

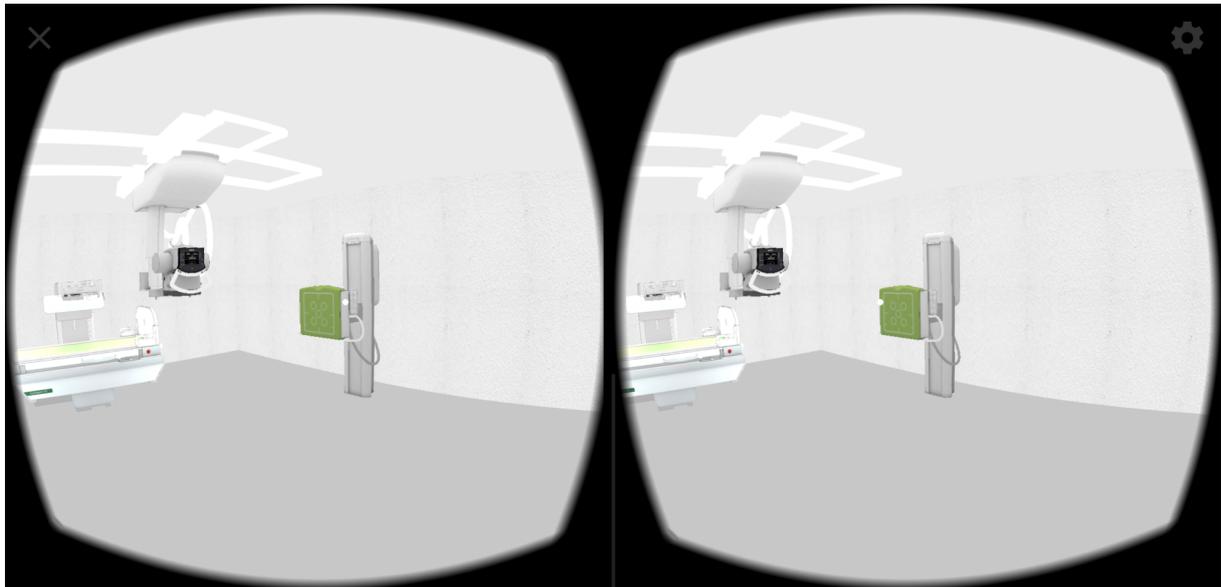


HOCHSCHULE LANDSHUT
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN

Fakultät für Elektrotechnik und Wirtschaftsingenieurwesen

Masterarbeit zum Thema

Rapid Prototyping Umgebung für VR-Umgebungen
in der Medizintechnik – Konzeptionierung,
Entwicklung und Test



vorgelegt von
Artur Maleta
aus Landshut

Eingereicht:

6.02.2018

Betreuer: Prof. Dr. Stefanie Remmele



HOCHSCHULE LANDSHUT
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN

Fakultät für Elektrotechnik und Wirtschaftsingenieurwesen

Erklärung zur Masterarbeit

(gemäß § 9 d, Abs. 3 APO)

Name, Vorname der/des

Student(in):.....

Hochschule Landshut

Fakultät Elektrotechnik und Wirtschaftsingenieurwesen

Hiermit erkläre ich, dass ich die Arbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benützt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

.....

(Datum)

.....

(Unterschrift der/des Student(in)en)



HOCHSCHULE LANDSHUT
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN

Fakultät für Elektrotechnik und Wirtschaftsingenieurwesen

FREIGABEERKLÄRUNG DER/DES STUDENT(IN)EN

Name, Vorname der/des Studentin/en:

.....

Hiermit erkläre ich, dass die vorliegende Masterarbeit in den Bestand der Hochschulbibliothek aufgenommen werden kann und

ohne Sperrfrist

oder nach einer Sperrfrist von

- 1 Jahr
- 2 Jahren
- 3 Jahren
- 5 Jahren oder länger

über die Hochschulbibliothek zugänglich gemacht werden darf.

.....

(Datum) (Unterschrift der/des Student(in)en)

Inhaltsverzeichnis

1 Ziel dieser Arbeit.....	1
2 Einleitung.....	1
3 Methode.....	3
3.1 Anforderungsdefinition.....	3
3.2 Systemspezifikation.....	5
3.3 Software-Architektur.....	6
3.3.1 Funktionsarchitektur.....	7
3.3.2 Spezifikation interner Komponenten.....	8
3.4 Entwicklung der Applikation Rapid Room Planing.....	19
3.4.1 Die GUI.....	19
3.4.2 Umsetzung interner Softwarekomponenten.....	20
3.4.3 Implementierung der 3D-Modelle.....	24
3.4.4 Umsetzung der Kollisionskontrolle.....	25
3.5 Systemtest.....	27
4 Ergebnis.....	29
4.1 Workflow der Applikation.....	29
4.2 Polygonreduktion der 3D-Modelle.....	33
4.3 Auswertung der Evaluation.....	34
5 Diskussion.....	38
Quellenverzeichnis.....	41
Abkürzungsverzeichnis.....	43
Abbildungsverzeichnis.....	44
Tabellenverzeichnis.....	46
Anhang A Quellcode.....	47
Anhang B Fragebogen der Evaluation.....	51
Anhang C Ergebnisse der Evaluation.....	54

1 Ziel dieser Arbeit

Das Ziel dieser Arbeit ist die Unterstützung der Raumeinrichtungsplanung medizinischer Behandlungsräume für die Installation technischer Medizinprodukte. Durch eine realitätsnahe Visualisierung der Raumplanung soll die Vorstellungskraft der Anwender unterstützt werden. Im Einsatzfeld soll die Kommunikation zwischen Vertriebsmitarbeitern und Kunden begünstigt und eine Diskussionsgrundlage für unterschiedliche Varianten einer Raumplanung geschaffen werden. Dazu soll ein computergestütztes System konzipiert werden, welches die Einhaltung der Distanzvorgaben von Einrichtungsgegenständen fördert. Die Anwender sollen eine Hilfestellung bei der Verifizierung der Raumplanung erhalten und Änderungen an der Raumplanung vornehmen können.

Der Fokus wird in Zusammenarbeit mit der Firma Philips Medical Systems DMC GmbH auf die Raumplanung und Einrichtung von Produkten aus dem Diagnostic X-Ray Portfolio gelegt. Zwei Produkte – dabei handelt es sich um Durchleuchtungsgeräte aus der Medizintechnik – sollen bei der Raumplanung zur Auswahl verfügbar sein. Des Weiteren soll die Machbarkeit eines Rapid-Roomplanning-Ansatzes auf Basis von Mobilgeräten untersucht werden.

Das System wird aus einem Smartphone mit der Erweiterung zu einem HMD (Head Mounted Display) und der Entwicklung einer Smartphone-Applikation aufgebaut. Das bedeutet, dass der Prototyp dieser Anwendung einen Untersuchungsgegenstand für den Einsatz mobiler VR-Technologie im unternehmerischen Kontext darstellen wird.

Aspekte der Usability sollen einen intuitiven Umgang mit dem System gewährleisten und die Implementierung von 3D-Modellen einen Eindruck über die Produkte verschaffen, indem bereitgestellte 3D-Modelle in der Smartphone-Anwendung zur Raumplanung verwendet und in einer realitätsnahen Visualisierung betrachtet werden können.

2 Einleitung

Im Investitionsgüterbereich nutzen Firmen unterschiedliche Instrumente in der B-2-B Kommunikation. Neben der eigenen Homepage stehen Direct-Mailing, der persönliche Verkauf im Außendienst und Messeausstellungen im Vordergrund. Der AUMA e.V. konnte 2017 durch eine Befragung von 501 Investitions- und Konsumgüterherstellern im B-2-B-Bereich eine hundertprozentige Beteiligung auf Messen feststellen[1].

Vordergründiger Faktor in der Kundenkommunikation ist hierbei die Klärung von Fragen zwischen Anbieter und Zielgruppe:

Ein potenzieller Kunde betritt den Messestand von Philips Medical Systems DMC und möchte sich über die Auswahl und Funktionalität der neusten X-Ray-Produkte informieren. Nach einem umfangreichen Gespräch mit dem Vertriebsmitarbeiter stimmt ihn das Produktangebot positiv. Einige Fragestellungen beschäftigen den

Kunden weiterhin. Kann das Produkt in unserem Behandlungsraum untergebracht werden? Integriert sich das Design in unser visuelles Konzept? Wie viel Platz nimmt das Produkt in Anspruch? Können alle durch das Produkt vorgegebene Mindestabstände eingehalten werden?

Diese Fragen können auf eine konventionelle Art und Weise durch den Einsatz von Printmedien bei Planungsgesprächen beantwortet werden. Produktkataloge informieren mit Designvorgaben, Datenblättern und technischen Zeichnungen über das Produkt und versuchen den Kunden für das Produkt zu gewinnen.

Einen neuen Lösungsansatz bietet die VR-Technologie. Durch den Begriff der Präsenz kann Problemstellungen gezielt begegnet werden. Dabei kann der Anwender durch die visuelle Isolation zu der realen Umgebung das Gefühl der Anwesenheit im virtuellen Raum entwickeln. Die darin dargestellte Umgebung kann durch den Aspekt der Ortsillusion die räumlichen Vorstellungskraft unterstützen. Die technischen Eigenschaften des VR-Displays begünstigen dabei den Grad der Involvierung. Die hierbei entstehende Immersion kann beispielsweise durch auditive Reize erhöht werden und das Gefühl der Präsenz bei Anwendern stärken[2].

Heute werden unterschiedliche computergestützte Systeme für die Visualisierung unterschiedlicher Problemstellungen herangezogen.

Im medizinischen Bereich finden Formen der interaktiven 3D-Visualisierung Anwendung. Auf Grundlage von CT-Datensätzen (Computertomographie) werden virtuelle 3D-Modelle generiert. Diese 3D-Modelle werden in navigationsgestützten Resektions-Systemen für präoperative Planungen eingesetzt, um die Präzision invasiver Eingriffe zu erhöhen[3]. In der Industrie werden immersive VR-Systeme zur Unterstützung der Fabrikplanung eingesetzt. Geplante Produktionshallen werden in statischen räumlichen Zusammenhängen oder dynamischen Raum- und Zeitabbildungen visualisiert und tragen zur Erkenntnisgewinnung in der Planungsphase bei[4].

Der Einsatz von VR-Anwendungen zu Kommunikationszwecken ist als noch wenig erforschter Bereich zu betrachten. In einer durch Statista erhobenen Umfrage hatten in Deutschland 78% der Befragten im Jahr 2017 noch keine VR-Brille genutzt[5].

Durch die Integration der VR in eine nicht-immersive Anwendung werden die Aspekte der Raumplanung und der realitätsnahen Visualisierung abgedeckt. Hinsichtlich der Zielsetzung konnte gegenwärtig keine Lösung mit den genannten Merkmalen gefunden werden.

3 Methode

Um eine mobile Lösung zu entwickeln, bedarf es der Konzeption eines Lösungssystems. In diesem Kapitel werden die Anforderungen an das System definiert und Lösungsansätze zur Erfüllung der Anforderungen bestimmt. Anschließend wird das Gesamtsystem spezifiziert. Dieses setzt sich aus einer Hardware und einer Software zusammen. Durch die Definition der Funktionsarchitektur sowie der internen Softwarekomponenten wird der Aufbau der Applikation dargelegt. Darauf aufbauend wird die Entwicklung der Software-Applikation umgesetzt. Zuletzt wird ein Test des Gesamtsystems mittels einer Evaluation durchgeführt. Abbildung 1 zeigt das Konzept des Lösungssystems.

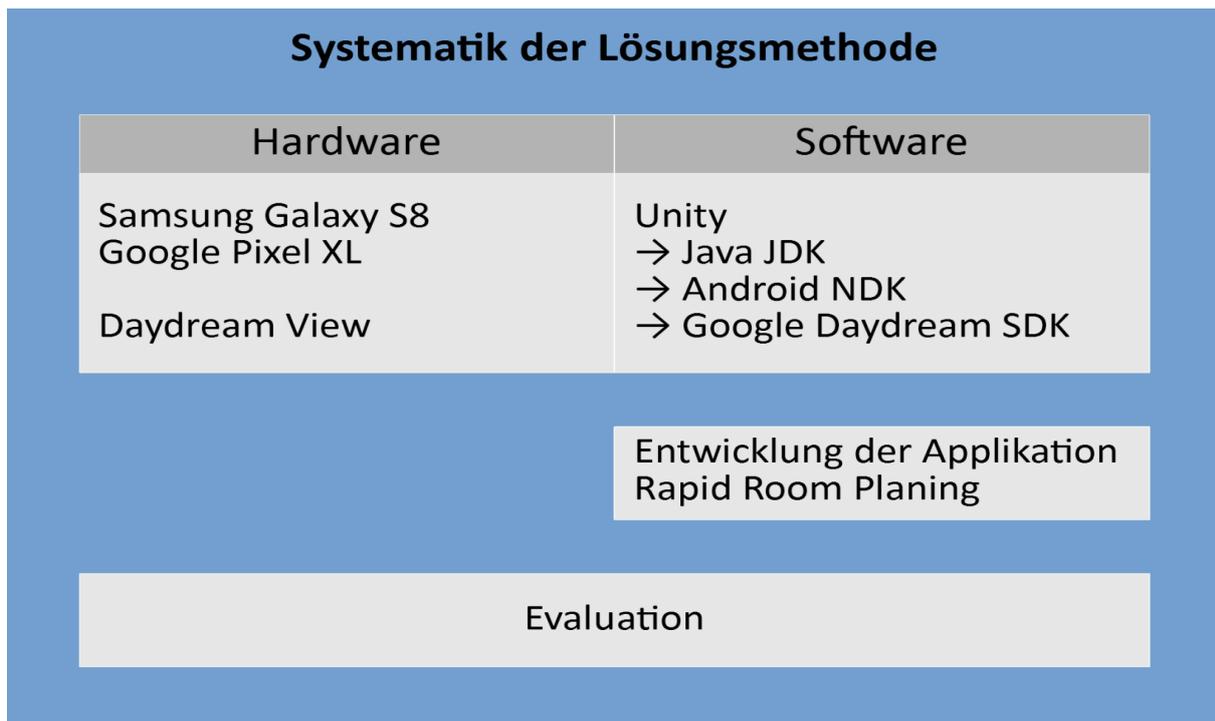


Abbildung 1: Konzept und Systematik der Methode

3.1 Anforderungsdefinition

Mit den vorliegenden Zielsetzungen soll der Anwender die Möglichkeit erhalten, eine gewünschte Raumplanung vorzunehmen und die Raumeinrichtung zu konfigurieren. Darauf folgend wird die Erstellung und Änderungen der Raumeinrichtung durch den Begriff Raumkonfiguration bezeichnet.

Zur Entwicklung des Konzepts werden zunächst Anforderungen definiert. Das zu entwickelnde System soll eine Planung sowie die Visualisierung unterstützen. Die funktionalen Anforderungen werden nach dieser Zielsetzung gegliedert und differenzieren die Anwendung in eine Planungsphase und eine Visualisierungsphase. Während in der

Planungsphase die Raumkonfiguration durchgeführt wird, soll das System den Anwender bei der Positionierung der Raumeinrichtung unterstützen und vor Kollisionen warnen. Nachdem die Raumkonfiguration durchgeführt wurde, wird die aktuelle Raumplanung in der Visualisierungsphase dargestellt. Dabei entsteht eine Abgrenzung zwischen der realitätsnahen Visualisierung durch VR und einer gemeinsamen Betrachtung der Raumplanung durch mehrere Personen. Um die Betrachtung aus verschiedenen Positionen zu ermöglichen, ist auch in der Visualisierungsphase eine Interaktion mit der Anwendung erforderlich.

Durch das Einsatzfeld des Systems entstehen ebenso Anforderungen an die Usability. Diese und die funktionalen Anforderungen an das Lösungssystem werden wie folgt formuliert:

Funktionale Anforderungen

- F1 Durchführung einer Raumkonfiguration.
 - F1.1 Implementierung von Gerätschaften aus dem Produktportfolio.
 - F1.2 Bestimmung der Raumgeometrie und der Raummaße.
 - F1.3 Signalisierung von Kollisionen unter Beachtung von Mindestabständen.
- F2 Visualisierung der Raumplanung aus verschiedenen Positionen.
 - F2.1 Darstellung der Raumplanung um eine Diskussionsgrundlage zu schaffen.
 - F2.2 Darstellung durch eine realitätsnahe Visualisierung.

Usability

- U1 Mobilität – Das Kommunikationsmedium soll ortsungebunden einsetzbar sein.
- U2 Akzeptanz bei Usern ohne oder mit wenig technischer Erfahrung. Daraus leitet sich die Anforderung an einen intuitiven Umgang mit der Applikation ab.

Um alle Anforderungen zu erfüllen werden Lösungsansätze formuliert. Eine Übersicht der Anforderungen und der geplanten Maßnahmen sind in Tabelle 1 dargestellt.

Tabelle 1: Gegenüberstellung der Projektanforderungen und der Lösungsansätze

Anforderung	Lösungsansatz
F1	→ Entwicklung einer Anwendung für die Raumplanung, mit Manipulation der Raumgeometrie und der Raummaße.
F1.1	→ Entwicklung einer Schnittstelle zur Produktauswahl.
F1.2	→ Bereitstellung von Benutzerinteraktionen für Raumkonfigurationen.
F1.3	→ Analyse der Vorgaben über Mindestabstände der Produkte und die Umsetzung einer Kollisionserkennung mit Kollisionsanzeige.

Anforderung	Lösungsansatz
F2	→ Entwicklung eines Verfahrens zur Festlegung der Betrachtungspunkte.
F2.1	→ Umsetzung einer perspektivischen Darstellung der Raumplanung.
F2.2	→ Präsentation der Raumplanung durch Virtual Reality.
U1	→ Konzeption und Entwicklung einer Smartphone-Applikation.
U2	→ Evaluierung durch Probanden.

3.2 Systemspezifikation

Um ein möglichst hochwertiges VR-Erlebnis bieten zu können, wird auf einen durch das Unternehmen Google definierten Standard mit der Bezeichnung Daydream[6] zurückgegriffen. Dieser Standard stellt die Leistungsfähigkeit der Smartphones für VR-Anwendungen sicher. Dabei wurden Hardwaremindestanforderungen sowie eine Mindestvoraussetzung an das Betriebssystem Android definiert, um erforderliche Schnittstellen bereitzustellen. Anforderungsgerechte Smartphones werden durch Daydream-Ready gekennzeichnet und in die Daydream-Plattform aufgenommen. Somit wird die Lauffähigkeit der Applikationen auf allen als Daydream-Ready gekennzeichneten Smartphones gewährleistet.

Der Daydream-Standard umfasst zudem eine 3D-Brille und einen Funk-Controller. Die 3D-Brille erweitert das Smartphone zu einem vollwertigen VR-Headset. Dieses kann für ein angenehmes VR-Erlebnis kabellos genutzt werden, jedoch konzeptuell ausschließlich in einer sitzenden oder stehenden Position. Somit wird eine physikalische Fortbewegung der Anwender nicht im Tracking-System des VR-Headsets erfasst. Der Funk-Controller realisiert die Schnittstelle zwischen dem Anwender und der VR-Umgebung. Der Controller muss dazu über die Bluetooth-Schnittstelle in der Daydream-App (Applikation) gekoppelt werden und erfüllt ferner die Funktion eines interaktiven Laserpointers in der VR-Umgebung.

Die Implementierung der Software erfolgt bei der Realisierung dieser Arbeit auf dem Samsung Galaxy S8 (Samsung Electronics Co., Ltd., Suwon, Südkorea) und dem Google Pixel XL (Google LLC, Mountain View, Kalifornien, Vereinigte Staaten). Beide Geräte erfüllen die Mindestanforderungen und sind dadurch auf der Daydream-Plattform zugelassen[7].

