigt werden und mit Punktoperationen verändert werden.	4
3.6	Der Hintergrund passt sich den Metadaten entsprechend an	5
3.7	Die zusätzlichen Bilder können auch mit lokalen und globalen Operationen gefiltert werden.	5
4. Übersicht (Musik)		
4.1	Unterstützt mit ID3-Tags versehene MP3.	1
4.2	Dateien werden alphabetisch gruppiert nach Artisten dargestellt	1
4.3	Dateien können nach Album, Genre und Jahr gruppiert werden	3
4.4	Alben können anhand ihres Covers wie Bilder dargestellt werden.	3
4.5	Unterstützt weitere Musikformate (Ogg Vorbis, FLAC, M4A, APE, TAK, etc.)	5
4.6	Dateien können ungruppiert dargestellt werden.	5
5. Detailansicht (Musik)		
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
6. Indexierung		
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	Dateien, die nicht mehr in einem der Medienordner liegen, werden aus dem Index gelöscht.	1
6.4	Die indexierten Ordner können innerhalb der Applikation angegeben werden	3
6.5	Die Indexierung kann innerhalb der Applikation durchgeführt werden.	4
7. Sonstige Funktionen		
7.1	Es können Filter über die Musik gelegt werden (Low-Pass, High-Pass, Compressor, etc.)	4
7.2	Bildergruppen können als "Diashow" angeschaut werden.	4
7.3	Es ist eine Netzwerkfunktion vorhanden, die es erlaubt, die Fotosammlung zu zweit anzuschauen.	5

Nr.	Beschreibung	Priorität
7.4	Gesamtstatistiken können angezeigt werden (Bildalter / Anzahl, Auflösung / Anzahl, Auflösung / Anzahl, Kartenchart mit Anzahl Fotos)	3
7.5	Online-Services können als Datenquelle angegeben werden (Flickr, Spotify, etc.)	5

Tabelle 3.1: Die Übersicht aller Anforderung von wichtig (1) bis unwichtig (5)

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.
- Die Speicherung der Daten erfolgt entweder über eine Serialisierung oder eine SQLite-Datenbank.
- Als Entwicklungsumgebung wird Visual Studio 2013 benutzt.
- Die Applikation ist auf Englisch.
- Es wird eine kurze Anleitung erstellt für die Bedienung der Applikation.

3.3 Use-Cases

In dieser Sektion wird eine Auswahl von Use-Cases vorgestellt, die zur Konkretisierung diverser Abläufe dienen soll.

3.3.1 UC1: Starten der Applikation

1. Der Benutzer startet die Applikation.
2. Er setzt sein Oculus Rift DK2 mit Leap Motion auf.
3. IMVR fordert den Benutzer auf, sich in eine angenehme Sitzposition zu bewegen und die Leertaste zu drücken.
4. Der Benutzer drückt die Leertaste.
5. IMVR gibt dem Benutzer eine Ansicht auf seine Bilder auf, sortiert nach Farbtönen.

3.3.2 UC2: Abbruch der Verbindung zur Leap Motion

1. Der Benutzer durchläuft UC1.
2. Die Leap Motion wird nicht (mehr) erkannt.
3. IMVR zeigt dem Benutzer eine Warnung als Overlay an.

3.3.3 UC3: Beendung der Applikation

1. Der Benutzer durchläuft UC1.
2. Er öffnet das Hauptmenü.
3. Er wählt den Eintrag für das Schliessen der Applikation.
4. IMVR zeigt einen Confirm-Dialog an.
5. Die Applikation schliesst, falls der Dialog bestätigt wurde.

3.3.4 UC4: Wechseln zur Musikansicht

1. Der Benutzer durchläuft UC1.
2. Er öffnet das Hauptmenü.
3. Er wählt das Element für Musik.
4. IMVR wechselt zur Musikansicht.

Alternative Idee: Klatschen

3.3.5 UC5: Hinzufügen eines Medienordners

1. Der Benutzer öffnet die Datei "library.conf".
2. Er fügt eine Zeile mit dem Pfad zu seinem Ordner ein.
3. Er speichert die Datei.
4. Der Indexer scannt beim nächsten Durchlauf den neuen Ordner.

