

Hochschule Offenburg

Fakultät Medien

Studiengang Medien und Informationswesen



Bachelorarbeit zum Thema:

**Konzeption und Erstellung einer Informationsplattform zur
Darstellung von Smart Home Anwendungen**

Zur Erlangung des Grades Bachelor of Science

Vorgelegt von:

Timo Mattes

Studiengang: Medien und Informationswesen

Zusammenfassung

Gegenstand dieser Bachelorarbeit ist die Konzeption und Erstellung einer digitalen Informationsplattform zur Darstellung von Smart Home Anwendungen und deren Funktionsweise.

Als Grundlage hierfür dient der von der Hahn-Schickard-Gesellschaft für angewandte Forschung errichtete und betriebene „Smarte Caravan (SmaC)“, in welchen zu Demonstrationszwecken einige Sensoren sowie Aktoren aus dem Smart Home Bereich eingebaut wurden.

Für die Erstellung der digitalen Informationsplattform wird daher zunächst eine Analyse sowie ein Vergleich der verschiedenen Möglichkeiten zur interaktiven Darstellung von Informationsinhalten durchgeführt.

Anschließend wird eine Aufstellung von Anwendungen und Szenarien gesammelt, welche in der interaktiven Darstellung integriert werden sollen.

Letztendlich liegt eine interaktive Smart Home Umgebung vor, mit der verschiedene Funktionalitäten des Smart Home erlebbar werden und Informationen darüber gewonnen werden können.

Schlüsselwörter:

Smart Home, Smart Living, Vernetztes Wohnen, Ambient Assisted Living, IoT, Modellierung

Inhaltsverzeichnis

Inhaltsverzeichnis.....	III
Abbildungsverzeichnis.....	V
1. Einführung.....	6
1.1. Ziel.....	7
1.2. Vorgehensweise.....	7
2. Grundlagen und Stand der Technik.....	9
2.1. Smart Home.....	9
2.1.1. Definition.....	9
2.1.2. Funktionen / Anwendungsfelder.....	10
2.2. 3D-Modellierung.....	10
2.2.1. Vertices.....	11
2.2.2. Polygone.....	11
2.2.3. Grundkörper.....	13
2.2.4. Texturen und Texture Mapping.....	13
2.3. Webbasierte Informationsplattform.....	14
2.4. User Experience.....	14
2.5. Smart Caravan (SmaC).....	14
2.6. Stand der Technik.....	15
3. Konzeption.....	19
3.1. Zielgruppe.....	19
3.2. Konzepte.....	19
3.2.1. Konzept 1 – One-Pager-Website.....	19
3.2.2. Konzept 2 – 360°-Rundgang.....	20
3.2.3. Konzept 3 – 3D-Spiel.....	21
3.3. Konzeptauswahl.....	21
3.4. Vorstellung Programmauswahl.....	22
3.4.1. 3D-Modellierungssoftware.....	22

3.4.2.	Spiel-Engine	23
3.4.3.	IDE & Code-Editor.....	23
4.	Umsetzung.....	25
4.1.	Modellierung	25
4.2.	Belichtung.....	30
4.3.	Design	34
4.3.1.	Farbpalette	34
4.3.2.	Logo	34
4.3.3.	UI-Elemente.....	35
4.4.	Programmierung	38
4.4.1.	Navigation	38
4.4.2.	Funktionalität	40
4.4.3.	UI-Elemente.....	42
4.4.4.	Szenarien	44
4.5.	Durchführung von Tests	48
5.	Schlussbetrachtung und Ausblick.....	49
	Literaturverzeichnis.....	VI
	Anhang 1 – Fragebogen zur Durchführung von Tests.....	VI

Abbildungsverzeichnis

Abbildung 1: Vertices	11
Abbildung 2: Polygone einer Low-Poly Kugel.....	12
Abbildung 3: Polygone einer High-Poly Kugel.....	12
Abbildung 4: Grundkörper des Pro-Builder Plug-Ins.....	13
Abbildung 5: SmaC der Hahn-Schickard-Gesellschaft für angewandte Forschung	15
Abbildung 6: Rundgang ECO System HAUS.....	16
Abbildung 7: BAR Technologies Ltd, WindWings Position2	17
Abbildung 8: BAR Technologies Ltd, WindWings Position 1	17
Abbildung 9: Grundriss aus Autodesk Revit	26
Abbildung 10: 3D-Modell aus Revit.....	26
Abbildung 11: Modellierter Fensterrahmen	27
Abbildung 12: Fertig modelliertes Fenster.....	27
Abbildung 13: Modell-Zwischenstand, High-Poly-Objekte, Badezimmer.....	28
Abbildung 14: Modell-Zwischenstand, High-Poly-Objekte, Wohnraum.....	28
Abbildung 15: Austausch hig-poly durch low-poly 3D-Modell einer Steckdose.....	29
Abbildung 16: Modellierung Dusche.....	29
Abbildung 17: Ambient Occlusion Effect deaktiviert	32
Abbildung 18: Ambient Occlusion Effect aktiviert	32
Abbildung 19: Farbpalette Smart Home Simulator.....	34
Abbildung 20: Erstellung Logo.....	35
Abbildung 21: Erstellung Info-Button.....	35
Abbildung 22: Info-Fenster.....	36
Abbildung 23: Erstellung Hauptmenü	37
Abbildung 24: Erstellung Aufgaben-Übersicht.....	38
Abbildung 25: Bewegungsrichtung in Abhängigkeit des Inputs.....	39
Abbildung 26: Aufgaben-Übersicht, geschlossen.....	43
Abbildung 27: Flussdiagramm zum Ablauf "Sturzmatten-Szenario"	45

1. Einführung

Systeme zur Kontrolle und Steuerung des menschlichen Umfeldes gewinnen in den vergangenen Jahren immer mehr an Interesse (Struck 2021). Sie werden jedoch auch schneller weiterentwickelt und bezogen auf die Bedienbarkeit einfacher gestaltet (Infraserv GmbH & Co. Höchst KG o.J.). Dadurch werden die Systeme immer benutzerfreundlicher, wodurch nicht nur Informatiker und technisch interessierte Personen Zugang erhalten, sondern auch die breite Bevölkerung und technisch unversierte Personen. Der Wunsch, sich seine Umgebung und seinen Lebensraum individuell und komfortabel anpassen zu können wird damit zunehmend in Erfüllung gebracht (Struck 2021).

Ziel ist die Steigerung der Lebensqualität, welche durch die Automatisierung von alltäglichen Aktivitäten erzielt werden soll, die die Privatperson bislang händisch durchführen musste (Luber und Litzel 2019). Besonders wichtig wird der Aspekt der Steigerung der Lebensqualität im Kontext von körperlich eingeschränkten Personen. Beispiele sind Personen mit körperlichen Behinderungen, mit Verletzungen oder die aufgrund ihres Alters die vollumfängliche Steuerung und Regulierung ihres Eigenheimes nicht bewältigen können (Ambient Assisted Living Deutschland o.J.).

Durch die Vernetzung verschiedener Technologien im häuslichen Umfeld können damit Abläufe automatisiert werden und müssen so von den Bewohnern manuell nicht mehr ausgeführt werden.

Doch neben der Steigerung der Wohnqualität soll die Installation eines Smart Home auch die Einsparung von Energie fördern. Dies kann durch das bequeme Steuern von Heizung, Fenstern, Beschattungs- oder Beleuchtungssystemen über ein mobiles Endgerät oder durch programmierte, automatisierte Szenarien erfolgen (Luber und Litzel 2019).

Letztlich kann durch im Smart Home System integrierte Überwachungs- und Alarmanlagen auch eine erhöhte Sicherheit für das Eigenheim geschaffen werden (Luber und Litzel 2019).

Trotz der vielen Möglichkeiten und Mehrwerte kämpfen die vollständige Gebäudeautomatisierung sowie zahlreiche Smart Home Gadgets gegen einen schlechten Ruf und viele Vorurteile an. Angst vor fehlendem Datenschutz oder das Abtun des Smart Home als unnütze Spielerei sind noch weit verbreitet in der Gesellschaft (Stadtwerke Düsseldorf AG o.J.) (Wolfangel 2019).

Unwissenheit zum Thema zu verhindern, die tatsächlichen Möglichkeiten und Mehrwerte eines Smart Home beispielhaft in einer digitalen Umgebung zu veranschaulichen und interessierten Personen Informationen über die Funktionsweise und Integration im eigenen Haus zu bieten, ist die Zielsetzung dieser Arbeit.

Aus dieser Zielsetzung lassen sich folgende Leitfragen ableiten:

- In welcher Form können die Thematik Smart Home sowie programmierte Szenarien und Abläufe dieses Kontexts dargestellt werden?
- Welche Informationen sind essentiell für das Verständnis von Smart Home und der Funktionsweise?

1.1. Ziel

In dieser Arbeit soll eine interaktive, digitale Informationsplattform erstellt werden, die es interessierten Endnutzern erlaubt, sich näher mit der Thematik Smart Home auseinanderzusetzen. Es sollen dabei Informationen zum Stand der Technik integriert, mögliche Sensoren und Aktoren beschrieben und unterstützende, automatisierte Szenarien dargestellt werden. Dabei steht die Visualisierung bzw. das Kennenlernen von Smart Home und möglichen Lösungen im Mittelpunkt. Um einen möglichst hohen Realbezug herstellen zu können, soll eine virtuelle, dreidimensionale Umgebung erstellt werden, welche die Realität widerspiegelt und damit die Integration eines Smart Home und dessen Anwendungen in den Alltag veranschaulicht.

Die Informationsplattform soll im Aufbau und Inhalt an den bereits existierenden ‚Smarten Caravan (SmaC)‘ der Hahn-Schickard-Gesellschaft für angewandte Forschung angelehnt werden. Somit steht auch mit dem digitalen Smart Home Simulator, der als Resultat dieser Arbeit entstehen soll, das Erleben von Smart Home im Mittelpunkt.

