HAW HAMBURG

Bachelorarbeit

Sebastian Peters

Entwicklung eines VR-Shops mit 3D-Replikaten realer Objekte

Fakultät Technik und Informatik
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Sebastian Peters

Entwicklung eines VR-Shops mit 3D-Replikaten realer Objekte

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Birgit Wendholt
Zweitgutachter: Prof. Dr. Stefan Sarstedt

Eingereicht am: 31. Dezember 2018

Sebastian Peters

Thema der Arbeit
Entwicklung eines VR-Shops mit 3D-Replikaten realer Objekte

Stichworte
Virtuelle Realität, Photogrammetrie, 3D-Scanning, Unity, HTC Vive, E-Commerce

Kurzzusammenfassung
Die vorliegende Bachelorarbeit beschreibt einen beispielhaften Entwurf eines Webshops in der virtuellen Realität. Dabei wird ein besonderer Fokus auf die Bereitstellung von 3D-Modellen realer Objekte gelegt, welche mithilfe von möglichst intuitiven 3D-Scannern automatisch rekonstruiert werden. Die Modelle werden als Ergebnis in den VR-Shop eingefügt, um dort virtuell betrachtet werden zu können. So entsteht für den potentiellen Käufer die Option, die angebotenen Artikel vor einer Kaufentscheidung besser zu bewerten. Eine prototypische Implementation zu diesem Entwurf wurde mithilfe von Unity und der HTC Vive erstellt.

Title of the Thesis
Development of a VR Shop with 3D Replicas of Real Objects

Keywords
Virtual Reality, Photogrammetry, 3D Scanning, Unity, HTC Vive, E-Commerce

Abstract
This bachelor thesis deals with an exemplary conception of a web shop in a virtual-reality environment. The main focus will be set onto the provision of 3D models of real objects, which should be automatically recreated by preferably intuitive 3D scanners. These models should be included into the VR shop, where they can be inspected in a virtual manner. This approach would aid the potential buyers in their judgement of the product prior to a purchase decision. A prototype implementation of this concept was created using Unity and the HTC Vive.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Motivation

Virtual Reality (VR) ist ein Forschungsgebiet, welches erst in den letzten Jahren drastisch an Präsenz gewann. Durch gestiegene Computerleistung sowie der Veröffentlichung von *Head-Mounted Displays (HMD)* speziell für Endverbraucher – z.B. die HTC Vive und die Oculus Rift – wurde das Gebiet so das erste Mal für ein größeres Spektrum an Interessengruppen zugänglich. Auch wenn die Technologien als solche noch keinen vollständigen Durchbruch erreicht haben, wird trotzdem viel daran geforscht und experimentiert. Denn auch wenn noch nicht jeder Mensch eine VR-Brille besitzt, gibt es doch mehr VR-Nutzer als je zuvor.

Dennoch steht VR noch weit vor Omnipräsenz, wie es z.B. bei Smartphones bereits der Fall ist. Dies ist einerseits durch die noch immer hohen Preise und Anforderungen an die Hardware begründet, andererseits aber auch dadurch, dass noch keine bedingungslose Akzeptanz gegenüber der Technik besteht. Die Folge daraus ist, dass Investoren sich nur vorsichtig an die Entwicklung von VR-Applikationen wagen, weswegen die Menge an aktiven Entwicklern gering ist.

Google zeigte aber mit seinem Spam-Bot-Schutz *reCAPTCHA*[a], wie der Anwender zum Entwickler werden kann: durch einfache Verwendung der Software (in diesem Fall, Training der künstlichen Intelligenz). Es werden keine bestehenden Kenntnisse über Software-Engineering dafür benötigt

Dieses Prinzip könnte auch auf VR abgebildet werden. Durch passend gestaltete User-Interfaces könnten die wenigen Besitzer einer VR-Brille zur Verbesserung der Technik beitragen, was zur Beschleunigung des Entwicklungsprozesses führt, da so die Menge an Mitwirkenden deutlich vergrößert wird.

[a] https://developers.google.com/recaptcha

Ein davon unabhängiges Problem ist durch *E-Commerce* entstanden. Die rasant steigende Präsenz von Online-Shops hatte zur Folge, dass Bestellungen heutzutage fast ausschließlich im Internet stattfinden. Die Option, sich in einem klassischen Warengeschäft einen optischen Eindruck aus erster Hand über das Produkt zu machen, findet kaum noch Beanspruchung.

Somit werden in vielen Fällen nur noch Produktbilder zur Meinungsbildung eingesetzt. Dabei ist der Kunde aber der Ehrlichkeit des Anbieters ausgeliefert, da leider in vielen Fällen digitale Bildbearbeitungsprogramme zur künstlichen Verschönerung der Produktbilder eingesetzt werden. Es ist dabei durchaus möglich, vorsätzlich die Abbildungen so zu manipulieren, dass eine unrealistische Präsentation des Artikels entsteht, um die Kaufentscheidung des Kunden zu beeinflussen.

Es entsteht also ein Widerspruch an Interessen:

(a) Kunden wollen fast nur noch online bestellen, aber
(b) Kunden wollen auch kein Produkt kaufen, das unfair beworben wurde.

Diese Bachelorarbeit verfolgt das Ziel, die beiden beschriebenen Probleme – mangelnde Mitwirkende für VR und mangelnde Produktinformationen in Online-Shops – kombiniert zu lösen: Ein VR-Shop, wobei durch Kunden erstellte 3D-Modelle einen unverfälschten Eindruck der Artikel gewährleisten sollen.

Entwürfe für virtuelle Einkaufsläden gibt es bereits, z.B. *ShelfZone VR*[b]. Es ist jedoch noch nicht geklärt, ob und wie diese einen regulären Webshop ersetzten bzw. diesen übertreffen sollen. Die wesentliche Fragestellung dabei ist: Welche Vorteile bietet ein VR-Shop überhaupt? In einer empirischen Studie zur Untersuchung des Konsumentenverhaltens in einem virtuellen Einkaufserlebnis[18] wurden zwei Hypothesen als mögliche Begründungen dieses Mehrwerts gegeben:

> Hypothesis 1. *The VR shopping mall significantly improves convenience, enjoyment and perceived quality assurance in comparison with the ordinary shopping mall.*
> Hypothesis 2. *The VR shopping mall results in better customer satisfaction than the ordinary shopping mall.* — K.C. Lee, N. Chung (2008)[18, S. 93]

Zusammengefasst wird also angenommen, dass VR-Shops dem Kunden ein besseres Einkaufserlebnis bieten, indem Produkte dreidimensional betrachtet werden können.

[b] https://www.youtube.com/watch?v=-2UT2KcnJiE

Dafür müssten aber zu den Tausenden von Produkten 3D-Replikate händisch erstellt werden – für den alleinigen Anbieter wäre dies eine Sisyphusaufgabe. Es gibt aber viele Personen, die helfen könnten: Kunden, die das Produkt bereits gekauft haben. Wenn es also eine einfache, intuitive und günstige Softwarelösung gäbe, mit der die Besitzer von gekauften Artikeln diese automatisch in 3D-Modelle verwandeln könnten, würde die Digitalisierung deutlich beschleunigt werden.

Die 3D-Modelle könnten dann in dem VR-Shop verfügbar gemacht werden, damit weitere interessierte Kunden einen unverfälschten, dreidimensionalen Eindruck erhalten, wie die Artikel ungefähr in der Realität aussehen. Durch die Darstellung in VR werden so auch Faktoren ersichtlich, die in einfachen Produktbildern nicht immer demonstriert werden können, z.B. die Größe des Artikels. Es entsteht ein Netzwerk nach dem Motto "Kunden helfen Kunden". Als beispielhafte Kompensation für den Ersteller des 3D-Modells könnten dann kleine Gutscheine von dem Shop-Betreiber verschenkt werden.

Bei Millionen von Artikeln wäre es immer noch unmöglich, 100% des Produktbestandes zu digitalisieren. Allerdings würde nach dem Pareto-Prinzip[c] mit der Digitalisierung von einer realistisch erzielbaren Menge des Produktbestandes schon der Großteil der meistgekauften und populärsten Produkte in einer 3D-Vorschau zur Verfügung stehen.

1.2 Zielsetzung

Ziel dieser Bachelorarbeit soll es sein, den in der Motivation vorgeschlagenen VR-Shop mit 3D-Modellen zu analysieren, zu entwerfen und letztlich zu implementieren.

Es gibt mittlerweile viele spezialisierte Technologien, die wirkliche Welt in eine virtuelle umzuwandeln. Viele dieser Lösungen haben dabei einen besonderen Fokus auf einfache Bedienung für den Benutzer und nehmen den größten Teil der Arbeit ab. Das fängt schon damit an, dass Kunden oft nicht mehr als ihr jetziges Smartphone brauchen und dann mit einer passenden Applikation Objekte als 3D-Modelle einscannen.

Die Bestrebung soll sein, ein einfaches und günstiges Verfahren zur Erstellung von 3D-Modellen durch Scans von Objekten aus der echten Welt zu finden. Diese sollen dann in einen von Grund auf entworfenen Virtual-Reality-Shop eingepflegt werden, damit ein potentieller Kunde die Möglichkeit hat, Artikel dreidimensional vor der Kaufentscheidung betrachten zu können.

[c] Pareto-Prinzip: Mit 20% des Gesamtaufwandes können bereits 80% der Erträge erzielt werden.

1.3 Aufbau der Arbeit

Der Hauptteil dieser Bachelorarbeit teilt sich in die folgenden drei Kapitel auf:

- Kapitel 2 stellt die theoretischen Grundlagen zum weiteren Verlauf der Arbeit vor. Dabei wird ein besonderer Schwerpunkt auf 3D-Scanning gelegt, um eine geeignete Lösung für die Zielsetzung zu finden. Weiterhin werden die essentiellen Anforderungen an Webshops und VR-Interaktionen vorgestellt.

- Kapitel 3 stellt für den anschließenden Entwurf die konkreten Rollen innerhalb des VR-Shops mit deren relevanten Use-Cases vor (funktionale und nichtfunktionale Anforderungen).

- Kapitel 4 stellt den eigentlichen Entwurf des VR-Shops vor. Dabei werden die spezifizierten Anforderungen aus der Anforderungsanalyse zunächst als System- und Softwareentwurf präsentiert. Abschließend zeigt eine prototypische Implementation die Umsetzung des Entwurfs für Unity und der HTC Vive. Dabei sollen auch etwaige Schwierigkeiten während der Entwicklung aufgeführt werden.

Kapitel 5 stellt abschließend die Ergebnisse dieser Bachelorarbeit zusammenfassend vor, gefolgt von möglichen Ausblicken für das Thema VR und VR-Shops in der Zukunft.

2 Grundlagen

In diesem Kapitel werden die Grundlagen für die späteren Anforderungen im Rahmen dieser Arbeit recherchiert, erörtert und bewertet. Dazu wird zunächst intensiv auf verschiedene Verfahren und Applikationen für 3D-Scanning eingegangen (Abschnitt 2.1) und die wichtigsten Anforderungen an qualitativ ansprechende 3D-Scans vorgestellt (Unterabschnitt 2.1.2). Aus diesen Erkenntnissen wird ein 3D-Scanner für den weiteren Verlauf der Bachelorarbeit ausgewählt (Unterabschnitt 2.1.3).

Anschließend werden die Grundlagen für die Gestaltung und den Aufbau von Online-Shops vorgestellt, welche essentiell für die Überführung in eine virtuelle Umgebung sind (Abschnitt 2.2). Dazu werden auch die grundlegenden Anforderungen an Interaktionen in virtuellen Welten vorgestellt (Abschnitt 2.3).

Das Kapitel wird abschließend zusammengefasst (Abschnitt 2.4).

2.1 3D-Scanning

Das 3D-Scanning ist ein Technologiegebiet zur Erfassung und Rekonstruierung von Szenarien und Objekten aus der realen Welt, um mittels EDV mehr Informationen aus diesen Daten gewinnen zu können. In vielen Fällen ist dabei das gewünschte Produkt dieses Prozesses die Erstellung einer möglichst exakten Nachbildung von eingescannten Objekten, dem 3D-Modell.

Durch die rasante Steigerung an Computerleistung, auch in kleineren Geräten, wurde über die letzten Jahre das Interesse um 3D-Scanning immer präsenter für kleinere Personengruppen. Heutzutage kann jeder Anwender mit relativ wenig Aufwand (sowohl auf finanzieller Ebene als auch durch Erfahrung) qualitativ ansprechende 3D-Modelle von diversen Objekten kreieren.[5]

Es werden dabei zwei konkrete Anwendungsfälle unterschieden: dem Scannen eines Objektes oder dem Scannen der Umgebung.

Ersteres wird zur Erfassung einer konzentrierten Menge an Informationen getätigt, während zweiteres i.d.R. für die Erfassung von Räumen und Flächen eingesetzt wird. Im Rahmen dieser Bachelorarbeit wird ausschließlich Fokus auf das Scanning von Objekten gelegt.

Die Qualität von fertigen 3D-Scans ist von vielen Faktoren abhängig, darunter nachfolgend aufgelistet nur einige:

(a) Äußerliche Einflüsse
(b) Gewähltes Scan-Verfahren
(c) Stand der Technik der eingesetzten Hardware/Software
(d) Korrekte Handhabung der Hardware während des Scans
(e) Optische Eigenschaften des zu scannenden Objektes
(f) Qualität und Quantität der erfassten Datenmenge

Nachfolgend werden diverse Scan-Verfahren beschrieben, verglichen und bewertet, mit dem Ziel, das Scan-Verfahren zu finden, das mit den genannten Punkten am Ehesten konform ist (Unterabschnitt 2.1.1).

Anschließend werden die Voraussetzungen für die in Punkt (a) genannten äußerlichen Einflüsse konkret beschrieben (Unterabschnitt 2.1.2).

Zum Schluss werden verschiedene 3D-Scanner vorgestellt und bewertet, damit ein Scanner gewählt werden kann (Unterabschnitt 2.1.3).

2.1.1 Vergleich von Scan-Strategien

Es gibt keine festgelegten Standards für die Erstellung von 3D-Scans, was zur Folge hat, dass unzählige Strategien und Verfahren verfügbar sind. Oft werden diese für spezielle Anwendungsfälle entworfen oder sind nicht öffentlich verfügbar.

Zusammen mit dem Bestreben, eine einfache und günstige Lösung zu finden, werden somit nur einige ausgewählte Strategien nachfolgend probatorisch beschrieben, die für diese Bachelorarbeit in Frage kommen könnten.

2.1.1.1 Kontaktmessung

Als *Kontaktmessung* wird die konkrete, physische Berührung eines Objektes als Anhaltspunkt zur Bestimmung der räumlichen Lage und somit der Oberfläche des Objektes bezeichnet. Dabei werden spezielle Maschinen eingesetzt, wie *Koordinatenmessgeräte* (Abbildung 2.1), die ihre eigenen Längen kennen und durch berührungsempfindliche Sensoren ab Kontakt mit einem Objekt dessen Höhe bestimmen können.[9]

Abb. 2.1: Koordinatenmessgerät[29]

Solche Maschinen sind sehr präzise, jedoch langsam (durch die nötige physische Bewegung) und vor allem sehr teuer. Außerdem ist die direkte Berührung von Objekten nicht immer unkritisch, beispielsweise bei Objekten mit elastischer oder zerbrechlicher Oberfläche. Dieses Messverfahren kommt deshalb nicht für die weitere Verwendung innerhalb dieser Bachelorarbeit in Frage.

2.1.1.2 Flugzeitmessung

Durch *Flugzeitmessung* (engl. "Time-of-Flight", kurz "TOF") wird ein zu scannendes Objekt über die physikalischen Gesetze der Lichtgeschwindigkeit räumlich bestimmt.

Eine Lichtquelle – i.d.R. im Infrarotbereich, damit diese unsichtbar erscheint – wird auf das Objekt projiziert. Das von dieser Bestrahlung reflektierte Licht wird von demselben Gerät wieder erfasst. Da sich eine Lichtquelle mit der konstanten und bekannten Lichtgeschwindigkeit c ausbreitet, entsteht eine für den Menschen nicht bemerkbare aber dennoch existierende Zeitverzögerung t zwischen dem Verschießen der Photonen bis zum Auftreffen auf dem Lichtsensor.

Die daraus entstehende Strecke s für einen Punkt ergibt somit durch: $s = ct/2$[31]

Der wesentliche Vorteil dieses Verfahrens ist die Kompaktheit und die schnelle Berechnung. Es wird nur ein Gerät benötigt, das auf einem fixierten Standort montiert ist und

eine kleine, vorgegebene Menge an Punkten berechnet. 3D-Scans können in annähender Echtzeit erfasst werden.

Der Nachteil dieses Verfahrens liegt auf der Hand: Lichtgeschwindigkeit ist so schnell, dass spezielle Hardware benötigt wird, die mit superhochfrequenten Lichtsensoren ausgestattet sind. Je präziser dieser Sensor ist, desto präziser ist der resultierende 3D-Scan. Gerade wenn feine Konturen in dem Scan erhalten bleiben sollen, ist dieses Verfahren – auch schon wegen der Hardwarekosten – ungeeignet.

Abbildung 2.2 zeigt eine mit einer TOF-Kamera aufgezeichnete Person. Die Konturen der Person sind zu erkennen und für Echtzeitmessungen geeignet, aber nicht für detaillierte Aufnahmen, da feine Details verloren gehen. Zudem ist die der Scan einseitig – die Rückseite der Person wurde nicht erfasst.

Abb. 2.2: *TOF-Kamera* im Einsatz[7]

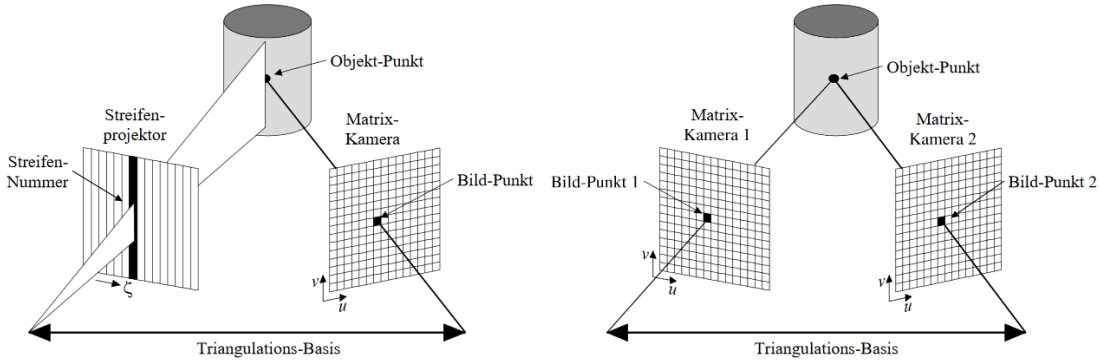
Eine verbreitete Implementation dieser Messtechnik ist die *Microsoft Kinect*, einer TOF-Kamera für die *Microsoft Xbox 360* und *Microsoft Xbox One*. Sie wird dort als Eingabegerät für Videospiele über Körperbewegungen eingesetzt. Da so die exakte Erhaltung einzelner Konturen nicht so entscheidend ist wie die effiziente und schnelle Bestimmung der spielenden Zielperson im Raum, ist das TOF-Messverfahren dort geeignet.

2.1.1.3 Triangulation

Als *Triangulation* wird eine allgemeine Messtechnik bezeichnet, das auf dem mathematischen Prinzip der Trigonometrie beruht und für die Standortbestimmung eines Punktes eingesetzt wird.

Das zu erfassende Ziel wird dabei durch mindestens zwei Messgeräte (“Matrix-Kameras”) so im Raum angepeilt, dass ein Dreieck aufgespannt wird. Dabei sind sowohl der Abstand zwischen den beiden Standorten (Triangulation-Basis) sowie deren erfassende Winkel bekannt, womit die Position des Objektpunktes im dreidimensionalen Raum bestimmt werden kann.[31]

Es wird zwischen *aktiver* und *passiver* Triangulation unterschieden.[31]



(a) Aktive Triangulation [31]

(b) Passive Triangulation [31]

Abb. 2.3: Grundidee von Triangulation und Vergleich zwischen aktiver und passiver Triangulation. Der Abstand beider Matrix-Kameras sowie deren Winkel sind untereinander bekannt, was die Bestimmung des Objekt-Punktes über einfache Trigonometrie ermöglicht. Aktive Triangulation verwendet eine eigene Lichtquelle, passive Triangulation nicht.

- **Aktive Triangulation** (Abbildung 2.3a) verwendet zwei verschiedene Geräte: eine Lichtquelle und einen Bildwandler. Die Lichtquelle projiziert einen gebündelten Strahl – z.B. einen Laser – auf das zu scannende Objekt. Der Bildwandler erfasst daraufhin über einen einen lichtempfindlichen Sensor die Reflexion der Lichtquelle auf dem bestrahlten Objekt. Die Positionen und Winkel der beiden Stationen sind untereinander bekannt. Zusammen mit der Position des erfassten Lichtes auf dem Bildwandler ist somit die Position des eingescannten Punktes im Raum bestimmbar.

- **Passive Triangulation** (Abbildung 2.3b) arbeitet mit zwei gleichen Stationen. Dabei wird keine Lichtquelle verwendet, sondern das natürliche Licht der Umgebung "passiv" erfasst – in diesem Zusammenhang werden deshalb fast ausschließlich Digitalkameras eingesetzt. Durch die Reflexion des Lichtes von dem Objekt können die Kameras somit Korrelationen zwischen den Bildern identifizieren (z.B. durch markante Merkmale wie Kontraste oder vorher platzierte Tracking-Marker). Im Gegensatz zur aktiven Triangulation können dabei eine beliebige Anzahl Stationen vorhanden sein, solange die verwendeten Geräte gleich beschaffen sind.

Da passive Triangulation keine eigene Hardware außer Kameras benötigt, eignet sich dieses Verfahren für den Anwendungsfall und wird nachfolgend näher betrachtet.

Das bekannteste Messverfahren der passiven Triangulation ist die *Photogrammetrie*. Dieses arbeitet mit einer Vielzahl von Fotos, welche so aufgenommen werden, dass zwischen diesen Überlappungen entstehen, wodurch eine räumliche Rekonstruierung des Scan-Ziels ermöglich wird.[31]

Die häufigste Variante ist dabei die *Nahbereichsphotogrammetrie*. Diese wird für 3D-Scannings aus kurzer Distanz eingesetzt wird. Sie ist daher für das Scannen von kleinen Objekten geeignet.

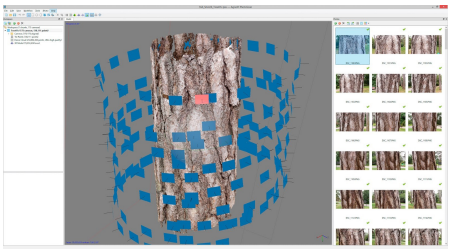


Abb. 2.4: Rekonstruierung eines Baumstamms durch Nahbereichsphotogrammetrie[25]

Abbildung 2.4 zeigt einen beispielhaften Einsatz von Nahbereichsphotogrammetrie zur Rekonstruierung eines Baumstamms. Jedes blaue Rechteck stellt dabei ein Foto dar, dessen Position automatisch bestimmt wurde. Es ist nicht unüblich, dass mehrere dutzend Fotos für adäquate Ergebnisse vorausgesetzt sind.

Nahbereichsphotogrammetrie kalibriert sich selbstständig. Der wesentliche Vorteil daran ist, dass nur eine Kamera benötigt wird. Außerdem ist eine akkurate Erhaltung von Farbinformationen bei den Scans einfacher, da die Informationen direkt aus den Bildern gezogen werden.

Wichtig ist, dass genügend Informationen bei den Aufnahmen erhalten bleiben, um ein möglichst präzises Ergebnis zu erzielen. Aus diesem Grund sollten möglichst hochauflösende, kontrastreiche Digitalkameras eingesetzt und mehrere dutzend Bilder aufgenommen. Allerdings haben die eingebauten Kameras von Smartphones mittlerweile eine Qualität erreicht, die für den Einsatz des Verfahrens vollkommen ausreichend sind.

Da es der Anspruch der Motivation war, potentielle Mitwirkende ohne großen Hardware-Aufwand zu gewinnen, ist die Nahbereichsphotogrammetrie somit ein Verfahren zum 3D-Scanning, welches für diese Bachelorarbeit geeignet ist.

2.1.2 Aufnahmebedingungen für 3D-Scans

Bei 3D-Scannern auf Basis von passiver Triangulation hängt die Qualität der Ergebnisse stark von einer ganzen Reihe äußerlicher Einflüssen ab. Der Benutzer muss für sauberes Scanning wesentliche Anforderungen erfüllen.[1;13]

Drei dieser Anforderungen sind:

- *Hohe Bildqualität*
- *Abhebung vom Hintergrund*
- *Gleichmäßige Beleuchtung*

Diese werden in diesem Abschnitt erläutert.

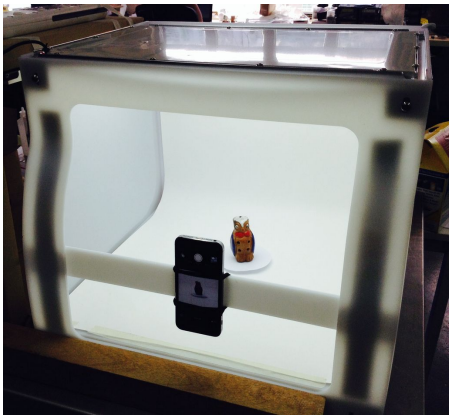


Abb. 2.5: "The Microwave"[13]

"The Microwave" (Abbildung 2.5) zeigt in einem konzeptionellen Hardware-Aufbau zusammenfassend, wie alle drei Anforderungen adäquat berücksichtigt wurden. Ein vergleichbarer Aufbau für ein Aufnahmestudio, welches später zum Scannen der Artikel für den VR-Shop eingesetzt werden soll, wird daher ebenfalls angestrebt.