3.3.6 UC6: Abspielen von Hintergrundmusik

1. Der Benutzer durchläuft UC4.
2. Er wählt einen Artisten aus.
3. IMVR wechselt zu einer Detailansicht und zeigt alle verfügbaren Alben des Artisten an.
4. Der Benutzer wählt ein Lied mit den Fingern aus.
5. IMVR spielt das Lied im Hintergrund ab und zeigt einen Fortschrittsbalken.

3.3.7 Use-Case Diagramm

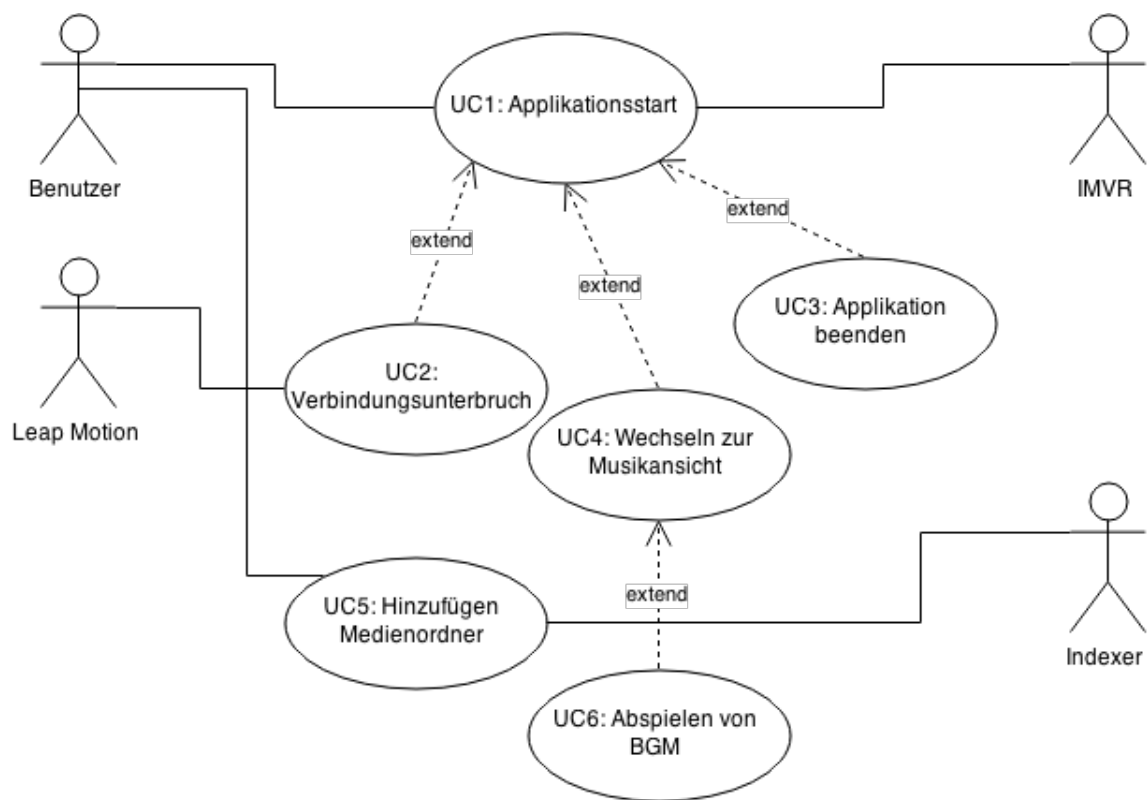


Abbildung 3.1: Eine kurze Übersicht der definierten Use-Cases.

3.4 Designskizzen

Es folgen ein paar frühe Skizzen, die den Aufbau der Applikation illustrieren.

