

## **IIT04: Collaborative BIM VR**

**Semesterarbeit von**  
**Brönnimann Nick, Semester 5**  
**Neumarktplatz 7, 5200 Brugg**  
[nick.broennimann@students.fhnw.ch](mailto:nick.broennimann@students.fhnw.ch)

**Renner Sebastian, Semester 5**  
**Neumarkt 2, 5200 Brugg**  
[renner.sebastian@students.fhnw.ch](mailto:renner.sebastian@students.fhnw.ch)

**FHNW**  
**Hochschule für Technik**  
**Informatik IModular**  
**Frau Prof. Dr. Doris Agotai**  
**Frau Kathrin Koebel**

**Windisch, 17. Januar 2020**

## **Abstract**

Dieser Bericht befasst sich mit der Fragestellung der Verwendung von VR zur Visualisierung von 3D Modellen der Bauindustrie. Der Bericht entstand im Rahmen der Projektarbeit IP5 und der strategischen Initiative *Digitaler Wandel Bau*. [1]

Auf Basis von wissenschaftlicher Forschung, um die Themen *Motion Sickness* und *User Interface und Interaktionen im virtuellen Raum* und anhand von technischer Forschung im Bereich des Imports von Bauplänen in VR und des Exports von User Feedback, wurden Konzepte entwickelt. Diese wurden evaluiert und getestet, was in einem finalen Konzept resultierte. Dieses wurde bestmöglich in einem Unity Projekt umgesetzt.

Aus dem Projekt ging ein funktionsfähiger Prototyp hervor, welcher alle gewünschten Funktionalitäten beinhaltet. Auf dieser Grundlage könnte in Zukunft ein vollumfängliches Produkt entstehen.

## **Vorwort**

Als erstes möchten wir uns bei der FHNW bedanken, dass sie dieses Projekt ausgestellt hat. Es hat uns viel Spass gemacht, für diese Problemstellung eine Lösung zu konzipieren und umzusetzen.

Ein spezieller Dank gilt natürlich unseren Betreuern, welche uns wertvolle Tips und Feedback geben konnten und uns stets unterstützt haben. Auch möchten wir uns bei Herr Dr. Wissam Wahbeh bedanken, dass wir von seiner Expertise im Bereich BCF profitieren konnten.

## Inhaltsverzeichnis

1	Einleitung	5
1.1	Problemstellung	5
1.2	IST-Zustand	5
1.3	Technologien	5
2	Projektrahmen	6
2.1	Ziele	6
2.1.1	Baupläne in Unity importieren	6
2.1.2	Feedback zum Bauplan	6
2.1.3	Feedback aus VR exportieren	6
2.1.4	Usability und User Experience	6
2.2	Methoden / Vorgehen	6
3	Forschung	8
3.1	Importieren Baupläne in Unity Tools	8
3.1.1	Unity Reflect	8
3.1.2	Tridify Convert	8
3.1.3	xBIM	8
3.2	VR Technologien	10
3.2.1	VR Hardware	10
3.2.2	SteamVR	10
3.2.3	Bim Collaboration Format (BCF)	10
3.2.4	BIM Collaborative Format (BCF)	10
3.3	User Interface und Interaktionen im virtuellen Raum	11
3.3.1	2D Interaktionen	11
3.3.2	3D Interaktionen	11
3.3.3	Sprachinteraktionen	12
3.3.4	Testschema	12
3.3.5	Auswertung	12
3.4	Motion Sickness	13
3.4.1	Begriff	13
3.4.2	Sensor Konflikt	13
3.4.3	Rest Frame	13

---

3.5	Feedback exportieren	14
3.5.1	BCF	14
3.5.2	Text basiert	14
3.5.3	Audio	14
<hr/>		
4	Konzeptentwicklung	15
4.1	Konzept	15
4.1.1	Bauplan Import	15
4.1.2	Feedback in VR	15
4.1.3	Bewegen in VR	16
4.1.4	Feedback exportieren	17
4.2	Konzept-Tests	17
4.2.1	Usability Tests	17
4.2.2	Interview	17
<hr/>		
5	Umsetzung	19
5.1	Entwicklungsumgebung	19
5.2	Bauplan Import	19
5.2.1	Import	19
5.2.2	Beleuchtung	19
5.3	Bewegung in VR	21
5.4	Feedback in VR	22
5.5	Feedback aus VR exportieren	22
5.6	Architektur	23
5.6.1	Projektordnerstruktur	24
5.6.2	Skripts	25
5.6.3	Plugins / Libraries	25
5.7	Adaptionen aus den Usability Tests	26
<hr/>		
6	Diskussion	28
6.1	Resultate	28
6.2	Reflektion	28
6.3	Wie weiter?	28
<hr/>		
7	Glossar	30
8	Quellenverzeichnis	31
9	Abbildungsverzeichnis	32

10	Tabellenverzeichnis	33
11	Eidesstattliche Erklärung	34

---

# **1 Einleitung**

## **1.1 Problemstellung**

Das Interpretieren und Verstehen von Bauplänen ist nicht einfach. Für Auftraggeber und Endbenutzer ist es schwierig aufgrund des 2D Bauplans das Endresultat zu erkennen. Um den Prozess zu vereinfachen, soll ein BIM kompatibler Bauplan mit Hilfe eines Tools in VR dargestellt werden. Allfälliges Feedback der Endbenutzer und des Auftraggebers sollen möglichst einfach dem Bauzeichner zur Verfügung gestellt werden.

## **1.2 IST-Zustand**

Es gibt Tools zur Darstellung von BIM kompatiblen Bauplänen in 3D und VR. Dazu gibt es Tools zum Erfassen von Kommentaren in BIM kompatiblen Bauplänen. Es gibt jedoch noch keine Tools, dass diese zwei Funktionalitäten kombiniert.

## **1.3 Technologien**

Als Plattform für VR wird Unity verwendet. Um BIM kompatible Baupläne in Unity zu importieren, wird Tridify verwendet. Damit das Feedback dem Bauplan zugewiesen werden kann, wird das BCFZip-Format verwendet. Von xBIM gibt es dazu eine Library, die das Erstellen eines BCFZip anbietet.

## 2 Projektrahmen

### 2.1 Ziele

Die nachfolgenden Ziele und Funktionalitäten wurden identifiziert und im Rahmen des Projekts in Form eines Proof of Concept erarbeitet.

#### 2.1.1 Baupläne in Unity importieren

Personen ohne grosses Vorwissen in Unity (z.B. Architekten oder Bauzeichnern) soll es möglich sein, auf möglichst einfachem Wege, IFC-Dateien in Unity zu importieren.

#### 2.1.2 Feedback zum Bauplan

Dem Benutzer soll auf intuitive Art und Weise die Möglichkeit geboten werden, Anmerkungen zu dem importierten Bauplan anbringen zu können, während er sich im virtuellen Raum befindet.

#### 2.1.3 Feedback aus VR exportieren

Das vom Benutzer angebrachte Feedback soll exportiert werden können, damit die zuständigen Bauzeichner die Änderungen im Bauplan vornehmen können.

#### 2.1.4 Usability und User Experience

Es soll zu Kollisionen mit Wänden, Motion-Sickness und Interaktionen mit UI-Elementen Konzepte erstellt werden, wie die Usability und User Experience durch Einhalten von Standards gesteigert werden kann.

### 2.2 Methoden / Vorgehen

Das Projekt wurde grob in vier Phasen unterteilt. Projektorganisation, Recherche, Konzeptentwicklung und Entwicklung. Konzept- und Usability Tests werden bilateral durchgeführt.

#### **Projektorganisation:**

Ziel dieser Phase ist es, dass sich alle am Projekt beteiligten Personen auf eine Projektvereinbarung einigen können. Auch soll die grobe Struktur des Abschlussberichtes in Form eines provisorischen Inhaltsverzeichnisses erarbeitet werden.

#### **Recherche:**

In der Recherche-Phase soll die Tauglichkeit von Tools evaluiert werden, welche den Import von IFC-Dateien anbieten. Ausserdem sollen Forschungsfragen definiert werden und in der Fachliteratur nach deren Lösung gesucht werden.

#### **Konzeptentwicklung:**

Das Projekt wurde in vier Teile zerlegt. Für jeden Teil soll ein eigenes Konzept erarbeitet werden:

- Baupläne in VR importieren
- Fortbewegung in VR
- Feedback in VR geben
- User-Feedback aus VR exportieren

Schlussendlich sollen alle Teile zu einem gemeinsamem Konzept zusammengefasst werden, welches es dann zu implementieren gilt.

**Entwicklung:**

In der Entwicklungsphase soll das oben erwähnte Konzept umgesetzt und entsprechend getestet werden.

## 3 Forschung

### 3.1 Importieren Baupläne in Unity Tools

#### 3.1.1 Unity Reflect

Wünschenswert wäre ein Tool, welches direkt in Unity integriert und von ihnen unterstützt würde. Dies soll in absehbarer Zukunft mit Unity Reflect geschehen. Dieses, von Unity Technologies in Zusammenarbeit mit Autodesk entwickelte Package, soll es ermöglichen BIM Modelle direkt aus Autodesk Revit in Unity zu importieren. Da zur Zeit der Durchführung dieses Projektes Unity Reflect noch nicht veröffentlicht ist und kein klares Releasedatum kommuniziert ist, kommt es leider für dieses Projekt nicht in Frage.