Missachtung dieser Anforderungen führt in aller Regel zu mangelhaften und unbrauchbaren 3D-Scans. Eine Auswahl dazu wird unter Anhang A vorgestellt.

2.1.2.1 Hohe Bildqualität

Photogrammetrie sucht wiederholende Merkmale, die zwischen den aufgenommenen Bildern mehrfach zu erkennen sind. Somit ist es zwingend erforderlich, eine hohe Bildqualität zu erreichen, damit diese Merkmale in den Aufnahmen gleichmäßig erhalten bleiben.

Dazu ist vor allem die Wahl der Kamera entscheidend. Wichtig ist, dass die Kamera vollständiges Licht-Management unterstützt, damit Bilder mit ausreichender Pixelbeleuchtung aufgenommen werden können. Zudem muss der Sensor eine ausreichende Größe haben, um die Bilder mit genügend Schatten und Kontrasten erfassen zu können. Eine High-End-Kamera führt deshalb nicht automatisch zu besseren Ergebnissen.[1]

Besonders wichtig ist zudem, dass zwischen den Bildern eine einheitliche Belichtung vorhanden ist und der Fokus der Linse sich nicht verändert. Diese Einstellungen sollten deshalb immer manuell vorgenommen werden.[13] In der Regel werden Spiegelreflexkameras empfohlen, die standardmäßig mit großen Sensoren ausgestattet sind. Moderne Smartphones können unter präziser Handhabung ebenfalls eingesetzt werden.

Sollten Smartphones benutzt werden, ist eine gleichmäßige Schärfe schwieriger einzuhalten. Dort tendieren die Bilder leicht zur Verwacklung durch instabile Handhabung. Deshalb empfiehlt sich die Montage an einer festen Position, etwa einem Kamerastativ. Da die Fotos kreisförmig um das Objekt aufgenommen werden müssen, das Smartphone durch ein Stativ aber fixiert steht, muss das Objekt selbst um die eigene Achse rotiert werden. Dies könnte händisch getätigt werden, was aber das Risiko in sich birgt, dass das Objekt aus dem Fokus gerät. Eine bessere Lösung ist de Einsatz einer langsam rotierenden Platte, auf der das Objekt gestellt wird, während die Aufnahmen getätigt werden.[13]

2.1.2.2 Abhebung vom Hintergrund

Eine der wichtigsten Anforderungen für qualitativ ansprechende 3D-Modelle ist, dass diese keine ungewollten Artefakte der Scan-Umgebung beinhalten. Dies geschieht über eine Zuordnung, welche Teile der Fotos das tatsächliche Objekt umfassen und welche nur den Hintergrund darstellen. Für die Möglichkeit, 3D-Modelle automatisiert erstellen zu können, muss ein entstehender Aufwand durch manuelle Entfernung des Hintergrunds zwingend vermieden werden.

Ein erster Schritt ist das Verwerfen von Informationen, bei denen zu wenige überlappende Merkmale erkannt wurden. Es ist aber nicht immer möglich, den kompletten Hintergrund zu vermeiden.

Eine erweiterte Lösung ist, der Anwendung “beizubringen”, was der Hintergrund ist. Es gibt mindestens zwei Arten, wie dies ablaufen kann. Eine Lösung ist die Entfernung einer bekannten Bildmenge aus den Fotos, etwa durch ein vordefiniertes Hintergrundmuster. Ein anderer Ansatz ist die Verwendung eines einfarbigen, weißen Hintergrunds, wodurch die Software nur noch das eigentliche Objekt erfassen kann. Dieser Trick wird in der professionellen Fotografie als “Infinity Wall” bezeichnet.[13]

Neben der Wahl des Hintergrunds ist noch ein Aspekt bei den einzuscannenden Objekten selbst zu berücksichtigen: Es ist nicht möglich, *glatte*, *reflektierende* oder gar *transparente* Objekte qualitativ ansprechend einzuscannen. Dies ist eine Einschränkung der Vorgehensweise von Photogrammetrie – Objekte werden anhand von Kontrasten und der Abhebung zum Hintergrund identifiziert. Dies hat die Konsequenz, dass die Scanning-Software bei nicht ausreichenden Merkmalen widersprüchliche Bildinformationen erkennt (z.B. bei einer Glasflasche).

In diesen Fällen könnte das Objekt mit einer speziellen Flüssigkeit behandelt werden, welche eine undurchsichtige, matte Oberfläche künstlich erzeugt. [22]

2.1.2.3 Gleichmäßige Beleuchtung

Der Baumstumpf in Abbildung 2.6 erscheint zwar auf dem ersten Blick erfolgreich eingescannt zu sein, da dieser sauber und hochauflösend dargestellt wird. Die Beleuchtung des Objektes zur Aufnahme der Fotos war jedoch das natürliche Licht der Sonne bei klarem Himmel aus einem Winkel. So entstand auf der linken Seite des Stammes Schatten, welcher fester Bestandteil der Textur geworden ist und nur durch aufwändiges Nachbearbeiten entfernt werden könnte.

Abb. 2.6: Einseitige ("eingebrannte") Schatten als Konsequenz ungleichmäßiger Beleuchtung (Quelle: Eigene Arbeit)

Unter Umstände sind solche "eingebrannte" Schatten kein Problem, z.B. wenn das Modell die natürlichen Umstände des Objektes beibehalten soll. Der VR-Shop soll aber Artikel darstellen, welche aus allen Winkeln eine gleiche Inspektionsqualität bieten. Für gute Scans wird somit nicht nur ausreichende, sondern vor allem auch *gleichmäßige* Beleuchtung erwartet.

Dafür können z.B. Lichtwannen (auch Softboxen genannt) eingesetzt werden. Dies sind Schweinwerfer, welche mit Papier oder weißem Stoff das geworfene Licht gleichmäßig verteilen (Diffusion). Sollte es sich um ein unbewegliches Objekt außerhalb handeln, wie der Baumstamm, ist eine Kompromisslösung das Scanning an einem bewölkten Tag, da eine verdeckte Sonne schwächere Schatten wirft. [25]

2.1.3 Bewertung verschiedener Applikationen für 3D-Scans

In den vorherigen Abschnitten wurde *Nahbereichsphotogrammetrie* als Scanning-Verfahren gewählt und dessen Anforderungen für qualitativ ansprechende Ergebnisse vorgestellt. Dazu wird nun eine Applikation auf Basis dieser Technologie gesucht, die für den Anwendungsfall des späteren VR-Shops adäquat ist.

In diesem Abschnitt werden dazu drei 3D-Scanner vorgestellt und bewertet. Dabei werden folgende Aspekte berücksichtigt:

- Bedienbarkeit, Komplexität und Intuitivität
- Qualität der 3D-Modelle (ohne manuelle Nachbearbeitung)
- Hardware-Anforderungen
- Kosten

Besonders der letzte Punkt – Kosten – bedarf eine Erläuterung.

Abbildung 2.7 zeigt einen industriellen, professionellen 3D-Scanner, welcher mehrere Tausend Euro kostet. 3D-Scanning ist, unabhängig von dem dafür gewählten Verfahren, ein sehr komplexer Vorgang. Aus diesem Grund ist über die letzten Jahre ein eigener Markt für spezialisierte Hardware entstanden.

Abb. 2.7: Artec Eva 3D Scanner (Gesamtpreis: 13.700)€[19]

Proprietäre 3D-Scanner wären die ideale Lösung für den gewünschten Anwendungsfall – sie sind einfach zu bedienen und bieten sehr gute Resultate. Allerdings sind sie für einen normalen Anwender zu teuer.

Aus diesem Grund werden nur Software-3D-Scanner vorgestellt, also solche, die keine eigene Hardware erfordern.

2.1.3.1 3DScann

3DScann[a] ist eine App zur automatischen Erstellung vom 3D-Modellen auf Smartphones. Dabei werden sämtliche Schritte – vom Aufnehmen der Fotos bis hin zum Berechnen und

[a] http://www.3dscann.co.uk

schlussendlich Exportieren des fertigen Modells – auf dem Handy getätigt.

Die App finanziert sich durch ein "Freemium"-Modell, d.h. einfache 3D-Modelle mit geringerer Auflösung können unbegrenzt kostenlos erstellt werden – höhere Exportqualität erfordert ein monatliches Abonnement.

Das Scanning-Verfahren basiert auf sogenannten "Tracking-Points" (Abbildung 2.8), also kontrastreiche Punkte in der Szenerie, durch die 3DScann jederzeit weiß, wo sich die Kamera im Raum befindet. Die Farbgebung der Punkte visualisiert die Genauigkeit des Scans.

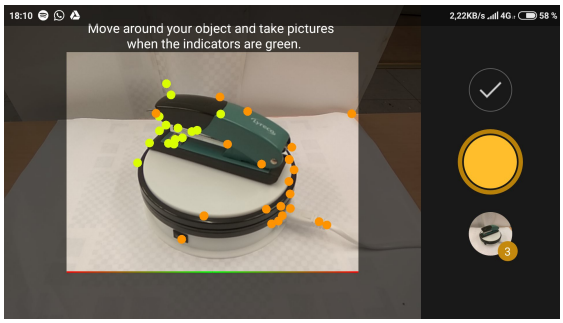


Abb. 2.8: 3DScann während der Aufzeichnung eines Heftgerätes. (Quelle: Eigene Arbeit)

Als notwendige Voraussetzung der App wird vor allem gefordert, dass die Kamera sich auch tatsächlich bewegt und nicht das zu scannende Objekt selbst. Für größere Scans – z.B. Räume – ist dies kein Problem. Da aber für den gewünschten Anwendungsfall des Shops Artikel eingescannt werden, also kleine Objekte, sind bei eigenen Test-Scans die fertigen Modelle fast gänzlich zur Unkenntlichkeit verkommen. Zudem hat es keine Funktion zur automatischen Entfernung von Hintergründen.

2.1.3.2 Qlone

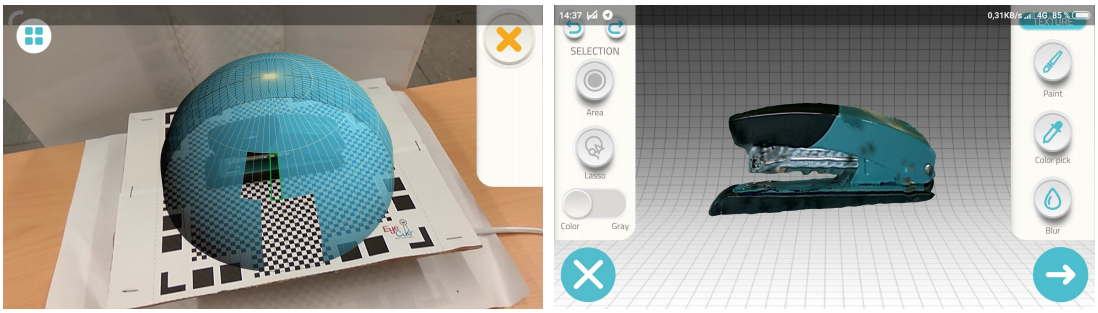
Qlone[b] ist eine App zur automatischen Erstellung vom 3D-Modellen auf Smartphones. Anders als 3DScann finanziert sich die App nicht durch höhere Qualität für erstellte Modelle gegen Geld, sondern erstellt alle Modelle innerhalb der Anwendung bereits in der höchstmöglichen Qualität.

Stattdessen werden für die weitere Benutzung der Modelle – also Exporte für andere Anwendungen – "Credits" benötigt, welche gegen Geld erworben werden können.

[b] https://www.qlone.pro

Die Arbeitsweise der App verfolgt einen fundamental anderen Ansatz als 3DScann, beruht aber auf dem Prinzip der Photogrammetrie und der Prozess findet immer noch auf dem lokalen Gerät statt. Anstatt die Kamera physisch um das zu scannende Objekt herum zu bewegen, wird stattdessen eine proprietäre Unterlage benötigt, die der Hersteller kostenlos zum Ausdrucken auf seiner Website zur Verfügung stellt. Diese Unterlage wird dann auf eine ebene Fläche gelegt und darauf wiederum das zu scannende Objekt.

Die Kamera des Smartphones erkennt die Unterlage und projiziert über *Augmented Reality (AR)* eine blaue Halbkugel (Abbildung 2.9a). Für den Scanprozess setzt sich dieser Körper dann aus dutzenden kleinen Rechtecken zusammen, die alle mit der Kamera einmal "gesehen" werden müssen, damit das Modell erstellt werden kann. Dabei ist es dem Programm egal, ob die Kamera – wie bei 3DScann – um das Objekt herumgeführt wird oder ob die Kamera fix justiert ist.



(a) Qlone während des 3D-Scannings (b) Das mit Qlone eingescannte 3D-Modell

Abb. 2.9: Screenshots von Qlone während und nach der Aufzeichnung eines Modells. Die proprietäre Matte liegt gut sichtbar zur Kamera, damit eine blaue AR-Halbkugel projiziert werden kann. Im Ergebnis sind Konturen und Farbgebung gut zu erkennen, aber Texturen wirken verwaschen und die Form ist an metallischen Stellen unsauber. (Quelle: Eigene Arbeit)

Für die Bedienung hat sich Qlone von allen Anwendungen mit Abstand hervorgehoben. Einerseits ist das Interface sehr intuitiv und andererseits erleichtert die Benutzung von AR den Scanvorgang erheblich. Damit Details nicht verloren gehen, unterstützt und empfiehlt die App, dass jedes Objekt ein zweites Mal – um 90° auf die Seite gedreht – gescannt wird. Gerade die Unterseite von Objekten wird ansonsten bei Scans oft vergessen.

Die Qualität der Modelle ist aber bestenfalls durchschnittlich; grobe Umrisse und Konturen bleiben erhalten, aber die Texturen wirken verwaschen (siehe das Heftgerät Abbildung 2.9b).

2.1.3.3 3DF Zephyr

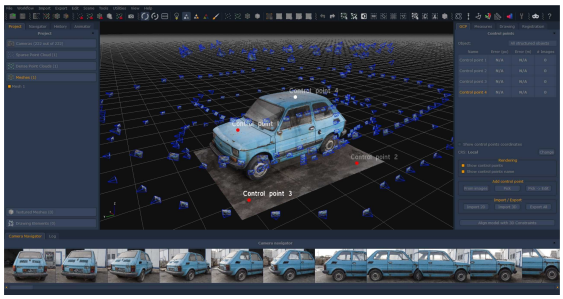


Abb. 2.10: Screenshot von 3DF Zephyr während der Bearbeitung eines Modells.[1]

3DF Zephyr[c] (Abbildung 2.10) ist eine kommerzielle 3D-Modellierungssoftware, die in mehreren Ausführungen verfügbar ist. Die Gratisversion unterstützt bis zu 50 Fotos, für weitere Fotos werden Kaufversionen benötigt.

Außer der Photogrammetrie ist die Funktionsweise dieser Software fundamental anders als bei den bisher genannten Scannern. Es handelt sich um keine Smartphone-App, sondern eine Software für Windows PCs. Sämtlicher Berechnungsaufwand wird auf die Grafikkarte ausgelagert – der Prozess wird erheblich beschleunigt. Als Nachteil ergibt sich aber dadurch mehr Aufwand für den Benutzer, da dieser nun die Bilder einer Kamera transferieren muss.

Als Grundvoraussetzung gilt weiterhin, Fotos aus vielen Perspektiven um das Objekt herum aufzunehmen – 3DF Zephyr bietet hier zusätzlich die Funktion an, aus einem Video die wichtigsten Bilder zu extrahieren, um den Aufnahmeprozess zu beschleunigen. Ein Video erleichtert vor allem den Austausch der Daten, da nur eine Videodatei zum Computer übertragen werden muss, statt mehrerer dutzend Fotos.

Es gibt viele Werkzeuge zur manuellen Nachbearbeitung der 3D-Modelle. So kann problemlos ein ungewollter Hintergrund und vereinzeltes Rauschen von unsauberen Aufnahmen entfernt werden. Das beschleunigt auch den Render-Prozess.

Das Programm agiert in mehreren Phasen, welche manuell vom Nutzer initiiert werden. Zunächst wird eine einfache Punktwolke berechnet (ca. 6.000 Punkte), welche einen ersten Eindruck bietet, ob die Fotos ausreichend für ein gutes Modell sind. Falls ja, wird anschließend eine dichtere Punktwolke berechnet (ca. 1.000.000 Dots), welche dann als Grundlage für die Erstellung ein Dreiecks-Mesh mit hochauflösenden Texturen dient.

[c] https://www.3dflow.net/3df-zephyr-pro-3d-models-from-photos

Allgemein lässt sich 3DF Zephyr als die beste Lösung für die Problemstellung dieser Bachelorarbeit bei den Punkten Qualität und Funktionalität erklären, doch sie ist aufgrund des hohen Preises nur für professionelle Umfelder zu gebrauchen. Die Free-Version bietet zwar die meisten Funktionen an, doch haben sich in eigenen Tests 50 Bilder nur mangelhaft für gute Ergebnisse herausgestellt. Zudem ist die Software nicht so intuitiv zu bedienen wie z.B. Qlone.

2.1.3.4 Bewertung der Applikationen und Entscheidung

Am Anfang dieses Unterkapitels (Unterabschnitt 2.1.3) wurden vier Anforderungen für die Auswahl des 3D-Scanners für diese Bachelorarbeit genannt. Diese sollen nun tabellarisch für die vorgestellten Anwendungen bewertet und verglichen werden:

Applikation	Bedienbarkeit	Qualität	Hardware	Kosten
3DScann	∅	−	+	+
Qlone	+	∅	+	∅
3DF Zephyr	∅	+	−	−

Tabelle 2.1: Tabelle zur Bewertung der drei getesteten Applikationen für 3D-Scanning. (Erklärung der Bewertung: + Gut, ∅ Neutral, − Schlecht)

3DScann ist etwas in der Bedienbarkeit eingeschränkt, da es physische Bewegung des Gerätes erfordert. Die 3D-Modelle besitzend nach dem Scannen zudem einem nicht automatisch entfernten Hintergrund.

3DF Zephyr ist eine umfangreiche Software mit sehr vielen Tools und erstellt hochauflösende 3D-Modelle. Dafür ist sie aber für qualitative Ergebnisse sehr teuer und vor allem schwer für einen Anfänger zu erlernen. Es wird außerdem zusätzliche Hardware in Form eines leistungsstarken PCs benötigt.

Qlone lässt sich somit im Schnitt als die geeignetste Software im Rahmen dieser Bachelorarbeit erklären. Zwar sind die Modelle teilweise sehr verwaschen und leiden unter Rauschen, aber dafür wird der Hintergrund durch die proprietäre Unterlage automatisch entfernt und die Anwendung ist für einen Einsteiger sehr intuitiv gestaltet. Es wird keine sonstige Hardware benötigt, da es eine Smartphone-App ist; lediglich die Unterlage muss ausgedruckt werden. Die Kosten sind vergleichsweise gering.[d]

[d] Der Hersteller von Qlone bietet für studentische Lehrzwecke die Exporte kostenlos auf iOS an, wovon in dieser Bachelorarbeit Gebrauch genommen wurde.

2.2 Grundlegende Anforderungen an Online-Shops

Abb. 2.11: Prozessbereiche des Online-Einkaufs. Der Fokus dieser Bachelorarbeit steht dabei vor allem auf den eingefärbten Bereichen: Produktsuche und der Sondierung von Alternativen, der Verwaltung eines Warenkorbs sowie der Prüfung & Empfehlung von Produkten.[21]

Online-Shops bzw. Webshops sind ein Teilgebiet des elektronischen Handels (E-Commerce). Sie gelten mittlerweile als eines der größten Wirtschaftsbereiche des Internets und unterliegen somit einiger Anforderungen für eine erfolgreiche Umsetzung. Diese verstehen sich als ubiquitär, d.h. die Konzepte sind für alle Webshops gleich zu behandeln.[21]

Nach Oliver Meidl[21] teilt sich der Online-Einkauf in drei wesentliche Geschäftsprozesse auf (Abbildung 2.11): *Pre-Sales*, *E-Sale* und *After-Sales*. Diese enthalten diverse Unterphasen, von denen die für den VR-Shop wichtigsten nachfolgend vorgestellt werden:

1. **Pre-Sales** (Vorverkaufsphase) fassen alle Prozessbereiche zusammen, die für die spätere Kaufentscheidung des Kunden relevant werden. Die Annahme ist dabei, der Kunde mit einem "Problem" den Shop aufsucht, also dem Bedarf, eine Lösung in Form eines passenden Produktes zu finden und zu erwerben. Damit dies möglich ist, muss dieses Problem schnellstmöglich erkannt werden. Die Lösung ist normalerweise das Vorhandensein einer geeigneten Suchfunktion, welche zielsicher Produktvorschläge aufzeigen kann.

2. **E-Sale** (elektronischer Verkauf) beschreibt den eigentlichen Kaufprozess. Ein Webshop ist, vereinfacht betrachtet, die digitale Überführung eines Einkaufserlebnisses von einem normalen Supermarkt bzw. Warengeschäft in eine digitale Umgebung. Aus einer großen Produktsammlung sucht sich ein Kunde eine Menge von Waren aus, die gekauft werden sollen. Das gebräuchlichste Modell ist dabei ebenfalls aus einem analogen Markt abgeleitet, dem Warenkorb. So hat der Kunde die Möglichkeit, Artikel vorab zu selektieren, um sie abschließend als Kollektiv erwerben zu

können. Es können mehrere Artikel des gleichen Produktes in den Warenkorb gelegt werden. Auch ist es möglich, Produkte jederzeit aus dem Warenkorb wieder zu entfernen. Die Artikel des Warenkorbs werden abschließend an der Kasse als Kollektiv bezahlt. Sobald der Auftrag aufgegeben wurde, womit eine Geschäftsvereinbarung zwischen Kunden und Anbieter entstanden ist, wird die Bestellung bearbeitet und versandt.

3. **After-Sales** (Nachkaufphase) sind eine abschließende Ansammlung von Prozessen, die rein optional in Anspruch genommen werden können. Nach Eintreffen der versandten Artikel hat der Kunde die Möglichkeit, etwaige Probleme mit der Bestellung zu reklamieren oder, im positiven Falle, die Bestellung positiv zu bewerten und somit das Interesse für weitere Käufer zu steigern.

Die in Abbildung 2.11 eingefärbten Bereiche spiegeln die wesentlichen Anforderungen für den später zu entwickelnden VR-Shop dar. So soll der Vordergrund vor allem auf der Interaktion mit der virtuellen Oberfläche liegen. Es muss ein ansprechendes Design geschaffen werden, mit dem Produkte gesucht, verglichen und schließlich zur Kaufentscheidung in einen Warenkorb gelegt werden können. Zur Unterstützung bei der Kaufentscheidung werden 3D-Modelle bereitgestellt, die von anderen Kunden in der letzten Phase im Rahmen einer Empfehlung erstellt wurden. Gerade positive Verkaufserlebnisse können zur Weiterempfehlungen führen.[21, S. 104]

Um eine effiziente und erfolgreiche Webpräsenz einer Einkaufsplattform zu erstellen, müssen zur konkreten Oberflächengestaltung diverse Aspekte berücksichtigt werden. Die Plattform muss u.a. schnell, visuell ansprechend und einfach zu bedienen sein und die Artikel selbst sollten möglichst viele Information bereitstellen, wie z.B. eine detaillierte Produktbeschreibung und Bilder oder Videos.[14]

Bei jedem Schritt muss der Kunde – beginnend ab der Sondierung bis hin zur eigentlichen Aufgebung der Bestellung – durch das Interface begleitet werden, um den Kaufprozess so einfach wie möglich zu gestalten. Es darf keine Abneigung gegenüber der Plattform aufgrund eines unseriösen Erscheinungsbildes oder einer frustrationsbehafteter Bedienung entstehen.

2.3 Intuitive Interaktion in virtuellen Welten

Virtuelle Welten sind ein Teilgebiet der Human-Computer-Interaction (HCI), deren Ziel es ist, unter Echtzeitbedingungen einen Informationsaustausch zwischen einem Anwender und einer virtuellen Umgebung zu vermitteln.[10] Als VR-Welten werden die wahrnehmbaren Eindrücke in VR bezeichnet, welche zusammen die Illusion der Anwesenheit in der Simulation ermöglichen (Immersion).[11]

Die grundsätzlichen Anforderungen entstammen ursprünglich aus der 3D-Computergrafik bzw. der Erstellung von Videospielen, wo ein hoher Wert auf optisch ansprechende Einzelbilder gelegt wird. Um aber eine möglichst interaktive Interaktion innerhalb von virtuellen Welten bereitstellen zu können, müssen zwei weitere Anforderungen zwingend berücksichtigt werden, welche nachfolgend erklärt werden: *Echtzeitfähigkeit* und *Interaktivität*.[16]

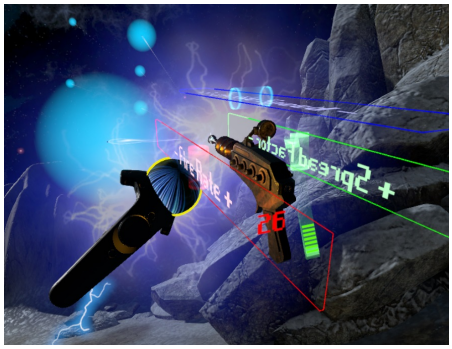


Abb. 2.12: Screenshot zu “Moderne Interaktion in Virtuellen Welten” (Quelle: Eigene Arbeit)

Abbildung 2.12 zeigt einen Screenshot eines prototypischen VR-Spiels mit dem Thema “Moderne Interaktion in Virtuellen Welten”, ursprünglich erstellt für die *Nacht des Wissens 2017*. Ziel war die Entwicklung eines intuitiv erlernbaren VR-User-Interfaces in einem Videospiel. Dabei wurde ein besonderer Schwerpunkt auf die Berücksichtigung der Anforderungen für intuitive Interaktion in der VR-Welt gelegt.