Die Zugänglichkeit wird dabei möglichst offen gestaltet, weshalb der Simulator über das Internet zugänglich gemacht wird.

1.2. Vorgehensweise

Die Erstellung der digitalen Informationsplattform kann in mehrere Abschnitte unterteilt werden.

Im ersten Abschnitt soll eine grundlegende Recherche zum Stand der Technik, den Möglichkeiten und Grundlagen in der Erstellung einer interaktiven, digitalen Informationsplattform durchgeführt werden. Darauf aufbauend sollen anschließend verschiedene Möglichkeiten und Tools zur Erstellung miteinander verglichen werden. Dabei soll die Art und Weise der Informationsplattform, deren Umfang sowie die Technik zur Erstellung festgelegt werden.

Anschließend wird die praktische Erstellung der Informationsplattform mit den ausgewählten Mitteln gestartet und schriftlich in Form einer Dokumentation festgehalten und erläutert.

In der gesamten Arbeit wird aus Gründen der Lesbarkeit ausschließlich die männliche Form für die Ansprache von Personen gewählt. Nichtsdestoweniger beziehen sich die Angaben auf Angehörige aller Geschlechter.

2. Grundlagen und Stand der Technik

Im nachfolgenden Kapitel werden grundlegende Begrifflichkeiten dieser Arbeit erläutert. Auch soll der Stand der Technik anhand von vergleichbaren Projekten ermittelt werden.

2.1. Smart Home

Für die Ausarbeitung dieser Arbeit spielen Smart Home Technologien und Anwendungen eine grundlegende Rolle. Jedoch ist Smart Home ein sehr breit angewandter Begriff, der, abhängig von Branche und Kenntnissen, von jedem etwas anders definiert wird.

2.1.1. Definition

Die Fokusgruppe *Connected Home* des Nationalen IT-Gipfels definiert den Begriff Smart Home wie folgt: „Smart Home dient als Oberbegriff für technische Verfahren und Systeme in Wohnräumen und -häusern, in deren Mittelpunkt eine Erhöhung von Wohn- und Lebensqualität, Sicherheit und effizienter Energienutzung auf Basis vernetzter und fernsteuerbarer Geräte und Installationen sowie automatisierbarer Abläufe steht.“ (BITKOM 2014, S. 2).

Dabei wird deutlich, dass es sich bei Smart Home um einen Begriff handelt, der viele verschiedene technische Anwendungen beschreibt, die in Wohnräumen installiert werden. Ein deutscher Begriff, der durch diese Definition synonym verwendet werden kann ist damit die Gebäudeautomation. Dieser beschreibt grundsätzlich die Automation eines Gebäudes und damit ein weitgehend selbst funktionierendes Gebäude (Karolin Wisser und Wisser 2018).

Auch lassen sich aus der Definition der Fokusgruppe *Connected Home* folgende Voraussetzungen, die für ein Smart Home erfüllt werden müssen, ableiten.

- Da es sich beim Begriff Smart Home um einen Oberbegriff verschiedener technischer Verfahren und Systeme handelt, spricht man von einem Smart Home erst dann, wenn viele Geräte, d.h. Sensoren und Aktoren, integriert und miteinander vernetzt oder vernetzbar sind
- Eingebundene Sensoren und Aktoren des Smart Home sind updatefähig und ggf. von außen steuerbar
- Es sind offene Schnittstellen vorhanden (sogenannte APIs)
- Es ist eine Verbindung zum Internet vorhanden
- Geräte können getagt werden
- Integrierte AAL-Funktionen sind barrierefrei
- Energieeffizienz und Nachhaltigkeit werden berücksichtigt

(BITKOM 2014, S. 2)

2.1.2. Funktionen / Anwendungsfelder

Wie bereits Eingangs erläutert, ist ein Smart Home ein Zusammenspiel technischer Verfahren mit unterschiedlicher Zielsetzung und Verwendung von verschiedenen Sensoren und Aktoren. Die Funktionen bzw. Anwendungsfelder der Sensoren und Aktoren und damit der Verfahren und Szenarien, lassen sich wie folgt unterteilen:

Beleuchtung: Das Einbinden der Beleuchtung in ein Smart Home System ermöglicht durch das automatisierte Abschalten von Lichtern auf der einen Seite eine Energieeinsparung. Diese kann durch die Nutzung von Bewegungs- und Präsenzmeldern erreicht werden, die wahrnehmen, ob sich eine Person im Erfassungsbereich befindet.

Auf der anderen Seite lassen sich die Lichtverhältnisse im Gebäude auch in Abhängigkeit des Tageslichts regeln (Konstantlichtregelung). Dazu werden die von Helligkeitssensoren gemessenen Werte an die Leuchten weitergegeben und beeinflussen die Helligkeit sowie die Farbe des Lichtes (Karolin Wisser und Wisser 2018).

Sicherheit: Ein sehr wichtiger Aspekt des Smart Home ist die Einbindung von Komponenten zur Steigerung der Sicherheit. Dabei stehen neben dem Schutz vor Einbrüchen auch Gefahren wie Schäden durch Brände, Schimmel oder Wasser im Fokus. Weit verbreitet sind dabei smarte Geräte wie Alarmanlagen, Türklingeln, Überwachungskameras oder Rauchmelder (homeandsmart GmbH o.J.).

Klimatisierung: Bei der Temperierung, d.h. Lüftung, Heizung und Kühlung und Be- und Entfeuchtung der Luft in Gebäuden spricht man von Klimatisierung. Bei der Automatisierung von Klimaanlage wird diese durch Informationen von beispielsweise Temperatursensoren und Zeitangaben automatisch geregelt. Dabei ergeben sich enorme Potentiale in der Energieeffizienz und der Einsparung von Energie. Mit der Einbindung ins Smart Home können verschiedene Modi, wie Schlaf-Modus, Zeit-Modus oder Außer-Haus-Modus dazu beitragen weniger Energie zu verbrauchen (VNR Verlag für die Deutsche Wirtschaft AG, Computerwissen 2021).

2.2. 3D-Modellierung

Der Begriff Modellierung stammt von dem übergeordneten Begriff Modell (lat. *modulus* = Maßstab, Muster), der im wissenschaftlichen Sprachgebrauch mehrere unterschiedliche Bedeutungen besitzt. Die gebräuchlichste ist die des *analogen Realitätsausschnitts* (Prof. em. Dr. Schlottke, Peter F. o.J.). Demnach beschreibt ein Modell einen Ausschnitt der Realität oder bildet diesen ab (Phillip 2020). Die Modellierung beschreibt daraus ableitend den Vorgang der Entwicklung eines Modells (Serlo Education e. V. o.J.).

Bei der 3D-Modellierung handelt es sich um den computergestützten Prozess der Entwicklung einer mathematischen koordinatenbasierten Darstellung von Oberflächen in einem dreidimensionalen simulierten Raum (Petty o.J.). Diese werden in drei Dimensionen, anhand von spezieller Software, durch das Manipulieren von Punkten (Vertices), Kanten (Edges) und Flächen (Polygons) erstellt.

2.2.1. Vertices

Ein Vertex beschreibt als mathematisches Gebilde eine Koordinate im dreidimensionalen Raum. Die Koordinate besitzt daher einen x-, y- und z-Wert, welche die Position im Raum beschreiben. Aus Vertices wird die gesamte Geometrie eines Körpers aufgebaut, weshalb sie als die Grundlage der Geometrie in der 3D-Computergrafik gelten (Vertex – DGL Wiki 2014) (Wenke 2012). Die in Abbildung 1 zu sehenden gelben Punkte stellen die Vertices der zu sehenden Kugel dar. Jeder dieser Vertices beschreibt einen Punkt im dreidimensionalen Raum und kann unabhängig der anderen Vertices verschoben werden.

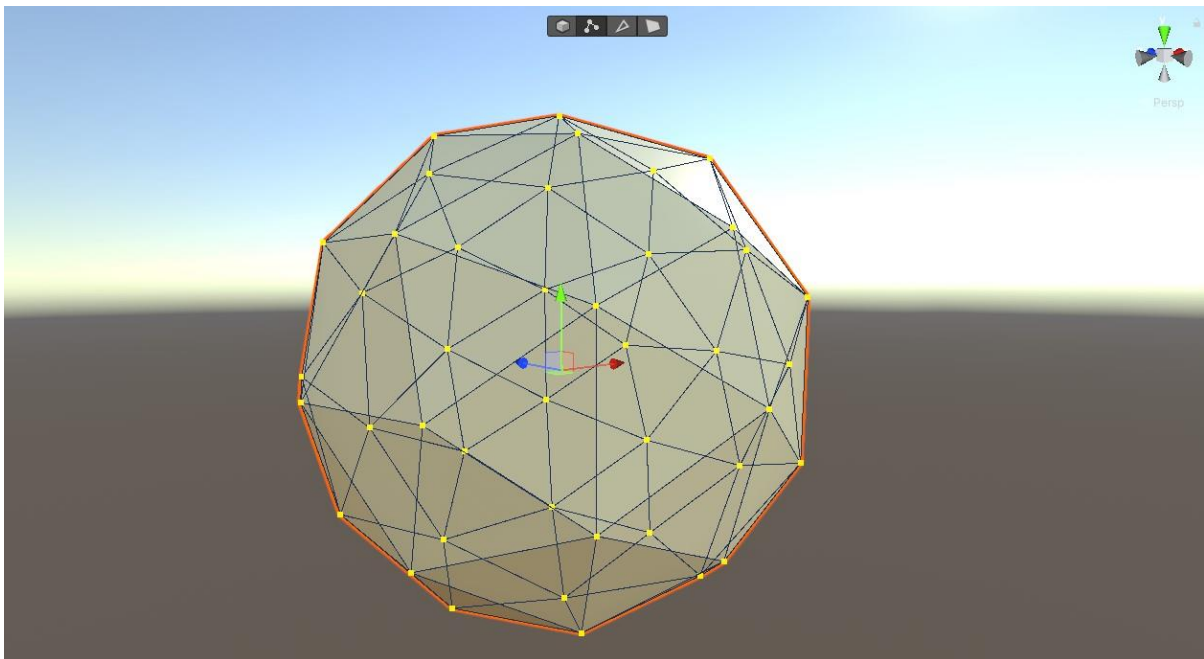


Abbildung 1: Vertices

2.2.2. Polygone

Polygone werden zur Darstellung von Körpern und Flächen genutzt und werden durch eine beliebige Anzahl an Vertices definiert. Benachbarte Eckpunkte werden mit Linien verbunden, welche die Kanten des Polygons bilden.