#### 3.1.2 Tridify Convert

Eine bereits bestehende Alternative zu Unity Reflect ist Tridify Convert. Das in 2012 in Finnland gegründete Unternehmen hat es sich zur Aufgabe gemacht, einen möglichst einfachen Weg zu bieten, um auf Basis von BIM Plänen interaktive 3D-Modelle oder gebündelte VR Assets zu generieren.

Der Generierungsprozess gestaltet sich äusserst simpel. Die IFC-Datei(en) werden im Tridify Portal hochgeladen und dort konvertiert. Anschliessend kann das Modell als interaktives 3D-Modell betrachtet werden oder innerhalb einer VR Entwicklungsumgebung (wie Unity) als Asset importiert werden. Eine detaillierte Schritt-für-Schritt Anleitung zu Tridify Convert befindet sich im Anhang dieses Dokumentes.

Wir haben somit einen zuverlässigen Weg gefunden, um IFC-Dateien schnell und ohne komplexen Konfigurationsaufwand in VR darzustellen.

Tridify bietet zusätzlich noch ein Plug-in für Unity. Dieses ermöglicht es u.a. die importierten VR-Räume generisch zu beleuchten, was diesen Prozess für einen ungeschulten Benutzer deutlich vereinfacht.

Einziger Nachteil, dessen wir uns bis anhin bewusst sind, ist das wir nicht steuern können, ob wirklich alle Attribute aus dem Bauplan übernommen werden. Ausserdem ist man auf die Verfügbarkeit und die korrekte Funktionsweise der Tridify Plattform angewiesen.

#### 3.1.3 xBIM

Um die Oberhand über die komplette Import Pipeline zu haben, müssten wir eine Library wie xBIM verwenden. xBIM ist eine Open Source Library für .NET, welche es ermöglicht IFC Dateien direkt via Code Snippet in Unity zu importieren. Es wäre jedoch mit enormem Zeitaufwand verbunden sich in die Library einzuarbeiten und die ersten Dateien in Unity zu überführen. Aufgaben, welche fertige Tools wie Tridify und Unity

Reflect out of the box anbieten. Auf der anderen Seite ist man auf keine Software von Drittparteien angewiesen.

Wir kommen zum Schluss, dass es durchaus Vorteile mit sich bringen würde, wenn der Import mit xBIM durchgeführt wird, jedoch würde es den Rahmen dieses Projektes sprengen, wenn wir Import Pipeline komplett eigenhändig programmieren müssten. Trotzdem könnten einige Funktionen der Library später noch von Nutzen sein.

## 3.2 VR Technologien

### 3.2.1 VR Hardware

Die Entscheidung mit welcher VR Hardware das Projekt durchgeführt werden soll, wurde durch die Auswahl der im Media Lab der FHNW zur Verfügung stehenden Headsets beeinflusst. Zur Auswahl standen die HTC Vive Pro, die Samsung Odyssey und die Oculus Rift. Grundsätzlich könnte das Projekt mit jeglichem dieser Modelle durchgeführt werden.

Der Entscheid fiel schlussendlich auf die HTC Vive Pro, da sie gegenüber der Oculus Rift über eine höhere Auflösung und Bildwiederholfrequenz verfügt [2][3] und die Samsung Odyssey primär für Mixed Reality ausgelegt ist.

### 3.2.2 SteamVR

#### 3.2.3 Bim Collaboration Format (BCF)

SteamVR ist die Implementation und Erweiterung von OpenVR von Valve. Dank der Unterstützung von OpenVR ist SteamVR nicht Hardware gebunden. Es werden verschiedenste VR-Brillen unterstützt, darunter: Valve Index, HTC Vive (Pro), Oculus Rift (S) und Microsoft Mixed-Reality Brillen [9].

Alternativ zu SteamVR gibt es VRTK (Virtual Reality Toolkit), welche Lösungen zu verschiedenen wiederkehrenden Problemen mit VR bietet.

Spezifizierte Frameworks wie beispielsweise jenes, welches von Oculus [10] angeboten wird, schränken die Hardware Auswahl zu stark ein und kommen nicht in Frage.

#### 3.2.4 BIM Collaborative Format (BCF)

Das BIM Collaboration Format wurde zusammen mit dem BIM-Format ins Leben gerufen. Es erlaubt Änderungen in Form eines Issues/Ticket mit anderen Personen, welche an demselben Projekt beteiligt sind, zu teilen. Das BCF wird in einem komprimierten Container namens BCFZIP übermittelt. Dieser enthält auch noch weitere Dateien, welche helfen sollen, den Fehler möglich präzise zu beschreiben. Alle Dateien basieren auf XML. Das BCFZIP gliedert sich wie folgt:

Auf der obersten Ebene des Ordners befinden sich zwei Dateien (bcf.version und project.bcfp) und ein Ordner, welcher nach der GUID des referenzierten Objektes benannt ist.

##### **bcf.version:**

Diese Datei muss in einem BCFZIP vorhanden sein. Sie definiert, welche Version des BCF Standards verwendet wird.

##### **project.bcfp:**

Diese Datei ist optional. Sie beinhaltet lediglich Informationen zur Identifikation des Projektes.

Im Ordner betitelt mit der GUID des Objektes sind alle restlichen Dateien abgelegt:

**markup.bcf:**

Diese Datei ist das Herzstück des BCF. Es beinhaltet weitere Informationen zum referenzierten Objekt und der gewünschten Änderung. Auch werden hier gewisse Meta-Informationen, wie der aktuelle Status des Issues, dessen Autor und das Erstellungsdatum gespeichert. Es definiert auch den Pfad des zugehörigen Screenshots und allfälligen Viewpoint.

**snapshot.png:**

Ein vollständiges BCF beinhaltet einen Screenshot. Dieser muss per Definition snapshot.png genannt werden. Es soll eine Momentaufnahme der Mängel im Modell sein.

**viewpoint.bcfv:**

Der Viewpoint hingegen ist optional. Er gibt u.A. die Kameraposition und -winkel an, in dem der Screenshot erstellt wurde, um dieselbe Ausrichtung später in einem CAD Programm wie Revit wieder zu rekonstruieren.

Alle Informationen wurden der Dokumentation des BCF des Herausgebers buildingSMART entnommen [4].

### **3.3 User Interface und Interaktionen im virtuellen Raum**

Es gibt viele verschiedene Arten wie Feedback gegeben werden kann. Im Paper *What User Interface To Use for Virtual Reality? 2D, 3D Or Speech - A User Study* wurden drei mögliche Interaktionsansätze mit 30 Personen im Alter zwischen 18 und 49 Jahren untersucht. [11]

Die analysierten Techniken werden in den folgenden drei Kapitel genauer beschrieben.

#### **3.3.1 2D Interaktionen**

Als 2D Interaktionen werden die herkömmlichen Tools bezeichnet. Fenster, Icons, Menus und Zeiger. Da sich der Benutzer im virtuellen Raum frei bewegen können soll, eignet sich eine Maus dazu nicht. In VR wird stattdessen einer oder beide Controller für die Interaktion mit den UI-Elementen verwendet. Um weiter entfernte UI-Elemente auszuwählen, bietet sich eine Implementation wie der Laserpointer an, mit dem das UI-Element ausgewählt wird.

#### **3.3.2 3D Interaktionen**

Mit 3D Interaktionen ist gemeint, dass der Benutzer Anpassungen am virtuellen Objekt gleich vornehmen kann. Beispielsweise das Ziehen an einer Ecke vergrössert das Objekt oder auch das Anwenden eines Pinsel mit Farbe auf das Objekt verändert dessen Farbe.

### 3.3.3 Sprachinteraktionen

Als Sprachinteraktion ist ein Home-Assistent gemeint. Der Benutzer aktiviert durch Drücken eines Buttons auf dem Controller das Mikrofon der VR-Brille. Ein Signal (akustisch oder visuell) gibt an, dass das System zuhört. Der Benutzer kann die Aktion, die ausgeführt werden soll, aussprechen. Das System interpretiert die Sprache, identifiziert Subjekt (das virtuelle Objekt) im virtuellen Raum und verändert es entsprechend der interpretierten Aktion. Dem Benutzer kann den verstandenen Auftrag in Textform im virtuellen Raum präsentiert werden.

### 3.3.4 Testschema

Den 30 Testteilnehmern wurden für alle drei Interaktionen je 17 Aufgaben präsentiert. Die Reihenfolge war für jeden Probanden zufällig bestimmt. Im Anschluss füllten die Probanden ein Fragebogen zu Verwendbarkeit, Lernbarkeit und weiteren Kriterien aus. Die Probanden konnten einen Wert auf der Skala von 0 *Trifft nicht zu* bis 10 *Trifft zu* auswählen.

### 3.3.5 Auswertung

Die Daten der Fragebogen wurden zusammengetragen und verglichen. Dabei hat sich die Sprach Interaktion als äusserst beliebt etabliert (siehe Tabelle 1).

**Tabelle 1 Auswertung der Fragebogendaten (vgl. Tabelle 15 in Paper)**

Aufgabe	2D Interaktionen	3D Interaktionen	Spracheinteraktionen
Positionierung und Rotation	+	o	-
Manipulation	+	o	-
Text	-	-	+
Erlernbarkeit	-	o	+
Spas	-	+	o

Unkomplizierte Handhabung	o	o	+
Immersion	o	+	+
Natürlich und intuitiv	0	+	+
Benötigte Zeit pro Task	0	+	o

Legende: - negativ, + positiv, o neutral

### 3.4 Motion Sickness

Im Zusammenhang mit Virtual Reality ist Motion Sickness nach wie vor ein wichtiges Thema. Es gibt unzählige Papers und Dokumente, die versuchen die Ursache und Lösungen in verschiedensten Anwendungsbereichen zu identifizieren und erforschen.

