IT DESIGNERS GRUPPE

HOCHSCHULE ESSLINGEN

Bachelorarbeit

Interaktive Visualisierung von Software-Requirements mit Augmented Reality: Eine Analyse der Usability und Effektivität

im Studiengang Softwaretechnik und Medieninformatik (SWB)
der Fakultät Informationstechnik
Sommersemester 2024

Kyle Mezger
Matrikelnummer: 765838

Zeitraum: 01.03.2024 bis 31.08.2024
Erstprüfer: Prof. Dr. -Ing. Andreas Rößler
Zweitprüfer: Prof. Dr. rer. nat. Dieter Morgenroth

Firma: IT Designers Gruppe

Betreuer: Stefan Kaufmann

Eidesstattliche Erklärung

Hiermit versichere ich, Kyle Mezger, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.
Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.
Weiterhin erkläre ich, dass ich für die Umformulierung einzelner Textpassagen und die Korrektur von Rechtschreib- und Grammatikfehlern die Hilfe von digitalen Tools, speziell ChatGPT und DeepL Write, in Anspruch genommen habe. Diese Unterstützung betraf ausschließlich die Formulierungshilfe und die Korrektur von Grammatik und Rechtschreibung, ohne dass die inhaltliche Eigenständigkeit und Urheberschaft meiner Arbeit dadurch berührt wurden.

Esslingen, den 8. April 2024 _____

Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Kurzfassung

2 Einleitung

2.1 Motivation

2.2 Zielsetzung

3 Grundlagen

Im folgenden Kapitel sollen konzeptionelle Grundlagen erläutert werden, welche für das Verständnis der Bachelorarbeit notwendig sind. Dabei wird auf die Themen Requirements Engineering, Augmented Reality und die Software reQlab eingegangen.

3.1 Requirements Engineering

Die Bachelorarbeit soll sogenannte Requirements, also Anforderungen, visualisieren. Daher ist es für das Verständnis der Arbeit wichtig, die Grundlagen des Requirements Engineering zu kennen.

Requirements

Grundlegend sind Requirements Anforderungen, die an ein System gestellt werden. Das International Requirements Engineering Board (IREB) definiert sie in ihrem Glossar mit drei Eigenschaften:

- Ein Bedürfnis eines Interesseneigners (Stakeholder).
- Eine Eigenschaft oder Fähigkeit, die ein System haben soll.
- Eine dokumentierte Repräsentation eines Bedürfnisses, einer Fähigkeit oder einer Eigenschaft.

[1, Def. Anforderung]

Sie sollen also die Bedürfnisse der Stakeholder an das System repräsentieren und dokumentieren.

Die Gestaltung von Requirements kann dabei je nach System und Anforderungen unterschiedlich sein. Chris Rupp nennt in ihrem Buch „Requirements-Engineering und -Management“ einige Beispiele für verschiedene Formen für Requirements:

- User-Stories
- Use-Cases
- Stories
- formalisierte natürlichsprachliche Anforderungen

- Anforderungen in Form von Diagrammen (semiformales Modell)

[2, S. 19]

Natürlichsprachliche Anforderungen können sehr einfach selbst formuliert werden, sind dadurch jedoch auch anfällig für Missverständnisse und Unklarheiten. Das Ziel von reQlab ist es, diese Missverständnisse und Unklarheiten zu erkennen und so die Qualität der Anforderungen zu verbessern. Daher werden im Unfamg dieser Bachelorarbeit nur natürlichsprachliche Anforderungen genutzt.

Zudem werden Requirements in funktionale und nicht-funktionale Requirements unterteilt. Funktionale Requirements beschreiben „die Funktionen, die das System leisten soll, die Informationen die es verarbeiten soll; das gewünschte Verhalten, welches das System an den Tag legen soll.“ [3, S. 12] Nicht-funktionale Requirements hingegen beschreiben alle Requirements, die nicht funktionaler Natur sind, also beispielsweise Performance, Sicherheit oder Zuverlässigkeit. Peter Hruschka beschreibt in seinem Buch Funktionale Anforderungen mit der Frage: „Was soll das System/Produkt tun?“. Auch unterteilt er nicht-funktionale Anforderungen in die zwei Kategorien Qualitätsanforderungen („Wie gut? Wie schnell? Wie zuverlässig? ...“) und Randbedingungen („Ressourcen, Wiederverwendung, Zukauf, geforderte Technologie ...“) “ [3, S. 13]. Diese Unterteilung ist hilfreich zur Strukturierung der Anforderungen und könnte im User-Interface der Visualisierung genutzt werden, um die Anforderungen zu kategorisieren.