2.3.1 Echtzeitfähigkeit

Echtzeitfähigkeit fasst all die Anforderungen zusammen, welche überhaupt die optische Illusion einer virtuellen Welt ermöglichen. Unser menschliches Auge verlangt, dass alle Bewegungen und erfassten optischen Eindrücke zeitgleich mit dem tatsächlichen, zeitlichen Auftreten der Informationen stattfinden. Einfacher ausgedrückt, wenn eine Person den Kopf neigt, muss sofort das Bild im Auge schräg erkannt werden. In der echten Welt ist dies ein natürlich gegebener Ablauf – das menschliche Auge erfasst durch die Netzhaut alle Lichtimpulse direkt.

Bei der Entwicklung von Anwendungen in VR ist dieser Aspekt nicht mehr gegeben. Ein HMD wird auf dem Kopf des Anwenders fixiert und isoliert dessen Augen vollständig von der Außenwelt, womit alle optischen Sinneswahrnehmungen ab diesem Zeitpunkt durch eine virtuelle Projektion ersetzt werden. Dies hat zur Folge, dass eine Kopfneigung während des Tragens eines HMDs auch eine Neigung der projizierten Welt innerhalb der Linsen der Brille zur folge haben muss. [6]

Abgesehen von der Neigung und Verschiebung des Bildes ist aber ein weiterer Punkt mindestens genauso kritisch für ein fließendes, immersives Erlebnis: die Latenz. Diese ergibt sich aus der zeitlichen Differenz zwischen Eingang von Information bis zum fertig gerenderten und somit wahrgenommenen Bild innerhalb des HMDs.

Im natürlichen Fall, d.h. ohne Headset, existieren Latenzen nicht, da unser Gehirn wahrgenommene Lichtimpulse sofort verarbeitet. Idealerweise ist eine Latenz in der virtuellen Welt ebenfalls nicht vorhanden, was aber aufgrund von Hardware-Beschränkungen und Berechnungsaufwand unvermeidlich ist. Deswegen muss der Wert durch geringe Berechnungszeit der Bilder so gering wie möglich gehalten werden. [2;3;6]

Als erlaubte Maximalzeiten für VR-Latenzen sei folgendes Zitat gegeben:

> [...] *more than 20 ms is too much for VR and especially AR, but research indicates that 15 ms might be the threshold, or even 7 ms.* — Michael Abrash (2012) [2]

Überschreitungen der Latenzgrenze stören nicht nur die Immersion in der virtuellen Welt, sondern wirken sich vor allem auch auf das Wohlbefinden des Anwenders aus. Die Diskrepanz zwischen Eingabe von Informationen und erfasstem Bild führt zu Gleichgewichtsstörungen des natürlichen Körpers, was zu Schwindelgefühlen oder im schlimmeren Fällen gar Übelkeit führen kann. [6] Dieses Phänomen wird im Allgemeinen als “VR-Krankheit” oder “Cybersickness” bezeichnet. [17]

Da die Gegebenheiten der Hardware zur Laufzeit einer VR-Applikation nicht veränderbar sind (die Auflösung der VR-Brille und die Rechenleistung von CPU und GPU), kann eine adäquate Latenz ausschließlich durch Begrenzung des Berechnungsaufwandes erzielt werden.

2.3.2 Interaktivität

Interaktivität bedeutet zusammenfassend, dass ein Anwender einen spürbaren Einfluss auf die virtuell dargestellte Welt hat. 3D-Objekte innerhalb der VR müssen eine direkte Interaktion zulassen, um so interagierbar zu sein.[16] Dabei gibt es keine universelle Lösung zu den Interaktionsmöglichkeiten – für jede VR-Anwendung muss der Umgang an den Anwendungsfall der Applikation angepasst werden.

Zum besseren Verständnis sei der Vergleich zu einer herkömmlichen 2D-Anwendung mit einer Maus gegeben. Trotz einer theoretisch unbegrenzten Menge an Auswahl- und Steuerungsmöglichkeiten innerhalb solch einer Anwendung, ist die Art der Interaktion in einem festen, bekannten Spektrum definiert, da ein Mauszeiger sich nur innerhalb der Bildschirmoberfläche bewegen kann und Aktionen nur durch wenige Tasten an dem Mausgerät ermöglicht werden.

VR-Welten bieten dagegen durch die höhere Menge an Freiheitsgraden – insgesamt sechs – eine wesentlich größere Menge an Umgangstechniken an. Das kann bei einem ungeübten Anwender oder einer ungeeigneten Gestaltung der Interaktionen zur Überwältigung führen. Viele VR-Anwendungen legen deshalb den Fokus auf solche Interaktionstechniken, die die meisten Anwender entweder bereits aus Desktop-Anwendungen kennen oder direkt aus der echten Welt kommen.[10]

Drei wesentliche Konzepte kommen zur korrekten Gestaltung von VR-Interaktivität auf: *Selektion*, *Manipulation* und *Navigation*.[10] Diese werden nachfolgend werden erklärt.

2.3.2.1 Selektion

> Selektion *bedeutet, dass der Nutzer einen Punkt, eine Fläche oder ein Volumen in der virtuellen Welt bestimmt* [...] *oder eine für ihn semantisch bedeutsame Teilmenge dieser Welt auswählt* [...] — Ralf Dörner et al. (2013)[10, S. 160]

Einfach formuliert bedeutet diese Anforderung, dass jede interaktive VR-Anwendung mindestens eine Technik benötigt, um Objekte oder Schaltflächen zielsicher anwählen zu können. Ein direkter Vergleich ist der Klick auf einer Maustaste, während sich der Mauszeiger auf einer Schaltfläche oder Datei befindet.

Eine gebräuchliche Interaktionstechnik für Selektionen agiert über ein Zeigegerät. Dies kann in speziellen Fallen der tatsächliche Zeigefinger des Menschen sein, ist aber häufig

eine sogenannte 3D-Maus – ein Controller in der Hand des Anwenders, dessen Position und Ausrichtungswinkel (Rotation) im Raum bestimmt und somit möglichst präzise in der VR-Welt nachgestellt werden kann.[10]

Sobald die Position des Gerätes bestimmt ist, folgt als nächster Schritt die eigentliche Auswahl von Zielen. Dazu wird eine digitale Kollisionserkennung benötigt, die eine Überschneidung mit dem Zielobjektes bestimmen kann. Da ein Zeigegerät räumlicher Einschränkungen unterliegt, was sich bei distanzierten Objekten problematisch bei der Auswahl herausstellen würde, werden zusätzlich noch Lösungen benötigt, weiter entfernte Interaktionen zu ermöglichen. Dazu werden Zeigegeräte oft wie Fernbedienungen behandelt: Von der Spitze des Gerätes ausgehend wird ein langer, dünner Kollisionskörper befestigt (welcher oft als Laserpointer dargestellt wird), womit eine Art unendlich großer Zeigestock entsteht. Dieser kann, auch auf Distanz, präzise die Zielobjekte auswählen.

Als zusätzliche Unterstützung für den Anwender empfiehlt sich ein Feedback darüber, dass ein Objekt tatsächlich selektiert wurde. Das kann z.B. durch farbliches Hervorheben des Objektes durch eine Umrandung geschehen oder auch durch Vibration des Zeigegerätes (haptisches Feedback).[10]

2.3.2.2 Manipulation

> Manipulation *von Objekten in einer virtuellen Welt definieren wir als interaktive Änderung von den das Objekt charakterisierenden Objektparametern* [...]
>
> — Ralf Dörner et al. (2013)[10, S. 165]

Manipulationstechniken werden benötigt, um einen Einfluss auf die virtuelle Welt zu haben und so mit ihr interagieren zu können. Die Konzepte schließen dabei direkt an Selektion an und sollten für ein geeignetes Modell somit auch gegenseitig aufbauend implementiert werden, damit ein nahtloser und intuitiv erlernbarer Übergang zwischen Selektion und Manipulation gegeben ist.

Im speziellen Anwendungsfall von VR wird eine sogenannte *direkte Manipulation* eingesetzt.[32] Diese beschreibt die Eigenschaft, dass eine Interaktion mit einem virtuellen Objekt unmittelbar und kontinuierlich ein Feedback zum Anwender liefert. Es soll keine unnatürliche Verzögerung zwischen Aktionszeitpunkt und visuellem Resultat entstehen, wie es auch beim Greifen und Aufheben eines Objektes in der realen Welt der Fall ist.

Ob die Art der Manipulation dabei eine reale Entsprechung besitzt spielt dabei keine Rolle. So können sowohl möglichst realistische Nachbildungen erstellt werden, als auch "magische Techniken", welche nur in einer VR-Umgebung möglich sind. Die Entscheidung zwischen diesen beiden Ansätzen ist in erster Linie durch den angestrebten Grad an Realismus innerhalb der Anwendung bedingt. Zwei Beispiele zu je einem dieser Ansätze werden nachfolgend aufgeführt:[10]

- **"Virtuelle Hände"** sind realistische Manipulationstechniken zur Interaktion mit Objekten. Dabei wird mit einem größtmöglichen Anspruch an Realismus das Greifen bzw. Berühren und den daraus resultierenden Veränderungen der Position und Rotation des Objektes angestrebt. Die reale Hand des Anwenders wird in der virtuellen Welt abgebildet und interpretiert. Da die Abbildung aus einem natürlichen Umfeld stammt, ist diese Technik sehr einfach zu erlernen.

- **"Zeigegesten"** sind magische Manipulationstechniken mit direktem Anschluss an die zuvor beschriebene Selektion eines Zeigegerätes. Ausgewählte Objekte können so aus einer theoretisch unbeschränkten Distanz gegriffen und bewegt werden. Dadurch entstehen mehr Freiheiten zur Bedienung der Oberfläche, da die Reichweitenbeschränkungen der menschlichen Arme keine Probleme mehr bereiten.

2.3.2.3 Navigation

Navigation besteht, im Gegensatz zu den bisher genannten Interaktionsanforderungen, aus zwei grundsätzlich unabhängigen, aber aufeinander aufbauenden Teilbereichen: *Wegfindung* und *Bewegungskontrolle*.

> *Die* Wegfindung [...] *ist die kognitive Komponente der Navigation und betrachtet auf höherer Abstraktionsebene Analyse, Planung und Entscheidung über Wege in der virtuellen Umgebung.* — Ralf Dörner et al. (2013)[10, S. 168]

Der Definition entsprechend ist eine gute Umsetzung zur Wegfindung stets eine Frage der visuellen Gestaltung der virtuellen Welt. Die Konzepte sind dabei keine Eigenheit von VR, sondern Abwandlungen aus der realen Welt, denn dort ist die Wegfindung eine unterbewusste Alltäglichkeit – einen Passanten nach der Richtung fragen oder das Lesen von Wegbeschreibungsschildern seien dabei nur zwei einfache Beispiele.

Ziel ist die Bildung einer "kognitiven Karte"[10] mithilfe der räumlichen Vorstellungskraft des Gehirns. In der realen Welt dient diese alltäglich zur Orientierung in der Umwelt.

Die Überführung auf die VR-Welt bedient sich dabei vor allem “Landmarken”. Dies sind eindeutige, auffällige und schnell wiederekennbare Objekte innerhalb der virtuellen Welt. Die Orientierung aus einem bekannten Punkt heraus ist eine natürliche Gegebenheit, die auch alltäglich regelmäßig zum Einsatz kommt (z.B. die relative Orientierung in einer Großstadt ausgehend von einem Hochhaus). Weiterhin wird “Routenwissen” zur Unterstützung eingesetzt, wie etwa wegweisende Pfeile oder auffällige Beleuchtung. [10]

Bei jedem Anwender ist die Qualität der kognitiven Karte – sowohl in den Bereichen Umfang, als auch die Geschwindigkeit des Erstellungsprozesses – unterschiedlich beschaffen. Eine geeignete Implementation erfordert deshalb grundsätzlich eine Abschätzung, wie routiniert ein durchschnittlicher Anwender der Zielgruppe sein muss. Die Wahl der Hardware hat ebenfalls einen Einfluss. So ist die direkte Widerspiegelung von menschlichen Bewegungen die fundamentale Idee von HMDs – die Fixierung des Gerätes am Kopf des Anwenders trivialisiert die visuelle Orientierung.

Nach der Wegfindung muss nun die eigentliche Bewegung ausgeführt werden können:

> *Die* Bewegungskontrolle [...] *ist die motorische Komponente der Navigation, d.h. man betrachtet nur die grundlegenden Aktionen, die benötigt werden, damit Position und Orientierung des virtuellen Kameraausschnitts passend verändert werden.*
>
> — Ralf Dörner et al. (2013) [10, S. 169]

Bewegungskontrolle ermöglicht es einem Anwender, von A nach B zu gelangen (Lokomotion). Sie ist von jeder interaktiven Anwendung, auch Computerspielen, ein Hauptmerkmal. Für eine geeignete Bewegungskontrolle versteht sich die Wegfindung dabei als notwendige Voraussetzung, da die Fortbewegung in eine gezielte Richtung in jedem Fall eine räumliche Einordnung benötigt.

Je nach eingesetzter Hardware ist eine natürliche Fortbewegung innerhalb des VR-Bereichs möglich, was aber schnell an Grenzen stoßen kann. Ein Hindernis von z.B. der HTC Vive ist die eingeschränkte Nutzfläche, weswegen nach wenigen Metern das Manövrieren durch Laufen nicht mehr möglich ist. [8]

In diesem Fall sind Alternativen entweder die “magische” Navigation (ähnlich der zuvor beschriebenen magischen Manipulation) oder die Gestaltung der virtuellen Welt muss schlicht auf den begehbaren Bereich beschränkt sein. Die gängigste Möglichkeit der magischen Fortbewegung über die Grenzbereiche hinaus ist die Teleportation. Der Anwender selektiert einen Punkt in der virtuellen Welt und wird unmittelbar zu diesem befördert.

Dies ist einfach zu implementieren und auch einfach zu erlernen, tendiert aber leicht zum Bruch der Immersion durch unrealistische Fortbewegung und kann bei ungeübten Anwendern zusätzlich zur Desorientierung führen.

Von der Navigation durch sukzessive Verschiebung der virtuellen Position pro Intervall, etwa durch gedrückt halten einer Steuerungstaste auf dem Controller, wird abgeraten. Wenn das wahrgenommen Bild eine Fortbewegung vortäuscht, der Anwender aber stillt steht, kann Cybersickness entstehen.[17]

2.4 Zusammenfassung

In diesem Kapitel wurden die Grundlagen vorgestellt, welche für die anschließenden Hauptkapitel benötigt werden. Die Ergebnisse werden nachfolgend zusammengefasst.

Zunächst wurde das Thema *3D-Scanning* ausführlich vorgestellt, mit dem Ziel, ein Verfahren und einen Scanner zu finden, welche für die Zielsetzung der Motivation adäquat sind. *Passive Triangulation* – speziell Nahbereichsphotogrammetrie – hat sich dabei als geeignetes Verfahren herausgestellt. Es folgte eine allgemeine Zusammenfassung der wichtigsten Anforderungen an die Aufnahmeumgebung für qualitativ ansprechende 3D-Scans, wobei sich *hohe Bildqualität*, *Abhebung vom Hintergrund* und *gleichmäßige Beleuchtung* als kritisch erwiesen haben. Ausgehend aus dieser Erkenntnis wurden verschiedene Applikationen für 3D-Scanning auf Basis von Nahbereichsphotogrammetrie vorgestellt und ebenfalls bewertet, wobei die Smartphone-App *Qlone* am besten abschnitt und für den weiteren Verlauf dieser Bachelorarbeit eingesetzt wird.

Nach dem Thema 3D-Scanning wurden grundlegende Anforderungen an Online-Shops vorgestellt, welche aus den drei Phasen *Pre-Sales*, *E-Sale* und *After-Sales* bestehen. Ein besonderer Fokus wurde dabei auf das Konzept des digitalen Warenkorbs gelegt.

Anschließend wurden die Anforderungen für die intuitive Interaktion in virtuellen Welten vorgestellt. Es wurde gezeigt, dass VR-Anwendungen auf jeden Fall die Rahmenbedinungen der *Echtzeitfähigkeit* einhalten müssen und geeignete Lösungsstragegien für *Interaktivität* benötigen werden.

Aus diesen Erkenntnissen kann nun eine geeignete Anforderungsanalyse (Kapitel 3) des VR-Shops erstellt werden, welche als Vorlage für den späteren Entwurf (Kapitel 4) dient.

3 Anforderungsanalyse

In diesem Kapitel werden die Anforderungen spezifiziert, die der VR-Shop später erfüllen soll. Dazu wird zunächst Bezug zu den verschiedenen Akteuren innerhalb des Systems hergestellt (Abschnitt 3.1) und anschließend die konkreten Anforderungen an die eigentliche Anwendung beschrieben – in Abschnitt 3.2 auf funktionaler Ebene (Design-Entscheidungen für die Oberfläche und dessen Bedienung) und in Abschnitt 3.3 auf nichtfunktionaler Ebene (technische Entscheidungen für Performanz und Sicherheit).

3.1 Rollenbeschreibung

Dieser Abschnitt beschreibt die wesentlichen Rollen für die Anwendung, jeweils mit den entsprechenden Aufgaben sowie deren Sicht und Berechtigungen auf das Gesamtsystem. Es werden ebenfalls die Motivationen der Rollen eingegrenzt.

Die Rollen teilen sich in die folgenden zwei bzw. drei Gruppen auf (Abbildung 3.1):

- Anwender:
  A – Kunde
  B – Modellbereitsteller
- Anbieter

Der *Anwender* versteht sich als Sammelbegriff für die zwei untergeordneten Rollen *Kunde* und *Modellbereitsteller*. Tatsächlich handelt es sich um dieselbe Personengruppe, jedoch wurden diese aufgrund von unterschiedlichen Interessen im Rahmen des späteren VR-Shops aufgeteilt.

Alle Rollen sollen im Rahmen dieser Bachelorarbeit vom Autor übernommen werden. Die Rollen werden nachfolgend einzeln beschrieben.

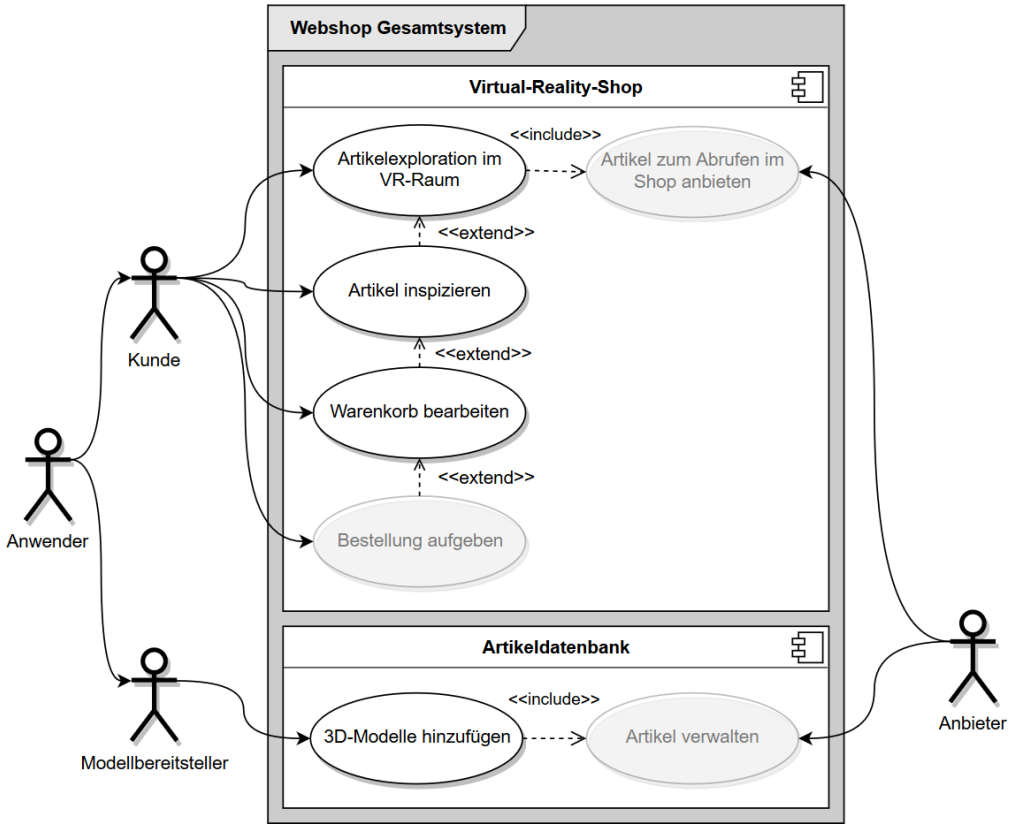


Abb. 3.1: Use-Case-Diagramm der drei Rollen und deren Zusammenspiel im System. Die ausgegrauten Use-Cases stellen nicht näher betrachtete Anforderungen dar, da sie eine Anbindung an einen echten Shop erfordern. (Quelle: Eigene Arbeit)

3.1.1 Anwender A – Kunde

Der Kunde ist ein Anwender des VR-Shops und erhält im Rahmen dieser Bachelorarbeit den größten Fokus, denn er versteht sich als Zielgruppe. Dies ist die Person, welche letztendlich die VR-Brille auf dem Kopf haben wird. Ein Anbieter hat das Bestreben, den Kunden möglichst zum Erwerb von Artikeln innerhalb des VR-Shops zu bewegen.

Dazu soll die Anwendung ohne Handbuch bedienbar sein – der Kunde soll den in der Motivation angestrebten Mehrwert durch die Benutzung von VR direkt spüren können.

Fast alle nachfolgenden Anforderungen um den Kunden spezifiziert. Dies ist damit begründet, dass der Kunde die einzige Rolle ist, welche die virtuellen Aspekte der Anwendung benötigen wird.

3.1.2 Anwender B – Modellbereitsteller

Der Modellbereitsteller versteht sich als eine alternative Form des Kunden, da die Rolle keine eigenen Rechte haben soll bzw. ein Kunde ebenfalls über dieselben Rechte verfügt. Jeder Kunde ist somit gleichzeitig ein potentieller Modellbereitsteller.

Ein Modellbereitsteller soll die Aufgabe haben, 3D-Modelle dem Shop hinzuzufügen, indem dieser die Modelle sowohl erstellt, als auch für den Anbieter hochlädt. Dabei soll außer Acht gelassen werden, wie diese Modelle erstellt werden, denn es soll keine festgelegte Arbeitsweise für diesen Prozess geben. Nach der Motivation liegt der Fokus auf die automatische Generierung von Modellen durch 3D-Scanner, aber es soll auch möglich sein, die Modelle mittels 3D-Modellierungstools manuell zu erstellen. Das einzige Kriterium, das zwingend berücksichtigt werden soll, ist, dass die Modelle in einem für den Shop kompatiblen Format angeboten werden.

Es soll einem Modellbereitsteller nicht möglich sein, neue Artikel in den VR-Shop einzufügen, sondern nur bestehende Artikel um die 3D-Modelle zu ergänzen.

3.1.3 Anbieter

Der Anbieter soll der (alleinige) Besitzer bzw. Bereitsteller der Shop-Software sein. Er soll die Verantwortung für Wartung und Vertrieb übernehmen. Hinzu kommt die Pflege der eigentlichen Inhalte des Shops, also die zum Verkauf angebotenen Artikel. Der VR-Shop ist als eine Erweiterung für einen etablierten Verkäufer gedacht.

Der Anbieter versteht sich als Administrator für das Gesamtsystem und ist aus eigenem Interesse dazu bewegt, dieses zu jedem Zeitpunkt verfügbar zu halten und weiterzuentwickeln. Dazu zählt auch, das Angebot des Shops zu erweitern.

Im Rahmen dieser Bachelorarbeit sollen die Anforderungen des Anbieters nur beschränkt betrachtet und spezifiziert werden, da keine Anbindung an einen echten Webshop vorliegt. Alle in Abbildung 3.1 ausgegrauten Use-Cases des Anbieters sollen lediglich zu konzeptionellen Demonstrationszwecken des VR-Shops simuliert werden.

3.2 Funktionale Anforderungen an die Anwendung

Nachfolgend werden die funktionalen Anforderungen ("FA") der Anwendung beschrieben. Das System soll eine virtuelle Umgebung eines Online-Shops darstellen, mit der Möglichkeit, Artikel in 3D-Form innerhalb dieser Welt betrachten zu können.

Jede funktionale Anforderung ist eine direkte Überführungen der im letzten Kapitel farblich gekennzeichneten grundlegenden Anforderungen an Online-Shops (Abbildung 2.11). Die Use-Cases in Abbildung 3.1 stellen je eine funktionale Anforderung dar. Alle Anforderungen direkt innerhalb des VR-Shops (d.h. die ersten drei) werden aufeinander aufbauen beschrieben – jede Anforderung ist abhängig zu seinen jeweiligen Vorgängern und wird als Erweiterung der bestehenden Funktionalitäten vorgestellt ("«extend»").

3.2.1 Anforderung FA1 – Artikelexploration im VR-Raum

Die einleitende Anforderung beschreibt die wesentliche Überführung der *Pre-Sales*-Phase (Abbildung 2.11), hauptsächlich der Komponente "Produktsuche". Dadurch ergibt sie sich als die umfangreichste Anforderungen des Gesamtsystems, da sie das umsetzten soll, was für den Kunden das Haupt-Interface darstellt.