Für die Entwicklung der Applikation wurde die Entwicklungsumgebung Unity (Unity Technologies, 30 3rd Street, San Francisco, CA 94103, Vereinigte Staaten) gewählt. Die Entwicklungsumgebung ermöglicht es, interaktive Inhalte mit 3D-Computergrafik-Modellen zu erstellen und weiterführend diese zu einer VR-Anwendung zu exportieren[8]. Grundlage dieser Entscheidung ist die Erreichbarkeit verschiedener Zielplattformen, zu welchen unter anderem mobile Geräte mit Android-Betriebssystemen zählen.

Um Unity als Entwicklungsumgebung für Daydream-Applikationen nutzen zu können, sind zusätzliche Softwareanwendungen notwendig. Zugleich sind für eine einwandfreie Kompatibilität zu den gewählten Smartphones die Versionen der Softwareprogramme von entscheidender Bedeutung. So können Konflikte zwischen den genutzten Software-Entwicklungsumgebungen Java JDK und dem Android Studio durch die Bereitstellung unterschiedlicher Funktionsbibliotheksversionen entstehen. Eine Kompatibilität wird durch die folgend dargestellte Konfiguration der Entwicklungsumgebung erreicht:

- Unity 2017.2.0p4 (64-bit)[9]
Dabei handelt es sich um eine Stand-alone-Entwicklungsumgebung. Unity verfügt über eine Schnittstelle um sogenannte Assets (Erweiterungen, Tools, Funktionssammlungen etc.) einzubinden um diese für die Entwicklung zu nutzen.
- Java JDK 1.8.0_151-b12[10]
Das Java Development Kit ist ein eigenständiges Entwicklungsprogramm. Es stellt Java-Funktionsbibliotheken bereit, die von Unity beim Kompilervorgang genutzt werden.
- Android Studio 3.0.1[11]
Das Android Development Kit ist eine eigenständige Entwicklungsumgebung, um Anwendungen für das Android Betriebssystem zu erstellen. Das Programm stellt Schnittstellen und Bibliotheken bereit, auf die Unity beim Kompilervorgang zugreift um eine Android-Anwendung zu erstellen.
- Google Daydream SDK 1.70.0[12]
Das Daydream Software Development Kit wird als ein Unity-Asset bereitgestellt. Es erweitert die Entwicklungsumgebung Unity mit Schnittstellen und Funktionen für die Daydream-Plattform.

3.3 Software-Architektur

Kernbestandteile der Applikation sind die durch Unity bereitgestellte Komponenten, Module, Tools und Klassen. Zusätzlich werden zur Umsetzung der Anforderungen Softwarekomponenten entworfen und definiert. Diese Designentscheidung der Komponentendefinition erlaubt die Festlegung von Zuständigkeiten und eine hierarchische Organisation. Dadurch wird bei der Entwicklung dieser Anwendung eine ausbaufähige Architektur geschaffen.

Im Folgenden wird die Funktionsarchitektur der Anwendung dargestellt. Diese unterteilt die Anwendung in die Planungs- sowie die Visualisierungsphase. Der Aufbau wird ferner durch die Komponentenspezifikation umgesetzt und die Entwicklung der Anwendung realisiert.

3.3.1 Funktionsarchitektur

In der Applikation werden die Anforderungen F1, F2.1 und F2.2 durch logische Strukturierung in drei Applikationsmodi umgesetzt. Daraus ergeben sich drei Grundfunktionalitäten der Applikation.

Die Raumplanung und die Konfiguration der Einrichtung werden in einem primären Modus ermöglicht und als 2D-Modus definiert. Dabei kann der Benutzer durch eine an den Grundriss von Technischen Zeichnungen angelehnte Darstellungsweise die Raumkonfiguration vornehmen. Dies ermöglicht die Einstellung von Größe und Form des Raumes durch Berührungsgesten am Bildschirm des Smartphones. Durch das Berühren der Raumeckpunkte kann der Benutzer die Änderung aktivieren und durch das Ziehen an der Bildschirmoberfläche die Position des aktiven Raumeckpunktes verändern. Durch diese Aktion wird die Breite und Länge des Raumes geändert und die Anforderung F1.2 umgesetzt. Im 2D-Modus wird die Schnittstelle für die Implementierung der Produkte aus dem Produktportfolio sowie weiteren Einrichtungsgegenständen über einen Katalog realisiert. Dadurch wird dem Benutzer eine Auswahl der Raumeinrichtung bereitgestellt und begegnet Anforderung F1.1. Der Anwender hat die Möglichkeit, eine Auswahl der Raumeinrichtung zu treffen und weitergehend durch Verschieben und Drehen der Gegenstände die Position und Lage der gewählten Einrichtungsgegenstände zu bestimmen. In diesem Modus wird auch die Umsetzung der Kollisionserkennung implementiert und Anforderung F1.3 erfüllt. Der 2D-Modus erfordert Informationseingaben zu den Raummaßen sowie zur Platzierung der gewählten Raumeinrichtungsgegenstände. Dadurch wird in diesem Modus die Eingabeschnittstelle für den Anwender bereitgestellt.

Das Betrachten der Raumplanung aus verschiedenen Positionen, entsprechend Anforderung F2, wird als Funktion *Viewpoint* in den Katalog implementiert. Das Setzen und Positionieren der *Viewpoints* erfolgt entsprechend der Positionierung der Einrichtungsgegenstände, wobei durch die Rotation eines *Viewpoints* die Orientierung der Blickrichtung festgelegt werden kann.

Der 3D-Modus bietet eine umfassende Betrachtung der Raumplanung. Der Benutzer erlangt die Möglichkeit den entworfenen Raum perspektivisch in einem horizontal orbitalen 360°-Umlauf zu betrachten. Zudem wird eine Subfunktion im 3D-Modus implementiert. Diese ermöglicht eine Kontrollmöglichkeit der Raumplanung. Die durch den Benutzer gesetzten *Viewpoints* können in einer perspektivischen 3D-Ansicht betreten werden. Damit entfällt die Notwendigkeit, das VR-Headset zur Prüfung der Raumkonfiguration zu tragen. Zugleich wird die Möglichkeit der gemeinsamen Betrachtung geschaffen (Anforderung F2.1).

Anforderung F2.2 wird durch die Darstellung der Raumplanung in der VR erfüllt und in einem weiteren Modus abgegrenzt. Im VR-Modus kann der Anwender den konfigurierten Raum durch das VR-Headset betrachten. Gerade das ist verfolgtes Ziel der Applikation.

Durch die Funktionsunterteilung der Anwendung wird dem Nutzer eine Validierung der Raumplanung im 3D-Modus ermöglicht und die Betrachtung in der VR durch den VR-Modus der Applikation klar abgegrenzt. Abbildung 2 stellt die Applikation schematisch dar und zeigt den Aktivitätsfluss der Anwender auf.

Die drei Funktionsmodi können bei einer abstrakten Betrachtung dem System-Input und dem System-Output zugeordnet werden. Der 2D-Modus stellt dabei die Schnittstelle für den System-Input dar. Der Output des Systems ist der Raumbetrachtung gleichzusetzen und wird durch die Modi 3D sowie VR realisiert.

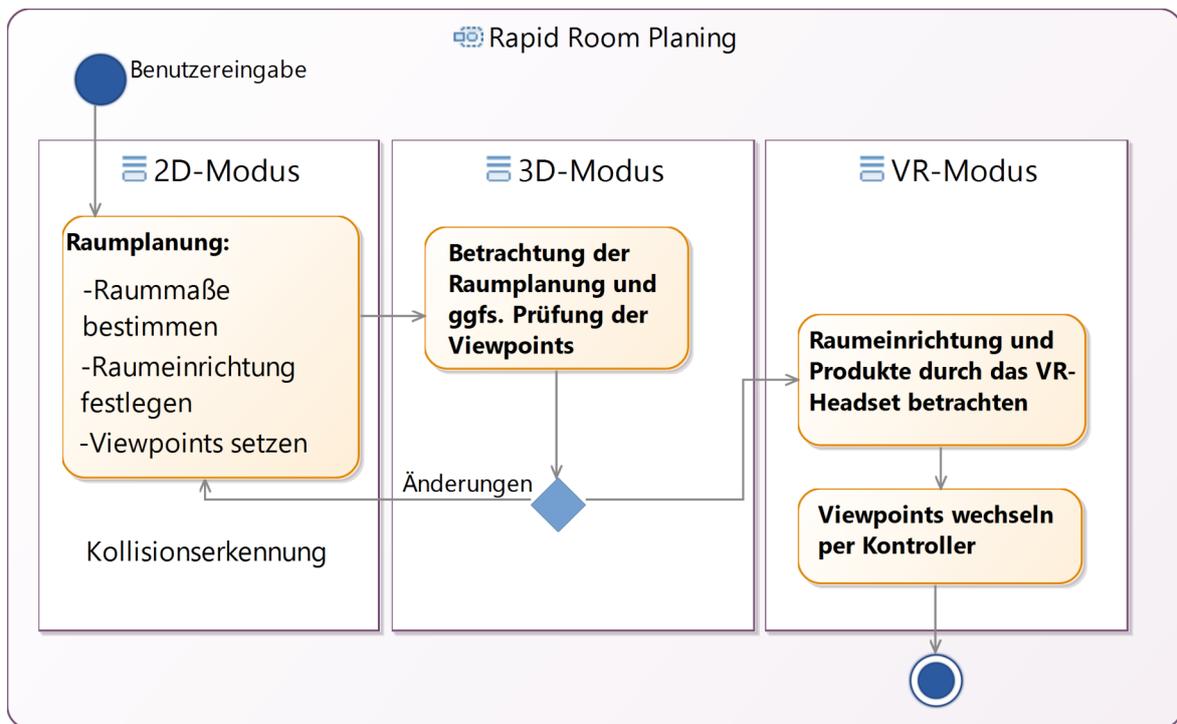


Abbildung 2: Aktivitätsfluss des Anwenders in den grundlegenden Funktionsmodi der Applikation Rapid Room Planing

3.3.2 Spezifikation interner Komponenten

Die Programmlogik der Applikation wird in Komponenten unterteilt und gewährleistet durch die Kohärenz der internen Komponenten zukünftige Funktionsanpassungen sowie -erweiterungen.

Tabelle 2 stellt eine abstrakte Definition der Komponenten dar. Die Zuständigkeit der Komponenten ist aus ihrer Zuordnung der Applikationsmodi entnehmbar.

Die Komponenten *Fingers* sowie *GvrController* sind in dieser Arbeit nicht entwickelt worden. Sie sind jedoch tragende Bestandteile des Lösungssystems und werden in dieser Ausarbeitung mitgeführt. Das Unity-Asset *Fingers – Touch Gestures for Unity* ist eine Erweiterung für die Entwicklungsumgebung. Das Plugin stellt Funktionsbibliotheken für die

Registrierung, Verwaltung sowie die funktionale Definition von Berührungsgesten bereit. Es wurde von Jeff Johnson entwickelt und ist im Unity-Asset-Store verfügbar[13].

Bei der Komponente *GvrController* handelt es sich um ein Modul aus der *Google VR SDK for Unity*[12]. Der *GvrController* implementiert den Funk-Controller von Daydream und stellt Informationen über Benutzerinteraktionen im VR-Modus bereit. Beide Komponenten werden zusammengefasst und bilden die Komponente Interaktionsverwaltung, welche weiter unten spezifiziert wird.

Tabelle 2: Definition interner Software-Komponenten zur Realisierung der Applikation Rapid Room Planing

Komponente	Kohärenz	Applikationsmodus	Abbildung durch Klasse(n)
Anwendungsverwaltung	Funktionen zur Verwaltung und Bereitstellung globaler Informationen und Daten	2D/3D/VR	GameController
Datenverwaltung	Verwaltung und Bereitstellung der Daten zur Raumausstattung	2D/3D/VR	DataPool ItemSO
Anwendungssteuerung	<ul style="list-style-type: none"> • Logik für Applikationsverlauf • Funktionen für interaktive Elemente der GUI • Konfiguration der Applikationsmodi 	2D/3D/VR	GameplayController
GUI* (Graphical user interface)	Interaktives Layout der Applikation	2D/3D	GUI
Interaktionsverwaltung	<ul style="list-style-type: none"> • Schnittstelle für Benutzerinteraktion • Definition und Verwaltung der Berührungsgesten • Funktionsbelegung des Funk-Controllers 	2D/3D/VR	InputController
→ Fingers*	Erkennung und Bereitstellung der Berührungsgesten	2D/3D	FingersScript
→ GvrController*	Schnittstelle zum Daydream-Funk-Controller	VR	GvrController
Raumverwaltung	<ul style="list-style-type: none"> • Raumrelevante Funktionsbereitstellung • Raumkonfiguration für 2D und 3D/VR-Modi 	2D/3D	RoomController Triangulator*
Kollisionssteuerung*	Erkennung und Signalisierung von Kollisionen	2D	CollisionDetector

(* Unity Komponente)

Unter Berücksichtigung etwaiger Zuständigkeiten werden funktionale Schnittstellen der einzelnen Komponenten durch Methoden für die differenzierten Anwendungsmodi definiert. Dabei wird der Zusammenschluss dieser Methoden als angebotene Schnittstelle deklariert und anderen Komponenten zur Verfügung gestellt.

Im Folgenden werden die Komponenten spezifiziert und im Nutzungskontext dargelegt. Die Abhängigkeiten der Komponenten und die definierten Schnittstellen können in Abbildung 3 eingesehen werden. Das Komponentendiagramm konzentriert sich auf die essentiellen Schnittstellen, um den Gesamtzusammenhang der definierten Komponenten zu verdeutlichen.

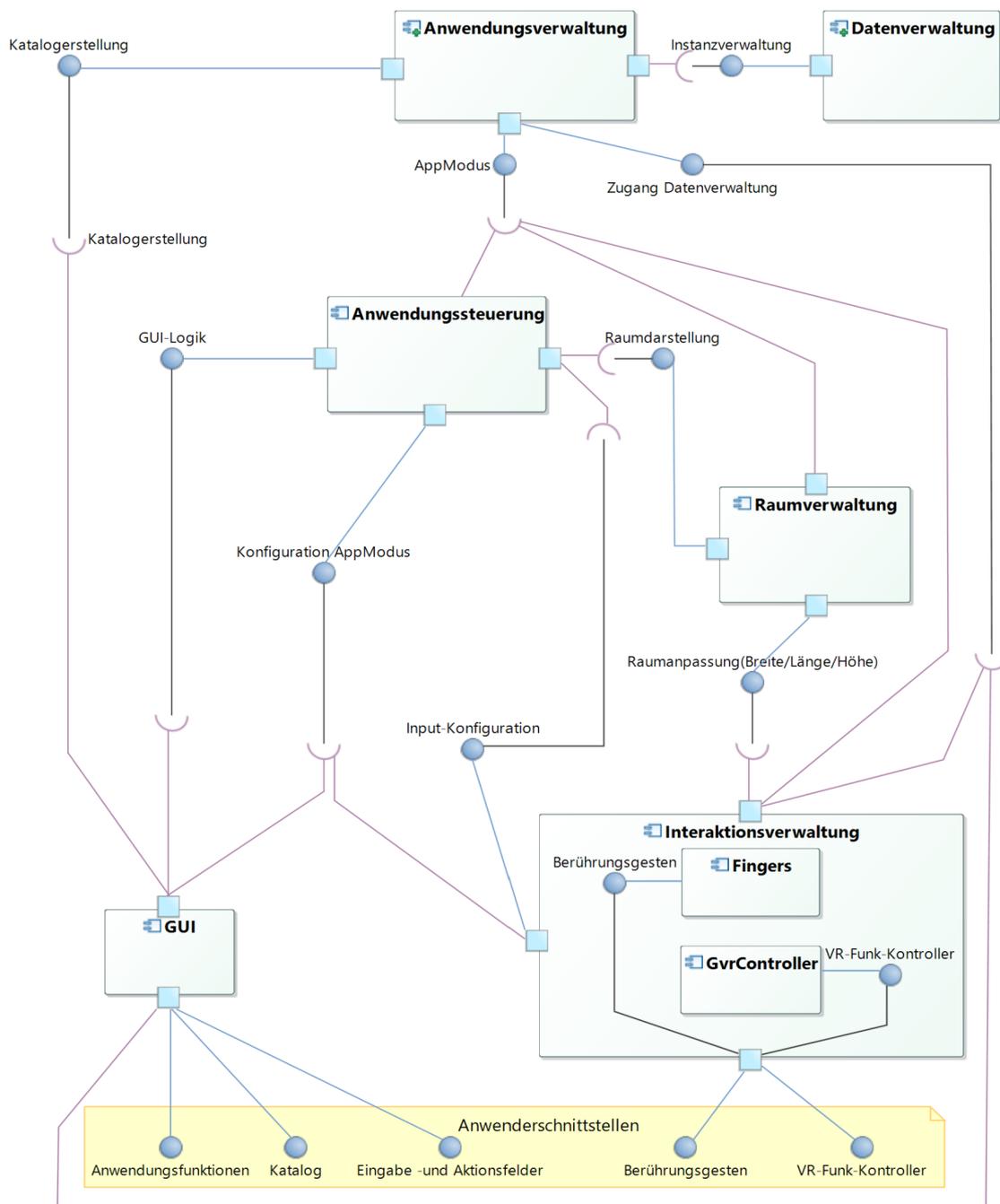


Abbildung 3: Übersicht interner Softwarekomponenten und Schnittstellen

Anwendungssteuerung

Mit der Komponente Anwendungssteuerung wird der Anwendungsverlauf organisiert. Sie kann als zentrale Steuereinheit der Applikation betrachtet werden. Dabei übernimmt die Anwendungssteuerung die Koordination über den Aktivitätsfluss des Anwenders und setzt diesen durch die Inanspruchnahme der angebotenen Schnittstellen der Komponenten GUI, Interaktionsverwaltung sowie Raumverwaltung um.

Die Anwendungssteuerung tritt indirekt mit dem Benutzer in Interaktion. Es werden notwendige Funktionen und Methoden der Komponente GUI durch die Schnittstelle *GUI-Logik* zur Verfügung gestellt. Die GUI bildet die Aktionsoberfläche für den Anwender und verbindet Benutzerinteraktionen mit der bereitgestellten Funktionslogik. Wird beispielsweise der Katalog durch den Anwender geöffnet, erfasst die GUI diese Aktion und ruft die entsprechende Methode der Anwendungssteuerung auf. Ausnahme hiervon ist das Einfügen der Einrichtungsgegenstände. Trifft der Nutzer eine Produkt- oder Einrichtungsauswahl, nutzt die GUI die durch die Anwendungsverwaltung angebotene Schnittstelle *Zugang Datenverwaltung* (vergleiche Abbildung 3).

Hauptmerkmal dieser Komponente ist die Konfiguration der Anwendung für die unterschiedlichen Applikationsmodi, wobei der Aspekt einer zentralen Steuereinheit zum Tragen kommt. Die Anwendungssteuerung bietet dazu über die Schnittstelle *Konfiguration AppModus* für jeden Applikationsmodus eine Methode. Primär werden diese über die GUI dem Anwender bereitgestellt und sind zudem über den Programmcode erreichbar. Das schafft die Möglichkeit den Applikationsmodus zu einer frei definierbaren Benutzerinteraktion zu wechseln. Beispielsweise kann für einen Wechsel von dem VR-Modus in den 2D-Modus diese Methode einer Funktionstaste des Daydream Funk-Controller zugeordnet werden.