#### 3.4.1 Begriff

Motion Sickness im Zusammenhang mit Virtual Reality hat viele Namen. So wurde für visuell ausgelöste Motion Sickness der Begriff *Cybersickness* definiert. Für Motion Sickness die durch die Hardware, Latenz o.ä. Einflüsse ausgelöst wird wurde der Begriff *Simulator Sickness* definiert. Beide Begriffe werden unter der *Virtual Reality Sickness* (VRS) zusammengefasst. [12] Die Symptome von Motion Sickness treten bei VRS unterschiedlich stark auf.

Es gibt verschiedene Hypothesen die den Grund für VRS zu erklären versuchen. Die zwei bekanntesten sind die Sensor Konflikt Hypothese und die Rest Frame Hypothese.

#### 3.4.2 Sensor Konflikt

Die Sensor Konflikt Hypothese begründet die Ursache von VRS mit dem Konflikt zwischen dem Gleichgewichtsorgan, den Augen und das Nervensystem. Beispielsweise; was in der VR-Brille gezeigt wird, erlebt der Körper nicht in echt oder auch (Mikro-)Bewegungen zur Aufrechterhaltung oder Wiederfindung der Balance werden im virtuellen Raum nicht wiedergegeben. Der Sensor Konflikt tritt häufig bei Bewegungen auf, wie beispielsweise bei einer simulierten Achterbahnfahrt.

#### 3.4.3 Rest Frame

Die Rest Frame Hypothese begründet VRS mit dem Fehlen von, in Konflikt stehenden oder sich bewegenden Referenzpunkte. Ein Rest Frame, oder auch ein Referenzpunkt, ist beispielsweise der Horizont, der Rahmen des Bildschirms oder auch eine Wand. Wichtig ist, dass sich der Referenzpunkt nicht mit den Bewegungen des Benutzers bewegt

(statisch ist). So dass der Benutzer immer ein oder mehrere Referenzpunkte hat, an denen er sich korrekt orientieren kann.

## **3.5 Feedback exportieren**

### **3.5.1 BCF**

Im Optimalfall sollte sich das Feedback als BCF-Datei exportieren lassen. BCF ist ein auf XML-basierender Dateiformat-Standard, welcher es ermöglicht, präzise Änderungsvorschläge an BIM Plänen zu kommunizieren. Dieses Format wird in der Baubranche standardmässig verwendet, um die Kollaboration zwischen den einzelnen Parteien (Architekt, Statiker, etc.) zu vereinfachen. Die Struktur der einzelnen XML-Dateien und der Aufbau der Ordnerstruktur des BCFZIP sind klar vom Herausgeber des Standards, BuildingSMART, definiert (Siehe Kapitel 3.2.3. BCF).

Da die einzelnen Dateien alle auf XML basieren, besteht die Möglichkeit, das BCF eigenhändig mit einem XML-Writer zu generieren bzw. zu befüllen. Da die BCF(ZIP)-Dateien jedoch immer komprimiert übermittelt werden, stellt sich noch die Frage welcher Kompressionsalgorithmus verwendet wird. Dies konnte bis anhin noch nicht ermittelt werden.

Aus der Recherche der Import Möglichkeiten kennen wir bereits die xBIM Library. Eine Sub-Library davon beschäftigt sich mit dem Import/Export von BCF-Dateien. Erste Recherchen und Tests haben ergeben, dass durchaus die Chance besteht, damit einen generischen Export von BCF Dateien zu erreichen. Trotzdem würde bei dieser Form von Export ein deutlicher Mehraufwand entstehen mit unklarem Endergebnis.

### **3.5.2 Text basiert**

Deutlicher einfacher hingegen wäre das Erstellen einer textbasierten Datei. Wir könnten selber definieren, welche Informationen wir extrahieren möchten. Fraglich ist jedoch, wie aussagekräftig ein Feedback in dieser Form für eine Drittperson wäre.

### **3.5.3 Audio**

Falls man sich beim Entgegennehmen des Feedbacks für eine Audioaufzeichnung entscheidet, liegt es auf der Hand, dass man diese auch für den Export verwendet. Die Verwendung einer Audioaufzeichnung bringt jedoch entscheidende Nachteile mit sich. Einerseits hat man keine Informationen, was die Person in diesem Moment gerade angeschaut hat und wie genau sie die Änderung haben möchte. Sie müsste dies sehr detailliert beschreiben. Ein weiterer Nachteil wäre, dass die Person, welche die Änderung dann schlussendlich vornehmen muss, sich die ganze Aufzeichnung anhören muss, obwohl nur ein Bruchteil davon relevant für die auszuführende Arbeit sind.

## 4 Konzeptentwicklung

In diesem Kapitel wird ein Fazit aus den einzelnen Forschungsthemen gezogen, um daraus ein einheitliches Konzept zu erstellen.

### 4.1 Konzept

#### 4.1.1 Bauplan Import

Die Recherche hat einen klaren Favoriten ergeben. Da Unity Reflect noch nicht veröffentlicht wurde und die Verwendung von xBIM mit einem zu grossem Zeitaufwand verbunden ist, bleibt nur noch Tridify Convert übrig. Wir sind zuversichtlich, dass Tridify Convert alle Anforderungen gerecht wird und die generierten Modelle eine gute Basis für dieses Projekt sind.

#### 4.1.2 Feedback in VR

Basierend auf der Forschung (siehe Kapitel 3.3 User Interface und Interaktionen im virtuellen Raum) bieten sich die drei Optionen Audioaufzeichnung, Tastatur und Kontextmenü an.

Aus dem Resultat der Forschung geht hervor, dass das Sprach-Interface bei den Probanden sehr beliebt ist. Dies ist auch in Tabelle 2 klar ersichtlich. Ein 3D Interface ist in diesem Anwendungsfall unangebracht, da davon auszugehen ist, dass den Benutzern das notwendige Wissen fehlt um den Bauplan direkt in VR bearbeiten.

Mit einer Audioaufzeichnung kann in relativ kurzer Zeit eine grosse Menge an Informationen aufgenommen werden. Auf der anderen Seite ist das Führen eines Audiologs für die Benutzer unter Umständen ungewohnt oder unangenehm und das Implementieren eines Sprach-zu-Text Interface wie im Paper [10] zu umfangreich. Situationsbedingt könnte die Option auch aufgrund von Störgeräuschen unangemessen sein.

Mit einer Tastatur kann eine geübte Person ebenfalls in relativ kurzer Zeit viel Informationen erfassen. Die Eingabe auf einer Tastatur in VR ist jedoch nicht angenehm und auch nicht einfach. Jeder Buchstabe wird einzeln mit dem Controller ausgewählt was den Prozess stark verlangsamt.

Wie bei einer Desktop-Applikation kann ein Kontextmenü geöffnet werden, welches vordefinierte Aktionen bereithaltet, die der Benutzer anklicken kann. Diese Aktionen können in einer einfach verständlichen Markup-Sprache definiert werden. Das Skript holt sich die Informationen aus dieser Datei und erstellt entsprechend das Kontextmenü. So kann mit einem simplen Klick ein vorgefertigtes Feedback abgegeben werden.

Tabelle 2 Vergleich Usability Faktoren aus dem Fragebogen (vgl. Tabelle 2 aus dem Paper)

Bewertungskriterium	2D Interaktionen	3D Interaktionen	Sprachinteraktionen
Unkomplizierte Handhabung	7.5 (1.9)	7.6 (1.8)	8.7 (1.6)
Konsistente Handhabung	8.5 (1.7)	8.9 (1.4)	9.0 (1.7)
Erwartete Funktionsweise	8.0 (1.9)	7.8 (1.7)	8.2 (1.5)
Eingabe ist nicht anstrengend	8.8 (1.4)	7.8 (2.7)	8.6 (1.9)
Zufriedenstellendes Feedback	7.0 (2.7)	8.2 (1.9)	8.3 (2.1)

Die Werte in Tabelle 2 für 2D und Sprachinterface sind in den meisten Kriterien nahe genug beieinander, was vermuten lässt, dass 2D keine schlechte Entscheidung ist. Zusätzlich kamen auch die Autoren zum Schluss, dass es nicht eine statistisch bessere Lösung für Interaktionen und User Interface in VR gibt. Die verschiedenen Technologien eignen sich für verschiedene Anwendungsbereiche. Die Autoren weisen auch darauf hin, dass die erhaltenen Antworten, und Interpretationen deren, stark von den Testaufgaben beeinträchtigt sind.

#### 4.1.3 Bewegen in VR

Die Forschung zu Motion Sickness (siehe 3.4 Motion Sickness) hat ergeben, dass besonders bei der Bewegung Motion Sickness ausgelöst werden kann.

Wie auch im Paper [12] erwähnt stellt Teleportieren eine Bewegungsart dar, die wenig Motion Sickness anfällig ist. Teleportieren in VR funktioniert wie man sich das im echten Leben vorstellt. Durch Drücken eines Buttons auf dem Controller wird der Spieler an die ausgewählte Position gesetzt.

Als Alternative dazu gibt es die Bewegungsart Locomotion. Der Benutzer bewegt sich durch virtuellen Raum indem er auf dem VR-Controller ein Joystick oder auch eine Richtungstaste in eine Richtung drückt. Der Spieler bewegt sich dann um eine fixierte Distanz oder Geschwindigkeit in diese Richtung. Die in der Forschung beschriebenen Sensor Konflikt Hypothese spricht grundsätzlich gegen diese Art, da der Benutzer eine Bewegung sieht, die er nicht erlebt. Auch die Rest Frames und bewegendes Referenzpunkte könnten den Effekt von Motion Sickness noch verstärken.