Ein Polygon im zweidimensionalen kartesischen Koordinatensystem besteht immer aus mindestens drei Eckpunkten. Damit ist das einfachste Polygon mit den wenigsten Punkten und

Kantenlinien das Dreieck. Im dreidimensionalen Koordinatensystem werden die Koordinatenwerte durch einen Z-Wert ergänzt.

Bei mehreren miteinander verbundenen Polygonen spricht man von Polygonnetzen (auch Mesh) (Bender und Brill 2006).

Je nachdem, aus wie vielen Polygonen ein Polygonnetz beziehungsweise ein 3D Modell besteht, lässt es sich als low oder high Poly(gon) Modell bezeichnen. Durch eine höhere Anzahl an Polygonen, die ein Modell beschreiben, wird dieses detaillierter und damit realgetreuer (Barth 2017). Jedoch benötigt die Modellierung, je mehr Polygone verwendet werden, einen höheren Zeitaufwand. Auch wird eine höhere Grafikleistung zur Berechnung benötigt (Michael 2021). Bei der Erstellung von Grundkörpern (siehe 2.2.3) kann ausgewählt werden, wie detailliert diese sein sollen, d.h. aus wie vielen Vertices und Polygonen diese bestehen sollen. Wie in Abbildung 2 und Abbildung 3 dargestellt, ist beispielsweise eine Kugel mit mehr Polygonen runder als eine mit nur sehr wenigen. Jedoch wird auch ein höherer Rechenaufwand zur Darstellung benötigt.

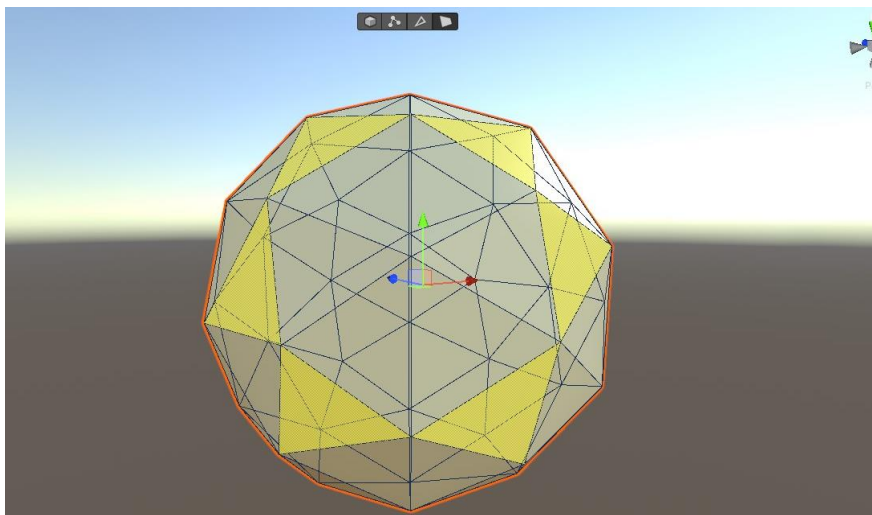


Abbildung 2: Polygone einer Low-Poly Kugel

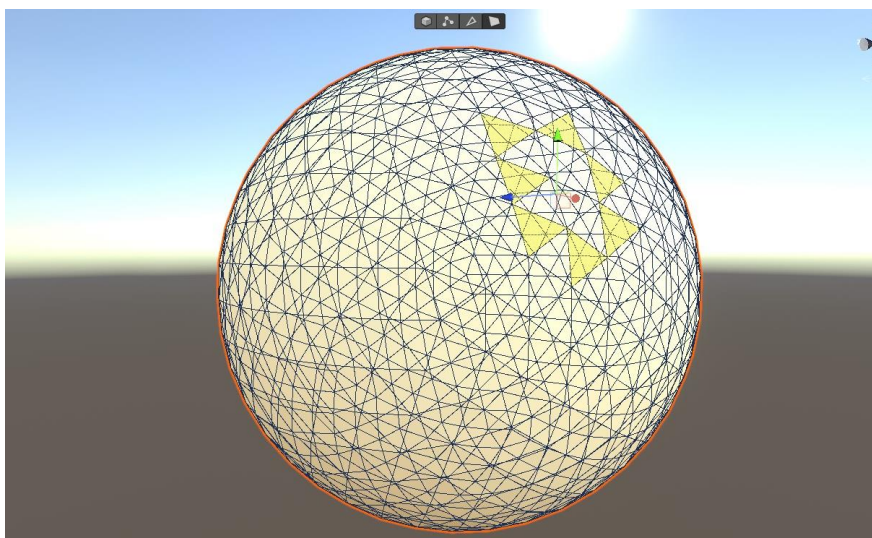


Abbildung 3: Polygone einer High-Poly Kugel

2.2.3. Grundkörper

Für die Modellierung von mathematisch exakten Körpern wie Kugeln oder Pyramiden können sogenannte Grundkörper als bereits fertig gebaute Objekte verwendet werden, bei welchen lediglich Zahlenwerte eingegeben werden müssen um beispielsweise die Größe einer Kugel oder das Seitenverhältnis eines Quaders vorzugeben. Grundkörper mit gebogenen Flächen benötigen außerdem die Information darüber, wie fein das Modell unterteilt werden soll, das heißt wie hoch die Flächen- und damit auch die Polygonanzahl sein soll (Garthe o.J.).

In Abbildung 4 sind Grundkörper der Auswahl des Pro-Builder Plug-Ins für Unity dargestellt. Darunter ein einfacher Würfel, eine Kugel, ein Zylinder, ein Prisma, ein Kegel sowie ein Ring.

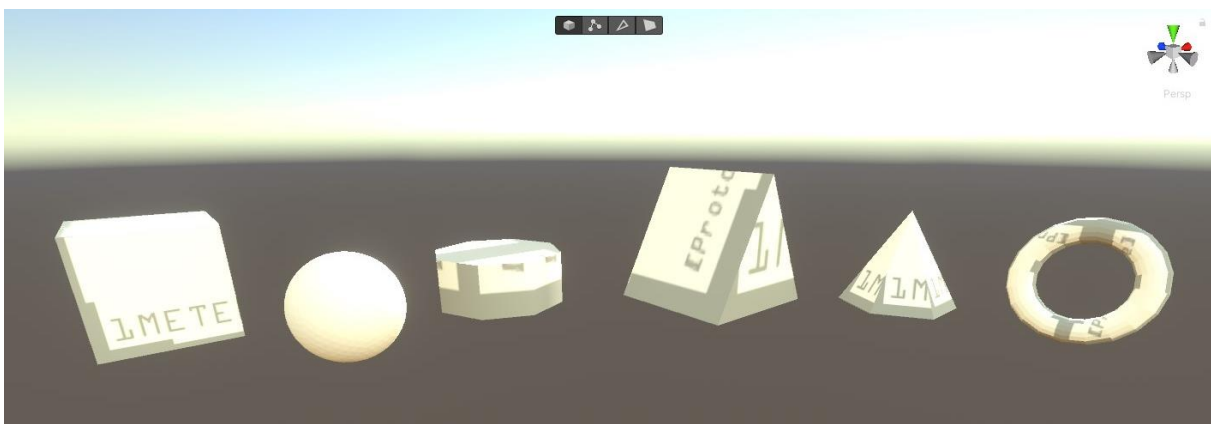


Abbildung 4: Grundkörper des Pro-Builder Plug-Ins

2.2.4. Texturen und Texture Mapping

Um einem 3D-Objekt eine bestimmte Oberfläche zuweisen zu können und damit ein Material veranschaulichen zu können, benötigt man Texturen. Bei diesen handelt es sich um einfache 2D-Grafiken, welche auf die Oberfläche eines 3D-Objektes projiziert werden sollen (VisCircle GmbH o.J.a).

Um eine 2D-Textur einem 3D-Objekt korrekt zuweisen zu können, werden spezielle Verfahren (Texture-Mapping) benötigt, um Bildabschnitte des 2D-Bildes den verschiedenen Flächen des 3D-Objektes zuordnen zu können (Garthe o.J.). Im Folgenden wird das verbreitetste Verfahren, das UV-Mapping genauer erläutert.

Beim UV-Mapping handelt es sich um die Technik der Erstellung einer 2D-Textur, sowie des Auftragens dieser auf ein 3D-Objekt. U und V beschreiben dabei die Achsen eines 2D-Bildes. Die Grundlage des UV-Mappings sind die Polygone des 3D-Objektes, dem eine Textur zugewiesen werden soll. Diese werden zunächst auf einer 2D-Ebene verteilt, sodass jedes Polygon einer bestimmten Fläche auf der 2D-Ebene entspricht. Anschließend kann die Textur bzw. das

Bild auf die Flächen gelegt werden, sodass jedes Polygon mit der dafür vorgesehenen Textur belegt wird (VisCircle GmbH o.J.b).