Stakeholder

Stakeholder können „Personen oder Organisationen sein, die die Anforderungen eines Systems beeinflussen oder die von dem System beeinflusst werden.“ [1]. Beispielsweise wären die Endnutzer eines Systems Stakeholder, welche durch das System beeinflusst werden. Sie haben also ein Bedürfnis an das System, können dieses jedoch nicht selbst umsetzen. Im Gegensatz dazu stehen die Auftraggeber beziehungsweise der Produkteigner (Product Owner), welche das System entwickeln und die Anforderungen festlegen.

Je nach der Größe und Komplexität des Systems kann es sehr viele Anforderungen geben. Durch die Menge an Anforderungen kann so schnell die Übersicht über das System und dessen Requirements verloren gehen, vor allem für Stakeholder, die nicht tagtäglich mit dem System arbeiten.

System

Die IREB definiert ein System als „Eine kohärente, abgrenzbare Menge von Elementen, die durch koordiniertes Handeln einen bestimmten Zweck erfüllen.“ [1] Das Wort System ist dabei ein Überbegriff für Produkte, Services, Geräte, Prozeduren und Werkzeuge und kann sowohl physisch als auch virtuell sein. Daher wird auch in dieser Bachelorarbeit das Wort System als Überbegriff für alle Arten von Systemen genutzt.

Requirements Engineering

Requirements-Engineering ist der Prozess, in dem Anforderungen an ein System erhoben, dokumentiert, analysiert, spezifiziert und validiert werden. Laut Chris Rupp besteht Requirements-Engineering dabei aus vier Haupttätigkeiten:

- Wissen vermitteln
- Gute Anforderungen herleiten
- Anforderungen vermitteln
- Anforderungen verwalten

[2, S.20]

Diese Bachelorarbeit soll versuchen einen neuen Ansatz in der Vermittlung von Anforderungen zu finden, um so langfristig die Qualität und Nützlichkeit der Anforderungen zu verbessern. Zudem soll die Visualisierung der Anforderungen helfen, bei großen Projekten die eine sehr große Zahl an Anforderungen besitzen, eine Übersicht über die Anforderungen zu erhalten und so eventuell eine bessere Kommunikation zwischen Stakeholdern und Entwicklern zu ermöglichen.

3.2 reQlab

Die Software reQlab ist ein Requirements-Engineering-Tool, welches von der IT-Designers GmbH entwickelt wird. Es dient dazu, Requirements automatisiert zu analysieren und zu bewerten. Dafür nutzt die Software ein Large-Language-Model (LLM), welches natürlichsprachliche Anforderungen analysiert und bewertet. Daher werden in reQlab alle Anforderungen als natürlichsprachliche Anforderungen verfasst um dann vom LLM verarbeitet werden zu können. Die Software analysiert diese Anforderungen und gibt eine begründete Bewertung aus, ob die Anforderung gut oder schlecht ist und gibt Verbesserungsvorschläge. Das Ziel von reQlab ist es, die Qualität der Anforderungen zu verbessern und so die Qualität des gesamten Systems zu steigern.

3.3 Virtuelle Realität

Für das Verständnis von Augmented Reality ist es wichtig, die Begriffe der virtuellen Realität zu kennen und zu verstehen. Dabei wird der Nutzer in eine virtuelle Welt versetzt, die durch Computer generiert wird. In virtueller Realität ist, im Gegensatz zu Augmented Reality die gesamte Umgebung digital. [vgl. 4, S.15] Zusätzlich wird unterschieden zwischen immersiver und nicht-immersiver virtueller Realität. Folgende auf den Nutzer bezogene Eigenschaften sind dabei Indikatoren für nicht-immersive virtuelle Realität:

- Der Nutzer steht nicht im Mittelpunkt.