Es handelt sich um eine Adaption eines regulären Webshops, wie dieser auf einem Browser betrachtet und benutzt wird, übertragen auf die virtuelle 3D-Umgebung. Die wesentlichsten Aspekte sollen dabei aus diesen Webshops identifiziert werden, worauf eine geeignete Abwandlung für den VR-Shop erörtert und spezifiziert werden kann.

3.2.1.1 Suche von Artikeln

Für einen kleine Ansammlung von Artikeln ist es kein Problem, diese alle auf einmal in dem Shop anzeigen zu lassen, z.B. bei Anbietern für sehr wenige Nischenprodukte. Für einen allgemeinen Webshop wäre dies aber eine sehr begrenzte Vorgehensweise, da es Tausende, wenn nicht gar Millionen von Artikeln geben könnte.

Alle Artikel auf einmal anzuzeigen kommt somit allein aus Gründen der Übersichtlichkeit nicht in Frage. Es soll also ein geeignetes System vorhanden sein, durch diese theoretisch große Sammlung an Artikeln möglichst mühelos navigieren zu können.

Freitextsuche per Spracherkennung

Jeder normale Webshop hat am oberen Ende seines Webauftritts eine Gemeinsamkeit: das Suchfeld. In diesem kann nach Belieben des Anwenders eine Freitexteingabe über die Tastatur getätigt werden, um schnell Suchergebnisse zur gewünschten Sammlung an Produkten anhand gezielter Worte zu finden. Ohne diese Möglichkeit der Suche und die Qualität der daraus zurückgelieferten Resultate würden Webshops nicht arbeiten, wie es heutzutage erwartet wird.

Es ist jedoch aus einem ganz einfachen Grund nicht möglich, das Konzept eines Shops in einem Browser direkt zu überführen: Die bereits erwähnte Tastatur zur Freitextsuche kann nicht verwendet werden, während eine VR-Brille getragen wird. Im Umfeld der virtuellen Realität gibt es keine allgemeine Lösung dieser Einschränkung. Deswegen wird eine geeignete Version einer Suchfunktion für VR-Shop essentiell und stellt die erste Interaktionsebene dar, mit der der Kunde konfrontiert wird.

Ansätze in anderen VR-Applikationen unterscheiden sich teils stark. So ist ein naiver Gedanke die Nachbildung einer Tastatur im digitalen Großformat, die durch virtuelle Kollision der einzelnen Tasten mit Zeigegeräten bedient wird. Allerdings ist die Schreibgeschwindigkeit für ungeübte Benutzer sehr langsam und es kann leicht zu Fehleingaben aufgrund mangelnder Haptik kommen, was wiederum Frustration verursachen könnte.

Aus diesem Grund soll die Suche über *Spracherkennung* agieren. Gegenwärtig sind Softwarepakete für Sprache-zu-Text-Konvertierung sehr weit fortgeschritten und können bereits diverse Sprachen – darunter Deutsch – gut erkennen. Da in den meisten HMDs bereits ein Mikrofon integriert ist – darunter die HTC Vive[15] und die Oculus Rift[12] – ist der Hardware-Aufwand für diese Anforderung kein Problem.

Die Spracherkennung soll durch einfache Betätigung eines leicht zu findenden Schalters ausgelöst. Der Text soll während des Sprechens innerhalb des Shops als symbolisches 3D-Textfeld dargestellt werden. Nach einer kurzen Zeit an Stille soll die Eingabe beendet und erkannte Text als Texteingabe zur Suche an den Shop gesendet werden, als wäre er auf einer Tastatur eingegeben worden. Eine gefilterte Auswahl an Artikeln soll anhand der Artikelnamen und der nachfolgend beschriebenen Kategorisierung herausgegeben werden.

Der VR-Shop soll *nur* über eine Spracherkennung die Artikel durchsuchen können – Tastatureingaben dürfen als Ersatzfunktion bei Problemen mit der Spracherkennung möglich sein, sind aber nicht explizit vorgesehen.

Kategorisierung zur leichteren Freitextsuche

Jeder Artikel in jedem normalen Webshop hat mindestens eine zugeordnete Produktkategorie zur Gruppierung ähnlicher Produkte. Diese können zur genaueren Filterung beliebig tief einen Baum an weiteren Unterkategorien aufspannen.

Eine saubere Kategorisierung ist aufwändig, hilft dafür aber bei der zielsicheren Suche von Artikeln. Bei einer Freitextsuche sucht der Anwender oft nicht nach konkreten Artikelnamen, sondern durch gezielte Eingabe eines einfachen Begriffes (z.B. Heftgeräte) nach einer Sammlung mehrerer ähnlicher Artikel, um sich dann für ein Ergebnis zu entscheiden.

Nachfolgend ist ein einfaches Beispiel aufgeführt, um zu demonstrieren, dass selbst bei vermeintlich simplen Produkten ein ganzer Strang an Oberkategorien folgen kann:

Arbeit und Büro → Büroartikel → Schreibtischbedarf → Heftgeräte

Deshalb soll der VR-Shop die Freitextsuche sowohl auf Basis von Artikelnamen als auch von Produktkategorien ermöglichen. Dabei sollen Oberkategorien (z.B. Büroartikel) alle untergeordneten Kategorien (Heftgeräte, Druckerpapier, Kugelschreiber usw.) allgemein mit einbeziehen.

Das Navigieren durch die Kategorien selbst ist zwar auch eine denkbare Lösung, doch findet auch auf Suchmaschinen von Webshops kaum Bedeutung, da die Freitextsuche auch dort die verbreitetste und effektivste Navigationsart zur Artikelsuche ist.

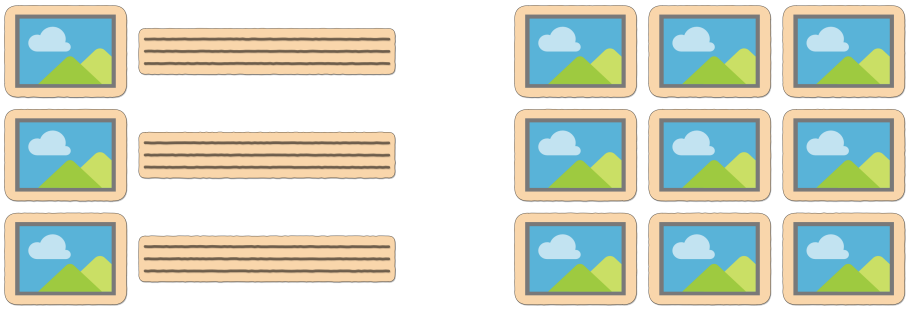
3.2.1.2 Oberflächenpräsentation des VR-Shops

Nachfolgend wird die wesentliche Konzeption der Oberfläche zur Darstellung der Suchergebnisse beschrieben. Damit der Shop auch intuitiv in der virtuellen Welt verständlich ist, muss zunächst herausgestellt werden, wie ein regulärer Webshop am Computer bedient wird und visuell aufgebaut ist.

Dazu werden nachfolgend zwei Konzepte eingeführt und erklärt: Die Artikelwand und dessen einzelne Kacheln, die Artikelmonitore.

Die "Artikelwand"

Das Verfahren zum Anzeigen mehrerer Artikel in einem Webshop ist gängig und wird fast überall identisch eingesetzt: Die Artikel werden zusammen mit ihren wichtigsten Informationen (Name und Preis sowie ggf. Beschreibung und Bewertung) in Kombination mit einem repräsentativen Produktbild für den schnellen Vergleich in einem Raster angezeigt. Die Anordnungsstruktur dieses Raster ist in den meisten Fällen eine von zwei Darstellungsformen: Listenansicht oder Galerieansicht



(a) Listenansicht (b) Galerieansicht

Abb. 3.2: Vergleich von Listenansicht zu Galerieansicht anhand konzeptueller Skizzen. (Quelle: Eigene Arbeit)

- In der Listenansicht (Abbildung 3.2a) sind die Artikel einzeln untereinander sortiert und haben, neben dem Hauptproduktbild, in der Regel mehr Raum für Informationen über den Artikel auf der horizontalen Ebene. Die Listenansicht ist i.d.R. häufiger zu in Webshops sehen, da die Interaktion mit einem Webbrowser von Natur aus vertikal verläuft – es ist eher ungewöhnlich, eine Webseite horizontal zu scrollen.

- Die Galerieansicht (Abbildung 3.2b) ist dagegen als Raster aufgebaut, wobei in jedem Feld ein Produkt angezeigt wird. Somit ist es möglich, mehrere Artikel nebeneinander zu platzieren, damit ein schnellerer Vergleich mit deutlich weniger Scroll-Aufwand möglich ist. Dies geschieht allerdings auf Kosten der Informationen, da die Artikel in der Galerieansicht in den meisten Fällen nur aus einem Bild und den wichtigsten Informationen bestehen.

Einige Webshops bieten die Galerieansicht gar nicht an bzw. nicht mehr, da die vertikale Natur der Listenansicht gerade auch für Smartphones deutlich geeigneter ist. Für den virtuellen Shop besteht dieser Zusammenhang allerdings nicht. Die Entscheidung auf die Listenansicht in einem dreidimensionalen Umfeld wäre nicht nachhaltig, da es nicht nur

einen kleinen zweidimensionalen Bildschirm vor dem Sichtfeld des Kunden gibt, sondern eine ganze Welt, die 360° um den Nutzer herum betrachtet werden kann. Es wäre eine massive Verschwendung von Platz, wenn trotz dieser riesigen Fläche nur ein vertikaler Stapel an Artikeln zu sehen wäre, der zudem noch mühselig durch Scrollen navigiert werden müsste.

Aus diesem Grund soll die Galerieansicht verwendet werden, allerdings in einer Form, die für das 3D-Umfeld aus der Ich-Perspektive geeignet ist und dabei die Vorteile von dieser Ansichtsform mit den Vorteilen der 3D-Welt verbindet. Hierbei werden die einzelnen Artikel, sinngemäß, in rechteckigen Feldern mit großen Produktbildern dargestellt und gleichmäßig verteilt.

Da sich der Anwender aber komplett um seine eigene Achse drehen können soll, muss der Abstand der einzelnen Monitore zu jedem Zeitpunkt gleich sein. Das erfordert also, dass die Artikel nicht planar – "in einer Ebene" – angeordnet sind, sondern rund. Dabei soll sich die Anordnung vertikal ebenfalls vollständig im vertikalen Sichtfeld des menschlichen Auges befinden, damit der Kopf nicht zu sehr nach oben und unten geneigt werden muss.

Ein sinngemäßer Vergleich aus der realen Welt ist eine riesige Wand aus Bildschirmen, die um den Betrachter herum angeordnet sind (Abbildung 3.3). Diese Wand wird im weiteren Verlauf dieser Bachelorarbeit als Artikelwand definiert.

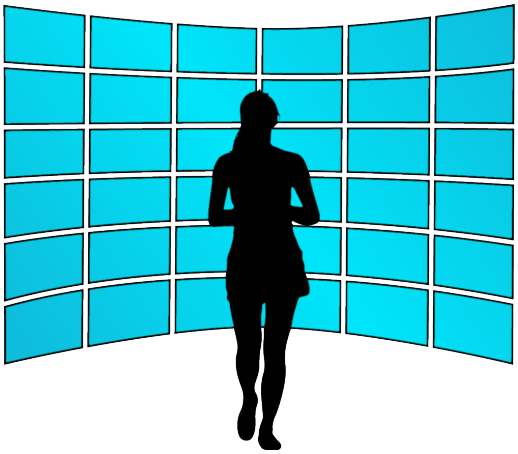


Abb. 3.3: Illustration einer rundlichen Anordnung von Bildschirmen um einen Anwender. Diese Grafik dient als Richtungsweiser für die Realisierung der Artikelwand für den VR-Shop. (Quelle: Eigene Arbeit)

Der "Artikelmonitor"

Bisher wurde die Artikelwand als Einheit beschrieben. Nun geht es um die einzelnen Kacheln innerhalb dieser Wand als solche, also die tatsächlichen Artikel, die dargestellt werden sollen. Diese werden nachfolgend als Artikelmonitore definiert.

Die Artikelmonitore sollen, Abbildung 3.3 entsprechend, einen gewissen Abstand zum Betrachter halten, während sie – vor der Selektion – noch Teil der Artikelwand sind. Zudem sollen sie, auf Grundlage der Galerieansicht, nur eine grobe Zusammenfassung der wichtigsten Informationen des angebotenen Artikels bereitstellen: *Name des Artikels, Preis* und ein *Produktbild*.

Eine Konzeptzeichnung eines Artikelmonitors ist unter Abbildung 3.4 zu sehen.

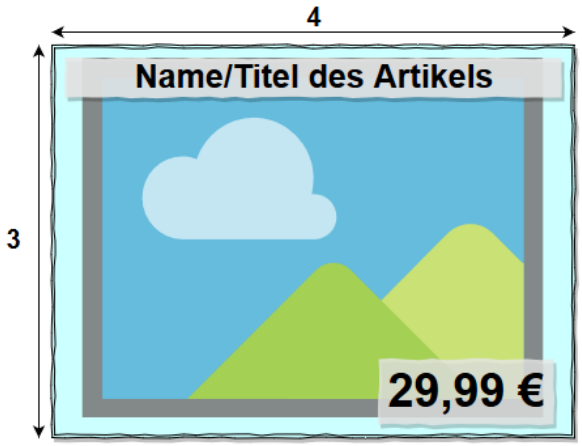


Abb. 3.4: Konzeptzeichnung eines Artikelmonitors mit Position und Größe für Titel und Preis (hier nur als Beispiel auf 29,99 € gesetzt). Das Seitenverhältnis ist als Vorschlag auf 4:3 gesetzt (in Anlehnung an echte Monitore). (Quelle: Eigene Arbeit)

Die ideale Größe der Monitore, Qualität der Bilder und Positionierung von Titel und Preis müssen durch manuelles Anpassen bestimmt werden, sind aber im Wesentlichen entlehnt aus bestehenden Praktiken existierender Webshops in der Galerieansicht. Üblicherweise nimmt das Produktbild den größten Teil des Kastens ein, während sich der Titel über oder unter diesem angezeigt wird. Der Preis ist hingegen häufig unten rechts angezeigt.

Beim Produktbild mag die Annahme naheliegend sein, dass dieses für den VR-Shop nicht notwendig sei. Schließlich ist das Ziel, die Artikel als 3D-Modelle betrachten zu können. Zudem wurde in der Motivation bereits die Gefahr von manipulierten Produktbildern.

Diese Anforderung entsteht aber als indirekte, logische Konsequenz aus der nichtfunktionalen Anforderung "Antwortzeitverhalten" (Unterabschnitt 3.3.3). Die Begründung wird im referenzierten Abschnitt später näher erläutert – zusammengefasst ist es wesentlich schneller, ein Bild aus dem Internet herunterzuladen, als ein komplexes 3D-Modell, das mehrere dutzend Megabyte groß sein könnte. Deshalb sollen nicht sofort die Modelle bei Rückgabe der Suchergebnisse geladen werden, sondern, nur Produktbilder mit den wichtigsten Information.

Oft ist auch ein Schalter für das sofortige Hinzufügen des Artikels in den Warenkorb vorhanden. Diese Funktion soll an dieser Stelle bewusst noch nicht bereitgestellt werden, da sich zunächst die 3D-Modelle angeschaut werden sollen, bevor die Kaufentscheidung getroffen wird. Zudem ist die zielsichere Interaktion durch Zittern der Arme nicht so einfach ist wie bei einer Maus am Desktop-Computer oder dem Display eines Smartphones.

3.2.2 Anforderung FA2 – Artikel inspizieren

Bisher wurde beschrieben, wie die Suchergebnisse des VR-Shops durch eine rundliche Anordnung mehrerer Artikelmonitore in einer Artikelwand dargestellt werden. Dabei handelt es sich allerdings nur um Zusammenfassungen der Suchergebnisse mit einfachen Produktbildern, nicht um die eingescannten 3D-Modelle.

Deshalb soll es, darauf aufbauend, nun die Möglichkeit geben, diese Artikel genauer betrachten zu können, d.h. die eingescannten 3D-Modelle für die Artikel tatsächlich zu inspizieren. Das erfordert, dass die Artikel aus der Artikelwand ausgewählt werden können und anschließend für eine geeignete Form der Untersuchung zur Verfügung stehen.

Diese Anforderung versteht sich als die Überführung der *Pre-Sales*-Komponente "Sondierung von Alternativen" (Abbildung 2.11), da hier nun die Möglichkeit vorhanden sein soll, Artikel zu vergleichen. Dafür ist eine geeignete Bedienung, die intuitiv zu erlernen und möglichst ohne filigrane Ausrichtung der Zeigegeräte ausgeführt werden kann, angestrebt.

Der Prozess wird, angelehnt an einen normalen Webshop, in zwei Ebenen unterteilt: Selektion und Detailbetrachtung.

3.2.2.1 Selektion von Artikeln

Es soll eine Möglichkeit vorhanden sein, durch Auswahl der Artikelmonitore mehr Informationen zu dem Artikel zu erhalten und diese kaufen zu können, da ein Warenkorb bisher bewusst vermieden wurde. Außerdem soll das entsprechende 3D-Modell für diesen Artikel in der Spielumgebung geladen werden können.

Die genaue Art der Selektion der Artikelmonitore ist eine Frage des Entwurfs und vor allem abhängig von der gewählten VR-Hardware. Es sollen deshalb keine festen Vorgaben zur Selektion vorhanden sein. Lediglich die Empfehlung, dass Zeigegerät als 3D-Maus zur Selektion verwendet werden kann (Unterunterabschnitt 2.3.2.1), sei gegeben.

Nach der Selektion soll der Artikelmonitor in einen neuen Zustand versetzt werden, der genauere Informationen über den Artikel bietet sowie weiterer Aktionsmöglichkeiten. Der Monitor soll sich von der Artikelwand lösen und sich in Richtung des Anwenders bewegen. Durch die somit entstehende kürzere Distanz ist es nun einfacher, den Monitor zu betrachten und über das Zeigegerät zu bedienen, da im dreidimensionalen Raum Objekte größer erscheinen, desto dichter sie an dem Betrachter positioniert sind (Perspektive).

Somit steht nun mehr Raum für den Monitor zur Verfügung. Dieser soll durch die Erweiterung um Warenkorbtasten und einer Produktbeschreibung ausgefüllt werden. Eine Konzeptzeichnung dazu ist unter Abbildung 3.5 zu sehen.

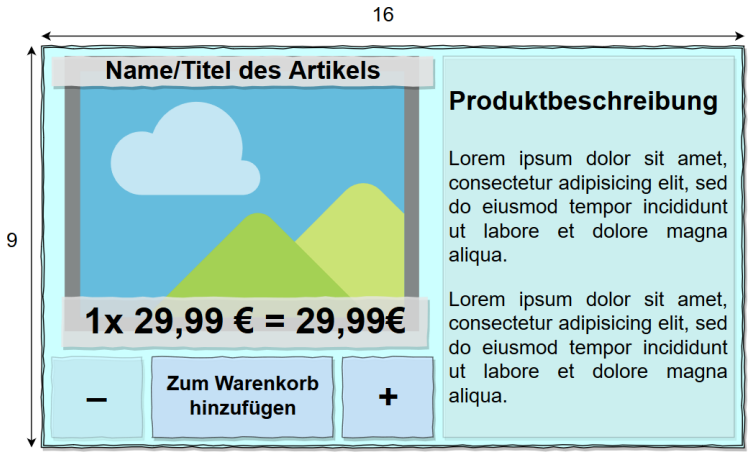


Abb. 3.5: Konzeptzeichnung eines expandierten Artikelmonitors (ausgehend von Abbildung 3.4), wo zusätzlichen zu den wichtigsten Information die Produktbeschreibung und die Tasten zum Hinzufügen des Artikels zum Warenkorb vorhanden sind. Das vorgeschlagene Seitenverhätlnis ist 16:9. (Quelle: Eigene Arbeit)

3.2.2.2 Detailbetrachtung von 3D-Modellen

Im Kern dieser Bachelorarbeit soll die Verbindung von einem Einkaufserlebnis mit der virtuellen Realität stehen. Eingescannte 3D-Modelle zu real existierenden Artikeln sollen in dem VR-Shop betrachtet werden können.

Die Detailbetrachtung setzt sich aus zwei Phasen zusammen: Abruf von 3D-Modellen und Manipulation (Unterunterabschnitt 2.3.2.2) des importierten Modells.

Import von gespeicherten 3D-Modellen in den Shop

Unterunterabschnitt 3.2.2.1 erklärte die Selektion von Artikelmonitoren. Der Abruf von 3D-Modellen soll zeitgleich mit dieser Selektion stattfinden. Dazu soll ein Hintergrundprozess gestartet werden, um die benötigten Dateien asynchron herunterladen zu können.

Nach dem Herunterladen soll unmittelbar ein Import in die Shop-Umgebung stattfinden. Dabei enthalten sind das Modell als solches sowie Texturen für die korrekte Einfärbung. Eventuell muss die relative Größe des Modells an die Umgebung angepasst werden.

Dieser Prozess ist dadurch begründet, dass der Anwender innerhalb des Shops eigenständig die volle Kontrolle darüber haben soll, wann dieser welche Modelle betrachten möchte. Zudem ist schon aus Gründen der Übersichtlichkeit selbstverständlich, dass nicht alle Modelle auf einmal geladen werden sollen – dies wäre aber hinsichtlich der erheblichen Performance-Einbußen in jedem Fall eine ungeeignete Lösung.

Betrachtung und Interaktion mit den 3D-Modellen

Nach dem Import sollen die 3D-Modelle zum Import freigegeben werden. Ein potentieller Käufer soll so die Möglichkeit haben, diese nach Belieben aus allen Winkeln betrachten zu können. Dabei ist es auch wichtig, dass der Betrachtungsabstand keine Rolle spielt, d.h. die Artikel können direkt vor den Augen des Anwenders positioniert werden, damit es möglich ist, feinste Details zu begutachten.

Da dies letztendlich die Anforderung ist, die je nach Komplexität bzw. Intuitivität die größte Hürde zur Akzeptanz darstellen wird, muss das Design darauf ausgelegt sein, so einfach und selbsterklärend möglich zu sein.

Die 3D-Modelle sollen deshalb ein realistisches Verhalten abbilden, als würden die Artikel in der echten Welt angefasst werden. Dies soll über eine Kollisionserkennung mit der virtuellen Welt sowie Gravitation erzielt werden.

Ziel ist die Erfüllung von vier Aufgaben. Die Modelle sollen

- gegriffen werden können,
- beim Greifen in den Händen rotiert werden können,
- in der Spielumgebung abgestellt werden können und
- geworfen werden können.

Das Abstellen dient dazu, dass mehrere 3D-Modelle verschiedener Artikel geladen werden können, damit sie parallel vergleichbar sind ("Sondierung von Alternativen").

Das Werfen dient zum Aufräumen der Umgebung. Der Anwender soll jederzeit die Möglichkeit haben, ein Modell mit Leichtigkeit aus der VR-Welt zu löschen (z.B. bei Nichtgefallen des Artikels). Dazu soll ein einfach zu treffendes Objekt als "Mülltonne" dienen.

Es soll keine Einschränkungen geben bezüglich der Menge an geladenen Artikeln geben (auch mehrere gleiche von demselben Artikel). Diese Entscheidung liegt alleinig beim Anwender.

3.2.3 Anforderung FA3 – Warenkorb

Diese Anforderung ist die direkte Überführung der einleitenden *E-Sale*-Komponente "Online-Warenkorb" (Abbildung 2.11). So gut wie jeder Webshop arbeitet auf dem Prinzip des Warenkorbs, einem virtuellen "Einkaufswagen", in dem Artikel beliebig hineingelegt und entfernt werden können, bevor sie als Kollektiv bezahlt werden.

Auch wenn sich der VR-Shop in diesem Fall in einer virtuellen Umgebung befindet, soll dieser dennoch bzw. gerade deswegen eine geeignete Lösung anstreben, das Modell des Warenkorbs in einem einfach zu bedienenden Interface darzustellen.

3.2.3.1 Warenkorb anzeigen

Der Warenkorb soll als ein virtueller Schalter symbolisiert werden, welcher an einem fixierten Ort in dem VR-Shop angezeigt wird. Auf diesem soll jederzeit die Anzahl sowie der Gesamtbetrag aller darin enthaltenen Artikel angezeigt werden.

Durch Auswahl der Schaltfläche soll der Inhalt als detaillierte Auflistung aller Artikel angezeigt werden. Hier befindet sich auch die Schaltfläche zum Aufgeben der Bestellung.[a]

3.2.3.2 Artikel in den Warenkorb legen

Diese Option soll nach dem Anwählen eines Artikelmonitors und der daraus resultierenden Expansion verfügbar werden. In Abbildung 3.5 wurde eine Konzeptzeichnung für einen expandierten Artikelmonitor bereits vorgestellt, welche Schaltflächen zum Hinzufügen von Artikeln zum Warenkorb ermöglicht.

Dabei soll direkt die gewünschte Quantität des Artikels angegeben werden. Anstatt aber direkt die exakte Menge an Artikeln anzugeben, soll ein Zähler verwendet werden, welcher die Menge inkrementiert bzw. dekrementiert. Diese Steuerung soll als Sammlung von selektierbaren Tasten direkt am Artikelmonitor vorhanden sein (ebenfalls in der Konzeptzeichnung zu sehen). Hintergrund dieser Idee ist, dass in einer virtuellen Umgebung zwei große Bedienfelder wesentlich leichter zu treffen sind als ein Nummernblock vieler einzelner Zahlentasten. Aus dem gleichen Grund wurde auch keine virtuelle Tastatur zur Freitextsuche gefordert (Unterunterabschnitt 3.2.1.1).