2.3. Webbasierte Informationsplattform

Die ursprüngliche Bedeutung des aus dem Französischen stammenden Wortes Plattform, eine „ebene Fläche auf hohen Gebäuden, Türmen o.Ä. (von der aus man einen guten Ausblick hat)“ (Bibliographisches Institut GmbH o.J.b) wurde im Sprachgebrauch zu einer Bezeichnung einer „Basis, [oder eines] Standpunkt[s], von dem bei Überlegungen, Absichten, Handlungen [...] ausgegangen wird“ (Bibliographisches Institut GmbH o.J.b). Im aktuellen Kontext kann diese Bedeutung auf die enthaltenen Informationen und Demonstrationen angewendet werden, die den Nutzer durch Informationsgewinnung zum Thema Smart Home aufklären sollen.

Ebenso hat sich die Verwendung des Wortes Plattform zur Beschreibung eines „Ort oder Personenkreis, der dem Austausch und der Verbreitung von Ideen, Anschauungen oder Produkten dient“ (Bibliographisches Institut GmbH o.J.b) etabliert. Daraus folgend beschreibt eine Informationsplattform meist eine Internetplattform (=Webseite) die die Verbreitung von Informationen und den Austausch von bestimmten Informationen zum Ziel hat (Bibliographisches Institut GmbH o.J.a).

Auch der Begriff webbasierte Informationsplattform beschreibt den aus dieser Arbeit resultierenden Simulator treffend, da der Besuch und Informationsaustausch letztlich mit Hilfe eines Browsers und über Webserver funktioniert.

2.4. User Experience

Für die Erstellung digitaler und interaktiver Plattformen spielt die User Experience eine zentrale Rolle, da diese das Erlebnis des Anwenders bei der Nutzung des Produktes beschreibt (Preim und Dachsel 2015). Das heißt beim User-Centered-Design-Prozess werden bereits in der Planung und Erstellung eines Produktes die subjektiven Wahrnehmungen, Bedürfnisse und Erlebnisse eines Anwenders miteinbezogen.

2.5. Smart Caravan (SmaC)

Grundlage dieser Arbeit ist der von der Hahn-Schickard-Gesellschaft für angewandte Forschung entwickelte Smarte Caravan (SmaC) (dargestellt in Abbildung 5). Ziel dieses zwischen 2018 und 2020 laufenden, staatlich geförderten Projektes war es, eine Transfer- und Innovationsplattform zu schaffen, mit der Informationen zu den Themen Smart Home, Smart Living und Ambient Assisted Living in die Bevölkerung getragen werden sollen. Darüber hinaus dient

der Wohnwagen als Plattform für Ideen und Innovationen, an welchen zusammen mit Praxispartnern verschiedener Branchen geforscht werden kann.



Abbildung 5: SmaC der Hahn-Schickard-Gesellschaft für angewandte Forschung

2.6. Stand der Technik

Beispiele zur digitalen Visualisierung von realen Umgebungen sowie zum Transport von Informationen finden sich aktuell bereits viele. Diese werden jedoch in den unterschiedlichsten Kontexten angewendet und folgen daher unterschiedlichen Zielsetzungen.

Aus diesem Grund werden folgend Branchen und Einsatzbereiche aufgeführt, in welchen man bereits vermehrt webbasierte, interaktive Informationsplattformen findet.

Die Suche wird dabei auf online und öffentlich verfügbare Webprojekte beschränkt, da für diese Arbeit die öffentliche Zugänglichkeit über das Internet ein zentrales Kriterium darstellt.

Eingeteilt werden können die gefundenen Beispiele in folgende Kategorien:

- Architektur & Bau
- Messen & Events
- Virtuelle Campusführungen
- Touren mit gesteigerter Interaktivität

Architektur & Bau

Häufig werden interaktive Webseiten, in welchen man sich durch Wohnungen und Häuser oder deren Konzepte klicken kann, im Rahmen von Architektur bzw. Baurundgängen verwendet. Dabei sollen Fertighäuser, Musterhäuser oder Konzepte von größeren Gebäuden dargestellt und erlebbar gemacht werden.

Die Interaktivität steht in diesem Kontext nicht im Vordergrund, vielmehr nur die Darstellung des Gebäudes an sich. Die Projekte basieren dabei meist auf bereits realisierten Gebäuden, welche mit 360°-Fotografie erfasst und dargestellt werden.

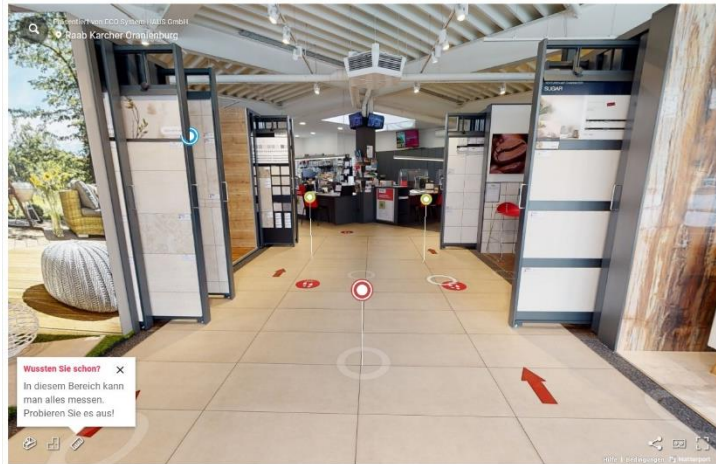


Abbildung 6: Rundgang ECO System HAUS

Beispiele finden sich in den virtuellen Besichtigungsmöglichkeiten der Musterhäuser des Unternehmens WeberHaus GmbH & Co. KG oder in den virtuellen Musterhäusern und Ausstellungen der Firma ECO System HAUS GmbH (siehe Abbildung 6).

Messen & Events

Spätestens seit der im Jahr 2020, durch die Corona-Pandemie hervorgerufenen Einschränkungen von Menschenansammlungen und damit dem massiven Ausfall jeglicher Messen, Ausstellungen und sonstigen Live Events, ist die Verbreitung von virtuellen Messen und Live Events stark gestiegen.

Hier liegt der Fokus vor allem auf dem Transport von Informationen und der Darstellung von Produkten oder einem gesamten Unternehmen. In Fällen von Live Events oder Live Messen ist außerdem das Einbetten von Live-Streams oder Chat-Funktionen ein wesentlicher Bestandteil. Daher ist neben der dreidimensionalen Simulation von Messeräumlichkeiten wichtig, Informationen auf verschiedene Arten zu integrieren und eine Live-Kommunikation zu ermöglichen.

Grundlage sind hier meist 3D-modellierte Umgebungen, welche eine Messe widerspiegeln sollen. Informationen werden anhand von Informations-Buttons oder klickbaren Icons eingebettet.

Virtuelle Campusführungen

Virtuelle Rundgänge und Besichtigungen bieten sich dazu an, Personen, Besuchern oder möglichen Kunden bereits im Voraus Einblicke in seine Räumlichkeiten zu gewähren. Genutzt wird diese Möglichkeit daher mittlerweile häufig von Hochschulen und Universitäten, um Studieninteressierten einen Eindruck des Campus zu geben.

Anwendung findet dieses Konzept beispielsweise bei der Technischen Hochschule Ingolstadt. Diese bietet anhand von 360°-Bildern aus den verschiedensten Gebäuden, Räumen und Perspektiven des Campus einen virtuellen Einblick in die Hochschule.

Touren mit gesteigerter Interaktivität

Die Vorgehensweise zur Erstellung der bisher vorgestellten Beispiele gleicht sich insofern, dass der Visualisierung der Umgebung 360°-Aufnahmen zugrunde liegen. Durch den Einsatz von Bild-Aufnahmen ist die Navigation stets in Form von vorgegebenen Standpunkten eingeschränkt. Daraus folgt außerdem, dass sich an der Umgebung selbst nichts ändern lässt. Touren mit gesteigerter Interaktivität hingegen basieren nicht auf statischen 360°-Aufnahmen, sondern bilden eine 3D-Umgebung mit darin enthaltenen 3D-Modellen in Laufzeit ab. Das ermöglicht auf der einen Seite eine freie Navigation durch die Umgebung und bietet außerdem die Möglichkeit die Umgebung und ihre Modelle zu verändern.

Die Firma BAR Technologies Ltd nutzt zur Darstellung ihrer WindWings-Technologie eine dreidimensionale Umgebung, in welcher die Technologie interaktiv demonstriert wird. Dargestellt wird ein Computer-modelliertes Schiff, ausgestattet mit den WindWings. Außerdem finden sich zwei auswählbare Dropdown-Menüs. Mit diesen kann die Kamera und deren Position sowie die Position der WindWings verändert werden.

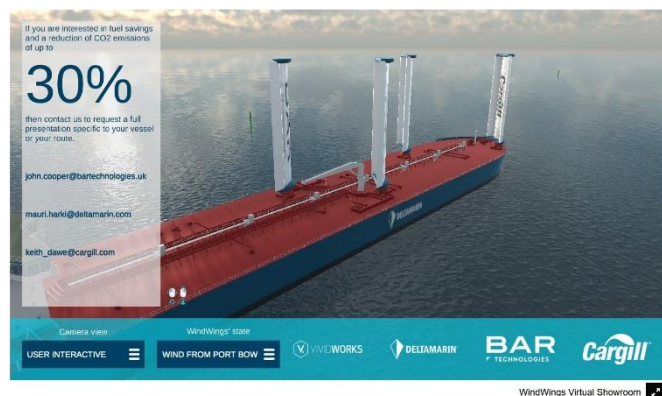


Abbildung 7: BAR Technologies Ltd, WindWings Position 2



Abbildung 8: BAR Technologies Ltd, WindWings Position 1

Ein weiteres, in der Navigation völlig freies Beispiel bietet das Deutsche Zentrum für Luft- und Raumfahrt e.V. (DLR) in der virtuellen Ausstellung zur Rosetta Mission der europäischen Welt- raumorganisation ESA. Darin werden Informationen über die Raumsonde Rosetta sowie über die Mission veranschaulicht dargestellt.