- Der Nutzer ist nicht vollständig von digitalen Inhalten umgeben.
- Der Nutzer erfährt die virtuelle Realität als Beobachter anstatt als Teilnehmer.

Bei immersiver virtueller Realität sind alle äußeren Einflüsse so weit reduziert wie möglich und alle Indikatoren sollten auf immersive virtuelle Realität hinweisen. Der Nutzer sollte bei immersiver VR das Gefühl haben, sich selbst in der virtuellen Umgebung zu befinden. [vgl. 5, S.23-24]

3.4 Augmented Reality

Augmented Reality (AR) ist eine Technologie, die die reale Welt mit digitalen Informationen erweitert. Dabei wird ein ähnlicher Ansatz wie bei Virtueller Realität verfolgt, jedoch wird die reale Welt nicht komplett ersetzt, sondern nur erweitert. Der Nutzer sieht also weiterhin seine reale Umgebung, diese wird aber durch digitale Informationen ergänzt.

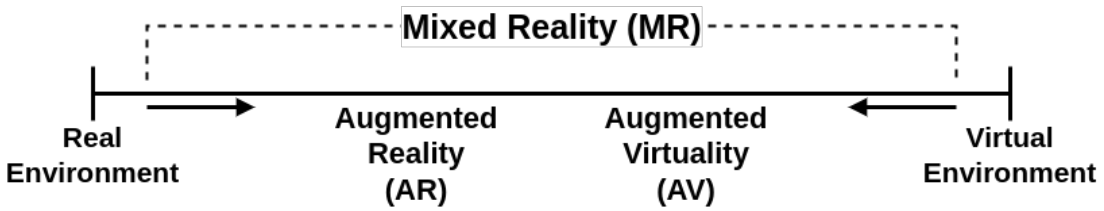

Abbildung 1: Realitäts-Virtualitäts-Kontinuum nach Milgram

Auf dem Realitäts-Virtualitäts-Kontinuum von Milgram, welches, wie in Abbildung 1 dargestellt, einen fließenden Übergang zwischen Realität und Virtualität beschreibt, liegt Augmented Reality zwischen der realen Welt und der virtuellen Welt [vgl. 6, S.9]. Daher fällt Augmented Reality unter den Überbegriff der Mixed Reality, da die reale Welt mit der digitalen Welt gemischt wird. Für diese Erweiterung der Realität müssen die Anzeigegeräte auch Informationen über die echte Umgebung sammeln können. Will man beispielsweise virtuelle Objekte in die reale Welt einfügen, so muss das Anzeigegerät die eigene Position kontinuierlich bestimmen können und die Position und Rotation des virtuellen Objekts anhand der Bewegungen des Nutzers anpassen.

Anzeigegeräte für Augmented Reality haben viele Gemeinsamkeiten mit Anzeigegeräten für Virtuelle Realität. Die Besonderheit von AR-Anzeigegeräten ist jedoch, dass sie die reale Welt mit digitalen Informationen erweitern. Das heißt sie müssen dem Nutzer auch eine Sicht auf die reale Welt ermöglichen und in diese Informationen einblenden. Beispielsweise kann das Display eines Anzeigegeräts transparent sein, sodass der Nutzer durch das Display hindurch sehen kann. Alternativ kann das Gerät eine oder mehrere Kameras besitzen, dessen Aufnahme auf undurchsichtige Bildschirme projiziert wird. Diese Methode nennt sich auch Image Passthrough. Es gibt verschiedene Arten von Anzeigegeräten, die

für Augmented Reality genutzt werden können, wobei man allgemein zwischen 3 Haputkategorien unterscheiden kann: Head-Mounted Displays, Hand-Held-Devices und Spatial Displays [7, S. 346]. Im Folgenden werden diese drei Kategorien genauer erläutert und einige Beispiele genannt.

3.4.1 Head-Mounted Displays

Head-Mounted Displays (HMDs) sind grundsätzlich Bildschirme, die direkt auf dem Kopf des Nutzers getragen werden. Durch ihre Nähe zu den Augen des Nutzers können sie ein großes Sichtfeld abdecken, ohne dabei selbst große Displays zu nutzen. Dadurch sind sie für volle Immersion im Normalfall kosteneffektiver als große Displays, wie bspw. eine Leinwand.