Ist die Anwendungssteuerung angewiesen einen Modus zu wechseln, werden in der entsprechenden Methode folgende Schritte durchgeführt:

- 1) Prüfung des aktuellen Applikationsmodus in der Anwendungsverwaltung.
In Bezug auf den aktuellen Applikationsmodus wird weitergehend,
- 2) eine Konfiguration an den Elementen der GUI vorgenommen,
- 3) die Interaktionsverwaltung über die Schnittstelle Input-Konfiguration veranlasst benötigte Benutzerinteraktionen zu konfigurieren,
- 4) die Raumverwaltung per Schnittstelle Raumdarstellung angewiesen, den Raum für die Planung(2D-Modus) oder zur Betrachtung(3D- und VR-Modus) zu konfigurieren
und
- 5) der neue Applikationsmodus in der Anwendungsverwaltung verzeichnet.

Die Methode für die Konfiguration des 2D-Modus ist in Kapitel 3.4.2 Tabelle 4 einsehbar.

Für den Betrieb des VR-Modus werden weitere Dienste aus der *Google Daydream SDK* notwendig. Bei Aufforderung lädt die Anwendungssteuerung die für den VR-Modus benötigten Module. Abbildung 5 stellt die Anwendungssteuerung dar.

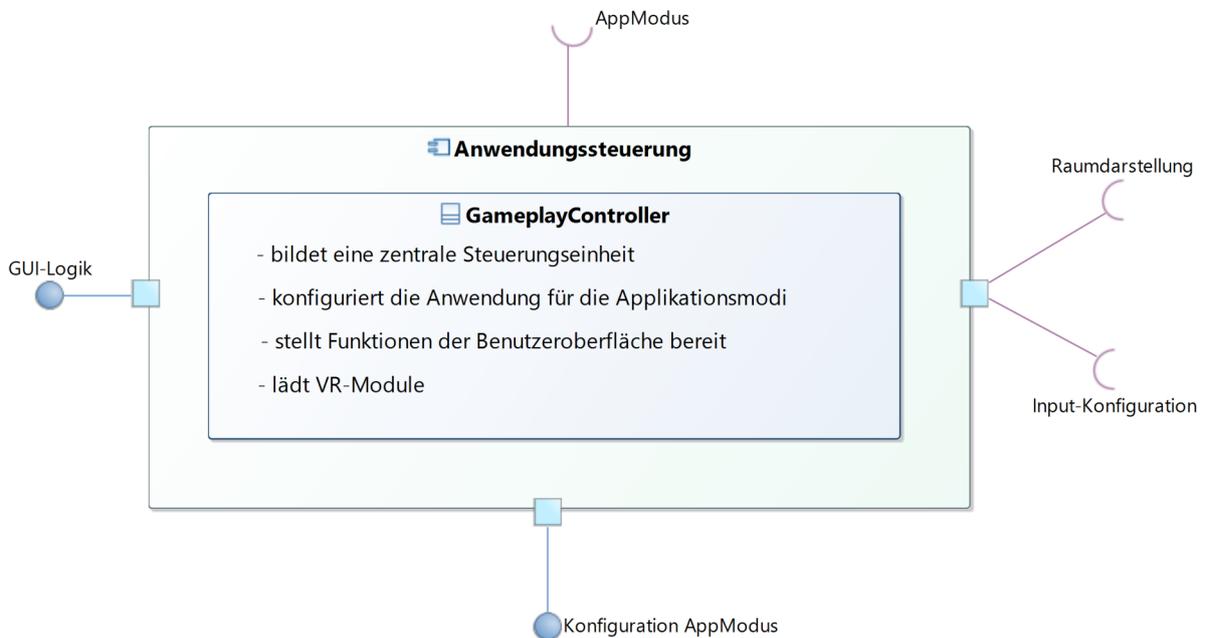


Abbildung 5: Spezifikation der Komponente Anwendungssteuerung

GUI

Die grafische Benutzeroberfläche bildet ein mit Funktionen belegtes Bediengerüst der Anwendung und ermöglicht es dem Nutzer mit der Applikation zu interagieren.

Mit der Komponente GUI wird das Layout der Benutzeroberfläche definiert, Aktions- und Eingabefelder konstruiert sowie Animationen für Änderungen am Layout festgelegt. Ferner beschränkt sich die GUI definitionsgemäß auf den 2D- und 3D-Modus der Applikation.

Hauptbestandteile der Komponente sind die Kopfleiste, der Katalog und die Fußleiste. In der Kopfleiste wird dem Benutzer eine Auswahl der drei Applikationsmodi durch drei Aktionsfelder angeboten. Zudem bietet die Kopfleiste eine weitere Schaltfläche zum Verlassen der Applikation an. Der Katalog wird als scrolling-fähige Seitenleiste aufgebaut, welche über eine an der linken Bildschirmseite platzierte Schaltfläche geöffnet werden kann. Die Seitenleiste stellt ihrerseits Sektionen, bestehend aus Aktions- und Textfeldern, in einer vertikalen Anordnung bereit. Jede dieser Sektionen repräsentiert ein Objekt, das in die Raumplanung durch den Anwender eingebunden werden kann. Bei diesem Objekt kann es sich um ein Produkt aus dem Portfolio, ein Inventarobjekt oder eine Funktion, wie das Setzen

der *Viewpoints*, handeln. Jede Sektion lässt sich entsprechend ihrer Funktionalität durch ein Miniaturbild sowie durch eine Bezeichnung identifizieren.

Der Katalog ist für die Raumplanung erforderlich und trägt zum System-Input in der Planungsphase bei. Demzufolge wird die Verfügbarkeit des Katalogs im 2D-Modus realisiert. Die Fußleiste bietet dem Nutzer kontextabhängige Funktionen an. In der Planungsphase lässt sich das Einstellungspanel sowie die Information über die Applikation aufrufen. Das Einstellungspanel ermöglicht dem Anwender, die Raummaße durch das Beschreiben der Eingabefelder für Breite, Länge und Höhe des Raums zu ändern. Des Weiteren kann die Anzahl der Raumecken verändert werden. Es besteht die Wahl zwischen vier oder sechs Raumecken, um die Raumform anzupassen. Im 3D-Modus werden dem Benutzer im Kontext der Raumbetrachtung lediglich im Falle vorhandener *Viewpoints* Aktionsfelder zur Verfügung gestellt, über die ein Betreten und Wechseln der *Viewpoints* ermöglicht wird. Abbildung 6 zeigt die Elemente der GUI im 2D und 3D-Modus.

Die Aktionsfelder der GUI nehmen Funktionen und Methoden der Schnittstelle *GUI-Logik* in Anspruch (Siehe Komponente Anwendungssteuerung). Die Manipulationen und Anpassungen der GUI werden ebenso durch die Anwendungssteuerung vollzogen.

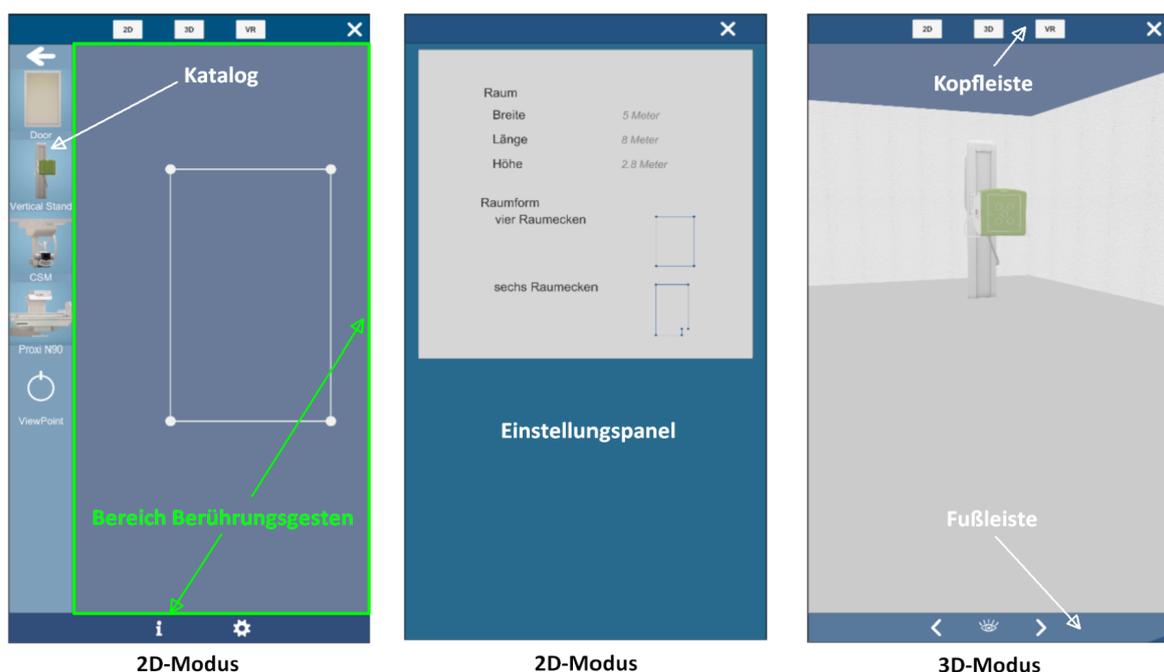


Abbildung 6: Spezifikation der Komponente GUI mit Kopfzeile, Katalog, Fußzeile und Einstellungspanel

Interaktionsverwaltung

Die Komponente Interaktionsverwaltung bildet trotz einer Abgrenzung zur GUI die Schnittstelle für Benutzerinteraktionen. Sie fasst die Komponenten *Fingers* und *GvrController* zu einer Komponente zusammen und ermöglicht dadurch eine einheitliche Definition sowie Integration der Benutzerinteraktionen abseits der GUI. Hier findet eine strikte Differenzierung der Benutzerinteraktionen in den unterschiedlichen Applikationsmodi statt. So bietet diese Komponente die Schnittstelle *Input-Konfiguration* der Anwendungssteuerung zur Konfiguration der Benutzerinteraktionen für den jeweiligen Applikationsmodus an. Eine weitere Unterteilung ist die Art der Benutzerinteraktionen. In den Modi 2D und 3D kommen Berührungsgesten zum Tragen. Der VR-Modus erfordert die Interaktionsfähigkeit via Funk-Controller.

Definition der Berührungsgesten

Die Subkomponente *Fingers* ist ein System für Berührungsgestenerkennung und deren Verwaltung. Sie bietet eine Schnittstelle für die Definition benötigter Berührungsgesten und ermöglicht die Determinierung der Funktionalität einzelner Gesten. An dieser Stelle erhält die Abgrenzung der Berührungsgesten von der GUI eine funktionale Relevanz. So werden die Berührungsgesten explizit für einen nicht durch die GUI abgedeckten Bildschirmbereich definiert und funktionale Überschneidungen für den Anwender ausgeschlossen (siehe Abbildung 6). Liegt der Subkomponente *Fingers* eine Definition der Gesten und deren Registrierung vor, findet eine interne Interpretation und Auswertung der vom Sensorbildschirm des Smartphones registrierten Ereignisse statt. Ist eine Interpretation entsprechend der definierten Bedingungen erfolgreich, wird die registrierte Gestendefinition ausgeführt. Die Applikationsmodi 2D und 3D erfordern unterschiedliche Funktionalitäten äquivalenter oder auch verschiedener Berührungsgesten. Die einzelnen Gesten können zudem eine kontextabhängige Funktionalität aufweisen. So ist beispielsweise im 2D-Modus die Pan-Geste von der Objektinteraktion abhängig. Eine Objektinteraktion liegt vor, wenn der Benutzer durch das Berühren eines Objekts per Tap-Geste eine Aktivierung durchführt. In diesem Kontext ist die Pan-Geste von ihrer Startposition abhängig. Beginnt diese an der Position des aktivierten Objekts, erfolgt eine Translation des aktivierten Objekts. Andernfalls wird eine Translation der gesamten Raumkonfiguration durchgeführt, was dem Verschieben des Bildschirmfokus entspricht. Die Definition kann in der Tabelle 3 eingesehen werden. Jede definierte Berührungsgeste ist in der Interaktionsverwaltung hinterlegt und zu unterschiedlichen Konfigurationsmethoden für die jeweiligen Applikationsmodi kumuliert. Die Interaktionsverwaltung führt entsprechend der Aufforderung durch die Anwendungssteuerung eine Konfiguration der Benutzerinteraktionen durch. Dabei werden die aktuellen Berührungsgesten bei der Subkomponente *Fingers* entfernt und die angeforderte Konfiguration registriert.

Tabelle 3: Definition der Berührungsgesten für die Applikationsmodi 2D und 3D

Berührungsgeste		Funktionalität im Applikationsmodus	
Bezeichnung	Interaktion mit Sensorbildschirm	2D	3D
Tap-Geste	Kurze Berührung mit einem Finger (Position: am Objekt)	Aktivierung / Deaktivierung einzelner Objekte für die Objektinteraktion	N.d.
Pan-Geste	Berührende Translation mit einem Finger (Position: am Objekt beginnend)	Translation aktivierter Objekte entlang der Ausführungsrichtung	Drehung der gesamten Raumkonfiguration in horizontaler(360°) und vertikaler(0-90°) Richtung um den Raummittelpunkt
	Berührende Translation mit einem Finger (Position: frei)	Translation der gesamten Raumkonfiguration entlang der Ausführungsrichtung	
Rotation-Geste	Berührende Rotation zweier Finger mit -oder gegen den Uhrzeigersinn (Position: frei)	Rotation aktivierter Objekte um die eigene Achse in positive oder negative Richtung	N.d.
Longpress-Geste	Anhaltende Berührung mit einem Finger (Position: am Raumeckpunkt)	Aktivierung der Raumeckpunkte zur Größenänderung des Raumes	N.d.
	Anschließende berührende Translation (Position: am Raumeckpunkt beginnend)	Translation des aktivierten und der benachbarten Raumeckpunkte	N.d.
Scale-Geste	Berührende Translation zweier Finger in eine zu sich entfernende Richtung (Position: frei)	Vergrößerung der gesamten Raumkonfiguration (zoom-in)	Entspricht 2D
	Berührende Translation zweier Finger in eine zusammenführende Richtung (Position: frei)	Verkleinerung der gesamten Raumkonfiguration (zoom-out)	Entspricht 2D
DoubleTap-Geste	Zweifache kurze Berührung mit einem Finger (Position:frei)	Bildschirmfokussierung auf Raummittelpunkt	N.d.

N.d.(Nicht definiert)

Definition der Interaktion am VR-Funk-Controller

Erhält die Interaktionsverwaltung die Anweisung, Benutzerinteraktionen für den VR-Modus zu konfigurieren, werden alle registrierten Berührungsgesten entfernt. Im VR-Modus nutzt die Interaktionsverwaltung die angebotene Schnittstelle der Subkomponente *GvrController*, um die durch den Benutzer am Funk-Controller ausgeführten Interaktionen auszuwerten. Der *GvrController* stellt dazu alle auf den Funk-Controller bezogenen Informationen in Echtzeit zur Verfügung. Dazu zählen die räumliche Position, Rotationsinformationen, sowie Informationen über die Nutzung der vorhandenen Tasten.

Für den VR-Modus wird eine Benutzerinteraktion wie folgt definiert und durch die Interaktionsverwaltung bereitgestellt: Wechsel zum nächsten *Viewpoint*.

Wird die Touch-Taste auf dem Funk-Controller gedrückt, erfragt die Interaktionsverwaltung bei der Komponente Datenverwaltung den nächsten *Viewpoint* und ändert die Position des Betrachters im virtuellen Raum. Hier kann der Nutzer die gesetzten *Viewpoints* in einer wiederkehrenden Abfolge betreten und die Raumkonfiguration aus unterschiedlichen Positionen betrachten. Sollte die Raumkonfiguration keine *Viewpoint* vorweisen, wird eine fest vereinbarte Position verwendet. Damit wird der Betrachtungspunkt auf einen Meter vor der südwestlichen Raumecke (2D-Modus) entfernt gesetzt. Abbildung 7 stellt die Komponente Interaktionsverwaltung mit den Subkomponenten *Fingers* sowie *GvrController* dar.

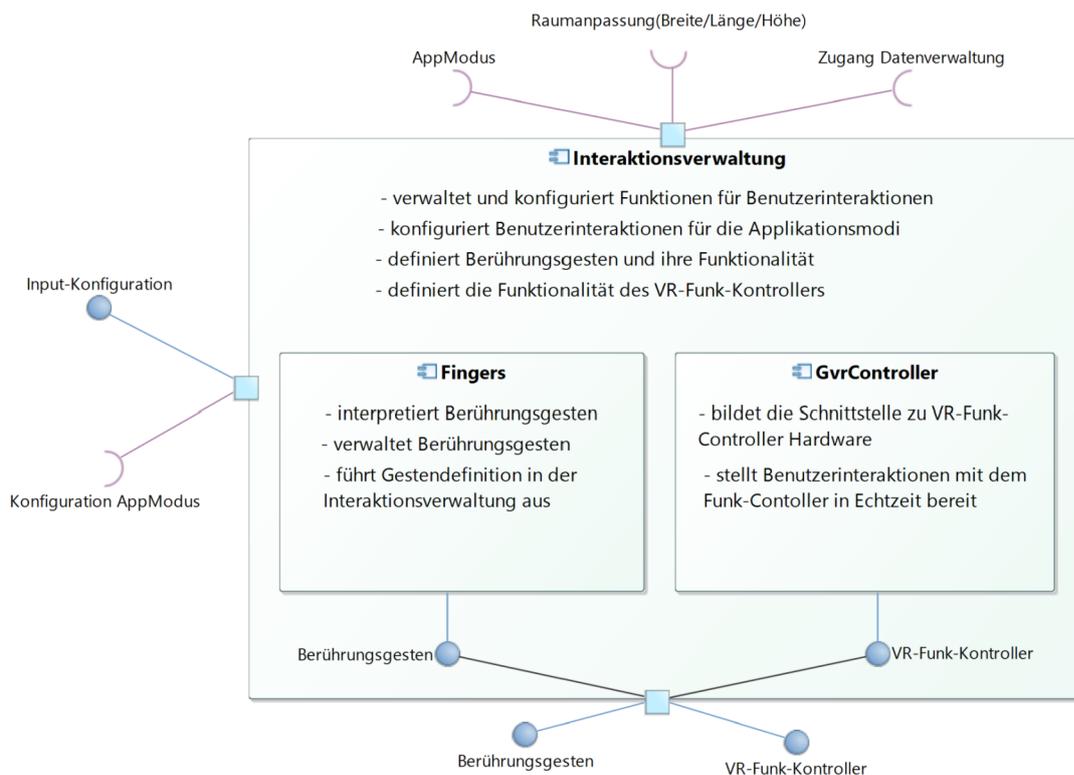


Abbildung 7: Spezifikation der Komponente Interaktionsverwaltung mit Subkomponenten *Fingers* und *GvrController*

Raumverwaltung

Durch die Raumverwaltung stehen der Anwendung Schnittstellen für die Raumkonfiguration zur Verfügung. Es wird primär zwischen zwei notwendigen Schnittstellen unterschieden. Die Schnittstelle Raumdarstellung bietet der Anwendungssteuerung Methoden zur Konfiguration der Raumrepräsentation an. Dabei wird streng zwischen Planungsphase(2D-Modus) und der Darstellungsphase(3D und VR-Modus) unterschieden. Im 2D-Modus ist eine zweidimensionale Raumdarstellung erforderlich. In der Darstellungsphase wird der Raum dreidimensional repräsentiert. Die Anwendungssteuerung nimmt somit beim Aufruf eines gewünschten Modus entsprechende Methoden der Raumverwaltung in Anspruch, welche die angeforderte Repräsentation der Raumkonfiguration erstellen. Über die Schnittstelle Raumanpassung werden Methoden zur Änderung der Raummaße bereitgestellt (siehe Abbildung 8). Auf diese Schnittstelle greift die Interaktionsverwaltung zu, um Informationen einer Änderung der Raumeckpunkte an die Raumverwaltung zu übermitteln. Dabei findet die Umsetzung dieser Änderung in der Komponente Raumverwaltung statt. Die Schnittstelle Raumanpassung wird ausschließlich in der Planungsphase in Anspruch genommen. In der Darstellungsphase werden definitionsgemäß keine Abänderungen am Raum vorgenommen, was die Relevanz dieser Schnittstelle auf den 2D-Modus beschränkt. Darüber hinaus bietet die Raumverwaltung Funktionen an, um die Anzahl der Raumecken zu ändern. So kann die Raumform angepasst werden, indem von vier auf sechs Raumecken gewechselt wird. Durch sechs Raumecken lassen sich L-förmige Räume oder auch rechtwinklige Vorsprünge an Raumecken abbilden. Die Raumverwaltung ermittelt zudem alle raumrelevanten Informationen und stellt diese über die Schnittstelle Rauminformationen zum Abruf bereit. Dazu zählen Raumbreite, -länge und -höhe sowie der Raummittelpunkt und die Raumeckpunkte.