Interaktive Websites

Eine interaktive Darstellung von Informationen im Web kann, abgesehen von 3D-Spielen und Animationen oder 360°-Rundgängen auch durch die Gestaltung einer einfachen Website erfolgen. Häufig werden hier sogenannte One-Pager gestaltet, bei welchen sich alle Informationen auf einer Seite befinden. Dabei kann durch Scroll-Activated-Animations eine Form der Interaktivität erzeugt werden. Ein Beispiel einer solchen Webseite bietet die Seite mit dem Titel *ATMOS – The Surrealist Flight Experiment*. Durch das Scrollen wird hier eine Animation eines fliegenden Flugzeugs gestartet, welches an Informationen in Form von Text vorbeifliegt.

Auch Apple nutzt zur Darstellung der Produkte und der Informationen darüber die Technik von Scroll-Activated-Animations. So werden, beispielsweise auf den Überblick-Seiten zu den jeweils aktuellen Produkten wie dem iPhone 13, Informationen in Textform erst beim Scrollen eingeblendet. Neben Texten werden Bilder beim Scrollen eingeblendet oder Videos bzw. Animationen gestartet. Ein interaktives Feature ist hier außerdem, dass die Farbe des iPhones an einer bestimmten Stelle über Buttons gewechselt werden kann. Dadurch wird jeweils das aktuelle Bild des Smartphones durch das Bild des Smartphones mit der gewünschten Farbe ausgetauscht.

3. Konzeption

In diesem Kapitel wird zunächst die Zielgruppe, die mit dieser Arbeit adressiert wird, definiert und beschrieben. Außerdem sollen die Möglichkeiten, Programme und Anbieter, die zur Erstellung des Smart Home Simulators in Frage kommen, recherchiert und verglichen werden. Auf Basis des Vergleichs soll letztlich eine Wahl der Art der Informationsplattform und der zu nutzenden Programme und Tools getroffen werden.

3.1. Zielgruppe

Die Zielgruppe, die mit dem Smart Home Simulator adressiert werden soll, besteht hauptsächlich aus interessierten Menschen mit einem technischen Grundverständnis. Aber auch Personen, für die aufgrund einer körperlichen Beeinträchtigung oder aufgrund ihres Alters eine Automatisierung von alltäglichen Aufgaben eine Steigerung ihrer Lebensqualität bedeutet, sind Teil der Zielgruppe. Da die Arbeit eine digitale Anwendung als Ergebnis haben soll, welche über das Internet in einem Browser aufgerufen wird, ist ein gewisses Verständnis für das Internet und dem Umgang damit eine Voraussetzung. Da bereits heute 100% der unter 50-Jährigen sowie 95% der zwischen 50- und 69-Jährigen deutschen regelmäßig das Internet nutzen (ARD/ZDF-Forschungskommission 2022) ist die digitale Darstellung von Smart Home Technologien auch durchaus für Personen relevant, welche aus Gründen der häuslichen Pflege interessiert an Unterstützung durch Smart Home Anwendungen sind.

3.2. Konzepte

Wie bereits in Kapitel 2.6 durch Beispiele verdeutlicht wurde, können für diese Arbeit relevante Möglichkeiten zur Erstellung einer digitalen, interaktiven Informationsplattform in drei Arten unterteilt werden. Zum einen kann eine einfache Website erstellt werden, in welche Informationen sowie multimediale Elemente eingebunden werden. Eine weitere Möglichkeit bietet das Erstellen eines 360°-Rundgangs. Die dritte Art, eine solche Informationsplattform zu schaffen, ist die Erstellung eines 3D-Spiels, welches über den Browser zugänglich gemacht wird.

Im Folgenden werden die drei Konzepte näher vorgestellt.

3.2.1. Konzept 1 – One-Pager-Website

Bei diesem Konzept soll eine Website zur Informationsvermittlung dienen. Dabei wird eine auf HTML basierende Website erstellt und durch die Skriptsprache JavaScript erweitert. Die Infor-

mationen sollen dabei auf lediglich einer Seite untergebracht werden, um eine Übersichtlichkeit zu erreichen. Beschrieben wird auf der Seite eine Auswahl an Sensoren und Aktoren sowie weiterer Technik aus dem Smart Home Bereich. Da ein Ziel der Informationsplattform auch die Demonstration der Funktionsweise der verschiedenen Akteure im Smart Home ist, sollen durch Bilder, Videos und Animationen Smart Home Szenarien veranschaulicht werden.

Unterteilt wird die Seite in die verschiedenen Räume eines Hauses, um eine gewisse Struktur in den Aufbau der Website zu bringen und die Verständlichkeit zu steigern.

Interaktionen werden unter anderem in Form von Buttons realisiert, die das Abspielen von Videos und Animationen oder das Wechseln von Bildern zur Folge haben.

Eine Möglichkeit ist das Erstellen eines Bilds mit demselben Ausschnitt einer Wohnung in zwei Varianten. Beispielsweise ein Bild eines Wohnzimmers mit an- und eines mit ausgeschalteten Lichtern. Diese lassen sich dann jeweils mit dem Klick auf einen Button austauschen, was das An- und Ausschalten eines Lichts simuliert.

Die gesamte Website wird dabei selbst erstellt, wozu lediglich eine integrierte Entwicklungsumgebung (IDE), wie beispielsweise Microsoft Visual Studio, benötigt wird. Die Seitenstruktur sowie der Inhalt wird anhand von HTML festgelegt. Die visuelle Darstellung sowie Formatierung werden anhand von Gestaltungsvorlagen wie CSS bestimmt. Um Benutzerinteraktionen auszuwerten oder Inhalte zu verändern und nachzuladen bzw. generieren zu können, wird zusätzlich die Skriptsprache JavaScript verwendet.

3.2.2. Konzept 2 – 360°-Rundgang

Beim zweiten Konzept wird ebenfalls eine Website erstellt. Diese basiert jedoch auf 360°-Bildern, die anhand von WebGL im Browser dargestellt werden. Jedes der Bilder ist dabei ein Standpunkt in einer im Voraus modellierten Umgebung. Zwischen den Bildern kann durch das Klicken eines Buttons navigiert werden, wodurch eine Art des Fortbewegens durch die Umgebung ermöglicht wird. Informationen können in verschiedenen Formen integriert werden. Durch das Klicken von Informationsbuttons können damit Dialogfenster mit Text, Videos, Fotos oder weiterführende Links und Dokumente geöffnet werden. Ein Button wird dabei einer bestimmten Stelle eines 360°-Bildes zugewiesen. So kann beispielsweise neben einem dargestellten Sensor ein Button angebracht werden, welcher beim Klick ein Informationsfenster zu dem Sensor öffnet.

Zur Erstellung der 3D-Umgebung wird spezielle Software zur 3D-Modellierung eingesetzt. Beispielsweise kann für die Modellierung von Gebäuden bzw. Gebäudeteilen spezielle Architektur Software, wie zum Beispiel Autodesk Revit, genutzt werden. Weitere 3D-Modelle werden in einem 3D-Modellierungs- und Computergrafikprogramm wie Autodesk 3Ds Max erstellt

und mit den Gebäudemodellen zu einer gesamtheitlichen Umgebung zusammengeführt. Aus dieser lassen sich an beliebigen Stellen 360°-Bilder exportieren und in einem der zahlreich verfügbaren Online-Tools zu einem 360°-Rundgang zusammenfügen. Hier werden nun auch Informationen in verschiedenen Formen eingebunden und über Buttons an der beliebigen Stelle angebracht.

3.2.3. Konzept 3 – 3D-Spiel

Der dritte Ansatz für die digitale Darstellung von Smart Home Anwendungen basiert auf der Erstellung eines 3D-Spiels. Dazu werden, wie auch bei Konzept 2, zunächst eine 3D-Umgebung und die darin enthaltenen 3D-Modelle modelliert. Zusätzlich werden anhand von Programmcode und Skripten innerhalb einer Spiel-Engine Funktionalität und Spielablauf implementiert. Zugänglich wird das Konzept ebenfalls über die API WebGL und einen Internet-Browser. Das Spiel wird dazu in eine HTML-Website als Canvas-Element integriert.

Durch das im Browser stattfindende Rendering ergibt sich die Möglichkeit, sich anhand von Tastatureingaben durch die 3D-Umgebung zu bewegen und 3D-Objekte während der Laufzeit zu verändern. Somit soll eine Navigation durch die erstellte Umgebung implementiert werden. Informationen werden zum einen über Informationsbuttons in die Umgebung integriert, so dass der Benutzer bei Belieben Informationen öffnen kann. Außerdem werden Informationen automatisch eingeblendet, beispielsweise wenn der Benutzer im Spiel einen neuen Raum betritt. Auch sollen komplette Smart Home Szenarien implementiert und dargestellt werden.

Die zunächst stattfindende Modellierung der Umgebung findet auch hier in spezieller Modellierungssoftware statt. Die daraus resultierenden 3D-Modelle werden anschließend in eine Spiel-Engine importiert. Innerhalb dieser ist es anhand von Plug-Ins ebenfalls möglich, einfache 3D-Modelle zu erstellen. In der Laufzeit- und Entwicklungsumgebung für Spiele werden vor allem jedoch Funktionalität anhand von Skripten implementiert sowie Animationen erstellt.

3.3. Konzeptauswahl

Um die Wahl für eines der drei Konzepte treffen zu können wird geprüft, anhand welchem Konzept die Zielsetzung der Informationsplattform am besten erfüllt werden kann. Auch soll das Konzept bestmöglich zu der vorliegenden Zielgruppe passen.