VR Headsets

Klassische VR-Headsets kann man sich vorstellen wie eine Skibrille mit 2 Bildschirmen, die auf dem Kopf getragen wird und üblicherweise 2 Displays hinter 2 Linsen hat, bzw. ein Display virtuell in 2 Displays aufteilt. Durch die 2 Bildschirme wird für jedes Auge ein eigenes Bild erzeugt, wodurch Inhalte in 3D dargestellt werden können. Außerdem können sie das volle Sichtfeld des Nutzers abdecken und so eine immersive Erfahrung schaffen.

Die Position des Headsets muss dabei kontinuierlich bestimmt werden, um die Bewegungen der Nutzer zu verfolgen und so die virtuelle Welt anpassen zu können. Dies geschieht meist durch Basisstationen (bspw. bei der Valve Index) oder durch Kameras im Headset (bspw. bei der Oculus Quest 3). Die Interaktion mit der Umgebung wird dann meist mit Controllern realisiert, deren Position und Rotation ebenfalls kontinuierlich bestimmt werden muss. In VR-Headsets mit eigenen Kameras ist es auch möglich, die Hände des Nutzers zu tracken und als Controller zu nutzen. Dabei können dann bestimmte Gesten oder Handbewegungen als Eingabe interpretiert werden.

Ursprünglich mussten VR-Headsets mit einem Computer verbunden werden, um die Rechenleistung für die Darstellung der Inhalte zu haben. Mit der Zeit wurden jedoch auch standalone VR-Headsets, wie bspw. die 2024 auf den Markt gebrachte Apple Vision Pro, die ihre eigene Rechenleistung haben und so unabhängig von einem Computer genutzt werden können.

Abbildung 2: Oculus Quest 3 VR-Headset

Quelle: https://unsplash.com/de/fotos/ein-mann-der-auf-einem-stuhl-sitzt-und-eine-virtuelle-brille-tragt-AZ-ND5uJ4S4

Jedoch sind nicht alle VR-Headsets für AR geeignet. Die meisten VR-Headsets besitzen keine Kamera, um die reale Welt aufzunehmen und dem Nutzer wiederzugeben. Nur Headsets wie bspw. die Oculus Quest 3, welche in Abbildung 2 dargestellt ist, oder die Apple Vision Pro besitzen eine Kamera nach außen, um die reale Welt aufzunehmen und so AR zu ermöglichen.

Durch all diese Technik, können die neusten VR-Headsets eine sehr hohe Immersion und Qualität für Anwendungen in AR bieten. Doch für lange Anwendungszeiten und die Verwendung in der Öffentlichkeit sind sie meist zu groß und schwer.

Smart-Glasses

Smart-Glasses sind Brillen, die digitale Informationen in das Sichtfeld des Nutzers einblenden. Grundsätzlich sind auch sie HMDs, jedoch werden sie aufgrund ihrer Sie stellen Informationen auf einer semitransparenten Fläche dar, sodass der Nutzer weiterhin die reale Welt sehen kann.

Da sie fast aussehen wie normale Brillen die unauffälligste Art von AR-Anzeigegeräten. Dadurch sind sie auch in der Öffentlichkeit unauffällig zu tragen und können so potentiell

auch im Alltag genutzt werden. Zudem sind sie, vor allem über einen längeren Zeitraum, meist deutlich leichter und angenehmer zu tragen als HMDs.

3.4.2 Hand-Held-Devices

Hand-Held-Devices sind Geräte, die ein Nutzer in der Hand hält. Diese Kategorie besteht heutzutage hauptsächlich aus Smartphones und Tablets, da die meisten dieser Geräte über eine Kamera und einen Bildschirm verfügen, können sie fast alle für die Darstellung von AR-Inhalten genutzt werden. Das Smartphone oder Tablet ist dabei wie ein Fenster in die digitale Welt, durch das der Nutzer die erweiterte Realität sehen kann.

Die Immersion ist bei diesem Anzeigegerät jedoch relativ gering, da der Nutzer immer die reale Welt sieht und das Smartphone oder Tablet nur ein kleines Fenster in die digitale Welt ist. Allein durch die Entfernung der Geräte vom den Augen der Nutzer können sie nur ein vergleichsweise kleines Sichtfeld abdecken.