Beim Betätigen des Schalters zum Hinzufügen von Artikeln in den Warenkorb wird der Artikelmonitor deselektiert und wieder teil der Artikelwand. Die Artikel werden entsprechend der gewählten Quantität dem Warenkorb hinzugefügt. Der angezeigte Gesamtpreis des Warenkorbs wird daraufhin aktualisiert.

3.2.3.3 Artikel aus dem Warenkorb entfernen

Ein Webshop muss ermöglichen, die in den Warenkorb hinzugefügten Artikel auch wieder zu entfernen. Diese Option soll während der Anzeige des Warenkorbs verfügbar sein.

Dazu soll neben jeden aufgelisteten Artikel zusätzlich ein großes, rotes Kreuz [X] stehen, das bei Selektion den Artikel in dieser Zeile aus dem Warenkorb entfernt. Danach soll der Gesamtbetrag aktualisiert werden.

[a] Für einen vollständigen Shop ist eine Möglichkeit der Bestellung essentiell. Da der Fokus dieser Bachelorarbeit aber auf der Interaktion in der VR-Welt liegt, wird eine Bestellkomponente nicht weiter beachtet (wie in Abbildung 3.1 auch schon angedeutet wurde).

3.2.4 Anforderung FA4 – 3D-Modelle hinzufügen

Diese Anforderung versteht sich als eine anteilige Überführung der finalen *After-Sales*-Komponente "Prüfung & Empfehlung" (Abbildung 2.11). Der Modellbereitsteller wird in diesem Fall als ein Kunde angenommen, welcher einen zufriedenstellenden Artikel gekauft hat und sich daraufhin bereit erklärt, den Artikel weiterzuempfehlen, indem ein 3D-Modell erstellt und dem VR-Shop hinzugefügt wird.

Es soll keine Vorgaben geben, wie die Modelle zu erstellen sind (Unterabschnitt 3.1.2). Im Rahmen dieser Bachelorarbeit soll der Fokus auf automatisch generierten 3D-Modellen durch die Smartphone-App *Qlone* liegen (Unterunterabschnitt 2.1.3.4). Die einzige Voraussetzung soll sein, dass die Modelle in einem Format hochgeladen werden, die für das spätere Programmiermodell geeignet sind.

Es ergeben sich zwei wichtige Anforderungen für den Upload-Prozess, damit der Modellbereitsteller durch das Interface unterstützt wird:

- 3D-Modelle sollen schnell bzw. mühelos in den Shop hochladen werden können.
- Probleme durch inkompatible Formate sollen weitestgehend bedeutungslos sein.

Besonders der zweite Punkt wird je nach gewünschter Zugänglichkeit des Hochladeformulars keine triviale Aufgabe, da es viele verschiedene Formate für 3D-Modelle mit unterschiedlichsten Spezifikationen gibt (z.B. `.obj`, `.stl`, `.x3d` und `.ply`).

Eine Ideallösung gibt es dafür nicht, eben auch durch die Freiheit in der Wahl des Erstellungsprozesses. Die wenigsten Konflikte entstehen durch Restriktion der erlaubten Modellformate. Als anspruchsvollere Erweiterung dazu könnte der Server die Modelle automatisch in ein einheitliches Format konvertieren.

Die aufwändigste aber auch beste Lösung wäre die Integration eines eigenen 3D-Scanners als Bestandteil der Smartphone-App des entsprechenden Webshops (sofern vorhanden), welcher die Modelle erstellen und hochladen kann. Allerdings ist die Erstellung von 3D-Replikaten so komplex, dass es den Rahmen dieser Bachelorarbeit sprengen würde. Zudem gibt es keine bestehende Shop-App, die erweitert werden könnte.

Deswegen soll der Prozess auf eine einfache, bewusst konzeptuell gehaltene Oberfläche ausgelagert werden, welche für diese Bachelorarbeit zudem auch eine Bearbeitungsoberfläche für die Artikel als solches sein soll. Eine Konvertierung soll ebenfalls noch keine Priorität haben – es soll ein einheitliches Dateiformat für die Exporte gewählt werden.

3.3 Nichtfunktionale Anforderungen an die Anwendung

Nachfolgend werden die nichtfunktionalen Anforderungen ("NFA") der Anwendung beschrieben. Der VR-Shop ist als digitales System zur Virtualisierung von 3D-Modellen realer Artikel konzipiert. Somit sind gewisse Aspekte für die frustfreie Handhabung des Anwenders zwingend zu berücksichtigen. Diese Anforderungen sind nicht sofort in der Funktionalität ersichtlich, leiten sich aber implizit aus den in Abschnitt 3.2 genannten funktionalen Anforderungen als Konsequenzen ab.

Die Entwicklung von Anwendungen im virtuellen Raum erfordert viel Leistung für eine flüssige Präsentation der Oberfläche. Nachfolgend wird beschrieben, aus welchen Gründen und wie diese Anforderung berücksichtigt werden muss. Außerdem werden grundlegende Anforderungen zu IT-Sicherheit in Online-Shops gegeben.

3.3.1 Anforderung NFA1 – Latenz und Framerate

Diese Anforderung versteht sich als Überführung der Anforderung "Echtzeitfähigkeit" (Unterabschnitt 2.3.1) für intuitive Interaktion in virtuellen Welten.

Ein HMD (darunter auch die HTC Vive[15]) besteht im Kern lediglich aus zwei Computerbildschirm direkt vor den Augen des Trägers. Diese unterliegen deshalb den gleichen Voraussetzung für ein klares, flüssiges Bild wie jener normaler Computerbildschirme. Zwei Konstanten sind dabei für den wesentlichen Berechnungsaufwand pro Bild entscheidend: *Bildwiederholrate* & *Auflösung (pro Bildschirm)*. Je höher diese Werte sind, desto höher ist der erforderliche Berechnungsaufwand. Pro Frame muss die Position des Anwenders samt HMD und Controllern über das Tracking-System bestimmt werden, diese Informationen an den Computer gesendet werden, von dort aus jeweils für ein Auge leicht versetzt als Bilder gerendert werden und dieses Resultat zurück an das HMD gesendet werden. Der Gesamtaufwand dieses Prozesses steigt mit der Menge an Polygonen der virtuellen Umgebung an und ist durch die verfügbare Leistung der GPU und CPU begrenzt.

Die Latenz soll zwingend unter *20 Millisekunden* als Maximalwert liegen, da bei zu hoher Verzögerung zwischen Bewegung und wahrgenommenen Bild Cybersickness auftreten kann. Zusätzlich sollen mindestens *90 Bilder pro Sekunde* gerendert werden, um die von den üblichen HMDs definierten Bildwiederholraten[12;15] zu erreichen und somit ein flüssiges Bild darstellen zu können.

3.3.2 Anforderung NFA2 – Ressourcenverbrauch

Die Anwendung soll sich während der Ausführung im Vollbildmodus befinden und wird, wie ein Computerspiel, mit großer Auslastung Systemressourcen daherkommen. Es ist durchaus möglich, die kompletten, verfügbaren Ressourcen vollständig auszuschöpfen, da nicht davon ausgegangen wird, dass der Anwender während der Interaktion mit dem HMD etwas anderes an dem Computer fertigstellen muss. Dennoch sollten nur die nötigsten Ressourcen in Anspruch genommen werden.

Ressourcenschonung wird wesentlich dadurch erreicht, dass die 3D-Modelle erst beim expliziten Anwählen der Produkte in den Speicher geladen werden (wie in Unterunterabschnitt 3.2.2.2 beschrieben).

3.3.3 Anforderung NFA3 – Antwortzeitverhalten

Wie in einem normalen Webshop, wo jeder Klick auf einen Link oder einen Button fast sofort eine Antwort zurückliefert, soll auch in dieser Anwendung die Bedienung keine großen Verzögerungen aufweisen. Das Herunterladen von 3D-Modellen aus einer Datenbank sowie das daraus erforderliche Importieren in die 3D-Umgebung wird dabei, je nach Komplexität des Modells, die meiste Zeit in Anspruch nehmen.

Eine Aktion innerhalb des VR-Shops soll nicht mehr als *2 Sekunden* in Anspruch nehmen. Die Einhaltung dieses Zeitlimits kann vor allem dadurch erzielt werden, dass die 3D-Modelle komprimiert und in ein geeignetes Format vorverarbeitet werden.

3.3.4 Anforderung NFA4 – IT-Sicherheit

Die Spracheingabe der Freitextsuche (Unterunterabschnitt 3.2.1.1) soll nur bei expliziter Initiierung der Suche gestartet werden. Statt einer Eigenimplementierung soll zudem eine Spracherkennung eines Drittanbieters über eine verschlüsselte Internetverbindung genutzt werden (was zudem den Vorteil bietet, dass die Spracherkennung genauer ist).

Als System für E-Commerce fordert der zudem VR-Shop einen sicheren Umgang mit Finanzen. Dies hat für diese Bachelorarbeit zwar keine Relevanz, da eine Bestellkomponente nicht vorgesehen ist (Abbildung 3.1), aber für eine vollständige Umsetzung sollten dennoch alle logistischen Verarbeitungen *nicht* vom VR-Shop übernommen werden, sondern für eine Bestellung an eine gesicherte, geprüfte Schnittstelle weitergeleitet werden.

3.4 Zusammenfassung

In diesem Kapitel wurden die Anforderungen an den VR-Shop gestellt, welche für den folgenden Entwurf umgesetzt werden. Die Ergebnisse werden nachfolgend vorgestellt.

Zunächst wurden die Akteure des VR-Shops vorgestellt, welche sich in zwei Varianten des *Anwenders* – *Kunde* und *Modellbereitsteller* – sowie der *Anbieter* des Shops aufteilen. Daraufhin wurden die eigentlichen Anforderungen an den VR-Shop vorgestellt. Diese teilten sich in jeweils vier funktionale und vier nichtfunktionalen Anforderungen auf.

Als funktionale Anforderungen ergab sich die *Artikelexploration im VR-Raum* zur kategorisierten Freitextsuche per Spracherkennung, mit anschließender Darstellung der Suchergebnisse (Artikelmonitore) in einer rundlichen Galerieansicht um den Kunden (Artikelwand), als umfangreichste Komponente. Die Selektion der Artikelmonitore soll das Inspizieren von 3D-Modellen mit realistischen Verhalten (Kollision und Gravition) ermöglichen. Artikel sollen in einer spezifizierbaren Quantität einem Warenkorb hinzugefügt werden, aus dem Artikel auch wieder entfernt werden können. Modellbereitsteller sollen neue 3D-Modelle über eine einfache Oberfläche hochladen können.

Die nichtfunktionalen Anforderungen ergaben sich als Konsequenz der funktionalen Anforderungen. Dabei ist vor allem die Einhaltung von Performanzbedingungen wichtig, welche durch Limitierung der Latenz auf 20ms, der minimalen Beanspruchung von Ressourcen und einem Antwortzeitverhalten von nicht mehr als 2s erzielt werden soll. Zudem soll die Spracherkennung der Freitextsuche nur aktiv sein, wenn sie benötigt wird.

Zusammen mit den Konzepten aus den Grundlagen (Kapitel 2) ergibt sich nun eine ausreichende Basis für einen geeigneten Entwurf (Kapitel 4).

4 Entwurf und Realisierung

In diesem Kapitel wird der eigentliche Entwurf des VR-Shops beschrieben. Dieser ergibt sich aus den in der Anforderungsanalyse (Kapitel 3) vorgestellten Use-Cases und technischen Rahmenbedingungen.

Zunächst werden in Abschnitt 4.1 die einzelnen Systemkomponenten sowie deren Zusammenhang im Gesamtsystem mit Beachtung der verwendeten Hardware betrachtet. Daraufhin ergibt sich in Abschnitt 4.2 der konkrete Software-Entwurf mit den Software-Komponenten der Funktionalitäten des VR-Shops. Zur Abbildung des Software-Entwurfs auf eine konkrete Programmiermodell wird anschließend unter Abschnitt 4.3 die beispielhafte Überführung des Entwurfs auf die Game-Engine *Unity* beschrieben.

Als Kulmination dieser Bachelorarbeit wird in Abschnitt 4.4 die Tragfähigkeit des Entwurfs anhand einer prototypischen Umsetzung demonstriert. Abschnitt 4.5 evaluiert den Umfang dieser Realisierung und stellt zukünftige Erweiterungsmöglichkeiten vor. Das Kapitel wird abschließend in Abschnitt 4.6 zusammengefasst.

4.1 Systemüberblick

Für den VR-Shop werden mehrere Systeme benötigt: Einerseits muss eine Unterstützung für Virtual Reality möglich sein, andererseits die Verwaltung der Datenbestände für 3D-Modelle sowie deren Artikel. Das Gesamtsystem soll für den Entwurf in drei Teilsysteme aufgetrennt werden: *Article System*, *Working Machine* und *Tracking System*.

Dafür wird in Abbildung 4.1 eine Übersicht der einzelnen Systemkomponenten vorgestellt und nachfolgend erläutert. Die ausgegrauten Komponenten in dem Diagramm wurden der Vollständigkeit halber mit einbezogen und werden nachfolgend erläutert, finden aber im weiteren Verlauf des Entwurfes keine weitere Beachtung oder wurden vereinfacht, da der Fokus auf der Oberfläche des VR-Shops liegen soll.

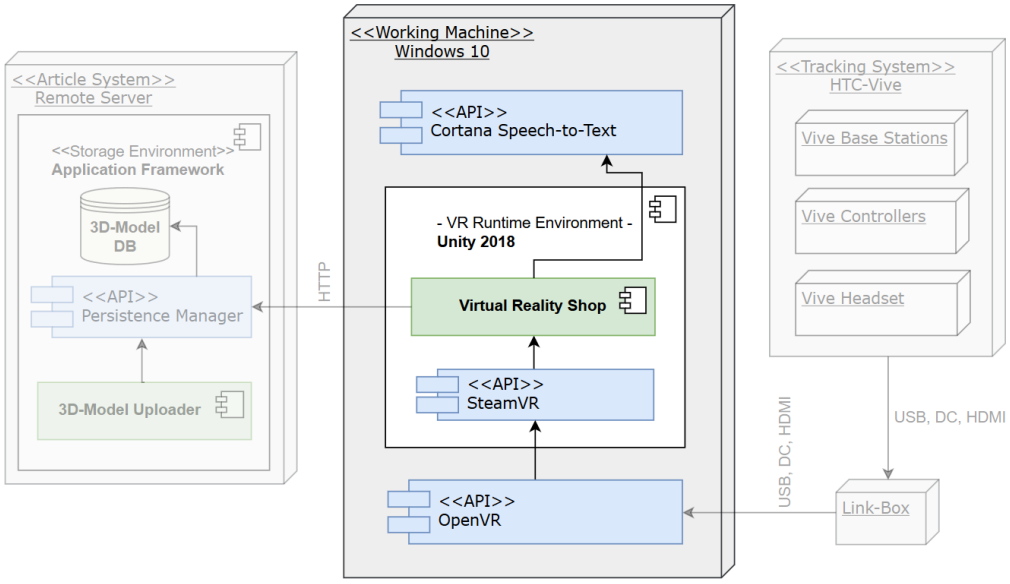


Abb. 4.1: Die Verteilung der einzelnen Systemkomponenten im Gesamtsystem mit grober Aufschlüsselung der enthaltenen Komponenten. (Quelle: Eigene Arbeit)

4.1.1 Article System

Das *Article System* soll ein externer Server sein, welcher über ein Applikations-Framework (z.B. Spring) ein "Persistence Manager"-Interface bereitstellt. Dieses soll dazu genutzt werden, sämtliche 3D-Modelle der Artikel zu speichern ("3D-Model DB"), damit diese später im VR-Shop abgerufen werden können.

Da angenommen wird, dass der VR-Shop eine Erweiterung eines bestehenden Webshops ist (Unterabschnitt 3.2.4), werden über diese API ebenfalls die Informationen der einzelnen Artikel abgerufen. Zum Hochladen neuer Modelle soll "3D-Model Uploader" zur Verfügung stehen, womit Modellbereitsteller (Unterabschnitt 3.1.2) die Möglichkeit haben sollen, zu bestehenden Artikeln neue Modelle über ein User-Interface einzupflegen.

4.1.2 Working Machine

Die *Working Machine* soll das System des eigentlichen VR-Shops darstellen ("Virtual Reality Shop"), welcher lokal auf einem Desktop-Computer ausgeführt werden soll. Dabei soll eine Installation von Windows 10 vorausgesetzt werden, um die Spracherkennungsfunktionalitäten von "Cortana" nutzen zu können.

Die Anwendung selbst soll in einer Game-Engine mit VR-Unterstützung umgesetzt werden (hier, "Unity 2018"), welche über die "SteamVR"-API eine konkrete Schnittstelle von "OpenVR" implementiert, womit die Anwendungsentwicklung für Tracking-basierte VR-Brillen ermöglicht wird. Der Shop soll über HTTP auf den Persistenz-Server zugreifen, um die 3D-Modelle abzurufen. Die Datenabfragen sollen über ein gekapseltes Interface ausgeführt werden, d.h. der Shop weiß nicht, auf was für eine Datenbank dieser zugreift; ausschließlich die Ergebnisse der Artikeldaten sollen relevant sein.

4.1.3 Tracking System

Das *Tracking System* ist das von der VR-Brille bereitgestellte VR-Hardware-System (hier die HTC Vive), welches im Kern aus dem HMD ("Vive Headset") besteht und direkt an die "Link-Box" angeschlossen ist. Die jeweils zwei "Vive Base Stations" und "Vive Controllers" sind derweil kabellos. Alle Informationen sollen gekapselt an die Working Machine über die OpenVR-API gesendet werden.

4.2 Software-Entwurf

Der **Virtual Reality Shop** in der Working Machine (Abbildung 4.1) soll alle selbst entwickelten Komponenten beinhalten und wurde in Abbildung 4.2 detailliert vorgestellt. Dabei soll eine Referenzarchitektur zur Unterstützung des Konzeptes gewählt werden, welche eine systemische Trennung der Anforderungen und Aufgabenbereiche fördert. Die Entscheidung fiel auf eine klassische Drei-Schichten-Architektur:

1. Oberfläche (GUI)
2. Business-Logik
3. Persistenz

Nachfolgend werden die einzelnen Komponenten der Schichten vorgestellt. Unabhängig von der gewählten Entwicklungsumgebung zur Implementierung dieser soll der Software-Entwurf einheitlich beschrieben und auf eine beliebigen, geeignete Entwicklungsumgebung für interaktive 3D-Applikationen ("Game-Engines") anwendbar sein.

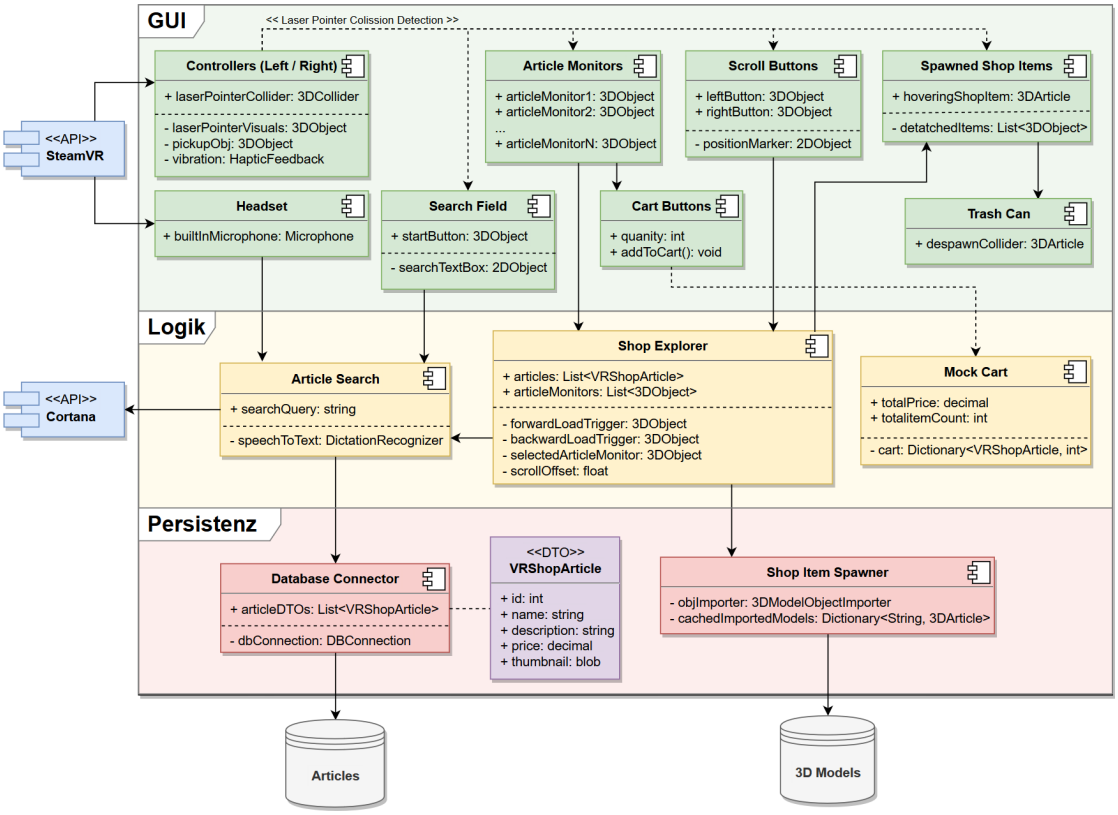


Abb. 4.2: Software-Entwurf des Virtual-Reality-Shops als Komponentendiagramm. Als Referenzarchitektur wurde eine klassische Drei-Schichten-Architektur gewählt: GUI → Logik → Persistenz. (Quelle: Eigene Arbeit)

4.2.1 GUI

Die GUI soll alle Komponenten zusammenfassen, welche in der VR-Oberfläche sichtbar sind oder direkt innerhalb der Oberfläche Interaktionsmöglichkeiten bereitstellen. Da vor allem der konzeptuelle Fokus dieser Bachelorarbeit auf die Bedienung und Usability der VR-Anwendung liegen soll, beinhaltet diese Schicht die meisten Komponenten.

Ausgangspunkt aller Interaktionen ist dabei die *SteamVR*-API. Diese stellt alle Daten zur VR-Peripherie – HMD und Controller – bereit und soll sie zur Darstellung und weiteren Verarbeitung an die Komponenten *Headset* und *Controllers (Left/Right)* senden. Das Headset hat ein eingebautes Mikrofon, was für die spätere Suche per Spracherkennung verwendet werden soll.

Alle Interaktionen sollen ausgehend von den Controllern angeleitet werden, weswegen sie die einzige Verbindung zu SteamVR besitzen. An den Controllern sollen Laserpointer mit Kollisionserkennung befestigt werden, damit die Interaktion mit den weiteren Schaltflächen der GUI ermöglicht wird. Weiterhin sollen die Controller die Möglichkeit haben, 3D-Modelle der Artikel aufzuheben und besitzen haptisches Feedback (Vibration).

Die Controller sollen folgende Komponenten zur Interaktion innerhalb der GUI haben:

- *Search Field* soll bei Auswahl die Suche per Spracherkennung initiieren.
- *Article Monitors* sollen die Überführung der Artikelmonitore in der Artikelwand darstellen, welche die Artikel mit den Produktinformationen darstellen.
  - *Cart Buttons* sollen an expandierten Artikelmonitoren die Mengenauswahl sowie das Hinzufügen von Artikeln zum Warenkorb ermöglichen.
- *Scroll Buttons* sollen die Artikelwand horizontal verschieben können (scrollen).
- *Spawned Shop Items* sollen beim Anwählen von Artikelmonitoren importiert werden und sollen innerhalb des Shops gegriffen, rotiert und geworfen werden können.
  - *Trash Can* soll eine Kollisionserkennung besitzen und die 3D-Modelle der Artikel beim Berühren aus der VR-Welt löschen.

Die Festlegung, ob Kollisionserkennung zur GUI- oder zur Logikschicht gehört ist dabei nicht klar einzuordnen, da zwar Berechnungslogik benötigt wird, die Selektion jedoch nur innerhalb der GUI angewandt wird. In diesem Fall wurde Kollisionserkennung deshalb weiterhin ausschließlich der GUI-Schicht zugeordnet.

4.2.2 Logik

Die Logikschicht soll alle Eingaben der GUI-Schicht technisch bearbeiten. Das Herz des gesamten VR-Shops und die eigentliche Kernkomponente des Gesamtsystems soll dabei der *Shop Explorer* sein – dieser verarbeitet alle getätigten Eingaben der Steuerung und behandelt die Selektion von Schaltflächen innerhalb der GUI auf Logikebene.

Dazu zählt in erster Linie die Artikelwand, deren zylinderförmige Anordnung durch den Shop Explorer berechnet und beim Scrollen angepasst werden soll – die Rotation soll durch einen manipulierbaren Offset berechnet werden. Durch die entstehende Verschiebung sollen die Artikelmonitore dann still ausgetauscht werden, weswegen *Load Triggers* benötigt werden. Dies erfordert, dass sowohl die Anzahl der Suchergebnisse der Artikel sowie die Anzahl der Artikelmonitore selbst bekannt sind. Der Shop Explorer soll

zusätzlich die Auswahl von allen Schaltflächen behandeln und bei der Auswahl von Artikelmonitoren den Import von den 3D-Modellen der Artikel an die Persistenzschicht initiieren.

Um den Shop Explorer nicht mit Funktionalitäten zu überladen, soll die Suchfunktionalität in eine eigene Komponente *Article Search* ausgelagert werden, wobei der Shop Explorer lediglich die Ergebnisse über eine Schnittstelle erhält. Der Artikelsuche soll über das Mikrofon des Headsets die Audioeingabe erhalten, welche dann mithilfe von der Cortana-Spracherkennung von Windows 10 in einen String umgewandelt werden soll. Dieser soll nach Erkennung des gesprochenen Textes an die eigentliche Komponente der Artikelsuche weitergeleitet werden.