Eine exotischere Art und Weise die Bewegung im virtuellen Raum zu ermöglichen ist die Bewegung mit Grab. Die Idee ist, das man sich selber respektiven den virtuellen Raum

durch ziehen bewegt. Es gilt dabei zu beachten, dass der Spieler sich nicht in den Boden, eine Wand, o.ä. Gegenstände bewegen können darf. Auch bei dieser Art der Bewegung könnten die bewegenden Referenzpunkte Motion Sickness verursachen.

Um Motion Sickness, ausgelöst durch die Bewegung, zu minimieren ist Teleportieren, wie auch im Paper erwähnt, die beste Variante.

#### **4.1.4 Feedback exportieren**

Es wäre wünschenswert, dass der Export als BCF-Datei stattfindet. Es würde die Nutzbarkeit des Endproduktes deutlich steigern. Jedoch ist unklar, ob sich ein Export in dieser Form im Rahmen dieses Projektes realisieren lässt.

Das Projektteam hat sich entschieden, sich der Herausforderung des Exports als BCF anzunehmen und im Falle des Scheiterns auf eine textbasierte Lösung zurückzufallen.

## **4.2 Konzept-Tests**

### **4.2.1 Usability Tests**

Um das Konzept ein erstes Mal zu Testen wurde der Proof-of-Concept Stand des Projektes mit zwei unserer Kommilitonen aus dem Informatik Studium getestet. Das Feedback bezog sich hauptsächlich auf Usability Aspekte der Applikation:

- Ein Feedback, welches Element vom ContextMenu erkannt wurde bzw. auf welches Element man zurzeit zielt wäre hilfreich.
- Das ContextMenu könnte noch ein wenig kleiner skaliert werden.
- Ein optisches Feedback, wenn ein Controller über einem Button des ContextMenus hovert würde die Usability verbessern.
- Ein visuelles oder auditives Feedback, wenn ein BCF erstellt wurde, könnte von Vorteil sein.

Dieses Feedback hat uns gezeigt, dass wir im Grossen und Ganzen auf dem richtigen Weg sind, jedoch dass aus Sicht der Usability noch einiges verbessert werden könnte. Die Verbesserungsvorschläge wurden aufgenommen und es wurde versucht, diese bestmöglich umzusetzen.

### **4.2.2 Interview**

Wir haben das entwickelte Konzept einer Person aus der Baubranche unterbreitet und sie nach der Tauglichkeit des Projektes befragt. Das Interview wurde auf die essentiellen Kernaussagen gekürzt.

#### **Karin Schoch, diplomierte Architektin ETH, interviewt am 25.12.2019:**

*Würden Sie unsere Software in Ihrem Arbeitsalltag verwenden?*

Ich würde es verwenden, um mir ein besseres Bild von gewissen überlappenden Elementen, zum Beispiel einem Dachstuhl, zu machen. Aber ich glaube nicht, dass ich darin arbeiten würde. Es würde mich mehr Zeit kosten, als wenn ich mit einem herkömmlichen Tool arbeite.

*Könnten Sie sich vorstellen unsere Software zu verwenden, um vom Auftraggeber oder dem Endbenutzer ein besseres Feedback zum entworfenen Gebäude zu erhalten?*

Wenn das Generieren von VR-Modellen mit eurer Software so einfach ist, wie du das beschreibst, ja. Ein VR-Modell zu erzeugen ist für uns derzeit immer noch ein grösseres Unterfangen.

Wenn der Auftraggeber eine Privatperson oder allgemein Personen ohne bautechnischen Hintergrund sind, ist es natürlich attraktiv, wenn man ihnen das künftige Gebäude in VR präsentieren kann, damit auch sie sich ein besseres Bild machen können.

*Wo sehen Sie Probleme mit unserer Software?*

Die meisten Architekturbüros besitzen keine VR-Ausrüstung, weil es sich mit den zurzeit verfügbaren Tools einfach nicht lohnt es einzusetzen. Ein grosses Büro wie Herzog und de Meuron kann sich das leisten, aber für uns hat es schlichtweg keinen Mehrwert.

*Verwendet Ihr das BCF-Format?*

Ja. Es ist ein essentieller Kanal, um Änderungen zwischen allen involvierten Parteien (Architekten, Polier, Statiker, etc.) auszutauschen. Innerhalb des Büros geht man eher kurz beim Kollegen vorbei.

*Denken Sie, dass sich VR in Zukunft in der Baubranche durchsetzen wird?*

Mein erster Chef hat bis zum bitteren Ende mit Papier und Bleistift gearbeitet. Er hat den Wechsel zum Computer bis zum Schluss nicht mitgemacht. Genauso werde ich wohl kaum noch den Wechsel zu VR - falls er den kommt - mitmachen. Aber ich kann mir gut vorstellen, dass es sich durchsetzen wird, wenn angehende Architekten darauf ausgebildet werden.

## **5 Umsetzung**

### **5.1 Entwicklungsumgebung**

Das Projekt wird mit Unity (in der Version 2019.2.13f1) durchgeführt. Unity wurde vor UnrealEngine bevorzugt, da das Projektteam bereits in einem vorhergehenden Projekt Erfahrung mit Unity sammeln konnten und weil Sie in der Programmiersprache C# mehr Vorkenntnisse besitzen als in C++.

Als Ausgangslage wurde das Standard Unity3D Projekt Template verwendet. Auch wurde in Betracht gezogen, die neue High Definition Render Pipeline (HDRP) von Unity zu verwenden. Erste Tests haben jedoch gezeigt, dass gewisse benötigte Assets sich nicht erwartungsgemäss verhalten. Es scheint als wäre die HDRP noch nicht ganz bereit für Projekte mit komplexer Programmlogik. Daher wurde in diesem Projekt darauf verzichtet, diese Technologie einzusetzen.

### **5.2 Bauplan Import**

#### **5.2.1 Import**

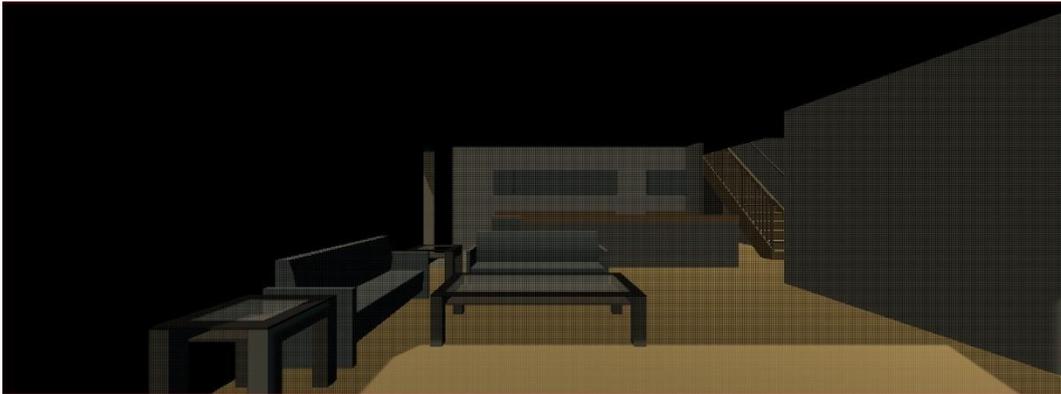
Wie bereits erwähnt, hat das Projektteam sich entschlossen, Tridify Convert als Import Schnittstelle zu verwenden. Um die auf Tridify hochgeladenen Baupläne in Unity zu importieren, muss das von Tridify bereitgestellte Unity Plug-in installiert werden. Die Version des Plug-ins, welche für dieses Projekt verwendet wurde, liegt dem Source Code bei.

Anschliessend kann über das Plug-In auf die hochgeladenen und konvertierten Baupläne zugegriffen und als Unity Asset importiert werden. Eine detaillierte Schritt-für-Schritt Anleitung zu Tridify und dem Import von Modellen liegt diesem Dokument als Anhang bei.

Das importierte Asset wird automatisch am Nullpunkt der ausgewählten Szene platziert.

#### **5.2.2 Beleuchtung**

Unity legt Wert darauf, dass ihre Szenen möglichst realistisch daher kommen. Das stellt insofern ein Problem dar, indem dass das künstliche Sonnenlicht logischerweise von den Wänden im und um das Haus geblockt wird. Auch wenn ein Gebäude grosszügig mit Fenster ausgestattet ist, reicht dies bei Weitem nicht aus, um alle Räume mit natürlichem Licht zu erhellen. Die Räume müssen also künstlich beleuchtet werden.



**Abbildung 1 Raum ohne künstliche Beleuchtung**

Wir erreichen dies, indem wir ein Grid an Light Probes [6] in jedem Stockwerk platzieren. Anschliessend werden an der Decke jedes Raumes Emissive Lights [7] positioniert. Das Floor Lighting Feature des Tridify Plug-ins unterstützt uns dabei, indem es automatisch die einzelnen Räume und Stockwerke identifiziert und die Light Probes und Emissive Lights entsprechend platziert.

Abschliessend müssen die ganzen Lichteinstrahlungen berechnet werden. Wir kalkulieren diese im Voraus und verhindern somit, dass es beim Starten der Applikation und während der Laufzeit nicht zu Performance-Einbrüchen kommt. Dieser Vorgang nennt sich pre-baking.