Jedoch ist die Nutzung von Smartphones für AR sehr weit verbreitet, da fast jeder ein Smartphone besitzt und so keine zusätzliche Hardware benötigt wird. Beispielsweise hatte das AR-Spiel Pokémon Go, welches 2016 veröffentlicht wurde, bereits Anfang 2019 über 1 Milliarde Downloads [8]. Bei dem Spiel werden Pokémon über die Smartphone-Kamera in die reale Welt projiziert, sodass der Nutzer sie fangen kann. Der Erfolg dieses simplen Konzepts zeigt das Potenzial der riesigen Nutzergruppe von Smartphones für AR-Anwendungen.

Auch können Smartphones als Displays für HMDs genutzt werden. Dabei ist jedoch meist die Kamera des Smartphones nicht mehr nutzbar, wodurch normalerweise die Wiedergabe von AR-Inhalten nicht möglich ist. Zudem ist die Pixeldichte von Smartphones meist geringer als bei speziellen AR-Anzeigegeräten, was die Immersion und Nutzererfahrung verschlechtern kann, da das Display sehr nah an den Augen des Nutzers ist.

3.4.3 Spatial Displays

4 Technologien

4.1 Oculus Quest 3

Im Laufe der Bachelorarbeit wird als Anzeigegerät für die AR-Anwendung die Oculus Quest 3 verwendet. Dabei handelt es sich um ein Standalone-AR- und VR-Headset, welches von Meta (ehemals Facebook) entwickelt wurde und Ende 2023 auf den Markt kam. Das HMD ist mit einem Qualcomm Snapdragon XR2 Gen2 Prozessor ausgestattet und ist dadurch selbst in der Lage auch anspruchsvolle Inhalte zu rendern. Es muss nicht wie andere HMDs an einen Computer oder ein Smartphone angeschlossen werden, sondern kann direkt verwendet werden. Über zwei Displays mit jeweils einer Auflösung von 2064 x 2208 Pixeln und einer Bildwiederholrate von 120Hz bietet das Headset eine hohe Bildqualität und eine flüssige Darstellung von Inhalten. Durch die Kombination durch geringe Latenz aufgrund des On-Board-Prozessors und der hohen Bildwiederholrate sollte das Gerät eine hohe Immersion und ein angenehmes Nutzungserlebnis bieten. Zudem hat die Oculus Quest 3 zwei Kameras auf der Vorderseite (also weg vom Nutzer gerichtet) sowie einen Tiefenprojektor, welche die Umgebung des Nutzers zu erfassen können und so AR-Anwendungen mithilfe von ImagePassthrough ermöglichen. Auch die Hände des Nutzers können mithilfe von Kameras, die nach unten gerichtet sind erfasst werden, sodass das Headset auch nativ Handtracking unterstützt. [9]

4.2 WebGL

WebGL ist eine JavaScript-API, die das rendern von 3D-Grafiken in einem Webbrowser ermöglicht. Die WebGL Spezifikation wird von der Khronos Group entwickelt, einer nonprofit Organisation, welche als ein Zusammenschluss aus über 180 Unternehmen aus der Industrie, darunter auch die Unternehmen hinter den größten Browsern Google (Chrome), Mozilla (Firefox), Apple (Safari) und Microsoft (Edge), technologiespezifische Standards entwickelt [10, 11]. Aufgrund dieser Mitarbeit verschiedener Unternehmen und der offenen Entwicklung der Spezifikation kann WebGL in fast allen modernen Webbrowsern verwendet werden.

WebGL basiert dabei auf der OpenGL ES Spezifikation, welche, ebenfalls von der Khronos Group, speziell für Embedded Systems, also Systeme ohne eigene Grafikkarte, entwickelt

wurde [12]. Aufgrund dieser Ausrichtung auf mobile Geräte, können mit WebGL Anwendungen entwickelt werden, die auf einfachen Geräten wie Smartphones oder aber auch auf VR- und AR-Headsets ohne zusätzliche Rechenleistung laufen [13, S.3].