Mock Cart soll der Warenkorb des Shops sein, welcher bewusst konzeptuell gehalten wird, da er nicht im Vordergrund der Arbeit steht und keine Anbindung an einen echten Shop für Bestellungen vorhanden ist.

4.2.3 Persistenz

Die Persistenzschicht soll in erster Linie die Aufgabe haben, den Rest der Anwendung möglichst entkoppelt von den Artikelbeständen zu halten. Im Falle dieser Bachelorarbeit soll über diese Schicht auch die Verschleierung der Abwesenheit einer echten Webshop-Datenbank behandelt werden, d.h. die Datenbank soll theoretisch austauschbar sein.

Dazu soll der *Database Connector* eine Schnittstelle anbieten, welche eine beliebige Datenbankanbindung (DBConnection) haben kann. Diese soll den erkannten Suchtext der Spracherkennung verarbeiten und die Anfrage an die eigentliche Datenbank stellen. Die Suchergebnisse sollen als repräsentative Datenklasse behandelt werden, weswegen sie vor der weiteren Verwendung zunächst in ein *VRShopArticle*-DTO umgewandelt werden sollen. Dieses soll neben der ID alle grundlegenden Informationen über die Artikel zur Darstellung in der Oberfläche enthalten (Name, Preis, Beschreibung und Produktbild).

Der *Shop Item Spawner* soll vom Shop Explorer eine Aufforderung erhalten, dass bei Selektion eines Artikelmonitors das entsprechende 3D-Modell geladen werden soll. Die IDs der VRShopArticle-DTOs werden hier benötigt – da jeder Artikel nur genau ein 3D-Modell besitzen soll, reicht die ID aus, um eine eindeutige Zuordnung zwischen Artikeln und Modellen zu gewährleisten.

Da das Importieren von 3D-Modellen in eine Game Engine nicht trivial ist und je nach Dateiformat der Modelle unterschiedlich ist, soll auf ein Plugin zurückgegriffen und über eine Schnittstelle (3DModelObjectImporter) zusammengefasst werden. Bereits importierte Modelle sollen in einem Cache aufbewahrt werden, selbst wenn sie aus der Oberfläche durch die Trash Can gelöscht wurden. So ist ein erneuertes Abrufen eines bestimmten Modelles performanter.

4.3 Abbildung des Software-Entwurfs auf das Programmiermodell Unity

Der Software-Entwurf wurde allgemein beschrieben. Dieser soll nun in ein konkretes Programmiermodell überführt werden. Dazu wird nachfolgend eine technische Abbildung auf *Unity* beschrieben. Die Entscheidung auf dieser Plattform ist vor allem aufgrund der bestehenden Vorerfahrung des Autors begründet (siehe Abbildung 2.12 als Referenzarbeit). Unity ist eine Game Engine bzw. IDE zur Entwicklung von Videospielen mit einem großen Funktionalitätsumfang und ist dabei sehr einsteigerfreundlich gehalten. Es bietet bereits ab Werk viele Konzepte der Computergrafik, die eine robuste und einfache Entwicklung von Spielen und somit dreidimensionalen Anwendungen ermöglichen. [28]

Im Kern von Unity steht dabei das **GameObject**, welches eine Art Containerklasse für Informationen in der digitalen Umgebung bereitstellt. Die GameObjects stehen in einer hierarchischen Anordnung zueinander – genauer, einer doppelt verlinkten Baumstruktur – wobei jedes GameObject seinen "parent" und seine "children" kennt, sofern vorhanden. Dieses Konstrukt wird in einem Szenengraphen gesammelt.

Standardmäßig ist ein GameObject leer und besitzt keine eigenen Eigenschaften, mit Ausnahme von obligatorischen Informationen zur Position, Rotation und Skalierung in der digitalen Umgebung (allgemein als **Transform** bezeichnet). Um weitere Funktionalitäten den GameObjects hinzuzufügen werden sogenannte **Components** eingesetzt, von denen beliebig viele an einem GameObject hängen können. Components werden technisch nicht differenziert, sollen aber für den Software-Entwurf des VR-Shops grundsätzlich in zwei Kategorien aufgeteilt werden:

- *Visuelle Components* sollen sämtliche Funktionalitäten zur Darstellung der Spielumgebung zusammenfassen. In Unity werden **MeshRenderer** zur Darstellung von 3D-Modellen eingesetzt. Die Modelle müssen somit eine Zusammenstellung

aus Meshes (dreidimensionalen Vernetzungen von Polygonen zur Darstellung von 3D-Körpern) und Materials (eine oder mehrere Bilddateien zur konkreten Einfärbung des Objektes) sein. Die Schaltflächen des VR-Shops, die Artikelmonitore und die 3D-Modelle der eingescannten Artikel sollen allesamt über den MeshRenderer visuell aufbereitet werden.

- *Technische Components* sollen alle Funktionalitäten zusammenfassen, welche nicht direkt in der Spielumgebung sichtbar sind. Dazu zählt die Kollisionserkennung, welche eingesetzt werden soll, um Schaltflächen selektieren und 3D-Modelle aufheben und bewegen zu können. Kollision spielt somit in erster Linie für die Selektion und Interaktivität der VR-Welt eine Rolle (Abschnitt 2.3). Technische Komponenten sollen auch sämtliche manuell implementierte Funktionen – alle Interaktionskonzepte der GUI-Schicht sowie die vollständige Logik- und Persistenzschicht – in Form von **Scripts** zusammenfassen, welche in Unity üblicherweise in *C#* geschrieben sind. Es ist möglich, in Scripts zu einem GameObject öffentliche Referenzen zu anderen GameObjects zu setzen, um so übergreifend auf Informationen anderer Komponenten zugreifen zu können. Wie im Software-Entwurf gezeigt wurde, sollen hierbei zyklische Abhängigkeiten vermieden werden.

Neben der einfach zu bedienenden Entwicklungsumgebung ist aber vor allem ein Aspekt entscheidend für die Auswahl von Unity als Plattform: Für Unity steht ein kostenloses, Open-Source *SteamVR*-Plugin[a] zur freien Verwendung zur Verfügung, welches direkt von *Valve Corporation* entwickelt und angeboten wird. Dieses Plugin beinhaltet eine einsatzbereite Zusammensetzung von mehreren GameObjects mit Scripten mit dem Namen **[CameraRig]**, um den direkten Einsatz von VR-Peripherie – HMD und Controller – ohne große, aufwändige Konfiguration zu ermöglichen.

Unity bietet zusätzlich noch eine eigene API zur Anbindung an die Spracherkennung *Cortana* von Windows 10 an, auf die die Komponente *Article Search* zugreift. Diese wird **DictationRecognizer** genannt.[b]

Der Import der 3D-Modelle ist trivial, da die Beschränkung auf ein einziges Format festgelegt wurde (Unterabschnitt 3.2.4). Für das `.obj`-Format steht dafür ein kostenloser Importer unter der MIT-Lizenz zur Verfügung.[c]

[a] https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647
[b] https://docs.unity3d.com/ScriptReference/Windows.Speech.DictationRecognizer.html
[c] https://github.com/gpvigano/AsImpL

4.4 Umsetzung

In diesem Abschnitt soll nun die Tragfähigkeit der Entwurfs zum VR-Shop anhand einer prototypischen Umsetzung in Unity demonstriert werden. Die Unterabschnitte beziehen sich dabei die funktionalen Anforderung aus Abschnitt 3.2.

Der Fokus der Umsetzung liegt zum Großteil auf den grafischen Komponenten des Software-Entwurfes (GUI-Schicht in Abbildung 4.2). Es werden dementsprechend sehr detailliert die ersten Anforderungen beschrieben, während nachfolgende lediglich zwecks der Vollständigkeit erläutert werden sollen. Eine Aufschlüsselung über den Umfang der Realisierung wird in der anschließenden Evaluation gegeben (Abschnitt 4.5).

4.4.1 Vorweg: Randbedingungen für die Umsetzung

Bevor die Umsetzungen der einzelnen funktionalen Anforderung beschrieben werden, sollen gewisse Rahmenbedinungen für den Aufbau des VR-Shops spezifiziert sein. Dies sind keine eigentlichen Anforderung an den VR-Shop, sondern allgemeine Entscheidungen zur Entwicklung von VR-Applikationen.

4.4.1.1 Interaktionsfläche und Spielumgebung

Der VR-Shop soll so umgesetzt werden, dass Einschränkungen von Tracking-basierten HMDs adäquat behandelt werden. Beispielsweise arbeitet die HTC Vive mit Infrarotsensoren, welche ein ca. 12qm großes Spielfeld aufbauen (Abbildung 4.3), in der sich der Anwender frei bewegen kann. Eine Navigation über die eingegrenzten Bereiche der echten Welt innerhalb der Spielumgebung hinaus würde jedoch eine magische Navigation wie Teleportation erfordern (Unterunterabschnitt 2.3.2.3). Da dies aber maßgeblich den Grad der Immersion beeinträchtigen würden und aus früheren Erfahrungen des Autors (Abbildung 2.12) keine frustfreie Handhabung der Applikation gewährleisten, soll die Anwendung entsprechend angepasst entworfen werden.

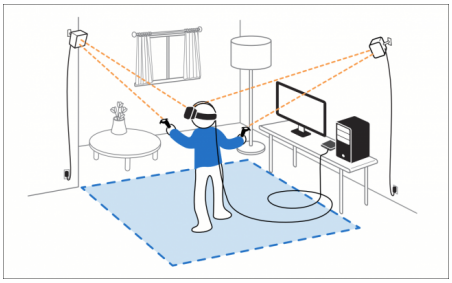


Abb. 4.3: Illustration der ungefähren Nutzfläche der HTC Vive[8]

Als Lösungsstrategie sei folgendes Zitat gegeben:

> [...] *these imperfections in the Vive's tracking output may not pose a problem. For instance, one can likely use the Vive for experiments in which the risk of losing tracking is small because the participant only moves in a small area.*
>
> — Diederick C. Niehorster (2013) [23, S. 20]

Aus diesem Grund soll die Umgebung "stationär" erstellt werden, d.h. der Interaktionsbereich ist gleich der virtuellen Umgebung. Der Anwender befindet sich im Mittelpunkt, während die Interaktionen mit dem Shop 360° um diesem herum stattfinden. Innerhalb des eingegrenzten Bereichs soll aus jeder Position und jedem Winkel eine Interaktion möglich sein – stehen oder sitzen soll ebenfalls keinen Einfluss haben.

Es soll nicht möglich und auch nicht nötig sein, sich innerhalb der Spielumgebung außerhalb des Tracking-Bereichs bewegen zu können. Diese Einschränkung bietet zwar weniger Fläche für die Anwendung, aber verhindert so auch Komplikationen durch unzureichende Bewegegungskontrolle (womit Cybersickness vermieden wird). Die eigentliche Spielumgebung soll möglichst so gestaltet sein, dass sie nicht von dem Shopping-Erlebnis ablenkt.

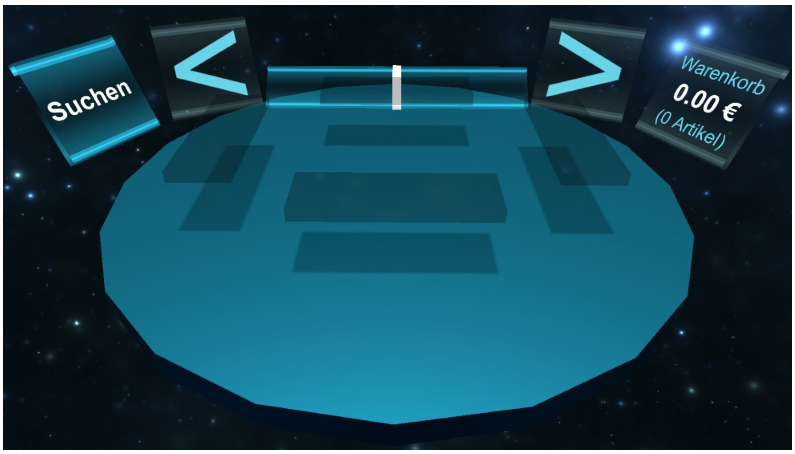


Abb. 4.4: Der initiale VR-Shop nach dem Start der Anwendung (Quelle: Eigene Arbeit)

Abbildung 4.4 zeigt die prototypische Umsetzung zur Einhaltung aller genannten Rahmenbedingungen. Auf eine aufwändige visuelle Gestaltung wurde verzichtet – stattdessen fiel die Wahl auf eine einfache, schwebende, runde Plattform mit einem schlichten, holografischen Design.[d]

[d] Es wurden während der Entwicklung Wälder und Häuser als Umgebung getestet, aber in allen Tests haben sich diese aber als zu ablenkend vom VR-Shop herausgestellt.

Der Anwender befindet sich im Zentrum dieser Plattform und ist zu keiner Zeit gezwungen, diese zu verlassen. Durch den bewussten Verzicht auf Realismus wurde ermöglicht, dass die in Unterunterabschnitt 2.3.2.2 genannten "magischen Manipulationstechniken" später eingesetzt werden können und gewährleistete zudem, dass der erste Eindruck der Anwendung schlicht ist.

Vier schwebende Plattformen sollen später dazu dienen, die 3D-Modelle abzustellen, wie auf einen Tisch. Der Hintergrund dieser Idee war dabei, dass die Modelle während der Entwicklung immer durch Bücken vom Boden aufgehoben werden mussten.

4.4.1.2 Steuerung

Die Umsetzung des VR-Shops soll mit expliziter Verwendung von der HTC Vive geschehen. Dieses VR-System wird mit zwei identischen Controllern geliefert, welche relativ genaue Positionserkennung im dreidimensionalen Raum ermöglichen. Zudem besitzen sie eine große Menge an Interaktionsmöglichkeiten durch die eingebauten Tasten.

Abbildung 4.5 zeigt eine einführende Übersicht der Controller mit Beschreibungen zu den jeweiligen Tastennamen. In den nachfolgenden Umsetzungen der einzelnen Anforderungen werden diese Tasten mehrfach referenziert.

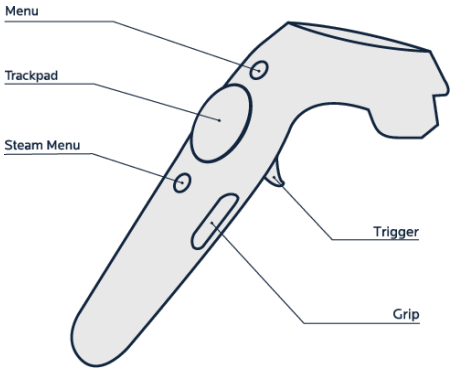


Abb. 4.5: HTC Vive Controller mit Beschreibung der Tasten [27]

Die einzelnen Tasten sollen für den VR-Shop folgendermaßen belegt werden:

- Trigger: Selektieren/Greifen
- Trackpad: Scrollen
- Menu: Suche starten
- Grip[e]: Aktion abbrechen
- Steam Menu: *–keine Belegung–*

[e] Die Grip-Taste soll bewusst optional für die Bedienung der Anwendung sein. Durch frühere Erfahrung (Abbildung 2.12) hatte sich eine intuitive Interaktion mit der Grifftaste als problematisch ergeben, da viele Personen diese nicht sofort verstanden oder gefunden haben.

4.4.2 Umsetzung FA1 – Artikelexploration im VR-Raum

Die Artikelexploration im VR-Raum ist die umfangreichste Komponente des Entwurfes. Sie soll alle Interaktionen mit der Oberfläche, die die Abbildung eines normalen Webshops auf die VR-Umgebung bilden, umfassen – von der Suche bis hin zu Auswahl eines Artikels.

4.4.2.1 Suche

Das Suchen von Artikeln soll über eine Freitextsuche agieren, angelehnt an ein Textfeld zur Suche in einem echten Webshop. Wie schon in der Anforderungsanalyse erwähnt wurde (Unterunterabschnitt 3.2.1.1), gibt es kein standardisiertes Verfahren, die gewohnten Eingabemöglichkeiten mit einer Tastatur in eine VR-Welt zu übertragen. Stattdessen soll eine Spracherkennung eingesetzt werden, welche verbale Sprache in Text konvertiert, als wäre diese durch eine Tastatur eingegeben worden.

Es hat sich zudem als relativ trivial herausgestellt, Spracherkennungen in Unity bereitzustellen, sofern ein Windows 10-Rechner als System für die Anwendung genutzt wird. Wie bereits in Abschnitt 4.3 erklärt wurde, wird dafür auf eine einfache API zugegriffen, die die Spracherkennung von Cortana nutzen soll und das eingebaute Mikrofon vom Headset der HTC Vive benutzt.

Die Suche soll durch den Schalter “Suche” (in Abbildung 4.4 links) oder universell durch die Betätigung der Menu-Taste (Abbildung 4.5) initiiert werden. Während der Suche soll ein Textfeld in der dreidimensionalen Welt angezeigt werden, dass in Echtzeit die erkannte Spracheingabe des Anwenders textlich darstellt. Nach zwei Sekunden Stille soll die Eingabe beendet werden und die Suche nach den Artikeln beginnen. Die Anzahl der Suchergebnisse soll nach Erhalt dieser in der Textbox angezeigt werden. Je nach Anzahl der Suchergebnisse soll die Textbox farblich umrandet sein – rot für keine Ergebnisse, blau für Erfolg. Die prototypische Umsetzung zur Suche im VR-Shop ist unter Abbildung 4.6 zu sehen.

Abb. 4.6: Das Suchfeld des VR-Shops mit beispielhaften Ergebnissen (Quelle: Eigene Arbeit)

4.4.2.2 Artikelwand und Artikelmonitore

Nach Erhalt der Suchergebnisse sollen diese nun adäquat dargestellt werden. In Unterunterabschnitt 3.2.1.2 wurde das aus Webshops abgeleitete Konzept der Galerieansicht vorgestellt sowie die rundliche Anordnung der Artikel als Kacheln um den Anwender herum, was jeweils als Artikelwand bzw. Artikelmonitore definiert wurde.

Artikelmonitore: Informationen darstellen

Abbildung 3.4 zeigte die konzeptuelle Skizze eines Artikelmonitors. Darin enthalten waren Informationen zum Namen und Preises des Artikels sowie eines Produktbildes.

Die Umsetzung in die 3D-Anwendung soll eine möglichst akkurate Überführung dieser Skizze sein. Eine vordefinierte Menge an instantiierten Artikelmonitoren (im nächsten Abschnitt näher erläutert) soll die Suchergebnisse auf die Oberfläche der Artikelmonitors an den entsprechenden Stellen positionieren. Die visuelle Umsetzung des befüllten Artikelmonitors für die prototypische Implementation ist unter Abbildung 4.7 mit einem beispielhaften Artikel zu sehen.

Abb. 4.7: Der Artikelmonitor mit einem beispielhaften Artikel (Quelle: Eigene Arbeit)

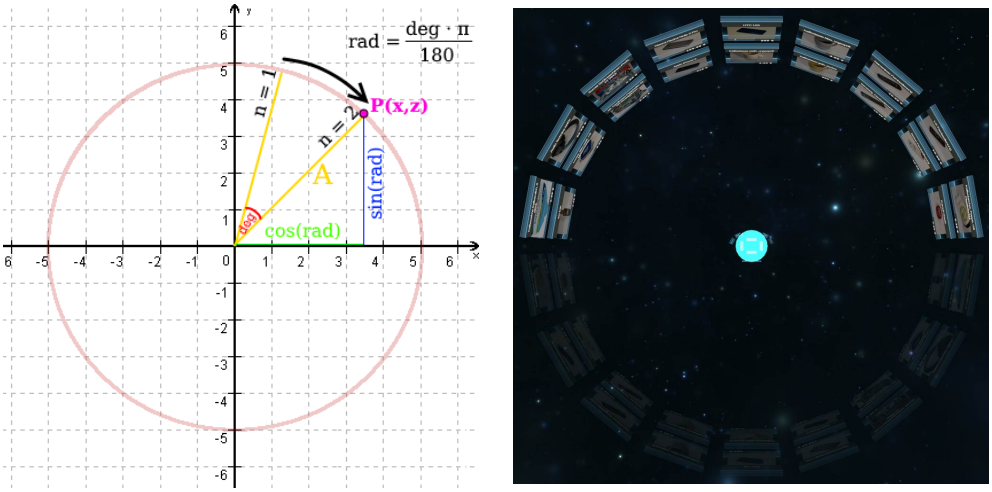
Artikelwand: Zylinderförmige Positionierung der Artikelmonitore

Die Artikelmonitore sollen, wie in Abbildung 3.3 skizziert wurde, in einer rundlichen Anordnung um den Anwender herum dargestellt werden. Dies ist die Hauptaufgabe der Kernkomponente *Shop Explorer* aus dem Software-Entwurf (Abschnitt 4.2).

Es war während der Entwicklung stets eine Frage, wie viele Artikelmonitore auf einmal gesehen werden können sollen. Der Anwender sollte einerseits nicht durch zu viele Artikel auf einmal überwältigt werden, andererseits aber auch einen Mehrwert aus der horizontalen Anordnung gewinnen können. Als Resultat von manuellen Justierungen stellten sich **18 Artikelmonitore**, aufgeteilt in zwei übereinanderliegenden Ringen, als akzeptabler

Mittelpunkt heraus. Da die Blickrichtung des Anwenders aber nur die Innenseite des Zylinders zu jeder Zeit sehen kann und damit nur eine Hälfte, sollen somit 36 tatsächliche Monitore erforderlich sein.

Die tatsächliche Anordnung der Artikelwand soll zylinderförmig sein. Da VR-Welt dreidimensional ist, kann jedes Objekt durch die Koordinaten (x, y, z) lokal beschrieben werden können. Da die Artikelmonitore ringförmig angeordnet sein sollen, ändert sich die vertikale Position y nach der Initialisierung nicht. Somit kann die Berechnung zweidimensional mit den Koordinaten x und z von oben als Kreis vereinfacht betrachtet werden (Abbildung 4.8a). Die Rotation richtet sich automatisch nach dem Ursprung.



(a) Berechnung (b) Resultat

Abb. 4.8: Ansicht der Artikelwand von oben. Links ist die Berechnung zusammengefasst und rechts das Resultat im VR-Shop. (Quelle: Eigene Arbeit)

Die Positionen ergeben sich durch einfache Trigonometrie aus

$$
\begin{aligned}
deg &= n \cdot A \pmod{360} \\
rad &= \frac{deg \cdot \pi}{180} \\
\mathbf{x} &= \cos(rad) \cdot D \\
\mathbf{z} &= \sin(rad) \cdot D,
\end{aligned}
$$

wobei n der n-te Artikelmonitor sein soll, A der konstante Abstand zwischen den einzelnen Monitoren und D die konstante Distanz zum Zentrum.[24] Abbildung 4.8b zeigt das Resultat der Umsetzung in der prototypischen Implementation.

Scrollen: Visualisierung von großen Datenmengen im virtuellen Raum

Im letzten Abschnitt wurde in Abbildung 4.8b bereits angedeutet, dass nur die Hälfte der Artikelwand sichtbar sein soll. Dies soll zunächst das Ziel verfolgen, den Fokus des Anwenders in eine feste Richtung zu lenken. Der Hintergrund ist aber eigentlich ein anderer: Die Suchergebnisse könnten mehrere hunderte Artikel umfassen, welche alle im Shop dargestellt werden sollen, die Anzahl der Artikelmonitore ist aber begrenzt. So wird also eine Möglichkeit der Navigation erwartet, welche, wie bei einem normalen Webshop, mehr Informationen anzeigen lässt, als der Bildschirm Platz bietet.

Eine Lösung dazu soll das Scrollen sein.[30] Wo der normale Webshop aber vertikal agiert, soll der Fokus beim VR-Shop auf horizontaler Bewegung liegen. Die Artikelwand soll zylinderförmig um den Anwender rotiert werden, wobei der Mittelpunkt senkrecht fixiert ist, wie die Achse bei dem Rad eines Autos. Somit kann der Anwender still stehenbleiben, während sich die Artikelwand um diesen dreht.

Während der Rotation sollen die Artikel durch **stilles Nachladen** ausgetauscht werden. Die nebenstehende Abbildung 4.9 zeigt einen konzeptionellen Entwurf dazu:

- **1** ist der Anwender, welcher in nördlicher Blickrichtung den sichtbaren Teil der Artikelwand betrachet und scrollt.
- **2** ist der sichtbare Teil der Artikelwand, während sie (vom Anwender aus) nach links gescrollt wird.
- **3** ist der Übergang in den unsichtbaren Teil der Artikelwand.
- **4** ist eine unsichtbare Passierzone, in der die Artikelmonitore still durch neue Artikel ersetzt werden.

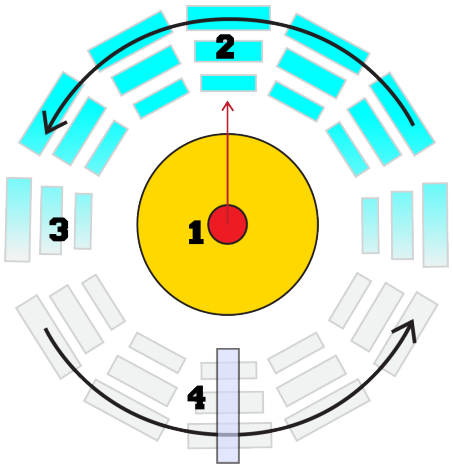


Abb. 4.9: Konzept des stillen Nachladens (Quelle: Eigene Arbeit)

Sobald sich ein Monitor direkt hinter dem Anwender befindet – unsichtbar – soll ein weiterer Artikel geladen werden und den bestehenden austauschen. So können problemlos im Hintergrund mehr Artikel dargestellt werden als Artikelmonitore vorhanden sind, ohne, dass der Anwender davon etwas merken würde. Eine Analogie dazu ist das Abwickeln einer Küchentuchrolle.