In den Light Probes werden die einzelnen Lichteinstrahlungen der jeweiligen Emissive Lights gemessen. Anhand von diesen Informationen wird berechnet, wie das Licht von Blocking Objects wie z.B. Wände reflektiert wird. Aus diesen Berechnungen resultieren Lightmaps, welche auf den Blocking Objects platziert werden. In der Computergrafik ist dieser Prozess bekannt als Global Illumination.

Nach Abschluss der Berechnungen können die Light Probes und Emissive Lights auf inaktiv gesetzt werden, damit sie in der Szene nicht mehr sichtbar sind.

Für die finale Beleuchtung des in unserem Projekt verwendeten Gebäudes wurde die Szene mehrmals gerendert, um die generelle Qualität des Lichtes zu erhöhen und um möglichst feingranulare Schatten zu produzieren.



Abbildung 2 Raum mit künstlich generierter Beleuchtung

Alternativ könnte man manuell jede einzelne Lichtquelle in der Szene platzieren. Dies würde jedoch (je nach Grösse des Gebäudes) mehr Aufwand bedeuten und es würde zu unregelmässigen Beleuchtung und Schattierungen führen.

### 5.3 Bewegung in VR

Wie im Konzept (siehe 4.1.3 Bewegen in VR) begründet, soll Teleportation implementiert werden. Es gibt verschiedene Libraries, die die Funktionalität des Teleportieren zur Verfügung stellen. Aus der Community wird vielfach SteamVR empfohlen. Aus diesem Grund und weil das Projektteam in einem vorangehenden Projekt bereits Erfahrungen mit SteamVR gemacht hat, wird SteamVR verwendet.

Das Teleportieren in SteamVR kann mit wenigen Scripts in einer eigenen Scene implementiert werden.

In der aktuellen Version von SteamVR können für einen Controller Actions definiert werden.

Die Action-Implementation folgt dem Observer-Pattern, eine Action entspricht einem Observable und ein Skript dem Observer. Dabei ist zu beachten, dass der Observer sich beim Observable registriert und auch wieder abmeldet.

Diese Actions werden ausgelöst, wenn der zugewiesene Button, Joystick, Touchpad, etc. betätigt wird.

So kann ebenfalls die Teleport Funktionalität an einen Controller gebunden werden.

Damit der Benutzer sich dann auch irgendwo hin im Gebäude teleportieren kann, gibt es von SteamVR zwei verschiedene Möglichkeiten.

Zum einen gibt es Prefabs für einzelnen Teleport-Punkt zu dem hin teleportiert werden kann, der Entwickler (Welt-Designer) muss diesen Punkt jedoch selber platzieren. In

Collaborativ BIM VR verwenden wir Teleport-Punkte um zwischen dem Erdgeschoss und der ersten Etage hin und her teleportieren. Die andere Möglichkeit ist eine sogenannte Teleport-Area. Dabei handelt es sich um ein Skript, welches eine Oberfläche eines GameObjects teleportierfähig macht. In Collaborativ BIM VR wird dieses Skript für den Boden verwendet, so dass sich ein Benutzer auf der gleichen Etage teleportieren kann. Voraussetzung für die Teleport-Area ist ein Collider auf dem GameObjekt. Das gewählte Beispielgebäude hat zwei Etagen. Damit man sich auch auf der oberen Etage teleportieren kann, benötigt der Boden, was zugleich die Decke der unteren Etage ist, einen Collider. Fügt man diesen Collider hinzu, kann man nicht mehr einfach zwischen den beiden Etagen hin und her teleportieren, da der Pointer, der aufzeigt wohin man teleportiert, nicht durch den Collider durchgehen kann.

Im Rahmen diese Prototypen haben wurde der Collider von Hand bearbeitet, so dass der Treppenbereich für den Etagenwechsel offen ist.

## 5.4 Feedback in VR

Wie im Konzept (siehe 4.1.2 Feedback in VR) begründet wird ein Kontextmenü implementiert.

Für die User Interface Elemente werden die von Unity zur Verfügung gestellten UI-GameObjects, für die Texte die TextMeshPro-GameObjects verwendet. Das Kontextmenü existiert als GameObject in der Scene. Wird der gebundene Button auf dem Controller gedrückt gehalten, wird das Kontextmenü-GameObject an die position der linke Hand gesetzt. Zugleich wird ein RayCast in die Mitte des Blickfeldes ausgesandt um das Objekt zu identifizieren, das angepasst werden soll. Damit der RayCast ein GameObject treffen kann, benötigt das GameObject einen Collider. Das führt dazu, dass sich der Benutzer nicht mehr frei auf der Etage teleportieren kann, da die Wände und Türen neu Colliders haben und der Pointer, der angibt wohin man teleportiert, nicht durch Colliders gehen kann.

Damit der RayCast nicht mit jedem Collider interagiert, kann er auf einem Layer ausgesandt werden. Beim Laden der Scene setzt ein Skript für die relevanten GameObjekte; Wände, Türen, Fenster und Einrichtungsgegenstände einen Layer für den RayCast.

Wird ein Button im Kontext-Menu gedrückt, wird durch den Event-Handler das Erstellen des BCFZip und Screenshot angestossen. Der Ablauf dieses Prozesses wird im Kapitel 5.5 erläutert.

## 5.5 Feedback aus VR exportieren

Wir haben bereits den Aufbau eines BCFs und die verschiedenen Implementierungsmöglichkeiten erläutert. Auch haben wir bereits erwähnt, dass die xBIM Sub-Library *xBIM.BCF* Funktionalitäten zum Importieren und Exportieren von BCF Dateien bietet. Die xBIM.BCF Library ist im NuGet Package Store verfügbar. Da ein Unity Projekt, kein normales .NET Projekt ist, muss hier ein Trick angewendet

werden, damit das Package importiert werden kann. Das Package muss manuell heruntergeladen und im */Assets/Plugins* Ordner abgelegt werden. Zusätzlich muss ein Linker XML-File im Root-Verzeichnis des Projektes angelegt werden, damit das Package auch jedes Mal beim Starten der Applikation geladen wird.

Die Library bietet leider nur das allernötigste an Konstruktoren und Setter-Methoden. Dies schränkt die Konfigurierbarkeit des Formates enorm ein bzw. sind wir gebunden an die Form, welche die Entwickler der Library als korrekt empfunden haben.

Das betätigen eines Buttons im Kontextmenü triggert auch das Erstellen der BCF-Datei. Es werden Parameter wie die GUID des anvisierten Elementes, dessen Namen und welche Änderung darauf vorgenommen werden soll, übergeben. Andere notwendige Informationen werden direkt dem importierten IFC-Modell entnommen. Als *CreationAuthor* wird beispielsweise der am System angemeldete Benutzer verwendet. Der effektive Änderungswunsch wird als *BCFComment* in der *Markup.bcf* - Datei hinterlegt.

Zu einem vollständigen BCF gehört auch ein Screenshot. Diesen nehmen wir direkt in VR auf. Damit man auf dem Bild alles erkennen kann, werden die Controller und das Kontextmenü für die Aufnahme ausgeblendet. Da in Unity Code immer nur am Ende eines Frames ausgeführt wird, müssen innerhalb dieser Subroutine zwei Frames gezielte werden, da die asynchrone Implementierung der Screenshot-Methode ansonsten blockiert.

Ein weiterer Teil eines BCFs ist der Viewpoint des Issues. Dieser beinhaltet die Kamera Position, Rotation und Up-Vector. Wir setzen diese Informationen anhand der aktuellen Lage und Position der VR Kamera des Player Objektes.

Da IFC-Dateien für ihre Objekte eine eigene Form von GUIDs verwenden, welche nicht dem universellen Standard [5] entsprechen, müssen diese enkodiert werden, da das BCF beide Arten an GUIDs verwendet. Die Enkodierung basiert auf Base64 und ist von Herausgeber des BCF, BuildingSMART, definiert.

Zur Benennung des BCFZIP verwenden wir den Namen des IFC-Objektes, auf welches es sich bezieht. Da die Namen der Objekte Zeichen beinhalten können, welche in einem Windows Dateisystem nicht zulässig sind, werden diese durch Leerzeichen ersetzt.

## 5.6 Architektur

Das Projekt kommt in der üblichen Struktur eines Unity3D-Projekt daher. Alles findet in einer einzelnen Szene statt. Ganz im Sinne der Best Practices [8] wurde darauf geachtet, dass möglichst viele Komponenten abgekoppelt vom restlichen System funktionieren. Auch wurden Logik und Präsentation strikt getrennt. Dies fördert die Erweiterbar- und Modifizierbarkeit der Applikation.

Jedes Skript verfolgt immer nur eine Aufgabe (Single Responsibility Principle). Diese modulare Architektur ist weniger fehleranfällig und Änderungen an den einzelnen Skripts

können bedingungslos vorgenommen werden, ohne dass Gefahr besteht, dass sie Auswirkungen auf andere Komponenten haben.