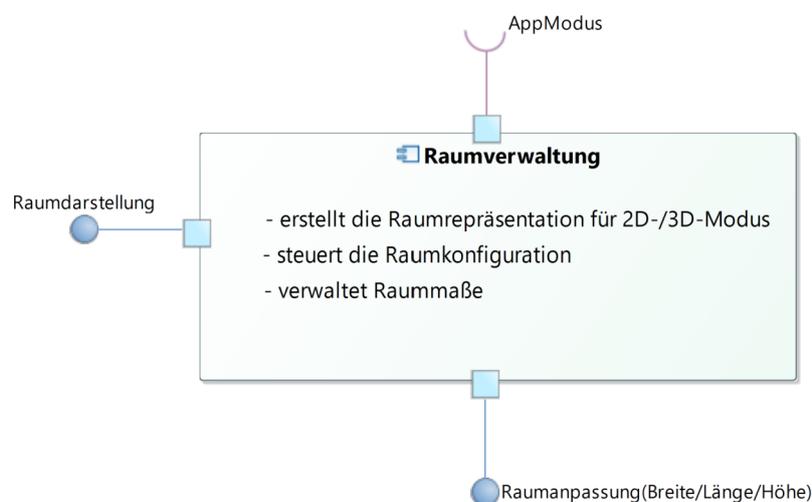


Abbildung 8: Spezifikation der Komponente Raumverwaltung

Kollisionssteuerung

Die Komponente Kollisionssteuerung steht in keinerlei Wechselwirkung mit den hier spezifizierten Komponenten. Sie hat zur Aufgabe, dem Anwender bei der Raumplanung das Überschreiten der Mindestabstände zu signalisieren. Im Kollisionsfall wird das zu positionierende Objekt farblich gekennzeichnet, bis der Anwender eine zulässige Positionsänderung vornimmt. Die Kollisionserkennung wird mit Entwicklungskomponenten der Entwicklungsumgebung Unity umgesetzt und unter Kapitel 3.4.4 erläutert.

3.4 Entwicklung der Applikation Rapid Room Planing

Unter Berücksichtigung des formulierten Ziels – der „Machbarkeit eines Rapid-Roomplanning Ansatzes auf Basis von Mobilgeräten“ – bedarf es einer Untersuchung der Realisierbarkeit der gegebenen Anforderungen durch die Systemlösung. Dazu wird die oben definierte Software-Architektur mit einer iterativen Vorgehensweise durch die Entwicklung der spezifizierten Softwarekomponenten umgesetzt. In der Entwicklungsumgebung Unity wird dazu das integrierte Konzept der Laufzeitumgebung genutzt. Hierbei werden die Softwarekomponenten in ihrer Entwicklungsphase in die Entwicklungsumgebung implementiert, durch interne Mechanismen vorab kompiliert und auf ihre definierte Funktionalität durch das Starten der Laufzeitumgebung überprüft.

Bei der Entwicklung wird der Fokus auf die Realisierung der Software-Architektur gesetzt, indem primär die grundlegenden Schnittstellen der Softwarekomponenten entwickelt und eine funktionale Basis bereitgestellt werden (vergleiche Abbildung 3). In weiteren Iterationen wird deren Funktionalität durch Differenzierung in weitere Subfunktionen erweitert. Dadurch erfolgt eine sukzessive Spezialisierung der Komponenten.

3.4.1 Die GUI

Die Realisierung der GUI erfolgt durch das UI-System (User Interface-System) in Unity. Es ermöglicht das Umsetzen einer Konzeption der grafischen Benutzeroberfläche und ist ein integraler Bestandteil dieser Entwicklungsumgebung.

Darin werden grundlegende Steuerelemente als 2D-Objekte für die Entwicklung einer GUI bereitgehalten. Zu diesen zählen Textfelder, Aktionsfelder, Eingabefelder, Felder für Darstellung von Grafikdateien oder Elemente um Scrolling bestimmter Sektionen zu realisieren. Eine Anordnung findet in einem *Canvas*-Objekt statt und ermöglicht eine Definition der Bildschirmauflösung, der Pixeldichte sowie die Bestimmung des Verhaltens bei Skalierung der Benutzeroberfläche.

Für die Interpretation der Benutzerinteraktionen ist das bereitgestellte *EventSystem* elementar und setzt im vorliegenden Fall Ereignisse am Sensorbildschirm in der Sektion der GUI in interne Input-Events um [14].

Entsprechend der Definition setzt sich die GUI aus mehreren Elementen zusammen und wird im Applikationsverlauf dynamisch organisiert. Die Kopfleiste wird hiervon nicht erfasst sondern liegt dem Anwender statisch zur Wahl der Applikationsmodi vor. Der Katalog und die Fußleiste werden hingegen dynamisch den zugrundeliegenden Benutzerinteraktionen angepasst. Diese Bereitstellung erfolgt hinsichtlich der UX (User Experience) mit zu vereinbarenden Effekten. Der Katalog wird durch ein Slide-In von der linken Bildschirmseite ausgehend präsentiert. Das Erscheinen des Einstellungspanels, welches über die Fußleiste aufgerufen werden kann, wird durch ein Slide-In von der unteren Bildschirmkante ausgehend festgelegt. Weitere über die Fußleiste aufrufbare Elemente werden dabei konsistent mit Effekten visualisiert. Bei der Umsetzung dieser Effekte wird auf das Animation-System der Entwicklungsumgebung zurückgegriffen. Das Animation-System bietet eine Schnittstelle, um zur Entwicklungszeit erstellte zeitabhängige Objektmanipulationen mit festgelegten oder programmierten Ereignissen auszulösen. Diese sogenannten Animationen können auf alle Objekte und somit etwa auf deren Merkmale sowie Attribute, Positions-, Rotations- und Skalierungsdaten angewendet und durch die Zeitabhängigkeit für die Erstellung von translatorischen Effekten genutzt werden.

Die Komponente Anwendungssteuerung stellt den Aktionsfeldern Funktionen bereit. Dabei wird beispielsweise das festgelegte Ereignis *Katalog öffnen* in einer Funktion der Anwendungssteuerung hinterlegt, welche wiederum einem Aktionsfeld zugeordnet und von diesem im Interaktionsfall aufgerufen wird.

3.4.2 Umsetzung interner Softwarekomponenten

Die Entwicklungsumgebung Unity stellt über eine Integration des Mono-Projekts einen .NET-Layer als Programmierschnittstelle bereit. Das Mono-Projekt ist eine Open-Source-Variante des von Microsoft entwickelten Frameworks .NET und gewährleistet eine plattformunabhängige Ausführung von Anwendungen, welche mit der .NET-Programmiersprache C# realisiert werden[15]. Die objektorientierte Programmiersprache C# wird bei der Entwicklung eingesetzt und bietet Zugriff auf Entwicklungsbibliotheken aus dem .NET-Framework sowie die darin entwickelten Funktionsbibliotheken für die Entwicklungsumgebung Unity.

Die Komponenten Anwendungsverwaltung, Anwendungssteuerung und Datenverwaltung erfüllen administrative Aufgaben. Sie werden primär in die Applikation implementiert. An dieser Stelle wird der modulare Aufbau der Anwendung fokussiert und die Anwendungssteuerung sowie die Raumverwaltung genauer dargelegt. Die Schnittstelle *Konfiguration AppModus* der Anwendungssteuerung stellt den drei Aktionsfeldern (2D / 3D / VR) der GUI-Kopfleiste drei Methoden bereit: LoadView2D, LoadView3D sowie LoadViewVR. Jede dieser Methoden nutzt weitere Konfigurationsmethoden der Schnittstellen Raumdarstellung und Input-Konfiguration, um den Applikationsmodus zu wechseln. Durch diesen hierarchischen Konfigurationsverlauf wird eine zentrale Konfigurationsschnittstelle geschaffen. Hier können weitere Funktionen neuer Komponenten in die Konfiguration

implementiert und beim Wechsel der Applikationsmodi aufgerufen werden. Um dies zu verdeutlichen ist in Tabelle 4 ein Auszug der Methode LoadView2D dargestellt. Der Aufbau der Methoden LoadView3D sowie LoadViewVR wird Analog zu LoadView2D gehalten und kann im Anhang A eingesehen werden.

Tabelle 4: Auszug der Methode LoadView2D aus der Komponente Anwendungssteuerung.
 Links: Quellcode. Rechts: Ablaufdiagramm

<pre> public void LoadView2D() { //configure mode-buttons _2D_Btn.interactable = false; //check if mode is not Loaded twice if (GameController.gameMode == GameController.GameMode._2D) return; // unload 3D-mode setup if (GameController.gameMode == GameController.GameMode._3D) { _3D_Btn.interactable = true; InputControllerScript.Instance. RemoveActivGestures(); if(GameController.database.isViewPoint) uiAnimator.SetTrigger("footer3dClose"); } // unload VR-setup by coroutine if (GameController.gameMode == GameController.GameMode._VR) { //configure mode-buttons _3D_Btn.interactable = true; _VR_Btn.interactable = true; StartCoroutine(UnloadDeviceVR("none")); AsyncOperation scene = SceneManager.UnloadSceneAsync("_VR"); SetupUiButtons(); } InputControllerScript.Instance. Config2DInput(); RoomController.Instance.ConfigRoom2D(); // GUI configuration _Cat_Btn.gameObject.SetActive(true); footerPanel2D.gameObject.SetActive(true); GameController.gameMode = GameController.GameMode._2D; } </pre>	<pre> graph TD Start(()) --> Config2D[Konfiguriere 2D-Modus] Config2D --> Deact2D[deaktiviere 2D-Button] Deact2D --> Is2D{ist 2D-Modus?} Is2D -- Ja --> End1(()) Is2D -- Nein --> Is3D{ist 3D-Modus?} Is3D -- Nein --> Act3D[aktiviere 3D-Button] Act3D --> RemGest[entferne aktive Berührungsgesten] RemGest --> Close3D[schlieÙe 3D-FuÙbleiste] Is3D -- Ja --> SetVP{sind Viewpoints gesetzt?} SetVP -- Nein --> Merge1{ } SetVP -- Ja --> Act3DVR[aktiviere 3D- und VR-Button] Act3DVR --> DeactVR[deaktiviere VR-Module] DeactVR --> RegGUI[registriere GUI-Funktionen] Merge1 --> RegGUI RegGUI --> Config2DInt[konfiguriere 2D-Interaktionen] Config2DInt --> Config2DRA[konfiguriere 2D-Raumdarstellung] Config2DRA --> ActCat[aktiviere Katalog-Button] ActCat --> Act2DF[aktiviere 2D-FuÙbleiste] Act2DF --> Set2D[setze 2D-Modus] Set2D --> End2(()) </pre>
--	--

Die Komponente Raumverwaltung bietet der Anwendungssteuerung die Methoden `ConfigRoom2D` und `ConfigRoom3D` über die Schnittstelle Raumdarstellung an. Die Raumverwaltung hält ein Referenzobjekt vor, welches die zweidimensionale Repräsentation einer Raumecke darstellt, und eine erste Definition der Raummaße über einen trivialen Raum mit vier rechtwinkligen Raumecken. Durch Anweisung der Anwendungssteuerung erzeugt die Funktion `ConfigRoom2D` zum Applikationsstart sowie beim Wechsel in den 2D-Modus die Raumdarstellung für den 2D-Modus. Eine rechteckige Anordnung der Raumecken wird anhand der vorliegenden Raummaße bestimmt. Die zweite Konfigurationsmethode `ConfigRoom3D` ist für die prozedurale Erstellung der dreidimensionalen Raumrepräsentation zuständig und nutzt dazu die Positionsinformationen der Raumecken sowie die Höheninformation des Raumes. Dabei entsteht die Notwendigkeit einer dynamischen Erzeugung des Raummodells. Denn der Anwender hat die Möglichkeit aus dem 3D-Modus in den 2D-Modus zu wechseln, um seine Raumplanung anzupassen. In diesem Fall ist das dreidimensionale Modell des Raumes nicht mehr maßgerecht und wird erneut generiert. In der Computergrafik werden Formen der 3D-Modelle durch ein *Mesh* (Polygonnetz) abgebildet. Ein *Mesh* wird aus Polygonen gebildet und durch dreidimensionale Vektoren dargestellt. Dazu bedarf es mindesten drei durch Vektoren repräsentierte *Vertices* (Punkte). Diese bestimmen die kleinste Teilmenge eines *Mesh*, ein *Triangle*. Ein Wandmodell wird durch vier *Vertices* festgelegt und muss im weiterem in *Triangles* unterteilt werden. Dabei legt die Reihenfolge bei der Indizierung der *Triangles* die Oberflächenseite eines Polygons fest. Die Polygonoberfläche wird in die Berechnung der Grafikdarstellung einbezogen um das Modell selbst, sowie Lichtreflektionen der Oberflächen zu bestimmen[2]. Eine Raumwand und ihre Oberflächenorientierung wird somit durch zwei *Triangles* bestimmt (Siehe Abbildung 9). Zwei Vektoren bestimmen hier eine Raumwand. Mit $Triangles_1=(0,3,1)$ und $Triangles_2=(0,2,3)$ wird die Oberfläche zur Raummitte festgelegt.

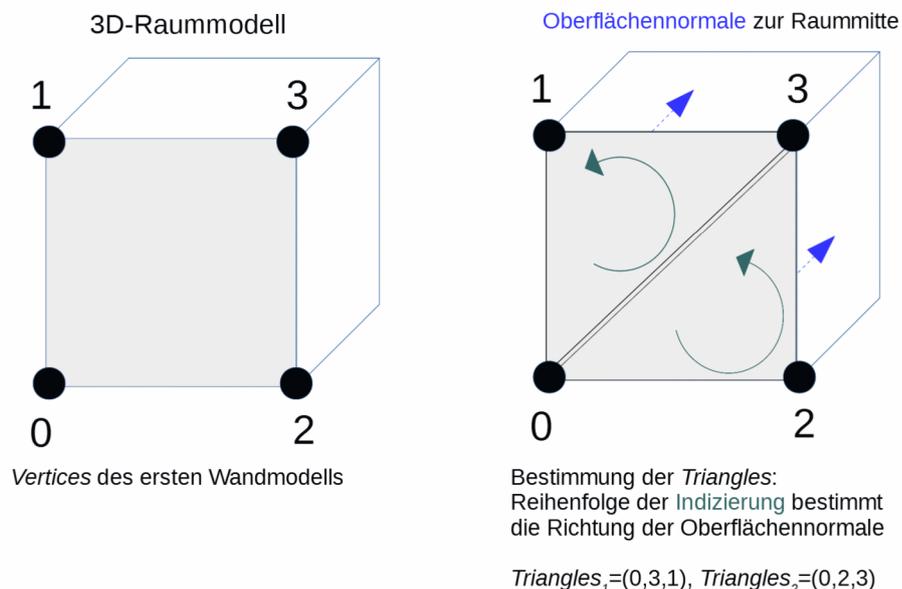


Abbildung 9: Triangulation des 3D-Modells einer Raumwand

Die Erstellung eines 3D-Modells der Raumplanung ist in drei Schritte unterteilt. Im ersten Schritt werden die Raumeckpunkte sukzessive interpretiert und planare Modelle der Raumwände erzeugt. Folgend wird ein Modell des Raumbodens erstellt, indem die Funktionen der Klasse *Triangulator*[16] genutzt werden. Im weiteren Schritt wird die Raumdecke durch Spiegelung des Raumbodens an der horizontalen Achse und unter Einhaltung der Raumhöhe positioniert. Die Anzahl der Raumeckpunkte kann durch den Anwender verändert werden. Um eine unbestimmte Anzahl von Raumecken bei der Erstellung des 3D Raummodells berücksichtigen zu können, wird ein Algorithmus definiert und in Abbildung 10 dargestellt.

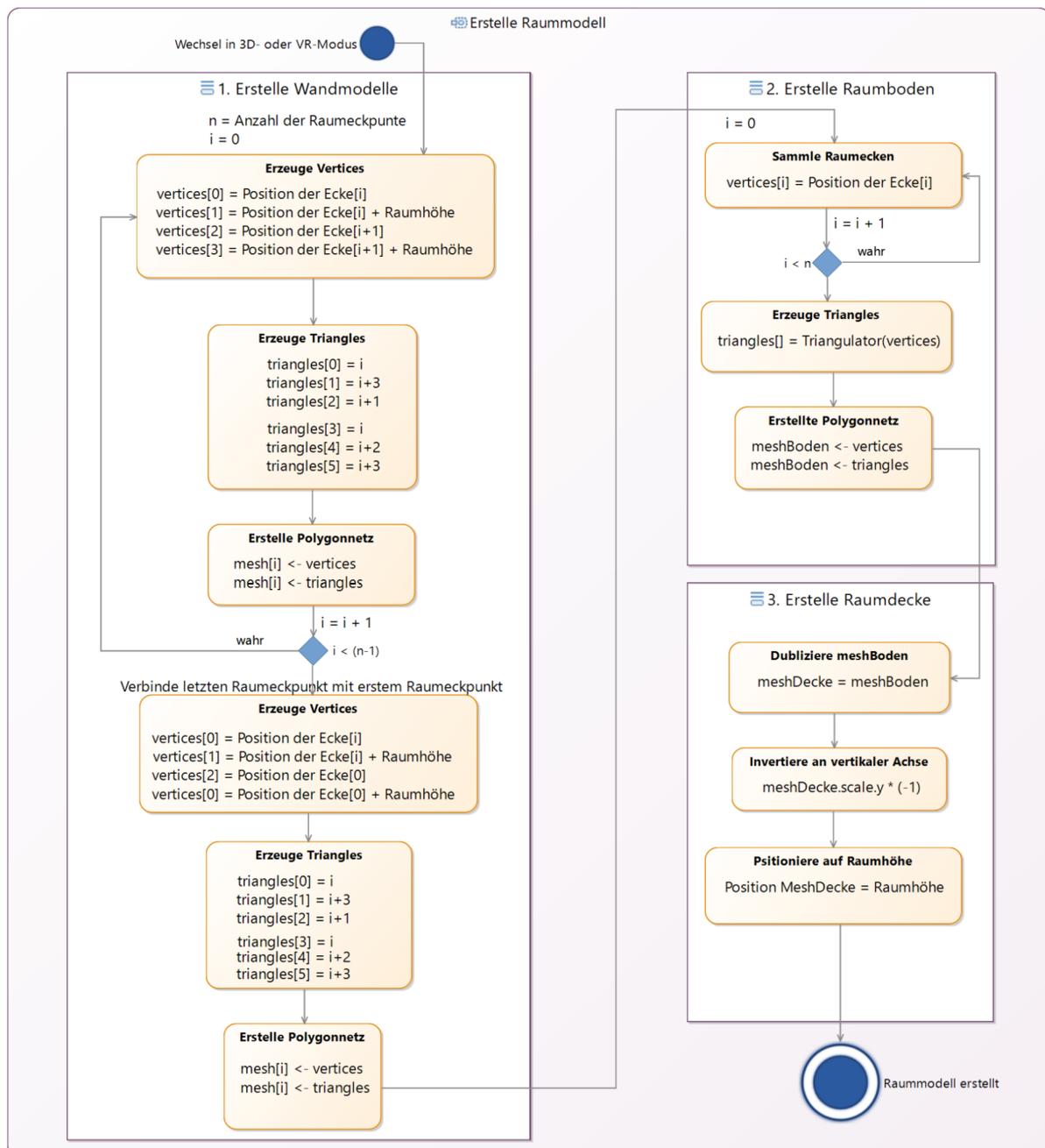


Abbildung 10: Algorithmus zur Erstellung von 3D-Raummodellen aus der Komponente Raumverwaltung

Bei der Erstellung des Raumbodens wird auf das Tool *Triangulator* zugegriffen. Damit kann eine beliebige Anzahl definierter *Vertices* trianguliert werden. *Triangulator* ist ein frei verfügbares Tool und kann im Anhang A eingesehen werden.