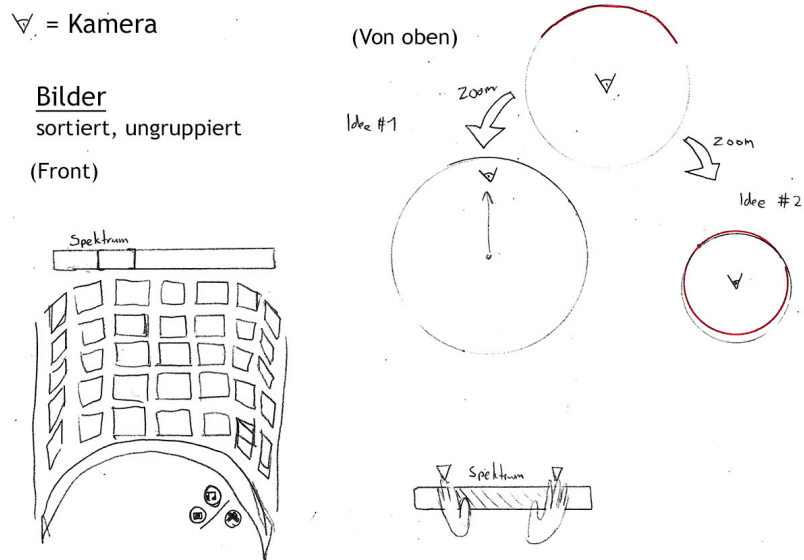


Abbildung 3.2: Bilderübersicht: Eine Leiste gibt Übersicht auf den momentan sichtbaren Bereich.

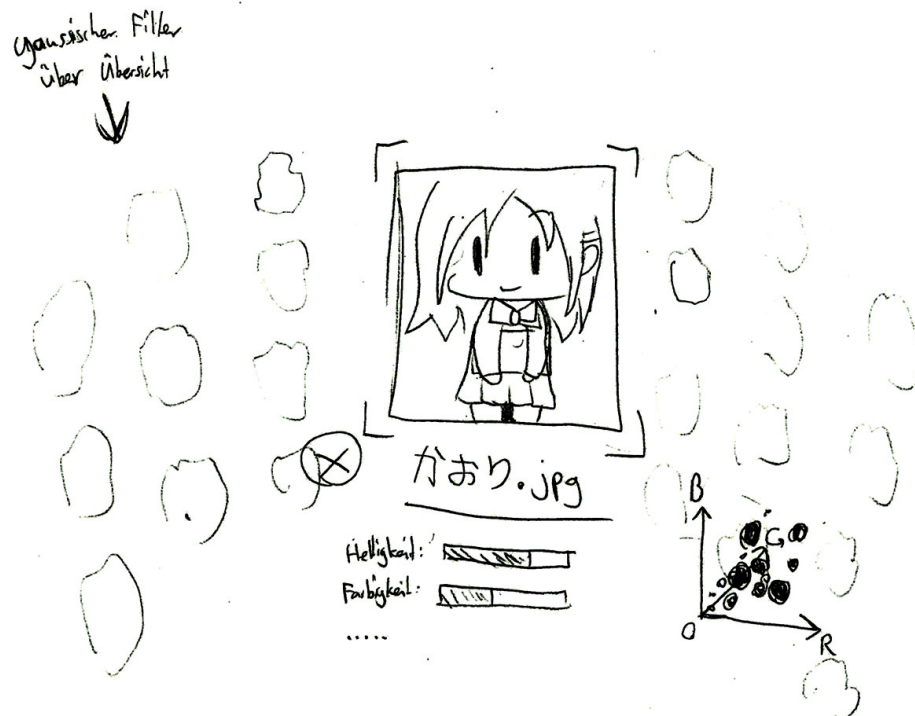


Abbildung 3.3: Bilderansicht: Das Bild kann frei bewegt werden und wird begleitet von Metadaten.