### 5.6.1 Projektordnerstruktur

Alle für das Projekt relevanten Dateien befinden sich im Root-Ordner `./Assets`. Dieser beinhaltet folgende Ordner:

**Tabelle 3 Projektordnerstruktur**

<b>Ordnername</b>	<b>Beschreibung</b>
Floors	Hier werden alle von Tridify generierten Modell Assets abgelegt
Plugins	Beinhaltet alle externen .NET Packages, wie zum Beispiel xBIM.BCF
Prefabs	Beinhaltet alle Prefabs (Player, Teleporting, ContextMenuCanvas, etc.)
Scences	Beinhaltet die Szene
Scripts	Beinhaltet alle Skripts
Standard Assets	Beinhaltet nur den Environment Teil des ansonsten riesigen Standard Assets-Package. Wird für die Texturen des Aussenbereiches benötigt
SteamVR	SteamVR Library
SteamVR_Input	SteamVR Input Handling Scripts
SteamVR_Ressources	SteamVR Settings File
StreamingAssets	SteamVR Controller Bindings
TextMesh Pro	Wird für das Stylen der Buttons des ContextMenus verwendet
Tridify	Tridify Unity Plug-in

### 5.6.2 Skripts

Alle von uns angefertigten Skripts sind im Ordner `./Assets/Scripts/` abgelegt. Dieser beinhaltet folgenden Skripts:

**Tabelle 4 Vorhandene Skripts und deren Zweck**

Skriptname	Zweck
BCFFactory	Validiert die übergebenen Parameter, macht einen Screenshot und stösst den Prozess zum Erstellen eines BCFZIP an.
BimHelper	Setzt beim Starten der Applikation einen definierten Layer für die Raycasts, so dass das richtige Kontextmenu geöffnet werden kann.
ContextMenu	Erzeugt das Kontextmenu für das Objekt im Fokus.
ContextPositioner	Positioniert das Kontextmenu an die Position des linken Controllers.
CreateBCF	Wird vom BCFFactory angestossen, um anhand der übergebenen Parametern ein BCFZIP zu erstellen.
FeedbackListener	EventHandler für die Buttons des Kontextmenu. Stösst die Erzeugung von Screenshot und BCFZip an.
IFCGUIDParser	Verwandelt eine IfcGuid in eine normierte GUID

### 5.6.3 Plugins / Libraries

Für das Projekt wurden folgende Plug-ins und Libraries verwendet:

**Tabelle 5 Vorhandene Plugins und Libraries und deren Zweck**

Name	Zweck
xBIM.BCF	Benötigt, um die Erstellung eines BCF zu ermöglichen

Tridify Unity Plug-in	Wird benötigt, um auf die konvertierten IFC Dateien von Unity aus zuzugreifen und für die initiale Generierung des Floor Lightings
SteamVR	Wird für die Verwendung des Players und dem Teleportieren benötigt
Unity Standard Assets	Beinhaltet Standard Texturen (wurde nur partiell importiert)
TextMesh Pro	Wird für das Stylen der Buttons des ContextMenus verwendet

### 5.7 Adaptionen aus den Usability Tests

Obwohl keine vollumfänglichen Usability Tests gemacht wurden, konnten durch die durchgeführten Testdurchläufe (siehe Kapitel 4.2.1. Usability Tests) nützliches Feedback gewonnen werden.

Im Zuge dessen wurde ein optisches Feedback implementiert, um dem Benutzer deutlich zu signalisieren, wenn er über einen Button des ContextMenus hovert. Beobachtungen haben erwiesen, dass sich Benutzer im virtuellen Raum ansonsten unsicher sind, ob sie den Button wirklich ausgewählt haben.

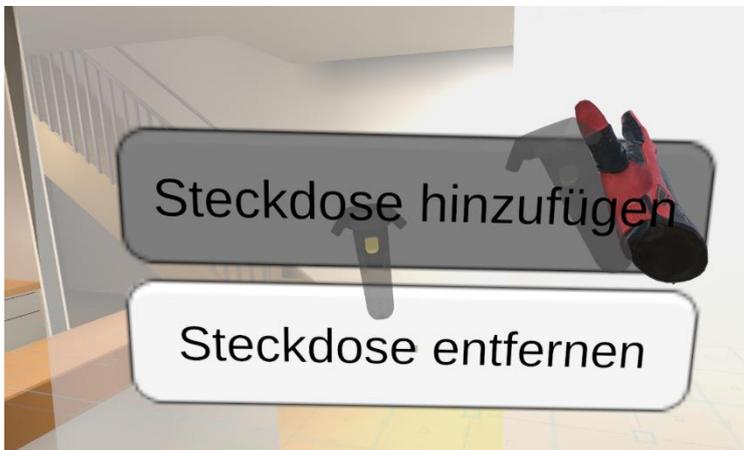


Abbildung 3 Kontextmenü Hover Effekt

Auch wurde die Grösse des ContextMenus generell überarbeitet, damit das Aufpoppen dieses UI Elementes den Benutzer nicht überwältigt.

Das Visualisieren des Rays, welcher verwendet wird, um zu erkennen, welches Element der Benutzer gerade betrachtet, konnte im Rahmen dieses Projektes leider nicht umgesetzt werden. Es würde jedoch sicherlich die Usability der Applikation steigern und sollte bei einer Weiterentwicklung des Projektes implementiert werden.

Ein weiterer Aspekt bezüglich der Benutzerfreundlichkeit wäre ein Feedback bezüglich der erfolgreichen Erstellung eines BCFZIPs während sich der Benutzer im virtuellen Raum befindet. Da aus technischen Gründen bei der Erstellung des BCFZIPs zwei Frames aufgegeben werden, stockt die Animierung für einen kurzen Moment. Dies kann als Indikator gewertet werden, dass das File korrekt erzeugt wurde. Da diese Verzögerung aber auch als Lag in der Applikation interpretiert werden kann, sollte man sich über eine weitere Form des Feedbacks Gedanken machen. Aus Sicht des Projektteams, kommen dafür drei Varianten in Frage.

Ein haptisches, optisches oder auditives Feedback, dass die Datei erfolgreich erstellt wurde. Auf ein weiteres haptisches Feedback (wie das Vibrieren der Controller) sollte verzichtet werden, da es bereits verwendet wird, um zu indizieren, dass sich die Controller in einem fest verbauten Strukturelement wie einer Wand befinden. Ein optisches Feedback könnte in Form eines kurz eingeblendeten Textes erreicht werden. Ein auditives Feedback würde voraussetzen, dass der Ton aktiviert ist und der Benutzer die Kopfhörer des Headsets effektiv richtig platziert hat (und nicht weggeklappt hat).

Es würde eine weitere Konzept-/Test-Phase benötigen, um diesbezüglich die richtige Entscheidung zu treffen, daher wurde dieser Aspekt des jetzigen Produktes offengelassen.

## 6 Diskussion

### 6.1 Resultate

Ein BIM kompatibler Bauplan kann mit Hilfe von Tridify in Unity importiert werden. Mit minimalen Anpassungen und mit Hilfe der von uns erstellten Anleitung, kann die Scene gestartet werden. Der Benutzer kann sich im virtuellen Raum im Gebäude bewegen und wo immer verlangt Feedback geben. Dieses Feedback wird einzeln in ein BCFZip geschrieben. Das BCFZip ist valid und kann von einem weiteren Tool (BCFcollab BCF Manager) beispielsweise in Revit geöffnet werden. Der Datendurchsatz bis dahin ist gewährleistet. BCFCollab hat zurzeit noch Probleme beim Interpretieren der ViewPort-Datei und wirft die Fehlermeldung, dass der Viewport unbenutzbar sei. Die genaue Ursache konnte bis anhin jedoch nicht erruiert werden.

Trotzdem sind wir der Meinung, dass das Projekt eine gute Basis zur Weiterentwicklung bietet und ein Erfolg ist.

### 6.2 Reflektion

Alle Ziele, welche am Anfang des Projektes definiert wurden, sind erreicht worden. Das Aufteilen des Projektes in vier Phasen, hat sicherlich dabei geholfen. Es war somit leichter, Meilensteine zu bestimmen und nach Beendigung einer Phase einen Abschnitt des Projektes hinter sich zu lassen.

Die Aufteilung der Arbeitspakete in jeglichen Aspekten des Projektes (Implementierung, Forschung, Dokumentation), war massgeblich dafür verantwortlich, dass wir unsere Meilensteine immer rechtzeitig erreichen konnten. Der Aufgabenteilung ist es auch zu verdanken, dass aus der Projektwoche ein funktionsfähiger Prototyp resultierte.

Das Schreiben der Dokumentation und die Aufarbeitung der wissenschaftlichen Quellen hätte man systematischer angehen sollen. Dies ist eine wertvolle Lektion, im Hinblick auf die Bachelor Thesis.

### 6.3 Wie weiter?

Das Projekt bietet noch viele Möglichkeiten, die Usability und generell die User Experience zu steigern. Da von der IFC Datei doch erheblich viele Informationen mit importiert werden, bietet es sich an, dass man diese im virtuellen Raum visualisiert. Somit könnten die Begebenheiten der einzelnen Strukturelemente angezeigt werden und so das Erforschen des Gebäudes interessanter gestaltet werden.

Ein Aspekt der sich sicherlich lohnt weiterzuverfolgen, ist das Erstellen eines funktionsfähigen BCF Viewpoint, um den Feedback Export als BCF zu komplettieren. Es ist gut möglich, dass dies in Zukunft mit einer neueren Version der xBIM.BCF Library möglich sein wird.

Eine weitere technische Erweiterung des Projektes wäre, den Import der IFC-Dateien und die Beleuchtung des Gebäudes komplett zu automatisieren. Dies würde die Benutzerfreundlichkeit für Endbenutzer, welche sich nicht gut mit Unity auskennen, bedeutend steigern. Um dies zu erreichen, müsste man jedoch die gesamte Import Pipeline neugestalten. Auch müsste man enorm viel Zeit investieren, um einen gleich stabilen Import, wie ihn ein professionelles Tool wie Tridify bietet, hinzubekommen.