Die dreidimensionale Repräsentation der Raumplanung ist in der Darstellungsphase (3D/VR Modus) erforderlich. Sie wird durch die Raumverwaltung bei jedem Wechsel in die Darstellungsphase erstellt und ist somit dem 3D-Modus zugeordnet.

3.4.3 Implementierung der 3D-Modelle

Die 3D-Modelle der zu implementierenden Produkte wurden durch die Firma Philips bereitgestellt. Bei den Produkten handelt es sich um zwei vollwertige Fluoroskopiegeräte mit den Bezeichnungen ProxiDiagnost N90 und CombiDiagnost R90. Beide Produkte können durch zwei weitere Funktionsgeräte erweitert werden. Der VS (Vertical Stand) und der CSM (Ceiling Suspension CSM) bilden gemeinsam die Funktionalität eines konventionellen Röntgengeräts, können jedoch in Kombination mit ProxiDiagnost N90 sowie dem CombiDiagnost R90 als funktionserweiternde Geräte installiert werden. Abbildung 11 stellt das CombiDiagnost R90 in Kombination mit dem VS und dem CSM dar.



Abbildung 11: Fluoroskopiegeräte von Philips. *Links*: Vertical Stand. *Mitte*: Ceiling Suspension CSM. *Rechts*: CombiDiagnost R90 (Quelle: [25])

Die vorliegenden Modelle weisen unterschiedliche Formate und Eigenschaften auf. Das CombiDiagnost R90 liegt im 3DM-Dateiformat vor. Das Dateiformat basiert auf NURBS-Daten (Non-Uniform Rational B-Splines)[17] und bedarf für die Nutzung in der Computergrafik einer Konvertierung in ein polygonbasierendes Datenformat. Hier wird das FBX-Format (Filmbox) bei der Datenaufbereitung herangezogen und bietet native Unterstützung in der Entwicklungsplattform Unity. Die Modelle ProxiDiagnost N90, VS sowie CSM sind im FBX-Format zur Verfügung gestellt worden und enthalten eine texturierte Modelloberfläche.

Durch die Graduierung eines Polygonnetzes wird der Detailgrad der Oberflächen eines 3D-Modells bestimmt. Ein detailreicheres 3D-Modell erfordert mehr Berechnungsleistung und lastet die Hardwarekapazität der Grafikeinheit aus[2]. Die Umsetzung der Lösung basiert auf Smartphone-Hardware, welche gegenüber PC-Systemen eine geringere Grafikleistung aufweisen. Daraus ergibt sich die Notwendigkeit die Polygonanzahl der 3D-Modelle zu reduzieren.

Die Datenkonvertierung des CombiDiagnost R90-Modells erzeugt eine Datenmenge von 556,8 MegaByte und entspricht dem 4,47-fachen gegenüber dem FBX-Modell des ProxiDiagnost N90. Die VS- und CSM-Modelle liegen in der Datengröße unter 100 MegaByte und deuten auf eine für die Computergrafik unverhältnismäßig große Datenmenge des CombiDiagnost R90 hin (Siehe Kapitel 4.2 Polygonreduktion der 3D-Modelle).

Die Optimierung der 3D-Modelle wird in einer Testversion der 3D-Software Maya durchgeführt. Dabei werden vorwiegend planare Polygonstrukturen durch Polygonnetze geringerer Dichte ersetzt. Um den Erhalt radialer Geometrien zu gewährleisten, wird die Polygonreduktion durch manuelle Modellierungsvorgänge realisiert.

3.4.4 Umsetzung der Kollisionskontrolle

Die Kollisionserkennung ist dem 2D-Modus zugeordnet und findet in der Raumplanungsphase ihre Anwendung. Dadurch soll das Einhalten der Mindestabstände einzelner Produkte bei der Raumkonfiguration sichergestellt werden. Der Anwender wird dabei über ein visuelles Feedback auf die Nichteinhaltung der vorgegebenen Kollisionsdaten hingewiesen, indem die 3D-Modelle rot eingefärbt werden. Die Kollisionserkennung wird mit dem Unity eigenen *Physics*-System realisiert. Das System ermöglicht die Kollisionserkennung durch die dargelegten Komponenten und unter Berücksichtigung der folgend genannten Konfiguration (siehe Abbildung 12).

Um Objekte in die Physikberechnung einzubeziehen, wird jedem Objekt die *Rigidbody*-Komponente hinzugefügt. In Kombination mit der *Collider*-Komponente werden die Objekte in die Kollisionsberechnungen aufgenommen. Die *Rigidbody*-Komponente bietet die Möglichkeit, die Physiksimulation pro Objekt zu deaktivieren. So können Objekte ohne Einfluss von Gravitations- und Trägheitsberechnungen bewegt werden.

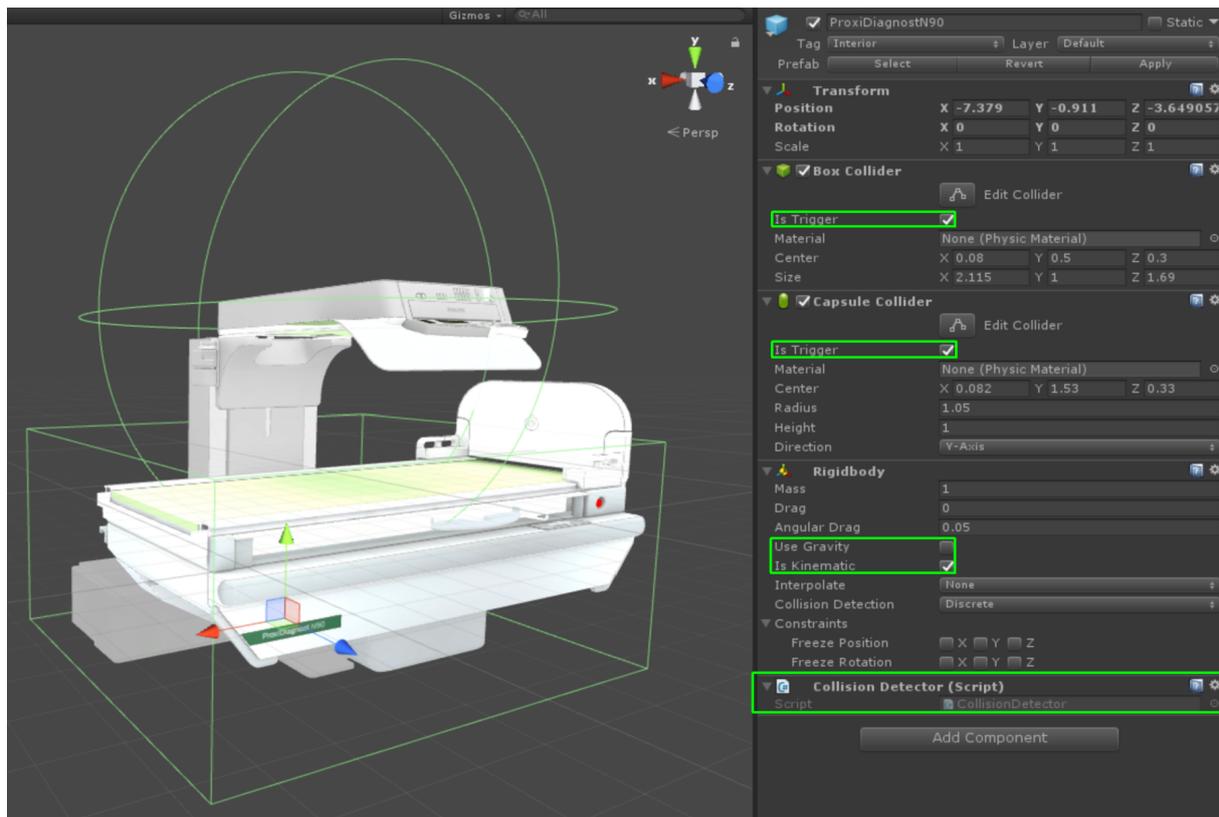


Abbildung 12: Konfiguration der Kollisionserkennung in Unity. *Links*: ProxiDiagnost N90 mit Kollisionskörpern. *Rechts*: Konfiguration der Unity-Komponenten als Kollisions-Trigger

Die Objekte werden mit kubischen und kugelförmigen *Collider*-Komponenten bestückt und die Mindestabstände der Gerätschaften entsprechend der Kollisionsdaten aus den Produktdatenblättern auf die *Collider*-Komponenten übertragen. Mit der aktivierten Trigger-Option in der *Collider*-Komponente kann die Physik-Engine als Kollisions-Trigger genutzt werden. Durch diese Konfiguration werden im Kollisionsfall Trigger-Events ausgelöst. Zusätzlich erhalten die Objekte das Skript *CollisionDetector*. Darin ist die Programmlogik für das Kollisionsverhalten definiert. Registriert die Physik-Engine eine Kollision zwischen den Kollisionskörpern, erfolgt eine Ausführung der hinterlegten Logik.

Es werden zwei Ereignisse zur Umsetzung der Kollisionserkennung herangezogen:

1. OnTriggerEnter färbt das kollidierende Objekt rot.
2. OnTriggerExit stellt die ursprüngliche Farbe des Objekts wieder her.

Im 2D-Modus wird der Raum über Raumeckpunkte repräsentiert. Bei der Raumkonfiguration ändern sich Raummaße oder die Anzahl von Raumecken. Dies erfordert eine dynamische Anpassung der Kollisionsgrenzen an Raumwänden. Dazu wird jede Raumecke mit einem *Collider* in die Kollisionserkennung eingebunden. Die Raumverwaltung organisiert die Kollisionskörper und bildet die Raumwände nach. So werden die Kollisionsgrenzen der Raumwände bei Änderungen der Raumdaten entsprechend angepasst.

3.5 Systemtest

Die Funktionalität der Anwendung wird einem explorativen Test unterzogen. Für die Befragung der Testpersonen wird ein Fragebogen in Schriftform erstellt. Dessen Auswertung soll eine objektive Aussage über die Zielerreichung ermöglichen. Die Versuchsreihe wurde sequentiell an 16 Probanden durchgeführt und unterliegt einem festgelegten Ablauf mit zwei Durchgängen an denen jeweils zwei Testpersonen beteiligt waren:

- 1) Einführung
Dem Probanden wird der Zweck der Anwendung erläutert und die Handhabung der Funktionen vorgeführt.
- 2) Einarbeitung
Der Proband führt eigenständig mit der Hilfestellung des Einweisers eine Raumplanung durch und betrachtet die Raumeinrichtung im VR-Modus.
- 3) Durchführung
In einem zweiten Durchlauf demonstriert der Proband der nachfolgenden Testperson die Handhabung der Anwendung und erläutert ihren Zweck.
- 4) Evaluierung
Bei der schriftlichen Bearbeitung des Fragebogens gibt der Proband gegebenenfalls der nachfolgenden Testperson eine Hilfestellung zur Einarbeitung.

Es wurden 22 Fragen bei der Versuchsreihe gestellt und in acht Kategorien gegliedert. Dabei wird die Lesbarkeit der verwendeten Zeichen und Symbole, die Informationsbereitstellung für den Applikationsablauf, die Selbstbeschreibungsfähigkeit der Software sowie die Konformität gegenüber der Teilnehmererwartungen untersucht. Die Kategorien Interaktion und Fehlertoleranz sollen Aufschluss über die Usability geben. Zudem werden die Teilnehmer zu ihren Erfahrungen und der Nutzung der VR in dieser Anwendung befragt. Abschließend soll die Kategorie „Anwendungsziel und VR“ Aufschluss über das Erreichen der gesetzten Ziele geben. Dabei wird die realitätsnahe Visualisierung der Raumplanung durch den Einsatz der VR bewertet und das Interesse über die implementierten Produkte festgestellt. Die Kategorien der Evaluation können Abbildung 13 entnommen werden. Das Bewertungssystem wird an die Likert-Skala[18] angelehnt, wobei die Probanden mit einer vollen Zustimmung (Wertigkeit 5) und einer Ablehnung (Wertigkeit 1) bewerten können.



Abbildung 13: Kategorien der Evaluation

Bei den teilnehmenden Probanden waren über 93% aus dem Studiengang Biomedizintechnik und befinden sich entweder im Bachelor oder Masterstudium. Ihr Alter liegt im Durchschnitt bei 22,0 Jahren, wobei die jüngste Person 18 Jahre und die älteste 41 Jahre alt war. Davon waren 43% der Probanden weiblich und im Schnitt 0,9 Jahre jünger als die männlichen Probanden. Tabelle 5 führt die Struktur der teilnehmenden Probanden auf.

Tabelle 5: Teilnehmende Probanden an der Evaluation

	Probanden (n=16)	durchschnittliches Alter	studierend
weiblich	43,75 %	22,0	100,00 %
männlich	56,25 %	22,9	88,89 %
Gesamt	100,00 %	22,0	93,75 %

Das Ergebnis der Auswertung dieser Versuchsreihe sowie die Fragen der Kategorien können in Kapitel 4.3 eingesehen werden.

4 Ergebnis

In diesem Abschnitt wird zuerst der Workflow der Anwendung erläutert. Darauf folgend werden die Ergebnisse aus der Polygonreduktion aufgezeigt und abschließend die Auswertung der Evaluation.

Mit der erstellten Applikation kann im 2D-Modus eine Raumplanung umgesetzt werden. Eine Darstellung erfolgt in einer 3D-Ansicht sowie in VR. Das Gesamtsystem besteht aus der 3D-Brille Daydream View[19] und dem Samsung Galaxy S8[20] und hat ein Gesamtgewicht von 456 Gramm. Wie in Abbildung 14 gezeigt, liegt eine mobile Lösung mit geringen Abmessungen vor.



Abbildung 14: VR-Headset bestehend aus Daydream View und Samsung Galaxy S8 im Verhältnis zu einem 30 cm langen Lineal

4.1 Workflow der Applikation

Die Applikation wird auf dem Smartphone als ausführbare App gelistet und kann durch das Berühren des Icons gestartet werden. Nach dem Start wird die Planungsphase (2D-Modus) mit einer rechteckigen Raumgeometrie gestartet. Im Weiteren wird der Workflow anhand einer Raumplanung erläutert.

Die Planungsphase

Mit der DoubleTap-Geste kann die Ansicht zu jeder Zeit auf den Raummittelpunkt zentriert werden. Durch das Berühren und Halten einer Raumecke wird die Raumgrößenänderung aktiviert. Eine unmittelbare Translationsbewegung ändert die Breite und Länge des Raumes (siehe Abbildung 15).

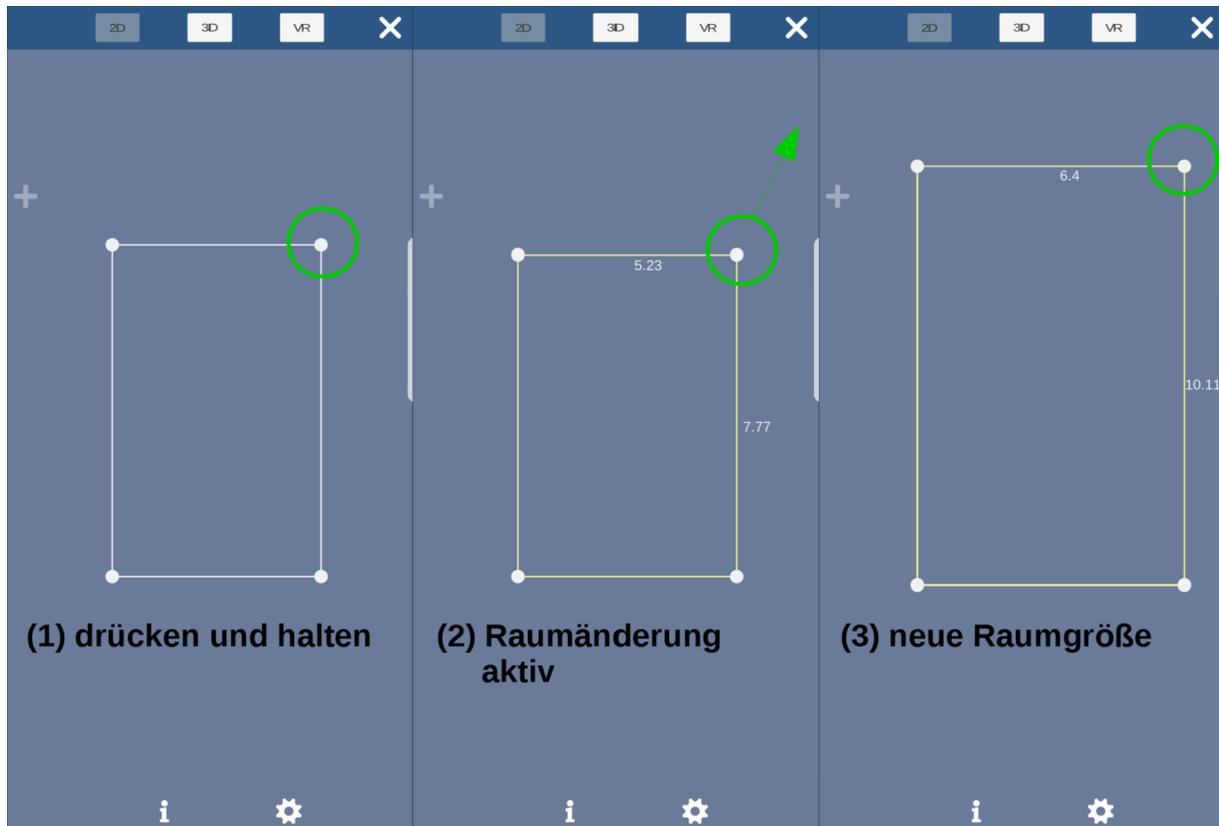


Abbildung 15: Workflow der Raumgrößenänderung in der Planungsphase

Durch die Berührung des Katalog-Icons öffnet der Katalog. So erhält der Nutzer die Auswahl der Produkte und die Möglichkeit *Viewpoints* zu setzen. Durch das Berühren einer Katalog-Sektion wird das Objekt im Raum platziert. Das Objekt ist nun im Bearbeitungsmodus und dadurch gelb gekennzeichnet. Nun kann es durch eine translatorische Berührung, beginnend am aktiven Objekt, im Raum positioniert werden. Eine Rotation erfolgt an jeder beliebigen Bildschirmposition durch eine Drehbewegung mit zwei Fingern. Der Bearbeitungsmodus kann durch eine Tap-Geste an einer freien Stelle am Bildschirm deaktiviert werden. Wird ein Objekt erneut berührt, so wird der Bearbeitungsmodus wieder aktiviert und ermöglicht eine weitere Positionierung. Alle weiteren Produkte können äquivalent im Raum positioniert werden.