Da im Mittelpunkt der Arbeit die Überlieferung von Informationen, sowie das Erleben von Smart Home steht, ist eine möglichst hohe Interaktionsmöglichkeit erforderlich. Diese ist vor allem bei der Umsetzung von Konzept 3 gegeben, da durch das Implementieren eigener Skripte gewünschte Interaktionen umgesetzt werden können.

Konzept 3 bietet zudem die Möglichkeit die Vorteile der Konzepte 1 und 2 zu integrieren und darüber hinaus ein individuelleres und umfangreicheres Produkt zu gestalten. So können in einem 3D-Spiel wie auch in 360°-Rundgängen feste Standpunkte in der Szene festgelegt werden, welche durch das Klicken eines Buttons erreicht werden. Dennoch kann zusätzlich eine Steuerung per Tastatur implementiert werden, wodurch ein freies Bewegen durch die Umgebung möglich wird. Auch Informationen können in Form von Buttons an gewünschten Stellen im Raum platziert werden. Jedoch kann die Gestaltung der Buttons, sowie der Inhalte und Aktionen beim Klick der Buttons selbst entworfen werden und ermöglicht dadurch das Erstellen eines individuellen Designs.

Um Abläufe und mögliche Szenarien eines vernetzten Zuhauses möglichst gut zu veranschaulichen sind Animationen und Zustandsänderungen von Gegenständen oder Lichtverhältnissen unumgänglich. Durch ein 3D-Spiel lassen sich diese im Gegensatz zu einer Webseite oder einem 3D-Rundgang zeitbasiert und teilweise zeitgleich in einen Gesamtablauf integrieren.

Aus diesen Gründen wird für das weitere Vorgehen Konzept 3 festgelegt.

3.4. Vorstellung Programmauswahl

Die Erstellung eines 3D-Spiels ist relativ umfassend und kann in die drei grundlegenden Teilschritte Modellierung, Animation und Programmierung unterteilt werden. Für jeden dieser Schritte sind verschiedene Programme notwendig, welche im Folgenden näher erläutert werden.

3.4.1. 3D-Modellierungssoftware

Autodesk Revit

Eine branchenspezifische Lösung für 3D-Modellierung ist Revit von Autodesk. Diese gilt als Standard für Architekten, Gebäudetechniker und Tragwerksplaner bei der Modellierung und Planung von Gebäuden (Oberste-Ufer 2020).

Interessant für diese Arbeit wird Revit, da für den Smart Home Simulator zunächst eine virtuelle Umgebung in Form eines Gebäudes, in das die Smart Home Anwendungen integriert werden sollen, erstellt wird.

Da Revit auf die Erstellung von Gebäuden ausgelegt ist, bietet dieses sich dazu an, das entsprechende Gebäude für den Smart Home Simulator zu erstellen.

Autodesk 3Ds Max

Eine professionelle 3D-Modellierungslösung, ebenfalls von Autodesk, ist 3Ds Max zur Modellierung, Animation, Simulation und Rendering in Bereichen wie Designvisualisierung, Architektur und Spielen. Anwendung findet 3Ds Max daher vor allem bei Film- und TV-Projekten, der Spieleentwicklung oder der Designvisualisierung (Autodesk, Inc. o.J.).

Damit bietet sich 3Ds Max für das Modellieren von sämtlichen 3D-Objekten an.

3.4.2. Spiel-Engine

Das bedeutendste Programm zur Erstellung von Spielen ist derzeit Unity der Unity Software Inc. Mit einem Marktanteil von über 50% ist die Software bereits ein Standard in der Spielentwicklung und kann durch ein großes Hilfsnetzwerk und Unity-spezifischen Foren glänzen (Maier 2021).

Anwendung findet die Unity Entwicklungsumgebung daher vor allem bei der Entwicklung von Spielen für verschiedene Plattformen wie PC-Betriebssysteme, Spielekonsolen, Mobile Betriebssysteme oder Webbrowser. Neben der Entwicklung von Spielen wird die Entwicklungsumgebung auch für die Erstellung von virtueller und erweiterter Realität bzw. audiovisuellen Simulationen der Realität oder experimentellen Medienanwendungen verwendet.

Außerdem bietet Unity auch Lösungen für Echtzeit-3D, die beispielsweise in der Visualisierung der Architektur- und Bau-Branche große Beliebtheit gewinnen, da sogenannte „digitale Zwillinge“ von Gebäuden erschaffen und mit Datenquellen wie Sensoren oder Metadaten kombiniert werden können. (Unity Technologies o.J.)

Besonders interessant für diese Arbeit ist die Möglichkeit der Entwicklung von Spielen und interaktiven Umgebungen für Webbrowser mit WebGL.

Dabei handelt es sich um eine auf OpenGL ES basierende JavaScript-Programmierschnittstelle, mit der 3D-Grafiken im Webbrowser ohne Erweiterungen dargestellt werden können (Parisi 2014).

Spiele und deren 3D-Szenen können damit direkt von Unity nach WebGL exportiert und in Browsern wiedergegeben werden.

3.4.3. IDE & Code-Editor

Um einer in Unity zusammengestellten Szene aus sogenannten GameObjects Funktionalitäten hinzufügen und auf Benutzereingaben reagieren zu können, werden Skripte in der Skriptsprache C# benötigt (Unity Technologies 2022). Grundsätzlich können diese mit jedem Texteditor erstellt werden. Für eine Erleichterung des Schreibens von Code durch eine übersichtlichere grafische Oberfläche, Syntaxhervorhebung, sprachspezifische Autovervollständigung und

Bug-Prüfung ist eine integrierte Entwicklungsumgebung notwendig (Red Hat, Inc. 2019). Für Unity und die Programmiersprache C# eignet sich hierzu Microsoft Visual Studio.

4. Umsetzung

In den folgenden Unterkapiteln wird die praktische Umsetzung der Arbeit mit den zuvor festgelegten Hilfsmitteln und Tools beschrieben.

4.1. Modellierung

Bei der Modellierung von 3D-Objekten wird in der Regel wie folgt vorgegangen.

Zunächst muss mit Hilfe eines Modellierungsprogramms wie beispielsweise Revit oder 3Ds Max ein 3D-Objekt erstellt werden. In einer typischen Modellierungssoftware kann dabei aus einigen Grundkörpern gewählt werden wie beispielsweise Würfeln (Cubes), Zylindern, Kugeln (Spheres) oder Torus etc.

Diese können anschließend anhand von verschiedenen Werkzeugen bearbeitet und verändert werden. Ist das 3D-Objekt mit der gewünschten Form erstellt, kann das Objekt in verschiedenen Dateitypen exportiert werden. Im Rahmen dieser Arbeit wird dazu das FBX-Dateiformat verwendet.

Da die Grundlage der zu erstellenden Informationsplattform der bereits vorgestellte SmaC darstellt, sollen auch in dem 3D-Modell die im Wohnwagen abgebildeten Räume zu finden sein. Es handelt sich dabei um die grundlegenden Räume Wohnbereich, Küche, Schlafzimmer und Badezimmer. Anhand dieser Räume lassen sich bereits alle gängigen Smart Home Szenarien abbilden und viele der wichtigsten Sensoren und Aktoren der Smart Home Technologie finden Anwendung. Auf weitere Räume oder Stockwerke wird bei der Raumplanung und der Festlegung der Größe des 3D-Modells verzichtet, um die Nutzung des Endproduktes in seiner Komplexität einzuschränken und zu verhindern, dass eingebundene Informationen durch verfrühte Abbrüche oder Nichtauffinden nicht vom Nutzer wahrgenommen werden können. Auch weil mit dem Smart Home Simulator die Möglichkeit der Implementierung in jede Art und Größe von Wohnungen vermittelt werden soll, wird eine Räumlichkeit mit lediglich den grundlegenden Zimmern erstellt.

Um Smart Home Komponenten auch für den Außenbereich einbinden zu können, wird die Wohnung als selbststehendes Einfamilienhaus entworfen.

Aus diesen Vorgaben entsteht der in Abbildung 9 gezeigte Grundriss der Wohnung.

Für den späteren Gebrauch in Unity werden nur die Wände, Decken und Böden der Wohnung sowie deren Aussparungen für Fenster und Türen benötigt. Somit werden in Revit ausschließlich diese durch vorgefertigte Modelle in der Größe angepasst und eingezeichnet, um ein fertiges 3D-Modell zu erhalten (siehe Abbildung 10).