Eingabemöglichkeiten zum Scrollen und Positionsvisualisierung

Das Scrollen zum Rotieren der Artikelwand und dem daraus resultierenden stillen Nachladens selbst soll auf mehrere Arten möglich sein. Alle nachfolgenden Interaktionsmöglichkeiten sollen über die Controller der HTC Vive geschehen (Abbildung 4.5).

(a) **Wischen über das Trackpad ("Swipe"):**
Generell ist die Interaktion durch Wischen über ein Touchpad bereits eine sehr erfolgreiche Methode, die bei jedem Smartphone täglich Anwendung sieht und bei den meisten Benutzern schon längst intuitiv geworden ist. Diese Funktionalität soll deshalb überführt werden: Durch horizontales Wischen über das Trackpad des Controllers soll gezielt und im eigenen Tempo die Artikelwand vor- und zurückgeschoben werden können. Dabei soll es auch möglich sein, durch sehr schnelles, wiederholtes Wischen rapide durch den Shop zu navigieren. Beim Loslassen soll der Bildlauf noch etwas "ausrollen", wie es auch beim Smartphone der Fall ist, damit die Navigation fließend ist (auch "träger Bildlauf" genannt). [30]

(b) **Seitenwechsel über das Steuerkreuz:**
Unterhalb des Trackpads ist zusätzlich noch ein Steuerkreuz eingebaut, das durch festes Drücken ausgelöst wird. So kann beim Druck auf die linke oder rechte Seite des Trackpads eine klassische Bewegung erzwungen werden, wie es bei einer Tastatur der bereits alltäglich ist. Die Wand soll so ebenfalls langsam vor- oder zurückgeschoben werden.

(c) **Seitenwechsel durch Betätigen virtueller Schalter:**
Diese Methode soll das Steuerkreuz als virtuelle Schaltflächen innerhalb des VR-Shops zur Verfügung stellen, welche als zwei Pfeiltasten links und rechts symbolisiert werden sollen. Der Bildlauf soll ansonsten identisch sein.

Zwischen den virtuellen Schaltern der letzten Scroll-Möglichkeit soll die Verbildlichung von der aktuellen Position sein (Abbildung 4.10). Dieser Ansatz entspricht der Motivation einer Bildlaufleiste eines Computers ("Scrollbar"). [4] Nach dem Laden der Suchergebnisse soll die Position der Artikelwand links sein und stetig nach rechts verschoben werden, bis durch alle Artikel gescrollt wurden. Die Tasten sollen sichtbar deaktiviert werden, sobald ein Ende der Artikelwand erreicht wurde.

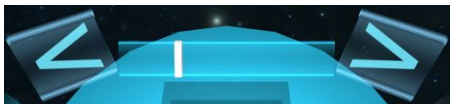


Abb. 4.10: Scroll-Buttons und Positionsmarker im VR-Shop (Quelle: Eigene Arbeit)

4.4.3 Umsetzung FA2 – Artikel inspizieren

Nachdem nun die Artikelexploration möglich ist, sollen die eigentlichen Artikel als eingescannte 3D-Modelle inspiziert werden können. Dazu werden nachfolgend Techniken zur Selektion und Interaktion beschrieben.

4.4.3.1 Selektion von Artikelmonitoren

Zunächst soll eine grundlegende Möglichkeit vorhanden sein, mit dem VR-Shop interagieren zu können, damit die Artikelmonitore selektiert und die 3D-Modelle geladen werden. Dazu wird zunächst eine Selektionstechnik benötigt (Unterunterabschnitt 2.3.2.1), welche ausgewählte Artikelmonitore in einen neuen Zustand versetzen.

Controller als Laser-Pointer

Es ist bereits ein gängiges Verfahren, dass die VR-Selektion durch zielsichere Ausrichtung der Controller auf virtuelle Bedienfelder stattfindet. Dabei ist die visuelle Darstellung in fast allen Fällen ein virtueller "Laserpointer", senkrecht ausgehend von der oberen Spitze des Gerätes. Auch bei diesem Shop soll dieses bestehende Prinzip möglichst unverändert übernommen werden, da es vom Kerngedanken mindestens schon seit Veröffentlichung der *Nintendo Wii* aus dem Jahre 2006 besteht und bis heute als eine gängige Lösung für die Bedienung eines Interfaces mit einer bewegungssensitiven Fernbedienung gilt.[26]

Abb. 4.11: Laserpointerselektion (Quelle: Eigene Arbeit)

Der Laserpointer soll standardmäßig rot sein. Zur immersiven Untermalung soll dieser leuchtend grün werden, sobald Kollision mit einer auswählbaren Schaltfläche auftritt – zusätzlich soll der Controller vibrieren (haptisches Feedback). Die prototypische Umsetzung ist in Abbildung 4.11 zu betrachten. Durch Betätigung der Trigger-Taste (Abbildung 4.5) soll die jeweilige Aktion der Schaltfläche ausgeführt werden, während der Laserpointer mit dieser kollidiert.

Auswahl des Artikelmonitors

Die Auswahl eines Artikelmonitors hat neben der Detailbetrachtung von 3D-Modellen noch das Ziel, mehr Informationen über das Produkt als solches sowie weitere Aktionsmöglichkeiten bereitzustellen. Dies wurde als "Expandierter Artikelmonitor" definiert (Abbildung 3.5).

Nachdem der Monitor durch den Laserpointer selektiert wurde, soll sich dieser sich vom Rest der Artikelwand lösen und sich an eine vordefinierte Position zur der Rückseite des Anwenders bewegen (da die Artikelwand nicht verdeckt werden soll).

Dabei soll es weiterhin möglich sein, durch den Shop zu scrollen, weswegen der ausgewählte Monitor als eigenständiges Objekt betrachtet werden soll, damit keine Konflikte entstehen, wenn das stille Nachladen einsetzt. Eine einfache Lösung dazu ist, dass der Artikelmonitor sich auf technischer Ebene gar nicht von der Wand löst, sondern einen Klon von sich selbst an gleicher Position erstellt. Dieser gibt zwar den Anschein, der ausgewählte Monitor zu sein, doch jener wird nur unsichtbar geschaltet und bewegt sich als Teil der Artikelwand weiterhin im Kreis um den Anwender. Das Klonen bietet zusätzlich den Vorteil, dass diese Kopie einfach gelöscht werden kann, sobald ein anderer Artikel ausgewählt wird. Der ursprünglich selektierte Artikelmonitor kann dann einfach wieder als sichtbar definiert werden.

Abbildung 3.5 skizzierte einen expandierten Artikelmonitor mit den zusätzlichen Schaltflächen für den Warenkorb sowie der Produktbeschreibung. Da der Artikelmonitor ein Quader sein soll (Abbildung 4.7), sollen diese auf der Rückseite dargestellt werden.

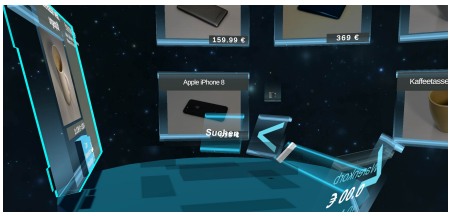


Abb. 4.12: Selektierter Artikelmonitor (Quelle: Eigene Arbeit)

Für die prototypische Implementation sollte dieser Ansatz verfolgt werden und die Umsetzung ist unter Abbildung 4.12 in Form einer perspektivische Außenansicht des VR-Shops mit einem ausgewählten Artikelmonitor zu betrachten: Links ist die Rückseite des geklonten Artikelmonitors zu sehen, welche den unsichtbaren Teil der zylinderförmigen Artikelwand verdeckt. Rechts ist ein Loch in der Artikelwand, an welcher Stelle sich der selektierte Artikelmonitor unsichtbar weiterhin bewegen soll. Durch drücken der optionalen Grip-Taste (Abbildung 4.5) soll der Artikelmonitor deselektiert und wieder in der Artikelwand dargestellt werden – der Klon wird gelöscht.

4.4.3.2 Detailbetrachtung von 3D-Modellen

Nun soll das eigentliche 3D-Modell zur Betrachtung in der VR-Welt freigegeben werden. Der Import des Modells soll zeitgleich mit der Selektion des Artikelmonitors geschehen. Da der Prozess einige Sekunden dauern kann, soll dieser visuell durch Ladeanimationen untermalt werden.

Nach dem Import soll das Modell entsprechend in der VR-Welt positioniert werden und stellt sich für die Interaktion bereit. Dazu soll jedes Objekt drei Phasen durchleben, die einen Lebenszyklus pro 3D-Modell innerhalb des VR-Shops darstellen, nachdem sie heruntergeladen und importiert wurden. Diese werden durch die prototypische Umsetzung in Abbildung 4.13 demonstriert und im Anschluss einzeln erläutert.

(a) Phase 1: Schweben (b) Phase 2: Greifen (c) Phase 3: Löschen

Abb. 4.13: Der Lebenszyklus eines 3D-Modells im VR-Shop ist in drei Phasen aufgeteilt: *Schwebendes Modell im Zentrum*, *Genauere Betrachtung des Modells durch Greifen* und *Wegwerfen des Modells*. (Quelle: Eigene Arbeit)

Phase 1: *Schwebendes Modell im Zentrum*

Zunächst soll das Modell nach dem Herunterladen an eine vordefinierte Position eingefügt werden und sich schwebend, etwa auf Augenhöhe des Anwenders, in der Luft befinden (Abbildung 4.13a). Dabei soll es sich, wie ein Ausstellungsstück, um die eigene Achse drehen.

Ziel ist es, dass somit immer ein einheitlicher Standpunkt für neu geladene Artikel existieren soll, damit der Anwender keine unnötige Zeit zur Suche benötigt. Zudem soll sich in diesem "Slot" jederzeit nur ein Artikel befinden; wird ein neuer Artikel ausgewählt, soll das bestehende Modell gelöscht und mit dem neuen Modell ersetzt werden.

Phase 2: *Genauere Betrachtung des Modells durch Greifen*

Möchte sich der Anwender das Objekt näher ansehen, soll dieses durch Drücken der Trigger-Taste (Abbildung 4.5) getätigt werden können. Das Objekt wird dann aus der schwebenden Position gelöst – ab diesem Zeitpunkt soll der Slot freigegeben werden und weitere Modelle geladen werden können. Hierbei soll das Objekt an den Controller fixiert werden, sodass durch Bewegung und Rotation das Objekt beliebig gedreht und betrachtet werden kann (Abbildung 4.13b).

Das Objekt soll solange in der Hand des Anwenders fixiert bleiben, bis die Trigger-Taste wieder losgelassen wird. Beim Loslassen wird das Objekt dann nicht wieder in den schwebenden Zustand zurückversetzt, sondern soll anfällig für Gravitation werden und zu Boden bzw. auf eine der vier Ablageflächen (Abbildung 4.4) fallen. So können nun mehrere Artikel gleichzeitig angezeigt und verglichen werden.

Phase 3: *Wegwerfen des Modells*

Die Modelle sollen ausschließlich als Hilfsmittel für den Kunden behandelt werden und keinen eigenen Einfluss auf das Kaufverhalten haben. Deswegen wird angenommen, dass regelmäßig Modelle geladen, betrachtet, verglichen und letztendlich nach Meinungsbildung verworfen werden.

Dafür soll sich in Spielumgebung eine riesige, sinnbildliche "Mülltonne" befinden, in welcher die Modelle problemlos hineingeworfen aus der Welt gelöscht werden können (Abbildung 4.13c). Da es dabei unter keinen Umständen zum Problem werden soll, dass dieses Ziel verfehlt werden könnte, ist die Mülltonne für die prototypische Implementation die gesamte Welt unterhalb der blauen Plattform des VR-Shops.

4.4.4 Umsetzung FA3 – Warenkorb

Die Umsetzung des Warenkorbs sollte von Anfang an von geringer Priorität für diese Bachelorarbeit sein – der Fokus lag vor allem auf der Artikelexploration und -interaktion, also den beiden zuvor erklärten Anforderungsumsetzungen, da es die einzigen funktionalen Anforderungen mit obligatorischer Relevanz für VR waren. Der Warenkorb wurde der Vollständigkeit halber in der Anforderungsanalyse (Unterabschnitt 3.2.3) konzeptuell mit vorgestellt.

Aus diesem Grund wurde der Warenkorb stark abstrahiert und nur zwei Funktionen für prototypischen Entwurf implementiert: Artikel dem Warenkorb hinzufügen und die Darstellung von Gesamtpreis und Artikelmenge des Warenkorbs. Die restlichen Funktionen (Warenkorb betrachten und Artikel aus dem Warenkorb entfernen) wurden nicht implementiert. Bereits im Software-Entwurf wurde diese Komponente deshalb bewusst als "Mock Cart" bezeichnet (Abschnitt 4.2).

Die Schaltflächen zum Hinzufügen eines Artikels in wählbarer Quantität sollen, nach der Skizze des expandierten Artikelmonitors (Abbildung 3.5), unten links auf diesem erscheinen. Zwei dieser Tasten sollen dafür bereitstehen, bei Betätigung einen Zähler mit der Anzahl der Exemplare zu erhöhen bzw. zu verringern, die gekauft werden sollen. Beim ersten Anwählen des Artikelmonitors soll dieser Zähler bei "1" für ein einzelnes Exemplar starten. Die Tasten zum Hoch- und Herunterzählen sollen durch die Wahl geeigneter Symbole selbsterklärend gestaltet sein. Zusätzlich soll gewährleistet sein, dass der Zähler nicht unter 1 fallen kann, indem die Minustaste beim Erreichen dieses Minimums deaktiviert wird. Die Umsetzung ist in Abbildung 4.14 zu betrachten. Hier wurden die Zählertasten durch $\boxed{-}$ und $\boxed{+}$ symbolisiert. Währenddessen wird über den Schaltflächen stets der Gesamtbetrag und die Anzahl der Artikel angezeigt.

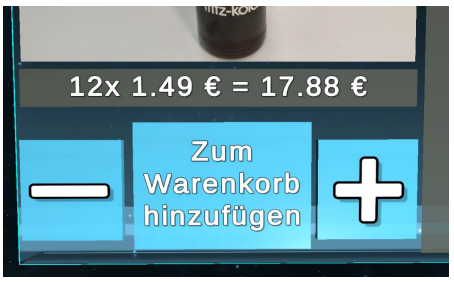


Abb. 4.14: Artikelmengenkontrolle (Quelle: Eigene Arbeit)

Die Schaltfläche des Warenkorbs soll jederzeit im VR-Shop sichtbar sein (in Abbildung 4.4 rechts) und eine Zusammenfassung des Gesamtpreises sowie der Anzahl aller enthaltenen Artikel anzeigen. Für den Prototypen ist die Schaltfläche deaktiviert und nicht selektierbar (Abbildung 4.15). Bei einer vollständigen Implementation soll beim Anwählen eine Übersicht aller Artikel angezeigt werden, inklusive der Möglichkeit, diese Artikel wieder aus dem Warenkorb entfernen zu können.

Abb. 4.15: Schaltfläche des Warenkorbs (Quelle: Eigene Arbeit)

4.4.5 Umsetzung FA4 – 3D-Modelle hinzufügen

Als finaler Schritt soll nun die Umsetzung für das Hinzufügen von neuen 3D-Modellen beschrieben werden. Das Hinzufügen von 3D-Modellen umfasst nur Aspekte, die nicht direkt innerhalb der VR-Umgebung stattfinden. Wie schon im System-Entwurf (Abbildung 4.1) angedeutet wurde, ist die Umsetzung auch hier – wie der Warenkorb – nur provisorisch zur Unterstützung der ersten beiden Anforderungen gewesen und wurde entsprechend abstrahiert.

4.4.5.1 Einscannen von neuen 3D-Modellen mit Qlone

Endergebnis dieser Bachelorarbeit soll es sein, 3D-Modelle von eingescannten Objekten (Produktartikeln) in virtueller Form betrachten zu können. Dabei soll es möglich sein, dass jeder Anwender mit minimalen Hardware-Anforderungen die Möglichkeit hat, eigene Modelle in den Shop hochzuladen. Da es in dieser Bachelorarbeit um einen konzeptuellen Entwurf zur Realisierbarkeit dieser Idee geht, musste ein geeignetes Modell zum Erstellungsprozess von 3D-Modellen zunächst gesucht und entwickelt werden.

In Abschnitt 2.1 wurden die Grundkonzepte zum 3D-Scanning beschrieben, wobei die *Nahbereichsphotogrammetrie* als das geeignetste Scanning-Verfahren für den hier beschriebenen Anwendungsfall gewählt wurde. Zu dieser Feststellung wurden drei Applikationen von Drittanbietern auf Basis dieser Technologie vorgestellt, verglichen und bewertet (Unterunterabschnitt 2.1.3.4), wobei die Smartphone-App **Qlone** als die geeignetste App bewertet wurde. Diese ist sehr einfach zu bedienen und erstellt Modelle mit ausreichender Qualität bei geringem Hardware-Aufwand, da nur ein Smartphone benötigt wird. Abgesehen davon benötigt Qlone lediglich eine ausgedruckte Unterlage mit einem proprietären Muster.[f]

Zum eigentlichen Aufbau sollen die Anforderungen an qualitative 3D-Scans berücksichtigt werden (Unterabschnitt 2.1.2). Als Hardware soll neben dem Smartphone und der Matte noch eine elektrisch rotierende Platte eingesetzt werden, welche ähnlich dem Prinzip von "The Microwave" (Abbildung 2.5) arbeitet, damit die Kamera zur Stabilisierung auf einem Stativ montiert werden kann. Da so die Qlone-Matte über die Platte hinausragt, muss diese durch ein Stück stabile Pappe verstärkt werden, damit sie weiterhin eine gleichmäßige Ebene bilden kann.

[f] https://docs.wixstatic.com/ugd/0dc13a_00f1c793e9274ea4897766276c116ca1.pdf

Idealerweise würde zur einheitlichen Ausleuchtung des Objektes eine Reihe von professionellen Softboxen eingesetzt werden. Da aber davon ausgegangen wird, dass Modellbereitsteller nicht bereit sind, solche Hardware extra zu erwerben, soll stattdessen eine kostengünstige Alternative verwendet werden. So kann ein gleichmäßig weißer Hintergrund bereits durch ein großes Blatt Papier (DIN A0) erzielt werden. Die Beleuchtung kann währenddessen mit einer einfachen Schreibtischlampe erfolgen – ein umwickeltes Stück Papier um die Lampe reicht dabei aus, um das Licht gleichmäßig zu verteilen.

Der finale Aufbau für die Umsetzung ist in Abbildung 4.16 zu betrachten. Ein zu scannendes Objekt – hier ein Kaffeebecher – wird auf die Qlone-Matte gestellt, welche durch ein Stück feste Pappe planar gehalten wird. Eine elektrisch rotierende Platte bewegt das Objekt und die Matte langsam um die eigene Achse, während ein Smartphone aus einem erhöhten Winkel, fixiert durch ein Stativ, diese filmt. Die gleichmäßige Beleuchtung geschieht über die bereits erwähnte, mit einem Stück Papier umwickelte Schreibtischlampe.

Abb. 4.16: Endgültiger Aufbau der Scan-Umgebung mit Qlone (Quelle: Eigene Arbeit)

4.4.5.2 Hochladen der 3D-Modelle

Ein Upload-Formular für 3D-Modelle zu den Artikeln wurde nicht implementiert. Für eine vollständige Implementation soll es dafür eine Web-Oberfläche geben, welche einen Upload-Formular für Modelldateien bereitstellt. Nach dem Hochladen soll die Datei auf Kompatibilität überprüft und ggf. verarbeitet/konvertiert werden. Ideal wäre es dabei auch, wenn sie zusätzlich optimiert oder komprimiert werden, um die Dateigröße zu verkleinern und das spätere Rendering im Shop zu beschleunigen. Ein möglicher letzter Schritt könnte sein, die Uploads vorher vom Anbieter zu bestätigen (Moderation), um qualitativ minderwertige oder unpassende Modelle zu unterbinden.

Für die prototypische Umsetzung hat es ausgereicht, die Exporte in einem zu dem Shop parallel liegenden Ordner auszulagern, um so das Resultat des Hochladens zu simulieren. Wie in Abschnitt 4.3 bereit beschrieben wurde, wurden nur `.obj`-Modelle verwendet.

4.4.5.3 Skalierung der 3D-Modelle

Es gab ein Problem, welches erst beim Export der 3D-Modelle offensichtlich wurde: Die Größe des eingescannten Objektes ist nicht immer akkurat zur Größe des physikalischen Originals. Da Qlone nicht weiß, wie groß die ausgedruckte Matte zum Zeitpunkt des Scans ist, kann dies nicht automatisiert werden. Als Resultat erschienen die ersten Scans noch unrealistisch groß im Shop. Das Phänomen wird bei der Verwendung von weiteren Scannern noch komplizierter – da jede Software eine eigene Basisgröße für die Exporte besitzt, kann ein entsprechender Skalierungsfaktor nicht generalisiert angenommen werden.

Die Lösung dieses Problems soll durch eine Zuordnung von manuell justierten Skalierungsfaktoren erzielt werden. Bei dem Upload-Formular soll ein Auswahlfeld vorhanden sein, das es dem Modellbereitsteller erlaubt, die Umstände des Scans zu definieren, d.h. der eingesetzte 3D-Scanner und, im Falle von Apps wie Qlone, die Größenwahl der ausgedruckten Unterlage.

Als Beispiel: Für die prototypische Umsetzung hat sich ein Scan mit Qlone auf einer ausgedruckten Matte der Größe DIN A4 ein Skalierungsfaktor von `0.00125` ergeben (Qlone exportiert standardmäßig sehr große Modelle).

4.5 Evaluation

In diesem Abschnitt wird die Umsetzung des VR-Shops evaluiert. Zunächst stellt Unterabschnitt 4.5.1 den Umfang der Entwurfsrealisierung vor. Unterabschnitt 4.5.2 stellt Grenzen, Probleme und Erweiterungsmöglichkeiten vor, welche sich bei der Entwicklung der prototypischen Implementation ergeben haben.

4.5.1 Umfang der Realisierung

Insgesamt bildet der Entwurf des VR-Shops die wesentlichen Aspekte der Anforderungsanalyse ab. Es gab jedoch einige nicht (vollständig) implementierte Anforderungen – sei es aus Zeitmangel, technischen Schwierigkeiten oder weil diese Punkte als irrelevant für die zu untersuchende Interaktion mit dem VR-Shop bewertet wurden.

Nachfolgend wird eine tabellarische Übersicht des Umfangs mit allen Anforderungen aus der Anforderungsanalyse vorgestellt:

Anforderung	Vollständigkeit
FA1 – Artikelexploration im VR-Raum	✓
Suche von Artikeln	✓
Oberflächenpräsentation des VR-Shops	✓
FA2 – Artikel inspizieren	✓
Selektion von Artikeln	✓
Detailbetrachtung von 3D-Modellen	✓
FA3 – Warenkorb	•
Warenkorb anzeigen	✗
Artikel in den Warenkorb legen	✓
Artikel aus dem Warenkorb entfernen	✗
FA4 – 3D-Modelle hinzufügen	•
NFA1 – Latenz und Framerate	✓
NFA2 – Ressourcenverbrauch	✓
NFA3 – Antwortzeitverhalten	•
NFA4 – IT-Sicherheit	✓

Tabelle 4.1: Tabellarische Übersicht des realisierten Umfangs der Anforderungsanalyse. (Bewertung der Vollständigkeit: ✓ vollständig erfüllt, • teilweise erfüllt, ✗ nur andeutungsweise erfüllt)

4.5.2 Grenzen, Probleme, Erweiterungen

Die prototypische Entwicklung des VR-Shops umfasst den Großteil der funktionalen Anforderungen, die in der VR-Welt abgebildet werden sollten. Bei Anforderungen außerhalb der VR-Welt gab es zum Teil jedoch erhebliche Schwierigkeiten. Diese werden nachfolgend vorgestellt und durch beispielhafte Lösungsansätzen als Ausblick ergänzt. Dazu werden zunächst alle fehlenden Funktionen der Entwurfsumsetzung und anschließend nicht gelöste Probleme des VR-Shops aufgeführt.

4.5.2.1 Fehlende Funktionen

Fehlendes externes Persistenz-System: Bereits Unterunterabschnitt 4.4.5.2 begründete, warum ein Persistenz-System mit Upload-Formular nie implementiert wurde: Für die Demonstration des VR-Shops hätte dies keinen Unterschied hervorgehoben. Auch die Anbindung an eine Artikeldatenbank wurde zwecks der Demonstration mit *SQLite*

simuliert. Sowohl das Formular als auch die Datenbankanbindungen müssten für eine Anbindung an einen realen Webshop ausgetauscht werden. Aus diesem Grund wurde im Software-Entwurf in der Persistenzschicht explizit die Entkopplung einer Datenbankabhängikeit durch die *DBConnection* gefordert (Unterabschnitt 4.2.3).

Fehlende Funktionalitäten des Warenkorbs: Der Warenkorb wurde nur exemplarisch als Mock-up behandelt (Unterabschnitt 4.4.4). Sollte der VR-Shop an einen echten Shop angebunden werden, müssten die Auflistung und Entfernung von beinhaltenden Artikeln zwingend nachträglich implementiert werden. Außerdem wird eine Logistikanbindung benötigt, um Bestellungen tatsächlich aufgeben zu können.