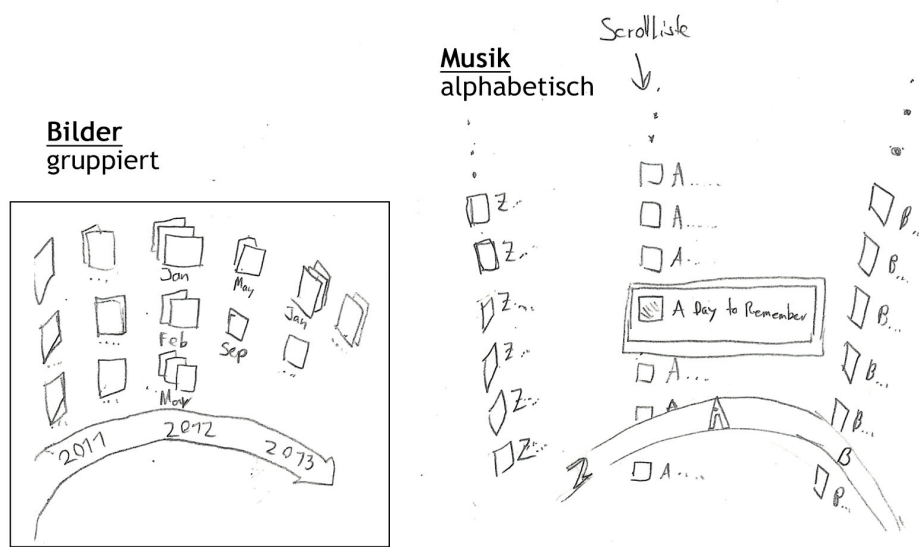


Abbildung 3.4: Gruppierte Bilderübersicht und Musikübersicht mit scrollbaren Elementen zur Auswahl des Artisten.

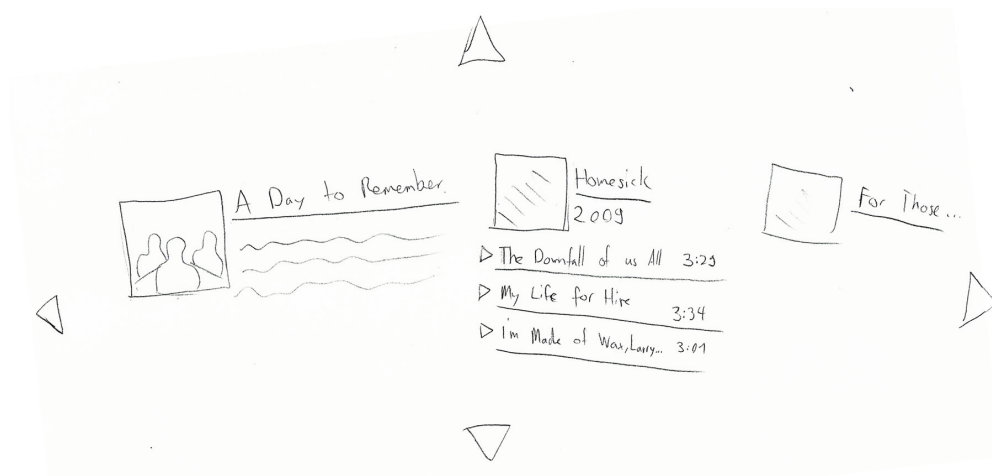


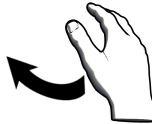
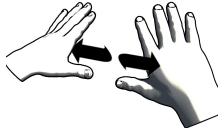
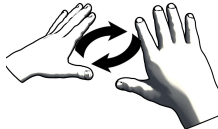




Abbildung 3.5: Detailansicht eines Artisten mit Liste von Alben und Metadaten.

3.5 Gesten

Es folgt eine Tabelle mit ein paar möglichen Gesten, die in der Applikation zum Einsatz kommen werden.

Kontext	Aktion	Geste	Sprachkommando
Überall	Öffnen des Hauptmenüs		Menu
Überall	Anzeigen von kontextsensitiven Statistiken		-
Übersicht	Scrollen		-
Detail	Skalieren des aktiven Bildes		-
Detail	Rotieren des aktiven Bildes		-
Detail	Ansicht verlassen		-
Dialog	Annehmen / Ablehnen		-

4 Prototyp

In der Vorbereitungsphase 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	Student	simon.meer@students.bfh.ch
Prof. Urs Künzler	Betreuer	urs.kuenzler@bfh.ch
Yves Petitpierre	Experte	yves.petitpierre@ericsson.com

5.2 Projektmanagement

- Das Projekt wird innerhalb eines Git-Projektes geführt. Ein privater, zentraler Klon wird auf GitHub abgelegt, welches unter der Adresse <https://github.com/EusthEnoptEron/IMVR> erreichbar ist.
- Meetings zwischen dem Studenten und dem Betreuer finden jeden zweiten Mittwoch in Biel statt.
- Gearbeitet wird durch den Studenten Montags bis Donnerstags an einem Arbeitsplatz im CPVR Labor in Biel.
- Das Projekt wird in Scrum-ähnlich mit Sprints vorangetrieben.