4.3 WebXR

WebXR ist eine standardisierte API für Webanwendungen, welche es ermöglicht, immersive VR und AR Anwendungen für das Internet zu erstellen. Das World Wide Web Consortium (W3C), welches die WebXR-Standards definiert beschreibt WebXR wie folgt: „Die WebXR Device API bietet die notwendigen Schnittstellen, damit Entwickler ansprechende, komfortable und sichere immersive Anwendungen im Web für eine Vielzahl von Hardware-Formfaktoren erstellen können.“ [aus dem Englischen mit DeepL 14, 1. Introduction] Mit WebXR entwickelte Anwendungen können als Webanwendungen direkt von einem Webbrowser eines AR- oder VR-Geräts aus aufgerufen werden, ohne dass eine zusätzliche Installation der Anwendung notwendig ist.

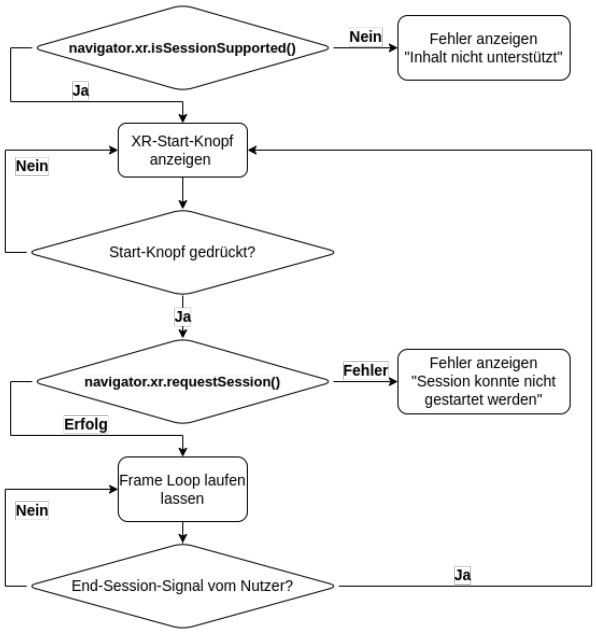

Abbildung 3: Application Flow von WebXR-Anwendungen
Quelle: Eigene Darstellung nach [14, Kap. 1.2. Application Flow]

Der Flow von WebXR-Anwendungen ist in Abbildung 3 dargestellt. Dabei wird zuerst beim Aufrufen der Anwendung geprüft, ob die angegebene Art von XR-Inhalt von der Hardware und dem User Agent (UA) unterstützt werden. Mit dem User Agent (UA) der Software, ist dabei die Software gemeint, mit welcher der Nutzer auf die Anwendung zugreift, also der Webbrowser des Nutzers. Ist die Art des XR-Inhalts von der Technik des Nutzers

unterstützt, kann die XR-Session vom Nutzer manuell über einen Knopf gestartet werden. Startet die Session erfolgreich, wird der Frame Loop der Anwendung gestartet. Der Frame Loop ist eine Schleifenfunktion, die kontinuierlich während der XR-Session ausgeführt wird um die einzelnen Frames, bzw. Bilder, für das XR-Display zu rendern. Dieser Frame Loop läuft, bis die Session durch den UA beendet wird. Befindet sich der UA noch auf der Seite der WebXR-Anwendung, wird wieder der Knopf zum Starten der XR-Session angezeigt, falls der Nutzer direkt wieder eine neue Session starten möchte.

4.3.1 Three.js

4.3.2 A-Frame

4.3.3 PlayCanvas

4.3.4 Babylon.js

Babylon.js ist eine Open-Source Rendering- und Game-Engine verpackt in einer JavaScript-Bibliothek die unter Anderem von einem Team von Entwicklern bei Microsoft entwickelt wird. Es bietet Support für WebGL 1.0/2.0 sowie WebGPU und hat eine Vielzahl von Funktionen, die es Entwicklern ermöglichen, detaillierte 3D-Modelle und -Szenen zu erstellen und zu rendern. Außerdem bietet es native Unterstützung und eine Dokumentation für WebXR, wodurch die Entwicklung von VR- und AR-Anwendungen vereinfacht wird [15].