Sollte ein Produkt nicht ausreichen Distanz zu einem anderen Produkt aufweisen, signalisiert die Kollisionserkennung dies durch Rotfärbung der betroffenen Gerätschaften (Vergleiche Abbildung 16 Punkt 5).

Ist der Bearbeitungsmodus bei einem Objekt aktiviert, kann dieses durch ein Berühren des Mülltonnen-Icons in der rechten Unterseite des Bildschirms wieder entfernt werden.

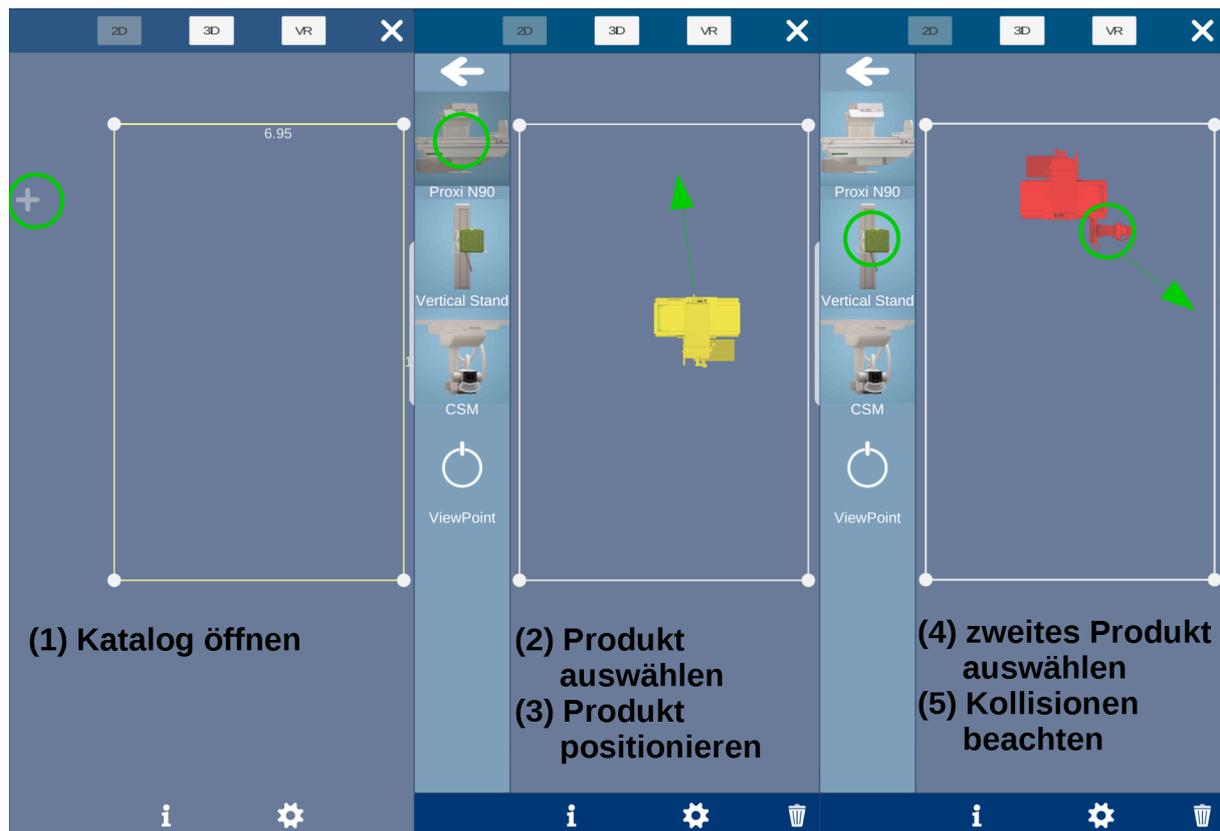


Abbildung 16: Workflow der Produktpositionierung unter Einhaltung der Mindestabstände in der Planungsphase

Ein *Viewpoint* wird immer am Bildschirmmittelpunkt mit der Orientierung zur Raummitte platziert. Das bedeutet, dass der Anwender den gesamten Raum verschieben kann, um anschließend einen *Viewpoint* am Bildschirmmittelpunkt in den Raum zu setzen. *Viewpoints* können des Weiteren wie Objekte aktiviert, positioniert und rotiert werden. Abbildung 17 zeigt das Setzen von *Viewpoints* und die anschließende Verifizierung im 3D-Modus.

Die Darstellungsphase

Im 3D-Modus kann die Raumkonfiguration in einer perspektivischen Darstellung aus frei bestimmbarer Richtung betrachtet werden. Es besteht eine Möglichkeit die gesetzten *Viewpoints* zu verifizieren. Dazu muss das *Viewpoint*-Symbol in der unteren Bildschirmmitte berührt werden. Nun kann der Anwender zwischen den *Viewpoints* wechseln und die Betrachtungspositionen für den VR-Modus prüfen bevor der VR-Modus aktiviert wird. Eine Änderung der Position oder der Orientierung kann im 2D-Modus vorgenommen werden. Für diese und weitere Änderungen an der Raumkonfiguration kann der Anwender zu jeder Zeit in den 2D-Modus wechseln.

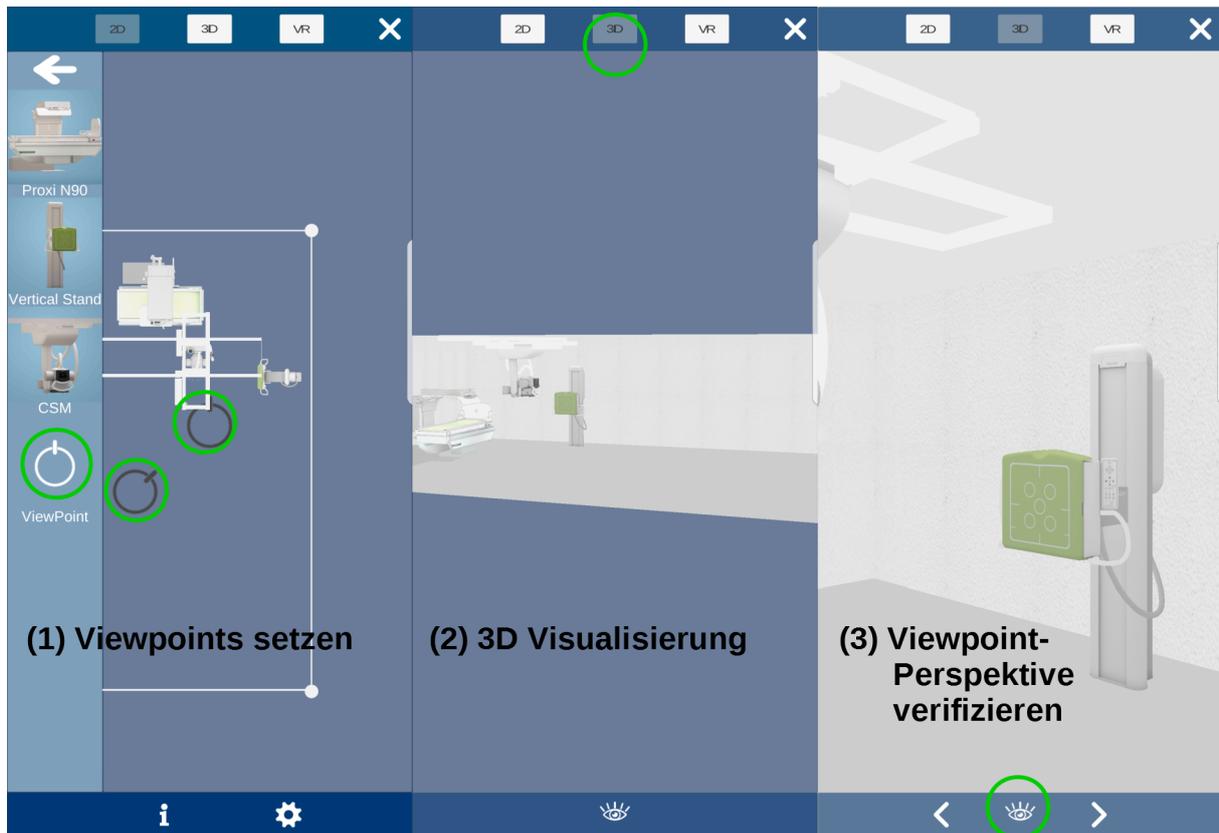


Abbildung 17: Workflow beim setzen und verifizieren der *Viewpoints*

Anschließend wird der VR-Modus über den VR-Button aktiviert. Hierzu wird das Smartphone in die 3D-Brille eingesetzt, um die Raumplanung in der VR betrachten zu können. Durch das Drücken der Touch-Taste am Funk-Controller können die Betrachtungspunkte der Reihe nach gewechselt werden. Abbildung 18 zeigt die stereoskopische Ansicht im VR-Modus.

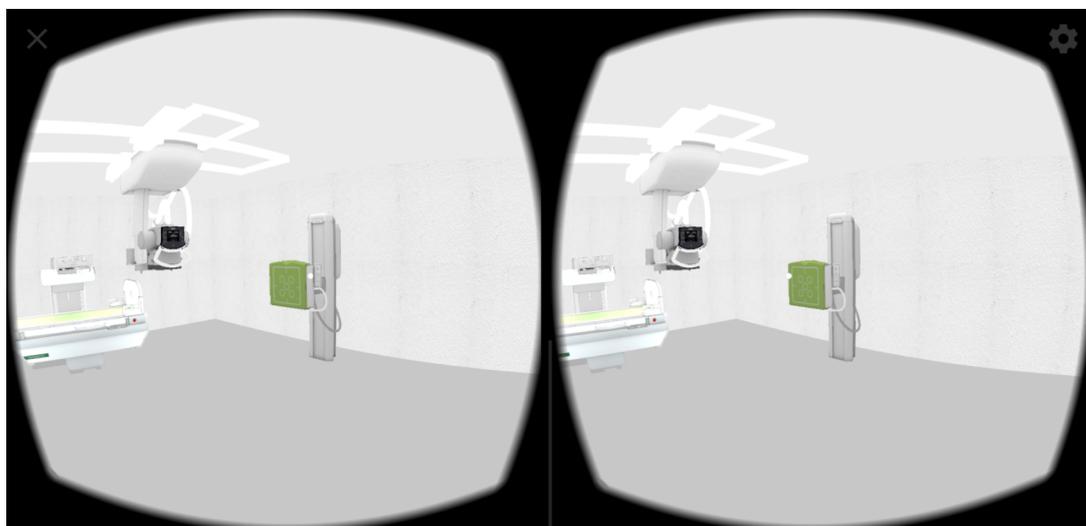


Abbildung 18: Stereoskopische Ansicht im VR-Modus

Der Workflow dieser Anwendung benötigt vier Berührungsgesten, um eine Raumkonfiguration in der Planungsphase umzusetzen. Durch die Einstellung „sechs Raumecken“ kann eine Vielfalt an Raumgeometrien (vergleiche Abbildung 19) mit nur zwei Berührungsgesten realisiert werden. Zudem besteht die Möglichkeit unmittelbar in den VR-Modus zu wechseln. So kann eine Raumplanung in wenigen Schritten erfolgen und ein Eindruck über die Produkte durch VR verschafft werden.

Zusätzliche Raumoptionen

Über das Einstellungspanel können die Raummaße durch Zifferneingabe mit einer Präzision von 10^{-2} Meter festgelegt werden. Hier findet auch die Bestimmung der Raumhöhe und auch der Wechsel der Anzahl von Raumecken statt.

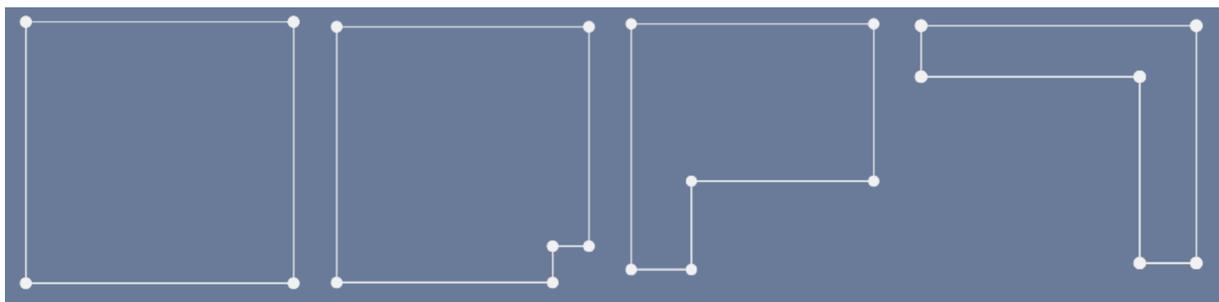


Abbildung 19: Beispiele zur Darstellung möglicher Raumgeometrien aus der Draufsicht

4.2 Polygonreduktion der 3D-Modelle

Durch die Reduzierung der Polygondichte um insgesamt 96 Prozent wurde eine Gesamtmenge der *Triangles* von 1.189.200 erreicht. Dabei fällt das NURBS-Modell des CombiDiagnost R90 mit einem Reduktionsfaktor von 0,11 am stärksten in die Gewichtung. Tabelle 6 stellt den Ausgangszustand sowie das Ergebnis der Optimierung dar.

Tabelle 6: Optimierung der 3D-Modelle durch Polygonreduktion

3D-Modell	Format	Ausgangszustand		Optimiert		Faktor der Polygonreduktion
		Dateigröße (in Kilobyte)	Triangles (10^3)	Dateigröße (in Kilobyte)	Triangles(10^3)	
ProxiDiagnost N90	FBX	124.639	176,30	52.656	98,80	0,56
CombiDiagnost R90	3DM	128.221	keine	–	–	–
	→ FBX	556.827	8.500,00	31.644	930,20	0,11
VS	FBX	11.661	36,70	7.994	9,70	0,26
CSM	FBX	50.394	82,50	39.193	73,00	0,88
Σ		743.521	8.619,20	131.487	1.189,20	0,14

Die optimierten 3D-Modelle sind in der Applikation implementiert und weisen auf den verwendeten Smartphone-Modellen keine Einschränkungen in der Lauffähigkeit auf. Bei der Vorbereitung der 3D-Modelle wurde die Oberfläche des CombiDiagnost R90 nicht texturiert und kann im vorliegenden Zustand nicht als visuelle Referenz zum Originalprodukt betrachtet werden. Weitere 3D-Modelle, welche in der Anwendung zum Tragen kommen, können bereits eine texturierte Oberflächenbeschaffenheit vorweisen. Ein Vergleich wird in Abbildung 20 dargestellt. Das ProxiDiagnost N90 (*links*) kann durch die Textur Farbübergänge an geraden Flächen abbilden. Den Oberflächen des CombiDiagnost R90 (*rechts*) sind lediglich Farbinformationen zugewiesen. Sie weisen einen geringeren Detailgrad auf.

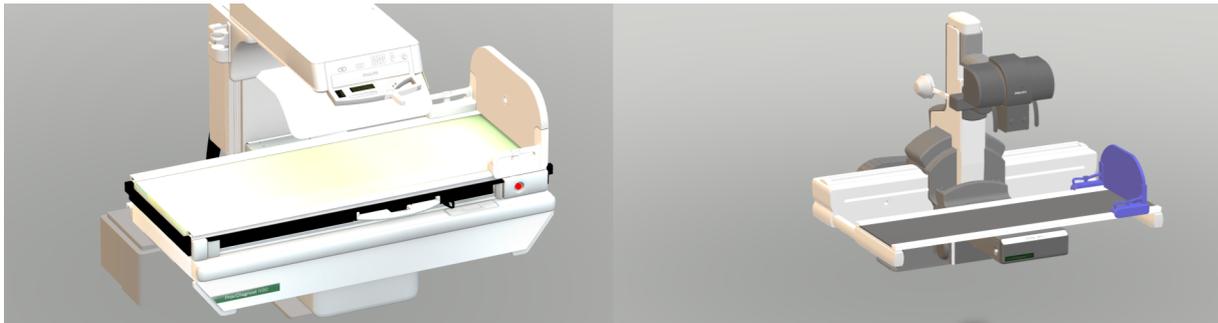


Abbildung 20: Vergleich zweier 3D-Modelle. *Links*: ProxiDiagnost N90 mit unterschiedlicher Farbintensität. *Rechts*: CombiDiagnost R90 mit homogener Farbintensität

In dieser Arbeit wurde eine Optimierung der Lichtverhältnisse für die Präsenz im VR-Modus nicht vorgenommen. Demzufolge werden keine Schatteneffekte realisiert. Bei abschließendem Systemtest konnte beim Betrieb des VR-Modus ein deutliches Flimmern an den Kanten der 3D-Modelle festgestellt werden. In dieser Ausarbeitung wurde der Schwerpunkt auf die Zielsetzung gelegt. Aus diesem Grund erfolgte keine weitere Untersuchung dieser Aspekte.

4.3 Auswertung der Evaluation

Bei der Versuchsdurchführung konnten sich die Probanden gemeinsam mit den Funktionen der Anwendung sowie mit der Durchführung einer Raumplanung auseinandersetzen.

Die Teilnehmer bewerten die Lesbarkeit der verwendeten Zeichen überwiegend positiv. Die Verständlichkeit der verwendeten Zeichen wurde mit einer Zustimmung von $4,25 \pm 0,93$ bewertet und die Darstellung der Symbol sowie der Farben mit $4,38 \pm 0,89$ (siehe Abbildung 21).

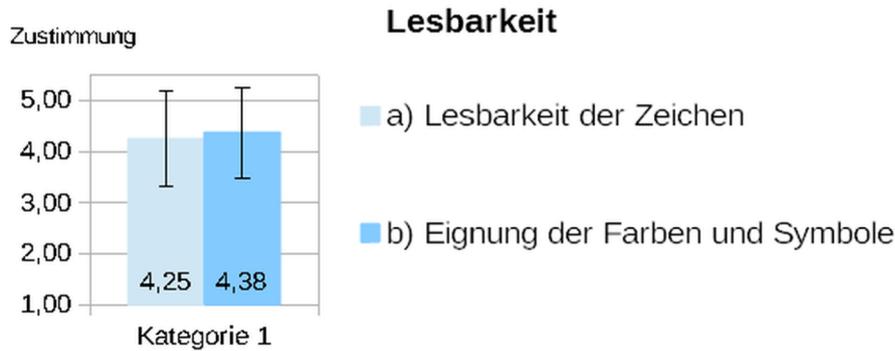


Abbildung 21: Ergebnis der Bewertung von Kategorie Lesbarkeit. Darstellung von Mittelwert und Standardabweichung

Bei der Raumkonfiguration ist die Handhabung der Anwendung mit $4,81 \pm 0,54$ zielführend. Die Informationsbereitstellung ist mit $4,56 \pm 0,63$ auf das wesentliche konzentriert, wobei mehr relevante Informationen gewünscht waren. So wurde die Vollständigkeit der Informationsbereitstellung zwischen 1,70 und 3,76 Punkten bewertet. Insgesamt ist der Selbstbeschreibungsfähigkeit der Anwendung mit 3,96 zugestimmt worden. Dabei wurden die Rückmeldungen der Software mit $3,63 \pm 0,89$ bewerten (vergleiche Abbildung 22).