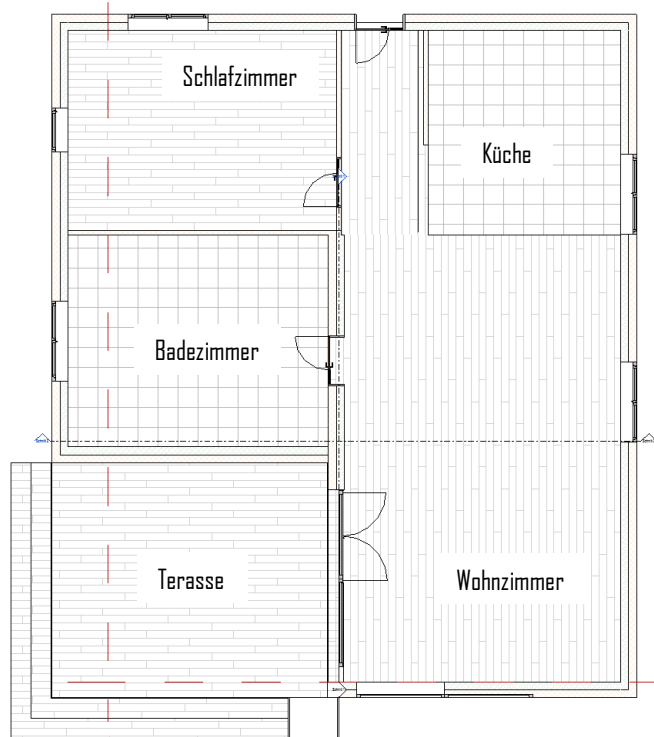


Abbildung 9: Grundriss aus Autodesk Revit

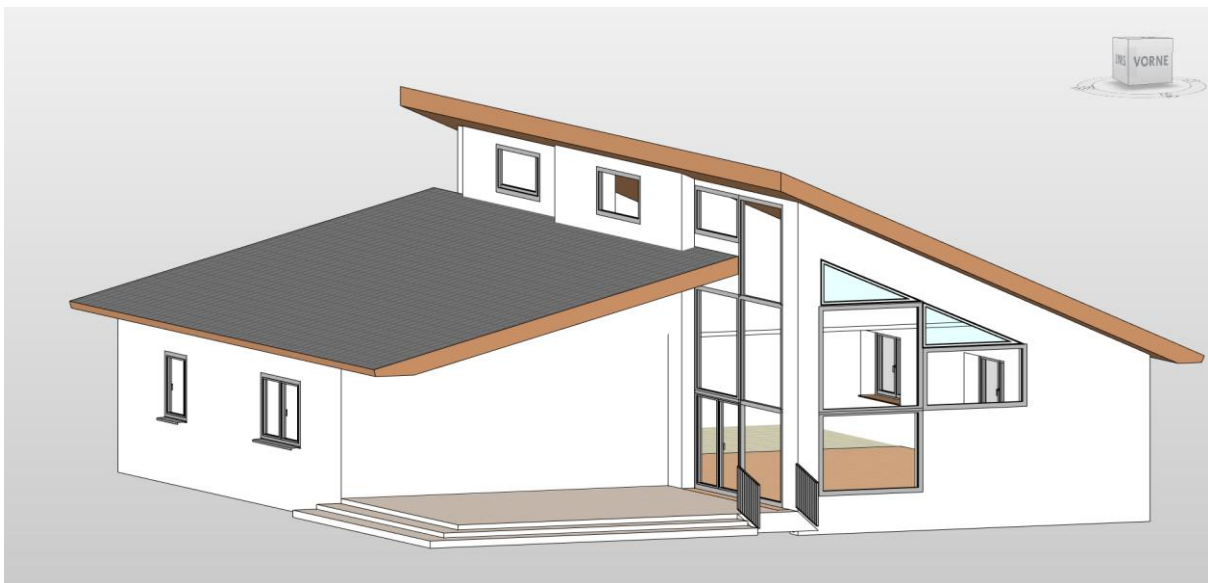


Abbildung 10: 3D-Modell aus Revit

Das in Revit erstellte 3D-Modell des Hauses wird dann als FBX-Datei exportiert und in ein neues Unity-Projekt geladen.

Für die einfache Modellierung von Objekten mit geringer Polygonanzahl wird ProBuilder, ein im Unity Asset Store verfügbares Modellierungs-Tool verwendet.

Den verschiedenen Komponenten des FBX-Modells des Hauses werden nach dem Import in Unity verschiedene Texturen zugewiesen.

Da Komponenten wie Fenster und Türen, die bereits in dem aus Revit exportierten Modell enthalten sind, im Endprodukt bewegbar sein müssen, müssen diese erneut modelliert werden. Diese sind als komplette Komponenten im Modell integriert, in welchen beispielsweise der Rahmen nicht vom Fenster selbst getrennt wird, sodass ein Öffnen des Fensters durch eine Animation nicht möglich ist. Dazu werden getrennt voneinander erst die Rahmen (Abbildung 11) und anschließend die Fenster (Abbildung 12) in Unity modelliert.

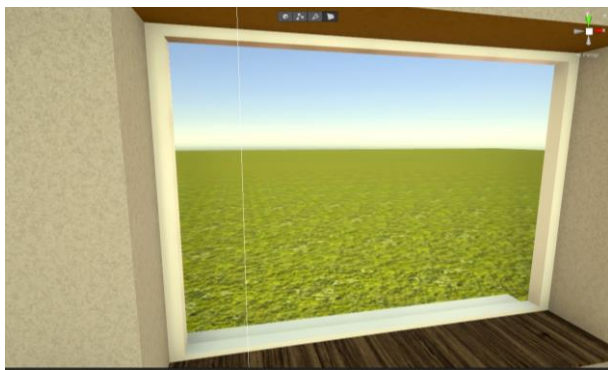


Abbildung 11: Modellierter Fensterrahmen

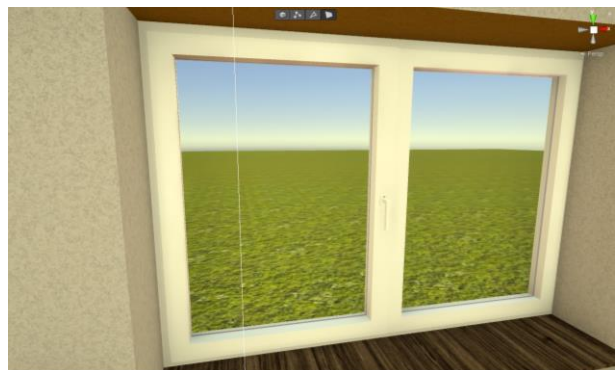


Abbildung 12: Fertig modelliertes Fenster

Um ein möglichst realgetreues Endergebnis zu erhalten, werden anschließend zahlreiche 3D-Modelle für die Innenausstattung des Gebäudes modelliert und platziert. Aufgrund des Ziels mit der Informationsplattform ein reales Smart Home zu simulieren und erlebbar zu machen, wird darauf geachtet, sehr detailreiche 3D-Objekte zu nutzen (siehe Abbildung 13 und Abbildung 14). Dazu werden, neben selbst modellierten Objekten auch kostenfreie Modelle Dritter verwendet.

Dabei wird zunächst auf 3D-Modelle mit hoher Polygonanzahl zurückgegriffen, um ein möglichst realitätsnahes Ergebnis zu erzielen. Auch bei selbst modellierten Objekten werden Flächen mit Rundungen durch eine höhere Anzahl von Polygonen erstellt, um diese möglichst glatt und detailliert wirken zu lassen.



Abbildung 13: Modell-Zwischenstand, High-Poly-Objekte, Badezimmer



Abbildung 14: Modell-Zwischenstand, High-Poly-Objekte, Wohnraum

Nachdem das Haus mit den wichtigsten Modellen ausgestattet ist, wird ein erster Test der Anwendung durchgeführt. Dazu wird eine Kamera in der Szene platziert. Zu dieser wird ein Skript hinzugefügt, das die Bewegung der Kamera anhand der Pfeiltasten der Tastatur ermöglicht. Dadurch ist ein Testen der Performance möglich.

Die in Unity erstellte Szene wird für den Test mit den enthaltenen Objekten als eine WebGL-Anwendung exportiert. Dabei wird eine HTML-Webseite mit der eingebundenen Anwendung erstellt. Um diese öffnen zu können, wird ein Server eingerichtet, auf dem die HTML-Seite und die notwendigen Daten der Anwendung gespeichert werden. Um die Webseite über den Browser öffnen zu können, muss nun die Adresse des Servers im Browser geöffnet werden.

Bei der Testung fällt auf, dass die benötigte Grafikleistung zum Rendern der Anwendung zu hoch ist für die Nutzung auf einem durchschnittlichen Computer mit einem Grafikspeicher von 6 Gigabyte. Das Steuern der Kamera durch die Szene ist aufgrund der hohen Anforderungen an die Grafikeinheit des Computers nicht flüssig möglich. Es fällt auf, dass wenn die Kamera in Richtung eines Raums mit vielen und besonders hochauflösenden Objekten gerichtet wird, die Render-Anforderungen die Leistung der Grafikkarte übersteigen und die Anzahl der Bilder pro Sekunde sehr gering wird.

Um die Performanz der Anwendung steigern zu können, werden folglich anspruchsvolle Modelle mit hoher Polygonanzahl ersetzt und selbst modelliert.



Abbildung 15: Austausch high-poly durch low-poly 3D-Modell einer Steckdose

Besonders Objekte, die in der späteren Anwendung keine tragende Rolle zugeschrieben bekommen, werden durch möglichst einfache Modelle ersetzt. Um den Verlust der Details möglichst gering zu halten, wird das im Abschnitt 2.2.4 beschriebene Texture-Mapping verwendet. Wie in Abbildung 15 zu sehen, können mit Hilfe des Texture-Mappings anspruchsvolle 3D-Modelle mit hoher Polygonanzahl durch einfache Würfel ersetzt werden. Auf diese wird eine Textur einer Steckdose gelegt.

Einige der in der Szene vorhandenen 3D-Modelle werden durch einfachere Modelle, die aus nur wenigen Flächen und Kanten bestehen, ersetzt. Andere Modelle, die lediglich der Dekoration der Umgebung dienen werden entfernt.

Viele der Modelle werden dabei in Unity durch das Pro-Builder Plug-In erstellt, wie die in Abbildung 16 dargestellte Dusche.