4.4 Android Debug Bridge

5 Implementierung

5.1 Entwicklungsumgebung für WebXR und Oculus Quest 3

5.2 Implementierung der Anwendung

5.2.1 Interaktionskonzepte für Requirements

Beispiel 1: Explodierende Bauteile

Beispiel 2: Wolken von Anforderungen

Beispiel 3: Anforderungen als 3D-Objekte

5.3 User Tests

6 Zusammenfassung

6.1 Ergebnisse

6.2 Fazit

6.3 Ausblick

Literatur

1. INTERNATIONAL REQUIREMENTS ENGINEERING BOARD (IREB). *CPRE Glossary* [online]. 2024 [besucht am 2024-03-20]. Abgerufen unter: `https://www.ireb.org/de/cpre/glossary/`. Definition übersetzt von Englisch auf Deutsch.
2. RUPP, Christine; SOPHISTEN, die. *Requirements-Engineering und -Management*. 7., aktualisierte und erweiterte Auflage. München: Carl Hanser Verlag GmbH & Co. KG, 2020. Abgerufen unter DOI: `10.3139/9783446464308`.
3. HRUSCHKA, Peter. *Business Analysis und Requirements Engineering*. 3., updated edition. München: Carl Hanser Verlag GmbH & Co. KG, 2023. Abgerufen unter DOI: `10.3139/9783446478190`.
4. DALTON Jeremy; Acker, Olaf. *Immersive Unternehmenswelten: Wie Augmented, Mixed und Virtual Reality die Wirtschaft transformieren*. Schäffer-Poeschel Stuttgart, 2023. Abgerufen unter DOI: `10.34156/978-3-7910-5689-0`.
5. WÖLFEL, Matthias. *Immersive Virtuelle Realität*. Karlsruhe: Springer Vieweg Berlin, Heidelberg, 2023. Abgerufen unter DOI: `10.1007/978-3-662-66908-2`.
6. MILGRAM, Paul; COLQUHOUN, Herman u. a. A taxonomy of real and virtual world display integration. *Mixed reality: Merging real and virtual worlds*. 1999, Jg. 1, Nr. 1999, S. 1–26.
7. CARMIGNIANI, Julie; FURHT, Borko; ANISETTI, Marco; CERAVOLO, Paolo; DAMIANI, Ernesto; IVKOVIC, Misa. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*. 2011, Jg. 51, Nr. 1, S. 341–377. ISSN 1573-7721. Abgerufen unter DOI: `10.1007/s11042-010-0660-6`.
8. LINDNER, Jannik. *Must-Know Pokemon Go Usage Statistics* [online]. 2023 [besucht am 2024-04-08]. Abgerufen unter: `https://gitnux.org/pokemon-go-usage-statistics/`.
9. META. *Meta Quest 3 Homepage* [online]. 2024 [besucht am 2024-04-03]. Abgerufen unter: `https://www.meta.com/de/quest/quest-3/`.
10. GROUP, Khronos. *WebGL Overview* [online]. 2024 [besucht am 2024-04-08]. Abgerufen unter: `https://www.khronos.org/webgl/`.
11. GROUP, Khronos. *About the Khronos Group* [online]. 2024 [besucht am 2024-04-08]. Abgerufen unter: `https://www.khronos.org/about`.
12. GROUP, Khronos. *OpenGL ES Overview* [online]. 2024 [besucht am 2024-04-08]. Abgerufen unter: `https://www.khronos.org/opengles/`.

13. BARUAH, Rakesh. *AR and VR Using the WebXR API: Learn to Create Immersive Content with WebGL, Three.js, and A-Frame*. Entering VR Through WebXR. Berkeley, CA: Apress, 2021. ISBN 978-1-4842-6318-1. Abgerufen unter DOI: 10.1007/978-1-4842-6318-1_6.

14. WORLD WIDE WEB CONSORTIUM (W3C). *WebXR Device API* [online]. 2024 [besucht am 2024-04-02]. Abgerufen unter: https://www.w3.org/TR/webxr/.

15. BABYLON.JS. *Babylon.js Features* [online]. 2024 [besucht am 2024-04-03]. Abgerufen unter: https://www.babylonjs.com/specifications/.