## 7 Glossar

Tabelle 6 Glossar

Begriff	Bedeutung
Autodesk Revit	Weit verbreitetes CAD Programm
Tridify (Convert)	Online Software zum Konvertieren von IFC-Dateien zu Unity Assets
Szene	Eine Szene bildet den Rahmen eines virtuellen Raumes
GameObject	Fundamentale Objekte in Unity, welche Komponenten repräsentieren.
Prefab	Ein Container, welcher GameObjects und deren Eigenschaften und Begebenheiten bündelt und speichert.
Asset	Repräsentation eines Gegenstandes in Unity. Dies kann ein 3D-Modell sein oder eine Textur, Audio-Datei, etc.
BuildingSMART	Non-profit-Organisation, welche die Austauschformate IFC und BCF definiert und weiterentwickelt haben.
IFC (Industry Foundation Classes)	Standardisiertes Datenmodell zur Beschreibung von Gebäude und Konstruktionen.  Platform-neutral und OpenSource
BCF (BIM Collaboration Format)	Format, um Änderungsvorschläge an Bauplänen präzise zu kommunizieren.
NuGet Package Manager	NuGet ist eine Plattform, wo von der Community entwickelte Packages für .NET ausgetauscht werden können.

## 8 Quellenverzeichnis

- [1] FHNW (2018): Digitaler Wandel Bau. <https://www.fhnw.ch/de/die-fhnw/strategische-entwicklungsschwerpunkte/strategische-initiativen/bauwesen/digitaler-wandel-bau> (abgerufen am 28.10.2019)
- [2] Binstock, Atman (15. Mai, 2015): Powering the Rift. <https://www.oculus.com/blog/powering-the-rift/> (abgerufen am 15.01.2020)
- [3] Vive (2018): VIVE Pro. <https://www.vive.com/us/product/vive-pro/> (abgerufen am 15.01.2020)
- [4] Sollien, Jon Anders (11. März 2019): BCF-XML Documentation. [https://github.com/buildingSMART/BCF-XML/tree/release\\_2\\_1/Documentation](https://github.com/buildingSMART/BCF-XML/tree/release_2_1/Documentation) (abgerufen am 28.11.2019)
- [5] Open Group (13. Januar 2013): Universal Unique Identifier. <https://pubs.opengroup.org/onlinepubs/9629399/apdx.htm> (aufgerufen am 02.01.2019)
- [6] Unity (08. Juni, 2019): Unity - Manual: Light Probes. <https://docs.unity3d.com/Manual/LightProbes.html> (abgerufen am 31.12.2019)
- [7] Unity (21. November, 2019): Unity - Manual: Emission. <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterEmission.html> (abgerufen am 31.12.2019)
- [8] Kalms, Mikael (Mai, 2019): How to architect code as your project scales. <https://unity3d.com/de/how-to/architect-code-as-your-project-scales> (abgerufen am 15.01.2020)
- [9] SteamVR (2020): Steam VR <https://www.steamvr.com/de/> (abgerufen am 16.01.2020)
- [10] Oculus (2020): Oculus Documentation <https://developer.oculus.com/documentation/> (abgerufen am 16.01.2020)
- [11] Weiß, Yannick. Hepperle, Daniel. Sieß, Andreas. Wolfel, Matthias (2018): What User Interface to Use for Virtual Reality? 2D, 3D or Speech—A User Study (2018):
- [12] Hunt, Xavier. Potter Leigh Ellen (2018): High Computer Gaming Experience May Cause Higher Virtual Reality Sickness

## **9 Abbildungsverzeichnis**

Abbildung 1 Raum ohne künstliche Beleuchtung	20
Abbildung 2 Raum mit künstlich generierter Beleuchtung	21
Abbildung 3 Kontextmenü Hover Effekt	26

## 10 Tabellenverzeichnis

Tabelle 1 Auswertung der Fragebogendaten (vgl. Tabelle 15 in Paper).....	12
Tabelle 2 Vergleich Usability Faktoren aus dem Fragebogen (vgl. Tabelle 2 aus dem Paper) .....	16
Tabelle 3 Projektordnerstruktur .....	24
Tabelle 4 Vorhandene Skripts und deren Zweck .....	25
Tabelle 5 Vorhandene Plugins und Libraries und deren Zweck .....	25
Tabelle 6 Glossar .....	30

## 11 Eidesstattliche Erklärung

Die Verfasser erklären an Eid statt, dass sie die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als die angegebenen Hilfsmittel angefertigt haben. Die aus fremden Quellen (einschliesslich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

---

Ort, Datum

---

Nick Brönnimann

---

Sebastian Renner

## **Anhang**

## Anleitung zum Importieren von IFC Files in Unity mittels Tridify

Sebastian Renner

14.11.19

### Inhaltsverzeichnis

<b>Tridify</b>	<b>2</b>
IFC Datei hochladen und konvertieren	2
Konvertierte Datei veröffentlichen	2
<b>Unity</b>	<b>2</b>
Hinzufügen des Tridify Plug-Ins	2
IFC Models in ein Unity Projekt importieren	3
<b>Floor Lighting</b>	<b>3</b>

In dieser Anleitung wird davon ausgegangen, dass Sie einen aktiven Tridify Account besitzen. Ansonsten kann dieser unter <https://convert.tridify.com/#/register> erstellt werden. Auch muss beachtet werden, dass Tridify nur IFC Dateien vom Schema **IFC2x3** unterstützt. Hochladen von Dateien im IFC4 Schema führt zu einem Validation Error.

## Tridify

### IFC Datei hochladen und konvertieren

Loggen Sie sich in Ihren Tridify Account ein. Erstellen Sie einen neuen Projektordner oder klicken Sie auf einen bestehenden Ordner (per default "My Files"). Ziehen Sie die IFC Datei auf das Drag&Drop-Feld. Die Datei wird nun hochgeladen. Nachdem die Validierung abgeschlossen ist, erscheint ein neuer Button "Start Conversion". Klicken Sie auf diesen Button. Im anschliessend auftauchenden Fenster sollten Sie beide Optionen auf An stellen. Beide Optionen machen hingegen nur Sinn, wenn die IFC Datei aus mehreren Stöcken besteht.

Start new conversion



#### OPTIONS



Split by storey

All storeys in the IFC file will be separated into their own files.



Generate  
Ceiling Slabs  
and Extras to  
Storeys

Upper storey floor slabs and lower storey stairs will be appended to each storey as 2 additional files. This is useful to get complete single storeys with less memory usage in Unity.

Klicken Sie danach auf "Start Conversion". Dadurch wird die Konvertierung gestartet. Je nach Grösse der Datei kann dies einige Minuten dauern.

### Konvertierte Datei veröffentlichen

Wählen Sie eine bereits konvertierte Datei aus. Klicken Sie auf den Reiter "Web Publishing". Wählen Sie die Option "Publish" an.

## Unity

### Hinzufügen des Tridify Plug-Ins

Für jedes neue Unity Projekt in dem Sie IFC Modelle importieren möchten, müssen Sie das Tridify Plug-In hinzufügen. Dieses kann [hier](#) heruntergeladen werden.

Öffnen Sie das Unity Projekt in demjenigen Sie das Plug-In hinzufügen wollen. Führen Sie das heruntergeladene Unity Package File aus und klicken Sie auf Import. In der Menüleiste von Unity sollte nun ein Reiter namens "Tools -> Tridify" vorhanden sein.

### **IFC Models in ein Unity Projekt importieren**

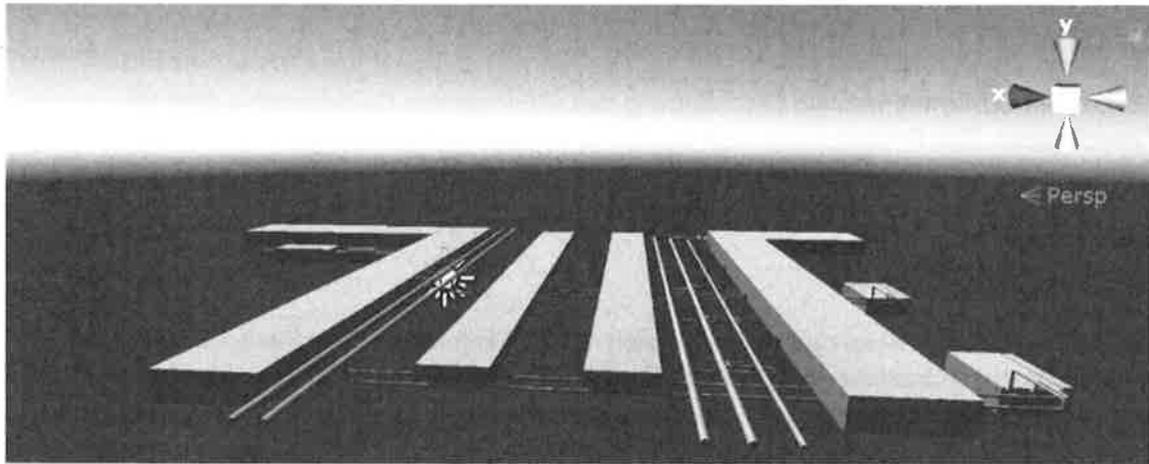
Wählen Sie in der Menüleiste “Tools -> Tridify -> Conversion Projects” aus oder benutzen Sie den Shortcut Ctrl+Alt+O. Anschliessend müssen Sie sich mit Ihrem Tridify Account anmelden. Wählen Sie den Projektordner aus in dem sich die veröffentlichte Datei befindet. Klicken Sie die gewünschte Datei an. Wählen Sie die gewünschten Etagen an, welche Sie importieren möchten und klicken Sie auf “Load selected”. Warten Sie bis die Assets importiert wurden. Das importierte Modell wird automatisch als Prefab am Nullpunkt der Szene platziert.