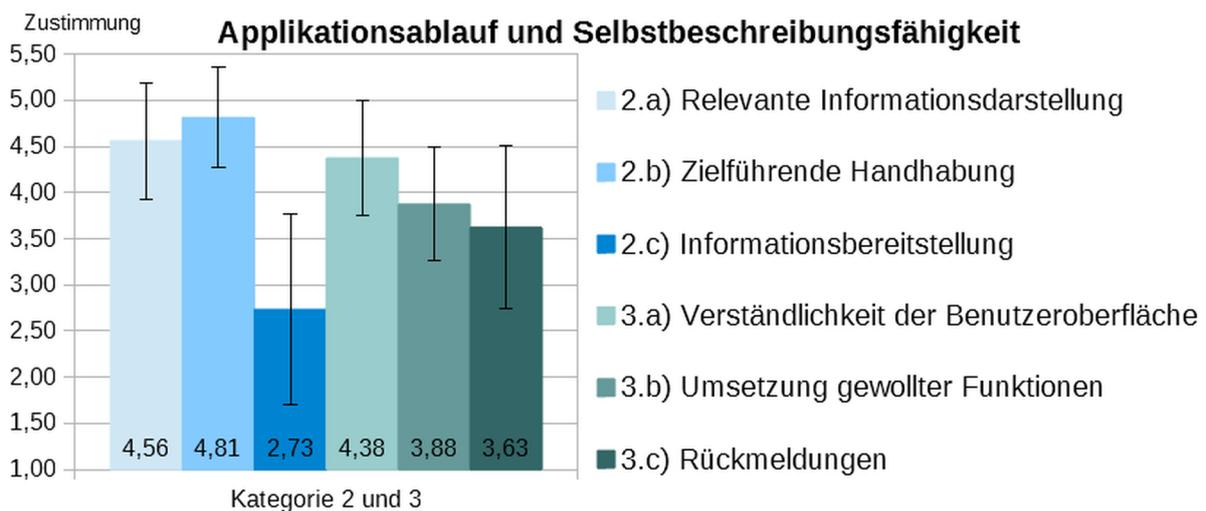


Abbildung 22: Ergebnis der Bewertung von Kategorie Applikationsablauf und Selbstbeschreibungsfähigkeit. Darstellung von Mittelwert und Standardabweichung

Die Interaktion mit der Anwendung wurde mit $3,75 \pm 0,93$ überwiegend als intuitiv gewertet. Eine Hilfestellungen durch die Software ist mit $3,75 \pm 1,0$ nur bedingt gegeben. Die Applikation kann jedoch nach Meinung der Probanden zu $3,44 \pm 0,89$ ohne Einarbeitung genutzt werden. Der Komfort der Steuerung ist dabei mäßig in einem Bereich zwischen 1,74 und 4,26 bewertet worden.

Der Schutz vor fehlerhafter Bedienung wurde mit $3,56 \pm 0,81$ bewertet. Die Nutzer konnten die Reaktionen der Anwendung unterschiedlich interpretieren und bewerteten das Auftreten erwarteter Reaktionen zwischen 2,69 und 4,81 (vergleiche Abbildung 23).

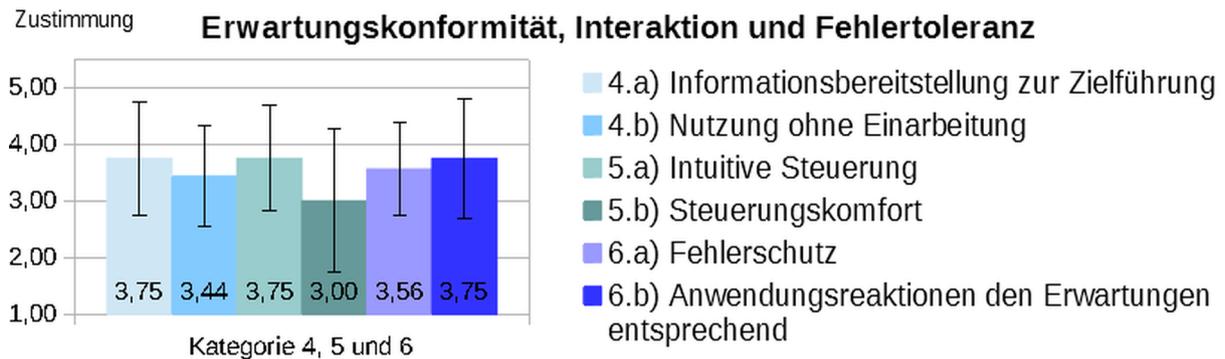


Abbildung 23: Bewertung der Kategorie Erwartungskonformität, Interaktion und Fehlertoleranz. Darstellung von Mittelwert und Standardabweichung

Die Teilnehmer können teilweise sehr wenig Erfahrung mit VR vorweisen. So befindet sich der Erfahrungswert in einem Bereich zwischen 1,44 und 4,42. Das VR-Erlebnis wurde mit $3,87 \pm 1,06$ als angenehm bewertet und entsprach mit $4,40 \pm 0,63$ den Erwartungen. Das VR-Headsets und der Funk-Controller konnten intuitiv genutzt werden und erreichten eine Wertung von $4,60 \pm 0,51$. Der Großteil der Teilnehmer empfand die Anwesenheit weiterer Personen beim Tragen des VR-Headsets mit $4,20 \pm 1,08$ als nicht störend (vergleiche Abbildung 24).

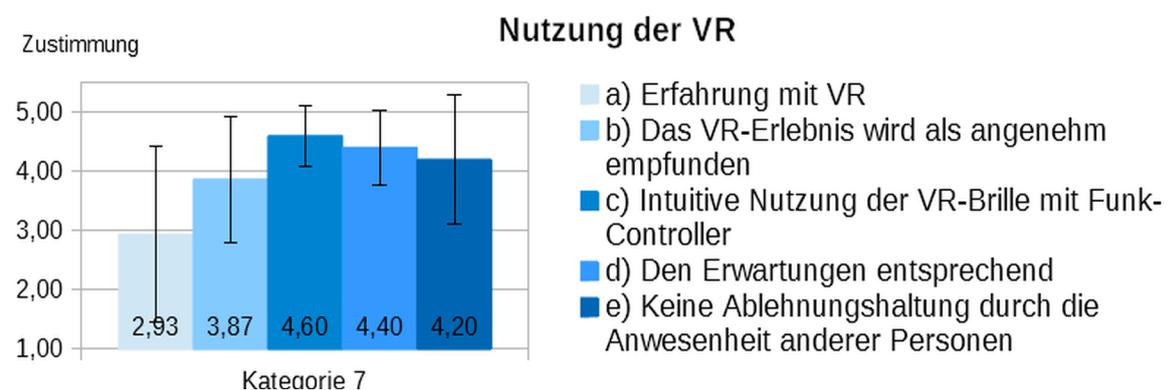


Abbildung 24: Bewertung der Kategorie "Nutzung der VR". Darstellung von Mittelwert und Standardabweichung

Bei der Befragung der Zielsetzungen erreichte die gesamte Kategorie „Anwendungsziel und VR“ 4,38. Der Anwendung wurde eine Unterstützung der Vorstellungskraft mit $4,60 \pm 0,63$ zugesprochen. Mit einer Wertung von $4,47 \pm 0,74$ konnten sich die Versuchspersonen einen

Eindruck über die implementierten Produkte verschaffen und werteten den Anreiz für ein Produktinteresse mit einer Zustimmung von $4,07 \pm 1,10$ (siehe Abbildung 25).

Damit konnte das System überwiegend die Vorstellungskraft der Probanden unterstützen und das entwickelte Raumkonzept verdeutlichen.

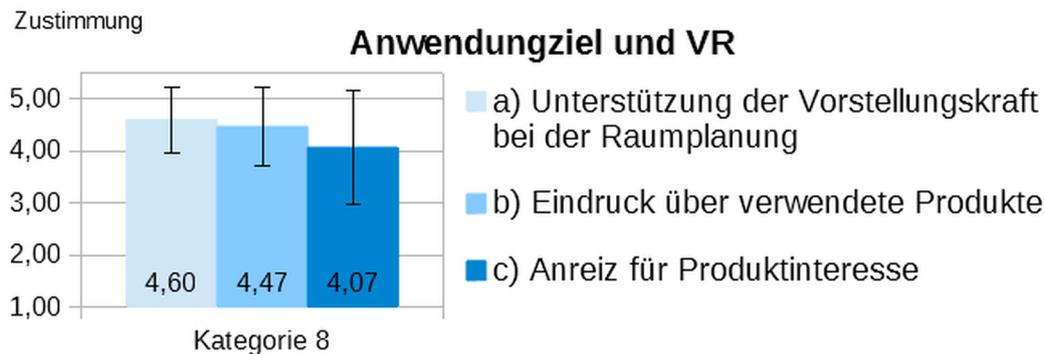


Abbildung 25: Bewertung der Kategorie "Anwendungsziel und VR". Darstellung von Mittelwert und Standardabweichung

In den Bemerkungen der Teilnehmer wurde häufig die Sensibilität der Steuerung bemängelt. Weitere Informationen, wie beispielsweise die Fläche des Raumes, sind erwünscht. Die Applikation wurde insgesamt positiv gewertet. Hinsichtlich der Zielerfüllung konnte eine positive Haltung der Teilnehmer festgestellt werden.

Bei weiteren Testdurchläufen konnte auf dem Samsung Galaxy S8 eine Verzögerung des Head-Trackings im VR-Modus sporadisch und bei schnellen Kopfbewegungen festgestellt werden. Bei Testdurchläufen auf dem Google Pixel XL konnte dieser Fehler jedoch nicht reproduziert werden.

5 Diskussion

Die Wahl der Hardware wurde von verschiedenen Aspekten beeinflusst. Das Mobilitätskriterium spielt dabei eine entscheidende Rolle. Durch die geringen Maße passt das System in jeden Aktenkoffer (vergleiche Abbildung 14). Zudem kann die Demonstration einer Raumplanung flexibel, kabellos und handlich ohne aufwendige Vorbereitungen und Installationen notwendiger Hardware durchgeführt werden. Ein weiterer Aspekt ist der vergleichsweise einfache und einer herkömmlichen Fernbedienung ähnelnde Daydream Funk-Controller. Dieser begünstigt eine intuitive Interaktion in der VR und eignet sich gegenüber anderen Eingabegeräten auch für Anwender mit keinerlei Gaming-Erfahrung. Die Interaktionskontrollen anderer VR-Systeme, wie das Oculus Rift[21] oder das HTC VIVE[22], weisen eine wesentlich höhere Anzahl an Aktionstasten auf.

Die Vorzüge hinsichtlich der Mobilität bringen Einschränkungen im Vergleich der Leistungskapazitäten mit sich. Das Smartphone Samsung Galaxy S8 kann eine Hauptprozessortaktung von 2,3 GHz und einen 4GB großen Arbeitsspeicher vorweisen[20]. Das Vergleichssystem HTC VIVE ist ein HMD und bildet mit einem PC-System die Gesamtheit eines VR-Systems. Im Zusammenschluss mit einem leistungsstarken Rechner werden hier berechnungsintensive Anwendungen und somit auch eine qualitativ hochwertigere Bilddarstellung möglich[23].

Die Applikation ermöglicht durch gerade mal zwei Berührungsgesten die Raumgeometrie in wenigen Schritten festzulegen. Die Positionierung der Produkte wird durch die Kollisionserkennung unterstützt. Sind die gewünschten Produkte platziert und die Raumkonfiguration abgeschlossen, kann mit einem Knopfdruck die Darstellungsphase in 3D oder VR gestartet werden. Die Visualisierung unterstützt die räumliche Vorstellungskraft und schafft Eindruck von den integrierten Produkten in der Raumeinrichtungsplanung.

Eine Alternative zu der Applikation Rapid Room Planing stellt die Lösung roomle[24] dar. Diese Plattform legt den Fokus auf die Einrichtungsplanung von Wohnräumen. Dazu werden Apps für verschiedene Hardwareformate wie das iPhone, iPad, Smartphones sowie Tablets mit Android Betriebssystemen zur Entwicklung und Einrichtung frei bestimmbarer Wohneinheiten angeboten. Dabei können in der 2D Draufsicht, vergleichbar mit dem 2D-Modus dieser Applikation, Wohnräume frei konzipiert werden. Die Planungsansicht bietet die Möglichkeit Raumecken frei zu positionieren und mit anderen Raumeckpunkten oder Wandsegmenten zu verbinden. Dazu stehen dem Anwender Berührungsgesten oder Eingabefelder zur Verfügung. Es bietet die Möglichkeit, Wände unter bestimmbar Winkeln zu zeichnen. Der Einrichtungskatalog ist an Möbelhäuser gekoppelt und bietet ein umfangreiches Einrichtungsangebot, welches in die Raumgestaltung integriert werden kann. Die Applikation besitzt einen Betrachtungsmodus, vergleichbar mit dem 3D-Modus von Rapid Room Planing, und einen walk-through-Modus. Dabei kann ein Rundgang in der Ich-Perspektive vorgenommen werden. Durch die AR-Funktion (Augmented Reality) können

einzelne Einrichtungsgegenstände an einem Fixpunkt der realen Umgebung platziert werden, um diese durch das Endgerät im gegenwärtig realen Raum zu betrachten. roomle erfordert eine Benutzerregistrierung und speichert die erstellten Raumkonfigurationen auf einem zentralen Server. Dieser bildet auch den Transferpunkt zur VR-Applikation von roomle. In dieser eigenständigen Anwendung erhält der Benutzer nach dem LogIn den Zugriff auf seine zentral gespeicherten Raumpläne und kann sich diese in Kombination mit einer 3D-Brille auf seinem Smartphone ansehen. Die Applikation roomle bietet eine ausgereifte Schnittstelle für Benutzerinteraktionen und ebenso qualitativ hochwertige Texturen der 3D-Modelle. Es ist jedoch festzuhalten, dass die Komponente VR nicht den Schwerpunkt dieser Plattform bildet und von der Applikation roomle getrennt angeboten wird.

Durch die Struktur der Software-Architektur ist eine funktionale Erweiterung einfach vorzunehmen. So kann die Komponente Raumverwaltung um weitere Funktionen zur Bestimmung der Raumgeometrie ausgebaut werden und dem Anwender beispielsweise das freie Setzen von Raumecken ermöglichen.

Die Kollisionserkennung gibt dem Anwender Hinweise, um Mindestabstände gegenüber anderen Objekten oder Raumwänden einzuhalten. Das System bietet jedoch keine Informationen der Abstandsmaße zu umliegenden Objekten oder Wänden und erschwert ein präzises Positionieren der Geräte. Eine präzise Positionierung der Produkte könnte über ein weiteres Einstellungspanel gelöst werden. Das Darstellen und das Bestimmen der Distanzmaße gegenüber den Raumwänden würden ermöglicht. Darüber hinaus könnten hier die vorgegebenen Mindestabstände der Gerätschaften dem Raumplaner eine weitere Hilfestellung anbieten.

Dem vorliegenden Prototyp fehlt noch die Möglichkeit eine Raumplanung vor dem Beenden der Applikation zu speichern. Durch die Smartphone Nutzung kann die Applikation allerdings durch den unmittelbaren Wechsel zum Home-Screen als Hintergrundprozess weiter geöffnet bleiben. Dies ist jedoch nur eine vorübergehende Abhilfe. Durch die Erstellung eines Speicherkonzepts könnten die Raumkonfigurationen dem Anwendender persistent zu Verfügung stehen. Dazu bieten sich zwei Szenarien an. Die Raumkonfiguration kann lokal auf dem verwendeten Smartphone gespeichert und aufgerufen werden. Eine andere Lösung stellt eine Dateiverwaltung auf einem zentralen Server dar. Durch die Einrichtung und Integration einer zentralen Datenbank könnten die konfigurierten Raumplanungen kumuliert zu Projekten oder Kunden gespeichert und verwaltet werden. Jeder Anwender der Applikation Rapid Room Planing hätte so die Möglichkeit auf diese Raumpläne zuzugreifen. Dadurch entsteht eine Variation von Präsentationsmöglichkeiten in der Kundenkommunikation.

Die Produktmodelle wurden einer Optimierung unterzogen und anforderungsgerecht in die Anwendung implementiert. 3D-Modellierung war kein Schwerpunkt dieser Arbeit. Demzufolge wurden bei der Polygonreduktion ausschließlich die redundanten Informationen aus planaren Polygonnetzen entfernt. Die Beschaffenheit der Modelloberflächen wurde aus

vorhandenen Textur-Dateien übernommen. Die visuelle Repräsentation der Produkte ist für weitere Tests hinreichend. Es ist jedoch erforderlich, weitere Optimierungen an den 3D-Modellen und deren Texturierung vorzunehmen. So kann die visuelle Repräsentation der Produkte verbessert werden und weiterhin Rechenleistung eingespart werden. Letzteres wird insbesondere bei der Erweiterung der Produktauswahl erforderlich. Diese Erweiterung wird durch das zentrale Datenmanagement in der Datenverwaltung und deren Integration im Katalog ermöglicht.

Das vorhandene Flimmern im VR-Modus kann durch weitere Untersuchungen und Optimierungen der Grafikeinstellungen unterbunden werden, um so ein visuell ansprechendes VR-Erlebnis zu gewährleisten.

Ob das Lösungssystem bei einer fachmännischen Raumeinrichtungsplanung unterstützend eingesetzt werden kann, konnte in der durchgeführten Versuchsreihe nicht gemessen werden. Die positiven Reaktionen der Teilnehmer zeigen das Potential dieses Systems auf. Der junge Altersdurchschnitt der Probanden lässt auf affinen Umgang mit Smartphones schließen. Es ist zudem festzustellen, dass die Teilnehmer teilweise noch keine Erfahrungen mit der VR-Technologie hatten. Demzufolge kann die durchgeführte Stichprobe für die Untersuchung der Usability als repräsentativ betrachtet werden. Die Steuerung könnte durch eine Änderung der Präzision von Berührungsgesten verbessert werden. Der Steuerungskomfort ließe sich durch die Implementierung einer Undo-Funktion erhöhen.

Der Prototyp Rapid Room Planning zeigt, dass sich immersive Anwendungen in nicht-immersive integrieren lassen und bedient mit der Kollisionserkennung ein differenzierteres Einsatzfeld. Eine Überarbeitung der Benutzerschnittstelle kann die Usability der Software weitergehend erhöhen. Ein weiterer Schritt wäre eine professionelle Evaluation des Systems in der Zielgruppe Sales. Dies würde Aufschluss über die Einsatzfähigkeit mobiler VR-Systeme im Vertriebssektor ermöglichen und spezialisierte Anforderungen an das System ermitteln.

Durch eine Erweiterung der 3D-Modelle kann das Produktangebot in der Applikation vielfältiger gestaltet werden. Zugleich würde die Installation von Türen, Fenstern sowie weiteren Einrichtungsgegenständen die Präsenz der Anwender im virtuellen Raum begünstigen. Darüber hinaus ist der Einsatz in anderen Zielgruppen denkbar und erfordert lediglich eine Anpassung der Produktmodelle.

Durch die genannten Maßnahmen kann das System verbessert und optimiert werden, so dass es unterstützend zur Raumplanung in der Medizintechnik und auch zur Kundenkommunikation eingesetzt werden kann.