5.3 Ordnerstruktur

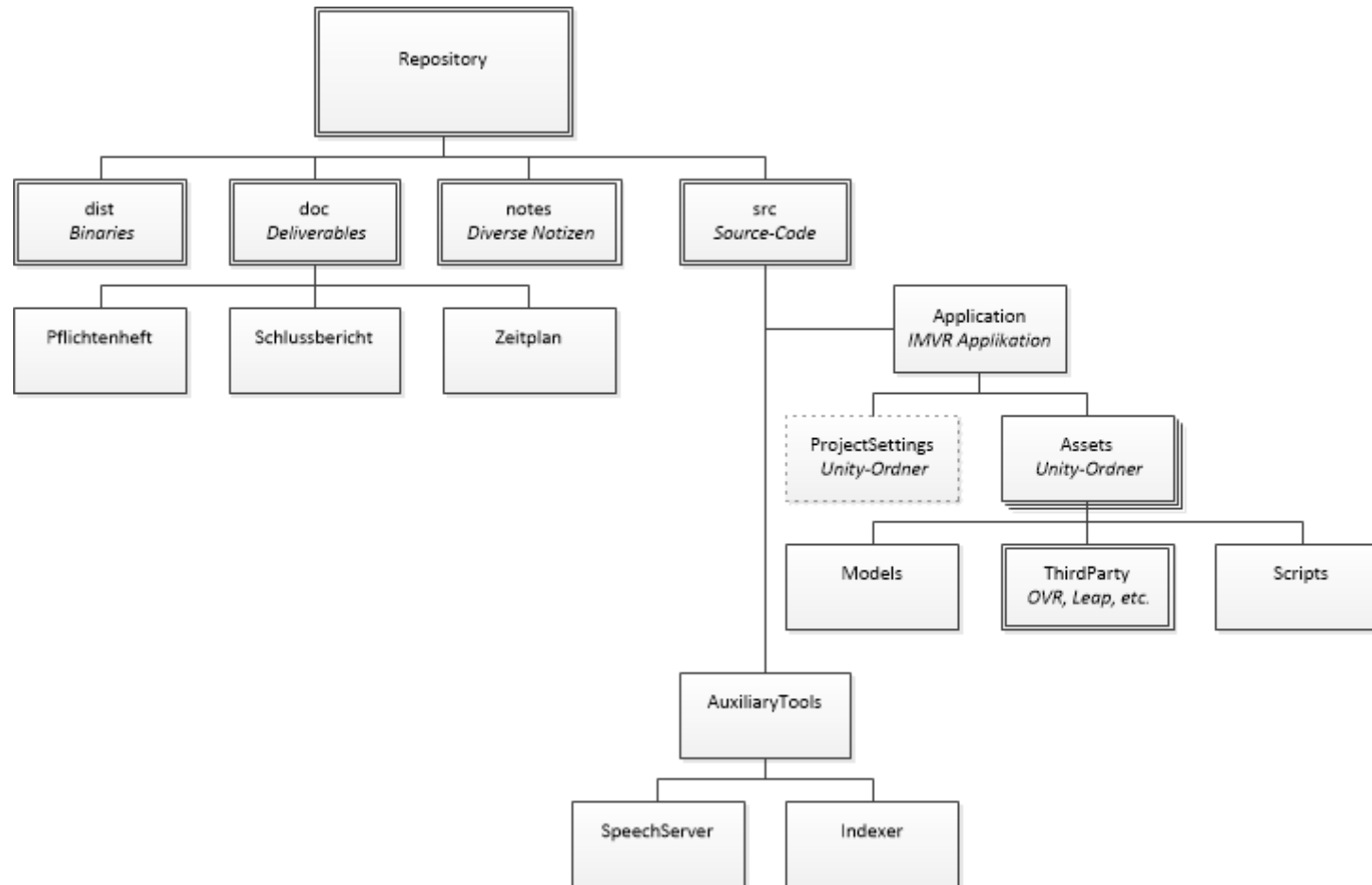


Abbildung 5.1: Die Ordnerstruktur des Projektes.