### **Floor Lighting**

Tridify bietet die Möglichkeit automatisch generierte Lichtquellen in importierte Gebäude zu platzieren. Es ist zu beachten, dass dafür nur ein Modell in der Szene vorhanden sein darf!

Wählen Sie in der Menüleiste “Tools -> Tridify -> Floor Lighting” aus oder benutzen Sie den Shortcut Ctrl+Alt+L. Nun können Sie die Ebene auswählen, an denen Sie Lichtquellen platzieren möchten. Wir empfehlen so viele wie möglich einzufügen. Es ist zu beachten, dass sich am Boden der untersten Ebene und an der Decke der obersten Ebene keine Quellen platzieren lassen. Klicken Sie anschliessend auf “Create Floor Lighting”. Dies führt dazu, dass alle Lichtquellen und “Light Probe”-Grids in der Szene platziert werden.

Wählen Sie in der Menüleiste “Window -> Rendering -> Lighting Settings” aus. Setzen Sie die Checkbox unter “Mixed Lighting -> Baked Global Illumination” auf False. Klicken Sie anschliessend auf “Generate Lighting”. Daraufhin werden alle Lightmaps berechnet. Dies wird einige Minuten dauern.



# Projektvereinbarung Collaborative BIM VR

Sebastian Renner  
Nick Brönnimann

Brugg, 14.10.2019

# Vorwort

In diesem Dokument wird die Durchführung des IP5-Projektes **Collaborative BIM VR** beschrieben. Das Projekt wurde vom Institut für Interaktive Technologien der FHNW ausgeschrieben, als Teilprojekt der Strategischen Initiative zum Digitalen Wandel im Bauwesen.

## Ausgangslage

In der Bauindustrie ist es nach wie vor üblich mit 3D-Modellen und schlussendlich mit ausgedruckten Konstruktionsplänen zu arbeiten. Der Sprung zu VR geht nur schleppend voran, obwohl das CAM/CAD optimale Voraussetzung dafür bietet. Auch wenn es vereinzelte Tools gibt, welche es ermöglichen Baupläne in VR darzustellen, werden diese zurzeit in der Branche kaum verwendet. Mit bestehenden Tools ist es ausserdem nicht möglich mit den Modellen im virtuellen Raum zu interagieren oder Anmerkungen zu hinterlassen.

Collaborative BIM VR soll das Verstehen der Konstruktionsplänen durch Darstellung im virtuellen Raum erleichtern. Die Endbenutzer sollen die Gebäude erleben und direkt Feedback abgeben können.

## Ziele

### Baupläne Unity importieren

Als Bauzeichner will ich die BIM-kompatiblen Baupläne auf möglichst einfache Art und Weise in Unity importieren können.

### Feedback zum Bauplan

Als Benutzer will ich auf intuitive Art und Weise direkt in der virtuellen Realität Feedback zum Bauplan abgeben können. Vorzugsweise kann beim Abgeben des Feedbacks die Ebene des BIM-Bauplans gesetzt werden.

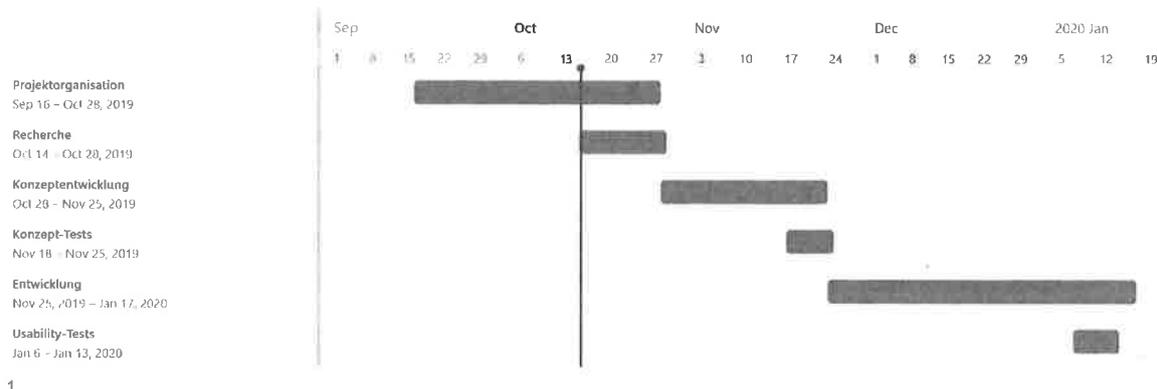
### Feedback aus VR exportieren

Als Bauzeichner will ich das Feedback der Benutzer aus dem virtuellen Raum exportieren, so dass notwendige Anpassungen am entsprechenden Bauplan vorgenommen werden können.

### Usability und User Experience

Es soll zu Kollisionen mit Wänden, Motion-Sickness und Interaktionen mit UI-Elementen Konzepte erstellt werden, wie die Usability und User Experience durch Einhalten von Standards gesteigert werden kann.

# Grob Plan / Meilensteine



## Roadmap der in Gitlab erfassten Epics

Abgebildet ist der grob Plan, aufgeteilt in die vier Projektphasen; *Projektorganisation*, *Recherche*, *Konzeptentwicklung* und *Entwicklung*. Ebenfalls eingetragen sind die Blöcke *Konzept-Tests* und *Usability-Tests*. Im nachfolgenden Abschnitt sind die Projektphasen mit den zugehörigen Meilensteinen und einer kurzen Erläuterung aufgelistet.

## Projektphasen und Meilensteine

### Projektorganisation

In der Phase *Projektorganisation* sind die folgenden Meilensteine enthalten:

- Projektvereinbarung; Die Projektvereinbarung ist erstellt und von allen Beteiligten unterzeichnet
- Aufbau Abschlussbericht; Das Dokument für den Abschlussbericht ist erstellt und die Struktur definiert

### Recherche

- IFC Dateien in Unity kompatible Dateien umwandeln; verfügbare Tools recherchieren und deren Tauglichkeit bestimmen. Auswahl für geeignetes Tool treffen.
- Definieren der Forschungsfragen und recherchieren von Fachliteratur, wissenschaftlichen Ressourcen, etc.

<sup>1</sup> Foto der Roadmap der auf Gitlab erfassten Epics, abgerufen am 14.10.2019

[https://gitlab.fhnw.ch/groups/ip5-collaborative-bim-vr/-/roadmap?label\\_name\[\]=Scope%3A%3AGrobplan](https://gitlab.fhnw.ch/groups/ip5-collaborative-bim-vr/-/roadmap?label_name[]=Scope%3A%3AGrobplan)

## Konzeptentwicklung

- Interagieren in VR; Verschiedene Möglichkeiten zum Interagieren in VR werden anhand ausgewählter Gesichtspunkte verglichen. Eine Variante wird ausgewählt und ein Konzept erstellt
- Bewegen in VR; Verschiedene Fortbewegungsmöglichkeiten in VR werden anhand ausgewählter Gesichtspunkte verglichen. Eine Variante wird ausgewählt und ein Konzept erstellt
- Feedback zu Bauplan in VR; Verschiedene Möglichkeiten zum Erfassen von Feedback werden verglichen und ausgewertet. Ein Konzept wird erarbeitet
- Exportieren des Feedbacks; Verschiedene Varianten zum Exportieren vergleichen und geeignete konzipieren.
- Die erarbeiteten Konzepte werden im Block *Konzept-Tests* mit Kommilitonen auf ihre Tauglichkeit überprüft

## Entwicklung

Die in der Forschungsphase erstellten Konzepte werden implementiert.

In der Umsetzungsphase ist ebenfalls der *Usability-Test-Block* enthalten, mit dem die Implementationen der Konzepte mit einer Testgruppe geprüft werden können. Zu den Meilensteinen gehören;

- Ende Projektwoche das Konzept des Feedbacks implementiert
- Ende Projektwoche die Konzepte zur Usability und User Experience weitestgehend implementiert
- 10. Januar die Konzepte prototypisch umgesetzt

## Risiken

Aus unserer Sicht ist das grösste Risiko, welches den Projekterfolg gefährden könnte, dass sich der Import der IRC-Dateien in Unity als ein grösseres Unterfangen herausstellt. Dazu zählt auch, wie sich die importierten Assets ansprechen und modifizieren lassen. Um dieses Risiko zu mindern, behalten wir uns weitere Tools aus der Recherche in der Hinterhand, als Option, falls sich das ursprünglich erwählte Tool als untauglich erweist.

Bei VR-Projekten oder bei Software-Projekten allgemein, besteht stets das Risiko, dass ein Feature mehr Aufwand generiert, als ursprünglich angenommen. Obwohl die Features in diesem Projekt überschaubar sind, besteht dieses Risiko trotzdem, da bei VR-Projekten schnell etwas vergessen wird. Dieses Risiko versuchen wir durch gute Planung und mittels genauer Recherche so gering wie möglich zu halten.

## Unterschriften

Mit der Signierung dieses Dokumentes einigen sich das Projektteam und die Betreuungspersonen auf die in diesem Dokument festgehaltenen Meilensteine und Ziele.

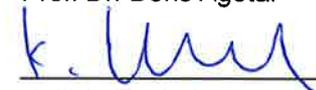
Brugg, 28.10.2019  
Ort, Datum

  
Nick Brönnimann

  
Sebastian Renner

Brugg, 17.10.2019  
Ort, Datum

  
Prof. Dr. Doris Agotai

  
Kathrin Koebel