Quellenverzeichnis

- [1] H. Hochheim und P. Neven, „AUMA | MesseTrend“, Berlin, 46, Feb. 2017.
- [2] R. Dörner, W. Broll, P. F. Grimm, und B. Jung, Hrsg., Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität. Berlin Heidelberg: Springer Vieweg, 2013.
- [3] S. Fehlberg, T. Lange, S. Eulenstein, P. M. Schlag, und P. U. Tunn, „Navigationsgestützte Resektion maligner Knochentumoren des Beckens“, *Der Onkologe*, Bd. 12, Nr. 2, S. 152–157, Feb. 2006.
- [4] U. Bracht, D. Geckler, und S. Wenzel, Digitale Fabrik: Methoden und Praxisbeispiele. Berlin: Springer, 2011.
- [5] Statista, „Statista Umfrage Augmented and Virtual Reality 2017“, Statista, Hamburg, Online Umfrage, Okt. 2017.
- [6] Google, „Daydream“. [Online]. Verfügbar unter: https://vr.google.com/intl/de_de/daydream/. [Zugegriffen: 31-Dez-2017].
- [7] Google, „Daydream – mit Daydream kompatible Smartphones“. [Online]. Verfügbar unter: https://vr.google.com/intl/de_de/daydream/smartphonevr/phones. [Zugegriffen: 22-Jan-2018].
- [8] Unity Technologies, „Unity“. [Online]. Verfügbar unter: <https://unity3d.com>. [Zugegriffen: 22-Jan-2018].
- [9] Unity Technologies, „Unity - Unity Patch Release 2017.2.0p4“. [Online]. Verfügbar unter: <https://unity3d.com/de/unity/qa/patch-releases/2017.2.0p4>. [Zugegriffen: 22-Jan-2018].
- [10] Oracle, „Java™ SE Development Kit 8, Update 151 Release Notes“. [Online]. Verfügbar unter: <http://www.oracle.com/technetwork/java/javase/8u151-relnotes-3850493.html>. [Zugegriffen: 22-Jan-2018].
- [11] Google, „Download Android Studio and SDK Tools | Android Studio“. [Online]. Verfügbar unter: <https://developer.android.com/studio/index.html>. [Zugegriffen: 22-Jan-2018].
- [12] Google, „Downloads and Samples | Google VR | Google Developers“. [Online]. Verfügbar unter: <https://developers.google.com/vr/develop/unity/download>. [Zugegriffen: 30-Jan-2018].
- [13] J. Johnson, „Fingers - Touch Gestures for Unity - Asset Store“. [Online]. Verfügbar unter: <https://www.assetstore.unity3d.com/en/#!/content/41076>. [Zugegriffen: 28-Dez-2017].
- [14] C. Seifert, Spiele entwickeln mit Unity 5: 2D- und 3D-Games mit Unity und C# für Desktop, Web & Mobile, 2., Aktualisierte und erweiterte Auflage. München: Hanser, 2015.
- [15] Mono Project, „Home | Mono“. [Online]. Verfügbar unter: <http://www.monoproject.com/>. [Zugegriffen: 09-Jan-2018].

- [16] runevision, „Triangulator - Unify Community Wiki“. [Online]. Verfügbar unter: <http://wiki.unity3d.com/index.php/Triangulator>. [Zugegriffen: 11-Jan-2018].
- [17] R. McNeel, „Rhinoceros - NURBS“. [Online]. Verfügbar unter: <https://www.rhino3d.com/de/nurbs>. [Zugegriffen: 13-Jan-2018].
- [18] H.-B. Brosius, A. Haas, und F. Koschel, Methoden der empirischen Kommunikationsforschung: eine Einführung, 7., Überarbeitete und aktualisierte Auflage. Wiesbaden: Springer VS, 2016.
- [19] Google, „Technische Daten zu Google Daydream View – Google Store“. [Online]. Verfügbar unter: https://store.google.com/product/google_daydream_view_specs. [Zugegriffen: 24-Jan-2018].
- [20] Samsung, „Technische Spezifikationen Samsung | Galaxy S8 und S8+ | Samsung DE“. [Online]. Verfügbar unter: <http://www.samsung.com/de/smartphones/galaxy-s8/spec-plus/>. [Zugegriffen: 17-Jan-2018].
- [21] Oculus VR, „Oculus Rift | Oculus“. [Online]. Verfügbar unter: <https://www.oculus.com/rift/#oui-csl-rift-games=mages-tale>. [Zugegriffen: 17-Jan-2018].
- [22] HTC, „VIVE | Discover Virtual Reality Beyond Imagination“. [Online]. Verfügbar unter: <https://www.vive.com/de/>. [Zugegriffen: 17-Jan-2018].
- [23] HTC, „VIVE™ Deutschland | VIVE Hardware kaufen“. [Online]. Verfügbar unter: <https://www.vive.com/de/product/>. [Zugegriffen: 17-Jan-2018].
- [24] Roomle, „Home“, *Roomle*. [Online]. Verfügbar unter: [//www.roomle.com/de/home](http://www.roomle.com/de/home). [Zugegriffen: 17-Jan-2018].
- [25] Philips, „View details of Philips CombiDiagnost R90“. [Online]. Verfügbar unter: <https://www.usa.philips.com/healthcare/product/HC709030/combidiagnost-r90-cross-functional-drf-system>. [Zugegriffen: 23-Jan-2018].

Abkürzungsverzeichnis

HMD	Head Mounted Display
CT	Computertomographie
App	Applikation
GUI	Graphical user interface
UI-System	User Interface-System
UX	User Experience
VS	Vertical Stand
CSM	Ceiling Suspension CSM
NURBS	Non-Uniform Rational B-Splines
FBX	Filmbox

Abbildungsverzeichnis

Abbildung 1: Konzept und Systematik der Methode.....	3
Abbildung 2: Aktivitätsfluss des Anwenders in den grundlegenden Funktionsmodi der Applikation Rapid Room Planing.....	8
Abbildung 3: Übersicht interner Softwarekomponenten und Schnittstellen.....	10
Abbildung 4: Spezifikation der Komponenten Anwendungsverwaltung und Datenverwaltung	11
Abbildung 5: Spezifikation der Komponente Anwendungssteuerung.....	13
Abbildung 6: Spezifikation der Komponente GUI mit Kopfzeile, Katalog, Fußzeile und Einstellungspanel.....	14
Abbildung 7: Spezifikation der Komponente Interaktionsverwaltung mit Subkomponenten <i>Fingers</i> und <i>GvrController</i>	17
Abbildung 8: Spezifikation der Komponente Raumverwaltung.....	18
Abbildung 9: Triangulation des 3D-Modells einer Raumwand.....	22
Abbildung 10: Algorithmus zur Erstellung von 3D-Raummodellen aus der Komponente Raumverwaltung.....	23
Abbildung 11: Fluoroskopiegeräte von Philips. <i>Links</i> : Vertical Stand. <i>Mitte</i> : Ceiling Suspension CSM. <i>Rechts</i> : CombiDiagnost R90 (Quelle: [25]).....	24
Abbildung 12: Konfiguration der Kollisionserkennung in Unity. <i>Links</i> : ProxiDiagnost N90 mit Kollisionskörpern. <i>Rechts</i> : Konfiguration der Unity-Komponenten als Kollisions-Trigger.....	26
Abbildung 13: Kategorien der Evaluation.....	28
Abbildung 14: VR-Headset bestehend aus Daydream View und Samsung Galaxy S8 im Verhältnis zu einem 30 cm langen Lineal.....	29
Abbildung 15: Workflow der Raumgrößenänderung in der Planungsphase.....	30
Abbildung 16: Workflow der Produktpositionierung unter Einhaltung der Mindestabstände in der Planungsphase.....	31
Abbildung 17: Workflow beim setzen und verifizieren der <i>Viewpoints</i>	32
Abbildung 18: Stereoskopische Ansicht im VR-Modus.....	32
Abbildung 19: Beispiele zur Darstellung möglicher Raumgeometrien aus der Draufsicht.....	33
Abbildung 20: Vergleich zweier 3D-Modelle. <i>Links</i> : ProxiDiagnost N90 mit unterschiedlicher Farbintensität. <i>Rechts</i> : CombiDiagnost R90 mit homogener Farbintensität....	34
Abbildung 21: Ergebnis der Bewertung von Kategorie Lesbarkeit. Darstellung von Mittelwert und Standardabweichung.....	35
Abbildung 22: Ergebnis der Bewertung von Kategorie Applikationsablauf und Selbstbeschreibungsfähigkeit. Darstellung von Mittelwert und Standardabweichung.....	35

Abbildung 23: Bewertung der Kategorie Erwartungskonformität, Interaktion und Fehlertoleranz. Darstellung von Mittelwert und Standardabweichung.....	36
Abbildung 24: Bewertung der Kategorie "Nutzung der VR". Darstellung von Mittelwert und Standardabweichung.....	36
Abbildung 25: Bewertung der Kategorie "Anwendungsziel und VR". Darstellung von Mittelwert und Standardabweichung.....	37

Tabellenverzeichnis

Tabelle 1: Gegenüberstellung der Projektanforderungen und der Lösungsansätze.....	4
Tabelle 2: Definition interner Software-Komponenten zur Realisierung der Applikation Rapid Room Planing.....	9
Tabelle 3: Definition der Berührungsgesten für die Applikationsmodi 2D und 3D.....	16
Tabelle 4: Auszug der Methode LoadView2D aus der Komponente Anwendungssteuerung. <i>Links</i> : Quellcode. <i>Rechts</i> : Ablaufdiagramm.....	21
Tabelle 5: Teilnehmende Probanden an der Evaluation.....	28
Tabelle 6: Optimierung der 3D-Modelle durch Polygonreduktion.....	33

Anhang A Quellcode

Methode LoadView3D der Anwendungssteuerung

```
public void LoadView3D()
{
    //configure mode-buttons
    _3D_Btn.interactable = false;

    //check if mode is not loaded twice
    if (GameController.gameMode == GameController.GameMode._3D)
    {
        return;
    }

    if (GameController.gameMode == GameController.GameMode._2D)
    {
        //configure mode-buttons
        _2D_Btn.interactable = true;

        //unload 2D Gestures
        InputControllerScript.Instance.RemoveActivGestures();

        //unload footer panel
        footerPanel2D.gameObject.SetActive(false);

        //close catalogue and hide catalogue button
        if (catalogueIsOpen)
        {
            CatalogueClose();
        }
        _Cat_Btn.gameObject.SetActive(false);
    }

    // configure 3D-setup
    InputControllerScript.Instance.Config3DInput();

    RoomController.Instance.ConfigRoom3D();

    if (GameController.database.isViewPoint)
    {
        uiAnimator.SetTrigger("footer3dOpen");
    }

    GameController.gameMode = GameController.GameMode._3D;
}
```

Methode LoadViewVR der Anwendungssteuerung

```
public void LoadViewVR()
{
    //configure mode-buttons
    _VR_Btn.interactable = false;
    _3D_Btn.interactable = false;
    _2D_Btn.interactable = true;

    //check if mode is not loaded twice
    if (GameController.gameMode == GameController.GameMode._VR)
    {
        return;
    }

    if (GameController.gameMode == GameController.GameMode._2D)
    {
        //unload 2D Gestures
        InputControllerScript.Instance.RemoveActivGestures();
        RoomController.Instance.ConfigRoom3D();
    }

    if (GameController.gameMode == GameController.GameMode._3D)
    {
        //unload 2D Gestures
        InputControllerScript.Instance.RemoveActivGestures();

        //unload 3D footer panel
        if (GameController.database.isViewPoint)
        {
            uiAnimator.SetTrigger("footer3dClose");
        }
    }

    //Load VR-setup by coroutine
    StartCoroutine(LoadDeviceVR("daydream"));

    GameController.gameMode = GameController.GameMode._VR;
}
```

Klasse Triangulator

```
using UnityEngine;
using System.Collections.Generic;

public class Triangulator
{
    private List<Vector2> m_points = new List<Vector2>();

    public Triangulator (Vector2[] points) {
        m_points = new List<Vector2>(points);
    }

    public int[] Triangulate() {
        List<int> indices = new List<int>();

        int n = m_points.Count;
        if (n < 3)
            return indices.ToArray();

        int[] V = new int[n];
        if (Area() > 0) {
            for (int v = 0; v < n; v++)
                V[v] = v;
        }
        else {
            for (int v = 0; v < n; v++)
                V[v] = (n - 1) - v;
        }

        int nv = n;
        int count = 2 * nv;
        for (int m = 0, v = nv - 1; nv > 2; ) {
            if ((count--) <= 0)
                return indices.ToArray();

            int u = v;
            if (nv <= u)
                u = 0;
            v = u + 1;
            if (nv <= v)
                v = 0;
            int w = v + 1;
            if (nv <= w)
                w = 0;

            if (Snip(u, v, w, nv, V)) {
                int a, b, c, s, t;
                a = V[u];
                b = V[v];
                c = V[w];
                indices.Add(a);
                indices.Add(b);
                indices.Add(c);
                m++;
                for (s = v, t = v + 1; t < nv; s++, t++)
                    V[s] = V[t];
                nv--;
                count = 2 * nv;
            }
        }

        indices.Reverse();
        return indices.ToArray();
    }
}
```

```

private float Area () {
    int n = m_points.Count;
    float A = 0.0f;
    for (int p = n - 1, q = 0; q < n; p = q++) {
        Vector2 pval = m_points[p];
        Vector2 qval = m_points[q];
        A += pval.x * qval.y - qval.x * pval.y;
    }
    return (A * 0.5f);
}

private bool Snip (int u, int v, int w, int n, int[] V) {
    int p;
    Vector2 A = m_points[V[u]];
    Vector2 B = m_points[V[v]];
    Vector2 C = m_points[V[w]];
    if (Mathf.Epsilon > (((B.x - A.x) * (C.y - A.y)) - ((B.y - A.y) * (C.x -
A.x))))
        return false;
    for (p = 0; p < n; p++) {
        if ((p == u) || (p == v) || (p == w))
            continue;
        Vector2 P = m_points[V[p]];
        if (InsideTriangle(A, B, C, P))
            return false;
    }
    return true;
}

private bool InsideTriangle (Vector2 A, Vector2 B, Vector2 C, Vector2 P) {
    float ax, ay, bx, by, cx, cy, apx, apy, bpx, bpy, cpx, cpy;
    float cCROSSap, bCROSScp, aCROSSbp;

    ax = C.x - B.x; ay = C.y - B.y;
    bx = A.x - C.x; by = A.y - C.y;
    cx = B.x - A.x; cy = B.y - A.y;
    apx = P.x - A.x; apy = P.y - A.y;
    bpx = P.x - B.x; bpy = P.y - B.y;
    cpx = P.x - C.x; cpy = P.y - C.y;

    aCROSSbp = ax * bpy - ay * bpx;
    cCROSSap = cx * apy - cy * apx;
    bCROSScp = bx * cpy - by * cpx;

    return ((aCROSSbp >= 0.0f) && (bCROSScp >= 0.0f) && (cCROSSap >= 0.0f));
}
}

```

Anhang B Fragebogen der Evaluation

Evaluation der Applikation Rapid Room Planing

(Explorativer Test)

Art der Methode: Subjektiv

Gegenstand der Evaluation:

- Benutzerschnittstelle der Software
- Arbeits- bzw. Anwendungsfluss
- Nutzung der VR

Ziel: Erfassen der Benutzerakzeptanz

	Stimme stark zu 1	Stimme zu 2	teils, teils 3	Lehne ab 4	Lehne stark ab 5
Lesbarkeit					
1. Die Zeichen auf dem Bildschirm sind gut lesbar und unterscheidbar.					
2. Die Farben und Symbole sind immer verständlich.					
Bemerkungen:					
Applikationsablauf					
3. Es werden nur Informationen dargestellt, die wirklich bei der Aufgabe benötigt werden.					
4. Die Software erfordert keine überflüssigen Schritte und Eingaben, um zum Ziel zu kommen.					
5. Es fehlen relevante Informationen.					
Bemerkungen:					

	Stimme stark zu 1	Stimme zu 2	teils, teils 3	Lehne ab 4	Lehne stark ab 5
Selbstbeschreibungsfähigkeit					
6. Bei der Software ist die Bedeutung der angezeigten Texte, Feldbezeichner, Auswahlmöglichkeiten, etc. verständlich und führt nicht zu Fehlinterpretationen.					
7. Es ist immer erkennbar, wie man das, was man tun will, auch tun kann.					
8. Die Software liefert immer Rückmeldungen, die verständlich und hilfreich sind.					
Bemerkungen:					
Erwartungskonformität					
9. Die Software liefert aufgrund von Eingaben immer die benötigte Information, um einen Schritt weiter zu kommen.					
10. Die App kann ohne Einarbeitung genutzt werden. .					
Bemerkungen:					
Interaktion					
11. Die Steuerung der App ist intuitiv und entspricht den Erwartungen.					
12. Die Interaktion mit der Anwendung wird als angenehm empfunden.					
Bemerkungen:					
Fehler- und Fehlertoleranz					
13. Die App schützt vor fehlerhafter Eingabe.					
14. Es treten unerwartete Reaktionen der Anwendung auf.					

	Stimme stark zu 1	Stimme zu 2	teils, teils 3	Lehne ab 4	Lehne stark ab 5
Nutzung der Virtuellen Realität					
15. Es liegen bereits Erfahrungen mit VR-Anwendungen vor.					
16. Das VR-Erlebnis wird als angenehm empfunden.					
17. Die Nutzung der VR-Brille und des Controllers ist intuitiv möglich.					
18. Das VR-Erlebnis entspricht den Erwartungen.					
19. Das Aufsetzen der Brille wurde in Gegenwart anderer Personen als unangenehm empfunden.					
Bemerkungen:					
Anwendungsziel und Virtuelle Realität					
20. Durch den VR-Modus der Applikation wurde das entwickelte Raumkonzept verdeutlicht und die Vorstellungskraft unterstützt.					
21. Es konnte sich von den implementierten Produkten ein räumlicher Eindruck verschafft werden.					
22. Durch die Virtualisierung der Produkte ist ein Interesse an der Produktfunktionalität geweckt worden.					
Bemerkungen:					
Vorschläge und Anregungen:					
Datum	Geschlecht	Alter	Beruf	Ausbildung	

Danke für Ihre Teilnahme

Anhang C Ergebnisse der Evaluation

Merkmal	m_{arithm}	Standard- abweichung
1. Lesbarkeit	4,32	
a) Lesbarkeit der Zeichen	4,25	0,93
b) Eignung der Farben und Symbole	4,38	0,89
2. Applikationsablauf	4,03	
a) Relevante Informationsdarstellung	4,56	0,63
b) Zielführende Handhabung	4,81	0,54
c) Informationsbereitstellung	2,73	1,03
3. Selbstbeschreibungsfähigkeit	3,96	
a) Verständlichkeit der Benutzeroberfläche	4,38	0,62
b) Umsetzung gewollter Funktionen	3,88	0,62
c) Rückmeldungen	3,63	0,89
4. Erwartungskonformität	3,60	
a) Informationsbereitstellung zur Zielführung	3,75	1,00
b) Nutzung ohne Einarbeitung	3,44	0,89
5. Interaktion	3,38	
a) Intuitive Steuerung	3,75	0,93
b) Steuerungskomfort	3,00	1,26
6. Fehler und Fehlertoleranz	3,66	
a) Fehlerschutz	3,56	0,81
b) Anwendungsreaktionen den Erwartungen entsprechend	3,75	1,06
7. Nutzung der VR	4,00	
a) Erfahrung mit VR	2,93	1,49
b) Das VR-Erlebnis wird als angenehm empfunden	3,87	1,06
c) Intuitive Nutzung der VR-Brille mit Funk-Controller	4,60	0,51
d) Den Erwartungen entsprechend	4,40	0,63

Merkmal	m_{arithm}	Standard- abweichung
e) Keine Ablehnungshaltung durch die Anwesenheit anderer Personen	4,20	1,08
8. Anwendungsziel und VR	4,38	
a) Unterstützung der Vorstellungskraft bei der Raumplanung	4,60	0,63
b) Eindruck über verwendete Produkte	4,47	0,74
c) Anreiz für Produktinteresse	4,07	1,10