5.4 Zeitplan

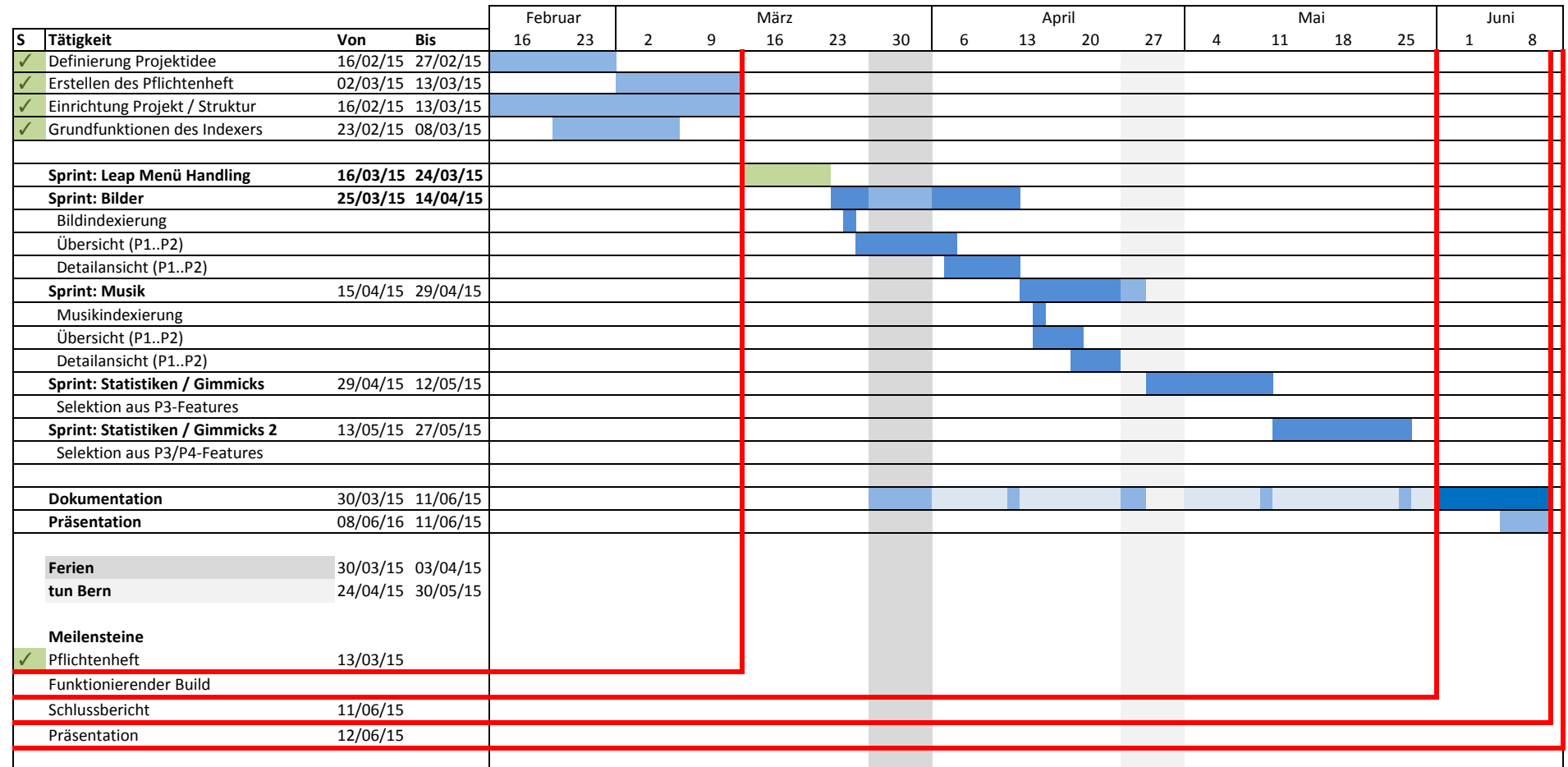


Abbildung 5.2: Der Zeitplan mit Sprints und Prioritäten.