4.5.2.2 Ungelöste Probleme

Mangelhafte Ergebnisse bei den 3D-Scans: Für den prototypischen Entwurf wurden gerade einmal *26 Scans* als qualitativ ausreichend eingestuft. Die 3D-Modelle haben nie einen Zustand von realistischer Qualität erreicht. So haben alle Scans Dellen, Löcher und verschwommene Texturen (ein Beispiel ist das Heftgerät unter Abbildung 2.9b). Durch die Scanning-Umgebung mit weißem Hintergrund und direkter Beleuchtung (Abbildung 4.16) wurden die Scans nur gerade so besser, dass sie zumindest andeutungsweise die Objekte widerspiegeln – vorher waren diese gänzlich entstellt und nicht wiederzuerkennen (Anhang A). Für eine realistische, kontemporäre Umsetzung des VR-Shops sind professionelle 3D-Scanner noch die beste Lösung (z.B. Abbildung 2.7).

Einbußen der Performanz beim Import von 3D-Modellen: Die Anforderung *Antwortzeitverhalten* (Unterabschnitt 3.3.3) wurde nur teilweise erfüllt. Zwar ist das Suchen und Darstellen von Artikeln auf der Artikelwand schnell, aber der Import der 3D-Modelle dauert wesentlich länger als erwartet. Dies liegt an der Natur der durch Qlone exportierten `.obj`-Modelldateien, da diese unkomprimiert sind und teilweise mehrere Millionen Polygone beinhalten. Die Modelle sollten also mit einem entsprechenden Tool komprimiert werden. Zudem ist Unity nicht Thread-Safe, weswegen der Import nicht asynchron stattfinden kann, was das System beim Import etwas verlangsamt. Als Zwischenlösung beider Probleme wurde ein Cache implementiert, welcher bereits geladene Modelle recycelt (Unterabschnitt 4.2.3). Eine richtige Lösung erfordert eine Vorverarbeitung, z.B. als sogenannte *AssetBundles*[g], einem Unity-Format, das bewusst für den Import von Assets aus externen Dateien konzipiert wurde.

[g] https://docs.unity3d.com/Manual/AssetBundlesIntro.html

Anspruchsvolle Kollisionsberechnung der Artikelobjekte: Dieses Problem hängt mit den bereits erläuterten unkomprimierten Modelldateien zusammen. Unity bietet zur Kollisionserkennung neben primitiven Körpern auch die Kollision mit den Meshes des Modells an ("MeshCollider"). Da die hohe Auflösung der Modelle das System erheblich verlangsamt hat, wurde stattdessen eine einfache Quaderkollision gewählt. Auch hier wäre die Lösung das Komprimieren der Modelle.

4.6 Zusammenfassung

Der Entwurf des VR-Shops wurde in diesem Kapitel ausführlich vorgestellt. Anhand einer prototypischen Implementation in Unity mit der HTC Vive wurde die Tragfähigkeit der Umsetzung demonstriert.

Die grundlegenden technischen Aspekte des System- und Software-Entwurfs wurden durch Diagramme abgebildet (Abbildung 4.1 und Abschnitt 4.2). Bei letzterem wurde eine klassische Referenzarchitektur aus den drei Schichten *GUI*, *Logik* und *Persistenz* gewählt. Anschließend wurde erklärt, wie dieser Entwurf auf das Programmiermodell Unity abgebildet wird (Abschnitt 4.3).

Die Anforderungen aus der Anforderungsanalyse (Kapitel 3) wurden anschließend einzeln umgesetzt (Abschnitt 4.4). Artikel werden mithilfe einer Freitextsuche durch die Spracherkennung von *Cortana* gesucht, welche anschließend für die VR-Welt aufbereitet werden. Dabei wurde ein besonderer Fokus auf die zylinderförmige Artikelwand und die Artikelmonitore gelegt sowie die Interaktionen mit diesen. Das Inspizieren von Artikeln ist durch Greifen der 3D-Modelle ermöglicht.

Die Vollständigkeit der Umsetzung ist für das provisorische Usability-Konzept ausreichend (Abschnitt 4.5). Ein Warenkorb wurde jedoch nur als Mock-Up entworfen. Die 3D-Scans wurden mit der App *Qlone* erstellt, es wurde aber kein Upload-Formular entworfen. Probleme entstanden zudem im Bezug auf die Performanz durch zu große, unkomprimierte 3D-Modelle, welche zudem auch mangelhafte Qualität aufwiesen. Zu allen Problemen wurden Lösungsansätze als Ausblick für die Weiterentwicklung des VR-Shops gegeben.

5 Zusammenfassung und Ausblick

Nachfolgend werden die Ergebnisse dieser Bachelorarbeit vorgestellt. Abschnitt 5.1 gibt eine Zusammenfassung der erarbeiteten Inhalte und Abschnitt 5.2 zeigt als Abschluss der Bachelorarbeit einen Ausblick auf Virtual Reality in der Zukunft.

5.1 Zusammenfassung der Arbeit

In dieser Bachelorarbeit wurde der beispielhafte Entwurf eines Webshops in der virtuellen Realität vorgestellt. Dabei wurde ein besonderer Schwerpunkt auf die Bereitstellung von 3D-Modellen realer Objekte gesetzt, welche mithilfe von möglichst intuitiven 3D-Scannern automatisch rekonstruiert werden sollten. Mithilfe der Unity Game-Engine und der HTC Vive VR-Brille wurde ein Prototyp implementiert.

Die Grundlagen (Kapitel 2) stellten zunächst 3D-Scanning vor. Durch Vergleich von verschiedenen Verfahren und Applikationen wurde die Smartphone-App *Qlone* auf Basis von *Nahbereichsphotogrammetrie* als eine geeignete Lösung für diese Bachelorarbeit gewählt. Anschließend wurden die grundlegenden Anforderungen an Online-Shops sowie die intuitive Interaktion in virtuellen Welten vorgestellt.

In der Anforderungsanalyse (Kapitel 3) wurden die Akteure des konzeptuellen VR-Shops vorgestellt. Daraufhin wurden jeweils vier funktionale und nichtfunktionale Anforderungen an den VR-Shop spezifiziert, welche sich durch einen besonderen Fokus auf die Interaktion mit dem VR-Shop auszeichneten. Die umfangreichsten Funktionen waren die Freitextsuche von Artikeln, die Darstellung der Suchergebnisse in einer für VR geeigneten Form und abschließenden die eigentliche Detailbetrachtung der 3D-Modelle innerhalb der VR-Welt.

Der anschließende Entwurf (Kapitel 4) konzipierte den VR-Shop dann ausgehend aus den bisherigen Erkenntnissen. Zunächst wurden Diagramme zur Übersicht des Gesamtsystems und des Software-Entwurfs präsentiert. Die Tragfähigkeit der Umsetzung wurde

anhand einer prototypischen Implementation mit dem Programmiermodell Unity und der HTC Vive bestätigt. Zuletzt wurden Probleme während der Entwicklung und mögliche Verbesserungsvorschläge an den VR-Shop beschrieben, wobei vor allem Einbußen der Performanz durch zu hochauflösende 3D-Modelle auftraten.

Der finale Prototyp des VR-Shops dieser Bachelorarbeit ist in Abbildung 5.1 zu sehen.

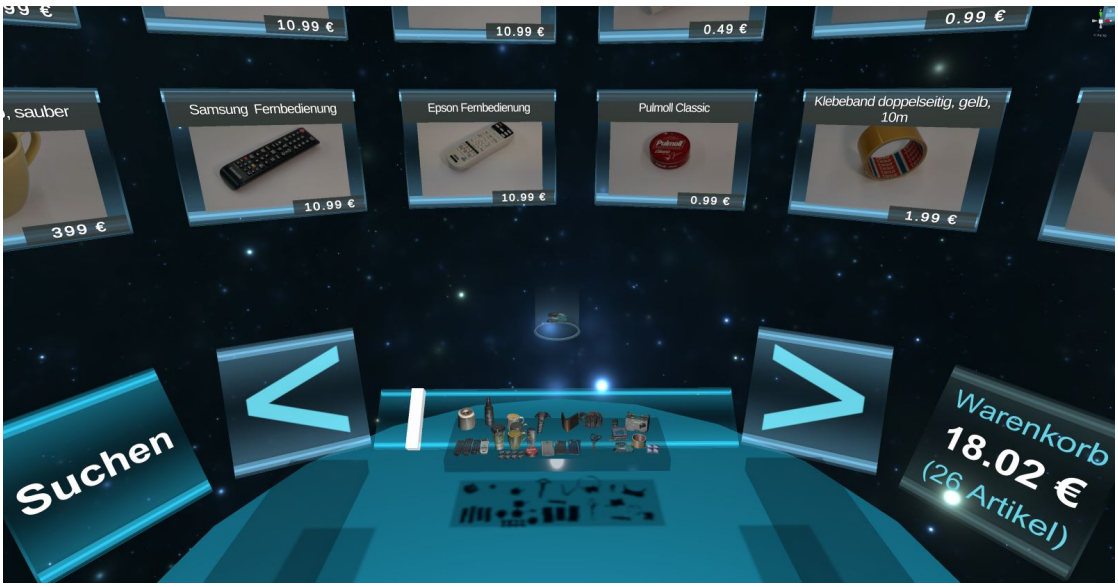


Abb. 5.1: Der finale VR-Shop in Unity mit allen 26 eingescannten 3D-Modellen. (Quelle: Eigene Arbeit)

5.2 Zukunftsvisionen

Der VR-Shop dieser Bachelorarbeit sollte demonstrieren, wie ein virtuelles Einkaufserlebnis in der Zukunft aussehen könnte. E-Commerce hat das stetige Bestreben hat, dessen mögliche Märkte auszuweiten und auch VR wird davon betroffen sein.

Zum aktuellen Zeitpunkt ist VR jedoch noch nicht weit genug fortgeschritten, um überhaupt für einen großen Markt an Kunden interessant zu sein. Dies liegt einerseits an den sehr hohen Hardware-Kosten bei gleichzeitig mäßiger Bildqualität, auf der anderen Seite aber vor allem auch an den bestehenden Mangel an umfangreichen VR-Anwendungen.

Bis auf wenige Ausnahmen handelt es sich meistens um kleine technische Demos oder Portierungen bestehender Software als VR-Anwendungen. Damit VR größere Akzeptanz findet, darunter auch der VR-Shop, muss für den Anwender ein sofort ersichtlicher Mehrwert vorhanden sein. Hätte diese Bachelorarbeit z.B. nur die Suche von Artikeln als Thema gehabt, ohne Beachtung von 3D-Modellen, wäre der Mehrwert verfallen.

Valve Corporation – Mitentwickler der HTC Vive – ist eins der wenigen Unternehmen im E-Commerce (in diesem Fall für Videospiele und Unterhaltungsmedien), welches ein starkes Interesse vertritt, den VR-Markt voranzutreiben. In einem Interview sagte CEO Gabe Newell zur Problematik des fehlenden Mehrwerts:

> *VR is not going to be a success at all if people are just taking existing content and putting it into a VR space.* — Gabe Newell (2017) [20]

Sobald genügend VR-Produkte verfügbar sind, welche diese Anforderung erfüllen und somit den großen Vorteil von virtueller Realität demonstrieren, ist auch die Ausweitung des Webshops in virtuelle Welt nicht mehr weit entfernt.

A Ungenügendes 3D-Scanning

Nachfolgend werden einige ausgewählte Bilder von 3D-Scans präsentiert, welche aufgrund von unzureichender Handhabung der Hardware oder Software zu entstellten und teils absurden 3D-Modellen geführt haben. Außerdem werden die ersten Versuchsaufbauten der Scan-Umgebungen mit der eingesetzten Hardware vorgestellt, jeweils mit Begründungen, warum diese nicht ausreichend waren.

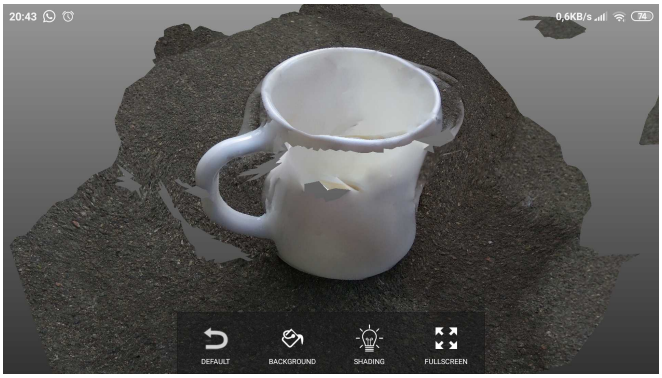
Dieser Anhang versteht sich als optionale Ergänzung der Rechercheergebnissen zu den Aufnahmebedingungen für 3D-Scans (Unterabschnitt 2.1.2).

Ungenügende Scan-Resultate:
Im ersten Versuch mit der App *3DScann* wurde eine weiße Kaffeetasse auf einem Asphaltboden eingescannt. Durch die reflektierende Oberfläche der Tasse sowie der stark kontrastreichen Textur des Bodens, ist die Tasse mit dem Boden "verschmolzen".

Im weiteren Versuch mit *Qlone* wurde eine schwarze Kaffeetasse eingescannt. Durch unsaubere Handhabung der Software wurde der Kaffeesatz in der Tasse als "Deckel" der Tasse interpretiert, während das untere Ende der Tasse nur partiell vorhanden ist.

Ungenügende Hardware-Aufbauten:
Im ersten Hardware-Aufbau des zu scannenden Heftgerätes steht dieses auf der Qlone-Matte, welche sich auf der rotierenden Platte befindet. Da die Aufnahmen aber draußen unter grellem Sonnenschein getätigt wurden, war die Beleuchtung auf dem Heftgerät zu ungleichmäßig.

Im zweiten Aufbau wurde das Smartphone auf einem Dreibeinstativ fixiert und die Umgebung wurde nach Innen verlegt. Allerdings hat die einseitige Bestrahlung des LED-Lichts immer noch eine zu ungleichmäßige Beleuchtung.



(a) Test-Scan mit *3DScann*

(b) Test-Scan mit *Qlone*

Abb. A.1: Qualitativ mangelhafte 3D-Scans.

(a) 1. Hardware-Aufbau

(b) 2. Hardware-Aufbau

Abb. A.2: Nicht ausreichende Hardware-Aufbauten für 3D-Scans.

B CD-Inhalt

Die beiliegende CD dieser Bachelorarbeit enthält folgende Ordner bzw. Dateien:

- `Bachelorarbeit_VRShop.pdf` ist die vorliegende Bachelorarbeit als PDF-Datei
- `Demonstration.mp4` ist ein kurzes Video zur Demonstration des Prototypen
- `Unity/` enthält das Projekt der prototypischen Implementation des VR-Shops:
  - `README.pdf` enthält Informationen zur Handhabung des VR-Shops, inklusive notwendiger Einstellungen für die Spracherkennung mit Cortana
  - `Articles/` beinhaltet die Informationen und 3D-Modelle der 26 Testartikel für den VR-Shop:
    - `vrshop.db` ist die SQLite-Datenbankdatei der probatorischen Artikelinformationen (Name, Preis, Produktbeschreibung und Bild)
    - Alle weiteren Ordner enthalten die eigentlichen 3D-Modelle der Artikel, wobei der jeweilige Ordnername der Artikel-ID innerhalb der Datenbank entspricht
  - `Assets/` enthält alle wichtigen Projektdateien:
    - `Scenes/BA_Scene.unity` ist die Unity-Szene des VR-Shops, welche in Unity geöffnet werden muss
    - `VRShop/` beinhaltet alle selbst entwickelten Unity-Dateien (hauptsächlich *C#*-Skripte unter `Scripts/`)
    - Die restlichen Assets werden lediglich zur visuellen Aufbereitung benötigt
  - Die restlichen Dateien des Projektordners wurden von Unity erstellt und sind obligatorisch für die ordnungsgemäße Funktionalität des Projektes

Glossar

3D-Modell Ein 3D-Modell bezeichnet die aus mathematischen Berechnungen entstehende Repräsentation einer beliebigen Oberfläche in einem dreidimensionalen Raum. Dies geschieht in der Regel durch spezialisierte Software. In dieser Bachelorarbeit wird die Bezeichnung auch als Synonym für automatisch generierte 3D-Scans aus echten Objekten verwendet..

Artikelmonitor Eine Kachel (d.h. ein konkreter Artikel) in der Artikelwand.

Artikelwand Allgemeine Bezeichnung für die im VR-Shop zylinderförmige Anordnung von Kacheln um den Anwender herum, die die einzelnen Artikel für ein Suchergebnis mit einem Vorschaubild darstellen. Beim Auswählen eines dieser Wände wird das entsprechende 3D-Modell geladen sowie die Möglichkeit, den Artikel in den Warenkorb zu legen.

HTC Vive Ein von HTC in Zusammenarbeit mit Valve Corporation entworfenes HMD. Besonderer Schwerpunkt bei diesem Lösungsansatz ist die Benutzung von sog. "Lighthouses" – Tracking-Stationen auf Infrarotbasis – mit denen die Position des Anwenders sowohl vom Headset, als auch von den zwei mitgelieferten Controllern bestimmt wird.

Oculus Rift Ein von Oculus VR entwickeltes und mittlerweile von Facebook Inc. aufgekauftes HMD..

Unity Eine von Unity Technologies bereitgestellte Spiel-Engine für grafische und interaktive 3D-Anwendungen. Die Zielplattformen sind dabei hauptsächlich Windows PCs..

VR-Shop Allgemeine Bezeichnung für den zu realisierenden Shop in Virtual Reality im Rahmen dieser Bachelorarbeit..

Literaturverzeichnis

[1] 3DFlow: *3DF Zephyr User manual*. 2013. – URL http://3dflow.net/zephyr-doc/3DF%20Zephyr%20Manual%204.000%20English.pdf. – (Abgerufen am 25.10.2018)

[2] Abrash, Michael: *Latency – the sine qua non of AR and VR*. 2012. – URL http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr. – (Abgerufen am 22.11.2018)

[3] Abrash, Michael: *Steam Dev Days | What VR Could, Should, and Most Certainly Will Be in Two Years*. 2014. – URL http://media.steampowered.com/apps/abrashblog/Abrash%20Dev%20Days%202014.pdf. – (Abgerufen am 15.10.2018)

[4] Beard, David V. ; II, John Q. W.: Navigational techniques to improve the display of large two-dimensional spaces. In: *Behaviour & Information Technology* 9 (1990), Nr. 6, S. 451–466. – URL https://doi.org/10.1080/01449299008924259

[5] Bernardini, Fausto ; Rushmeier, Holly: The 3D Model Acquisition Pipeline. In: *Computer Graphics Forum* 21 (2002), Nr. 2, S. 149–172. – URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.00574

[6] Buhr, Mathias ; Pfeiffer, Thies ; Reiners, Dirk ; Cruz-Neira, Carolina ; Jung, Bernhard: *Echtzeitaspekte von VR-Systemen*. Kap. 7, S. 197–198. In: Dörner, Ralf (Hrsg.) ; Broll, Wolfgang (Hrsg.) ; Grimm, Paul (Hrsg.) ; Jung, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR)*, Springer Berlin Heidelberg, 2013. – URL https://doi.org/10.1007/978-3-642-28903-3

[7] Captaindistance: *Image captured by a 3D-TOF-camera*. 2008. – URL https://de.wikipedia.org/wiki/Datei:TOF_Kamera_3D_Gesicht.jpg. – (Abgerufen am 24.10.2018)

[8] Crider, Michael: *Oculus Rift vs. HTC Vive: Which VR Headset Is Right for You? – The Vive Has Better Tracking Technology.* 2017. – URL https://www.howtogeek.com/246333/oculus-rift-vs.-htc-vive-which-vr-headset-is-right-for-you. – (Abgerufen am 13.09.2018)

[9] Curless, Brian: From range scans to 3D models. In: *ACM SIGGRAPH Computer Graphics* 33 (1999), nov, Nr. 4, S. 38–41. – URL https://doi.org/10.1145/345370.345399

[10] Dörner, Ralf ; Geiger, Christian ; Oppermann, Leif ; Paelke, Volker: *Interaktionen in Virtuellen Welten.* Kap. 6, S. 158–160. In: Dörner, Ralf (Hrsg.) ; Broll, Wolfgang (Hrsg.) ; Grimm, Paul (Hrsg.) ; Jung, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR)*, Springer Berlin Heidelberg, 2013. – URL https://doi.org/10.1007/978-3-642-28903-3

[11] Dörner, Ralf ; Steinicke, F.: *Präsenz und Immersion.* Kap. 2, S. 46. In: Dörner, Ralf (Hrsg.) ; Broll, Wolfgang (Hrsg.) ; Grimm, Paul (Hrsg.) ; Jung, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR)*, Springer Berlin Heidelberg, 2013. – URL https://doi.org/10.1007/978-3-642-28903-3

[12] Facebook Technologies, LLC.: *Oculus Audio SDK Guide | Oculus Hardware Capabilities.* 2018. – URL https://developer.oculus.com/documentation/audiosdk/latest/concepts/audiosdk-hardware. – (Abgerufen am 09.10.2018)

[13] Frayne, Shawn: *The Microwave: a Color 3D Scanner for Small Objects.* 2014. – URL https://www.instructables.com/id/The-Microwave-A-Color-3D-Scanner-for-Small-Objects. – (Abgerufen am 25.10.2018)

[14] Harbaugh, Ashley: *Ecommerce Website Requirements: Does Your Ecommerce Platform Have What Customers Need?* 2016. – URL https://us.hitachi-solutions.com/blog/ecommerce-website-requirements. – (Abgerufen am 27.10.2018)

[15] HTC Coorperation: *Produktbeschreibung der HTC Vive.* 2018. – URL https://www.vive.com/de/product. – (Abgerufen am 09.10.2018)

[16] Jung, Bernhard ; Vitzthum, Arnd: *Virtuelle Welten*. Kap. 3, S. 66–67. In: Dörner, Ralf (Hrsg.) ; Broll, Wolfgang (Hrsg.) ; Grimm, Paul (Hrsg.) ; Jung, Bernhard (Hrsg.): *Virtual und Augmented Reality (VR / AR)*, Springer Berlin Heidelberg, 2013. – URL https://doi.org/10.1007/978-3-642-28903-3

[17] LaViola, Joseph J.: A discussion of cybersickness in virtual environments. In: *ACM SIGCHI Bulletin* 32 (2000), jan, Nr. 1, S. 47–56. – URL https://doi.org/10.1145/333329.333344

[18] Lee, Kun C. ; Chung, Namho: Empirical analysis of consumer reaction to the virtual reality shopping mall. In: *Computers in Human Behavior* 24 (2008), Nr. 1, S. 88–104. – URL http://www.sciencedirect.com/science/article/pii/S0747563207000155. – ISSN 0747-5632

[19] Lievendag, Nick: *Artec Eva 3D Scanner Review*. 2018. – URL https://3dscanexpert.com/artec-eva-3d-scanner-review/. – (Abgerufen am 22.10.2018)

[20] Matulef, Jeffrey: *Valve is making three "full"VR games – Eurogamer.net*. 2017. – URL https://www.eurogamer.net/articles/2017-02-10-valve-is-making-three-fully-fledged-vr-games. – (Abgerufen am 09.12.2018)

[21] Meidl, Oliver: *Design globaler Webshop-Lösungen*. S. 44–49. In: *Globaler Webshop*, Springer Fachmedien Wiesbaden, 2015. – URL https://doi.org/10.1007/978-3-658-08327-4

[22] Motley, Darryl: *How to Scan Dark, Shiny, or Clear Surfaces with a 3D Scanner [With Video Demo]*. 2017. – URL https://gomeasure3d.com/blog/scan-dark-shiny-clear-surfaces-3d-scanner-video-demo. – (Abgerufen am 11.12.2018)

[23] Niehorster, Diederick C. ; Li, Li ; Lappe, Markus: The Accuracy and Precision of Position and Orientation Tracking in the HTC Vive Virtual Reality System for Scientific Research. In: *i-Perception* 8 (2017), Nr. 3, S. 20. – URL https://doi.org/10.1177/2041669517708205

[24] Pyka, Martin: *Berechnung von Rotationen*. 2017. – URL https://www-user.tu-chemnitz.de/~heha/viewchm.php/hs/SelfDXD.chm/directxgraphics/theorie/dg_ber.html. – (Abgerufen am 13.09.2018)

[25] Salvi, Anthony: *Go Scan the World! Photogrammetry with a Smartphone*. 2016. – URL https://www.allegorithmic.com/blog/go-scan-world-photogrammetry-smartphone. – (Abgerufen am 24.10.2018)

[26] Schou, Torben ; Gardner, Henry J.: A Wii remote, a game engine, five sensor bars and a virtual reality theatre. In: *Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artifacts and environments*, ACM Press, 2007, S. 232–234. – URL https://doi.org/10.1145/1324892.1324941

[27] Survios: *Simon Says: Know Your Controllers – HTC Vive Controls*. 2017. – URL https://survios.com/rawdata/guide/?doing_wp_cron=1537198037.0668509006500244140625. – (Abgerufen am 17.09.2018)

[28] Unity Technologies: *Unity - Manual: Working in Unity*. 2018. – URL https://docs.unity3d.com/Manual/UnityOverview.html. – (Abgerufen am 21.11.2018)

[29] Vulture19: *9.12.17 Coordinate measuring machine*. 2009. – URL https://de.wikipedia.org/wiki/Datei:9.12.17_Coordinate_measuring_machine.png. – (Abgerufen am 24.10.2018)

[30] Wendholt, Birgit: *Präsentation - Navigation - Interaktion*. 2018. – (Abgerufen am 01.11.2018)

[31] Wiora, Georg: Optische 3D-Messtechnik : Präzise Gestaltvermessung mit einem erweiterten Streifenprojektionsverfahren. (2001), S. 3–10. – URL http://archiv.ub.uni-heidelberg.de/volltextserver/1808

[32] Zeng, Liang: Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition) by B. Shneiderman and C. Plaisant. In: *International Journal of Human–Computer Interaction* 25 (2009), Nr. 7. – URL https://doi.org/10.1080/10447310903187949

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Ort	Datum	Unterschrift im Original