6 Testszzenarien

Dieses Kapitel definiert eine Reihe von Tests, mit denen das Projekt geprüft werden kann und soll.

6.1 Indexer

T1 (Anforderung 6.1) Der Benutzer fügt seine “Eigenen Bilder” und seinen Windows-Ordner zur Library hinzu.

- ▶ Der Indexer durchläuft den Ordner fehlerfrei und die Anzahl indexierter Files, die er ausgibt, stimmt.

T2 (Anforderung 6.2) Der Benutzer öffnet die Applikation.

- ▶ Alle hinzugefügten Bilder werden angezeigt und sind korrekt sortiert.

T3 (Anforderung 6.3) Der Benutzer schliesst die Applikation und löscht den Windows-Ordner von der Library und startet den Indexer.

- ▶ Die Applikation zeigt beim nächsten Start nur noch die Dateien im Ordner “Eigene Bilder”.

6.2 Bilder

- Der Benutzer fügt seine “Eigenen Bilder” zur Library hinzu.

T1 (Anforderung 2.1) Der Benutzer öffnet die Applikation IMVR.

- ▶ Ein 3D-Karussell (oder anders 3D-Konstrukt) mit seinen Bildern wird angezeigt.

T2 (Anforderung 2.2) Der Benutzer öffnet das Menü und sortiert nach Helligkeit.

- ▶ Die Elemente werden neu sortiert und erscheinen geordnet nach Helligkeit.

T3 (Anforderung 2.4) Der Benutzer berührt mit den Händen eines der Bilder.

- ▶ Die Übersicht verblasst und das selektierte Bild wird vergrößert.

T4 (Anforderung 3.1) Der Benutzer bewegt seine Hände auseinander entsprechend der Gestentabelle.

- ▶ Das Bild vergrößert sich entsprechend.

T5) Der Benutzer wischt mit der Hand entsprechend der Gestentabelle.

- ▶ Die Detailansicht wird verlassen und die Übersicht kommt wieder in den Vordergrund.

6.3 Musik

T1 (Anforderung 4.1) Der Benutzer fügt seine "Eigene Musik" zur Library hinzu und startet den Indexer.

- ▶ Alle MP3 Dateien mit korrekten ID3-Tags werden laut Log erkannt.

T2 (Anforderung 4.2) Der Benutzer öffnet die Applikation und wechselt in den Musikmodus.

- ▶ Die soeben hinzugefügte Musik erscheint alphabetisch geordnet.

T4 (Anforderung 1.5) Der Benutzer öffnet das Hauptmenü und beendet die Applikation.

- ▶ Die Applikation fragt einmal nach und beendet dann.

Glossar

Git Git ist ein verteiltes Versionskontrollsystem, welches hauptsächlich dazu dient, den Sourcecode eines Projekts dynamisch zu verwalten. Anders als z.B. SVN benötigt Git keine Verbindung zu einem zentralen Repository, da jede Kopie vollwertig ist..

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.

Leap Motion Ein auf Infrarotkameras basierter Handerkennungssensor.

Oculus Rift Ein HMD von Oculus VR.

Stereoskopie Die Stereoskopie ist die Wiedergabe von Bildern mit einem räumlichen Eindruck von Tiefe, der physikalisch nicht vorhanden ist. Umgangssprachlich wird Stereoskopie fälschlich als "3D" bezeichnet, obwohl es sich nur um zweidimensionale Abbildungen (2D) handelt, die einen räumlichen Eindruck vermitteln. Normale zweidimensionale Bilder ohne Tiefeneindruck werden als monoskopisch bezeichnet. (Quelle: Wikipedia).

Unity Unity ist eine Spielengine, die das einfache Entwickeln von 3D-Applikation für diverse Endgeräte erlaubt.

VR Virtual Reality.