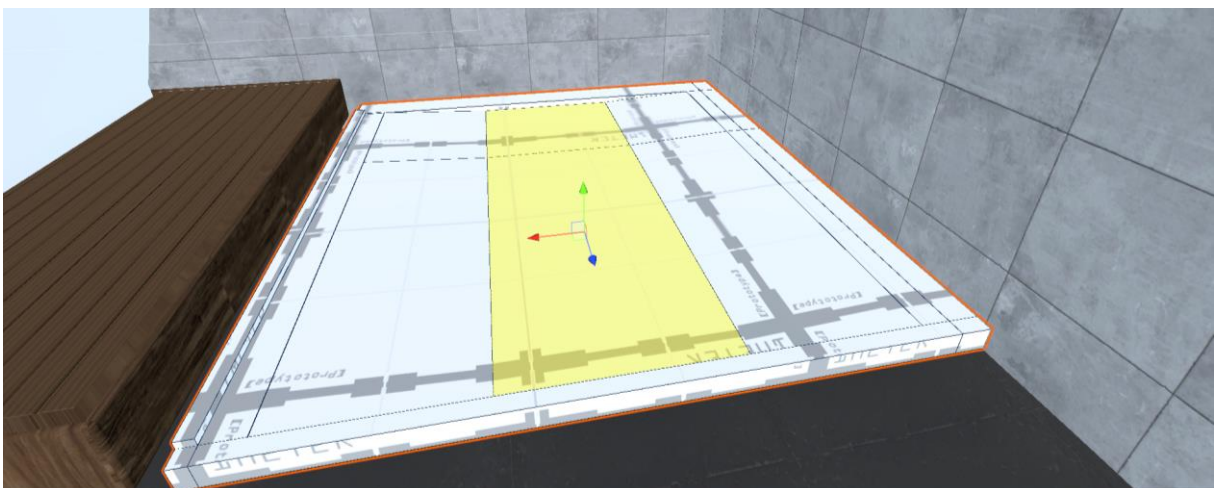


Abbildung 16: Modellierung Dusche

4.2. Belichtung

Ein wesentlicher Bestandteil eines intelligenten, vernetzten Zuhauses ist auch die automatisierte und zentrale Steuerung des Lichts. Auch dies lässt sich in Unity darstellen, wobei folgende Lichtquellen unterschieden werden können:

- **Point Light:** Befindet sich an einem Punkt in der Szene und emittiert Licht gleichmäßig in alle Richtungen.
- **Spot Light:** Befindet sich an einem Punkt in der Szene und emittiert Licht in Kegelform.
- **Directional Light:** Befindet sich unendlich weit entfernt und emittiert Licht in ausschließlich eine Richtung (als Sonnenlicht verwendet).
- **Area Light:** Ist als Rechteck mit bestimmter Größe in der Szene definiert, das Licht in alle Richtungen gleichmäßig über seine Oberfläche abstrahlt, jedoch nur von einer Seite des Rechtecks.
- **Emissive Material:** Ein Material, welches Objekten zugewiesen werden kann. Diese emittieren dann Licht über ihre Oberfläche.

Damit die in der Szene platzierten Lichtquellen der verschiedenen Typen auch einen Effekt auf die umliegenden Objekte und die Szene haben, muss ein Modus ausgewählt werden. Dabei wird wie folgt unterschieden:

- **Baked:** Für eine Lichtquelle im Baked-Modus berechnet Unity vor der Laufzeit die Beleuchtung und speichert die Lichtdaten in sogenannten Lightmaps. Während der Laufzeit werden ausschließlich die Lichtdaten gelesen und keine Berechnungen mehr durchgeführt, was zu einer Reduzierung des Aufwands während der Laufzeit führt.
- **Realtime:** Belichtung wird während der Laufzeit, jeden Frame berechnet. Es werden keine Berechnungen vor der Laufzeit durchgeführt. Eine Lichtquelle in Realtime kann jedoch während der Laufzeit geändert werden (beispielsweise in Farbe oder Intensität).
- **Mixed:** Elemente von Baked- und Realtime-Lights werden kombiniert. Somit werden Teile in der Laufzeit berechnet und können verändert werden, während andere Informationen bereits im Voraus berechnet werden können.

Bei der vorliegenden Anwendung ist eine gute Performance besonders wichtig, da die Anwendung auf der einen Seite über einen Browser wiedergegeben, das heißt in eine Website eingebunden und auf der anderen Seite von einer möglichst breiten Zielgruppe verwendet werden soll. Dabei ist es wichtig darauf zu achten, dass die Systemanforderungen nicht zu hoch sind, sodass die Anwendung mit einem durchschnittlichen Computer ausgeführt werden kann.

Einen wesentlichen Einfluss auf die Performance hat dabei die Belichtung und der Modus der Lichtberechnung.

Damit sind Baked Lights die beste Wahl um eine hohe Performance und trotzdem eine hohe Belichtungsqualität zu erreichen. Änderungen an den Lichtern während der Laufzeit sind nicht mehr möglich. Da dies aber eine gewollte Eigenschaft der Lichter dieser Arbeit darstellt, sind Baked Lights keine Option und die Wahl fällt auf eine Realtime Belichtung.

Für die Realtime Belichtung wird jedoch, aus Performance Gründen, das sogenannte Global Illumination deaktiviert. Bei diesem wird indirektes Licht in der Szene simuliert. Wird dies während der Laufzeit berechnet, fällt ein sehr hoher Berechnungsaufwand an, welcher die Performance stark beeinflusst.

Den verschiedenen Räumen des Hauses werden Spot Lights zur direkten Beleuchtung hinzugefügt. Die Position ist dabei durch die bereits modellierten Lampen in der Decke vorgegeben. Auch im Außenbereich des Hauses werden Spotlights zur Beleuchtung von Hof und Terrasse eingesetzt. Da die Spot Lights jedoch lediglich die sich im Radius befindenden Objekte mit Licht beeinflussen, jedoch keine Emissionsfläche besteht, wird diese anhand von Emissive Material erstellt. Dazu wird für jeden Raum, das heißt für jede Spot Light-Gruppe ein Material angelegt, welches den verschiedenen Lampen zugeordnet wird. Wird während der Laufzeit das Licht eines Zimmers aktiviert, wird automatisch auch die Emission des Materials aktiviert, wodurch eine leuchtende Fläche in den Lampen entsteht.

Um den Effekt des Leuchtens zu erhöhen, wird auf die Kamera ein sogenannter Bloom-Effekt, ein Post-Processing-Effekt für das Kamerabild, hinzugefügt. Bei diesem werden Objekte und Lichtquellen ab einer bestimmten Intensität mit einem hellen Schein umrandet, sodass ein leuchtender Effekt entsteht.

Um dem Qualitätsverlust des fehlenden indirekten Lichtes etwas entgegen zu wirken, wird der Kamera der Ambient Occlusion Post-Processing-Effekt, hinzugefügt. Dieser simuliert die leichten Schatten welche in Falten, Löchern und Oberflächen auftreten, welche nahe zueinander sind. In Abbildung 17 und Abbildung 18 ist der Unterschied zwischen der Umgebung ohne und mit Ambient Occlusion dargestellt. Besonders in den Ecken des Raumes sind die Schatten zu erkennen.



Abbildung 17: Ambient Occlusion Effect deaktiviert



Abbildung 18: Ambient Occlusion Effect aktiviert

Ein weiteres Detail, welches der Szene zusätzliche Realität verschafft, ist das Integrieren von Reflektionen. Dabei wird Objekten mit reflektierendem Material die Reflektion der Umgebung zugewiesen. Somit können Spiegel erstellt oder das Erscheinen von Glas realistischer gestaltet werden.

Um in Unity Reflektionen anzeigen zu können, müssen zunächst Reflection Probes der Szene hinzugefügt werden. Diese gelten für einen bestimmten, selbst einzustellenden Bereich in der Szene und beeinflussen alle sich darin befindlichen Objekte. Auch Reflection Probes können vom Typ Realtime oder Baked sein. Wie bei der Belichtung wird auch hier im Realtime Modus

die Reflektion jeden Frame neu berechnet und erstellt, während im Baked Modus die Berechnung vor der Laufzeit stattfindet und die Reflektion in sogenannten Cubemaps gespeichert wird. Um auch hier den Einfluss auf die Performance möglichst gering zu halten, werden ausschließlich Baked Reflection Probes angelegt, da diese via Script während der Laufzeit auch manuell aktualisiert, das heißt neu gerendert werden können. Reflection Probes werden für jeden Raum angelegt, um eine hohe Präzision in den Reflektionen zu erreichen.

Weitere Lichter, welche als indirekte Zierbeleuchtung sowie Signalbeleuchtung des Smart Home fungieren und sich im Schlafzimmer, dem Badezimmer und der Küche befinden, werden ebenfalls durch Emissive Materials dargestellt.

Da ein Emissive Material jedoch lediglich durch Lightmaps, das heißt nach dem Bake als Baked-Light einen Einfluss auf andere Objekte haben kann, wird zusätzlich zu den angelegten Lichtquellen mit Material noch ein Point Light in die Szene eingebunden. Damit entsteht wieder ein Paar aus einer sichtbaren Lichtquelle (Objekt mit Emissive Material) und einem Licht (Point Light).

Da viele zentrale Aufgaben und Szenarien eines Smart Home die (automatische) Steuerung und Änderung der Lichtverhältnisse im und um das Haus beinhalten, wird auch im vorliegenden Projekt die Lichtsteuerung mit aufgegriffen. Dies bedeutet, dass Eigenschaften wie Intensität und Farbe der bisher vorgestellten Arten von Lichtern während der Laufzeit verändert werden. Realisiert wird dies über implementierte Skripte, welche im folgenden Kapitel näher beschrieben werden.

4.3. Design

Da für die Interaktion mit dem Simulator ein umfangreiches User Interface, bestehend aus Dialog-Fenstern, Informationsfenstern und Buttons angelegt werden muss, wird dafür ein einheitliches Farbschema sowie Designs und ein Logo festgelegt.

4.3.1. Farbpalette

Um optisch eine Einheitlichkeit über die gesamte Arbeit hinweg erzeugen zu können, wird eine Farbpalette angelegt. Zukünftige User Interface-Elemente und Designs sollen dadurch in ihrer Farbgebung eingeschränkt werden, wodurch auch der Wiedererkennungseffekt gesteigert werden soll. Die Farben der Farbpalette basieren dabei auf dem Corporate Design der Hahn-Schickard-Gesellschaft für angewandte Forschung. Diese gibt die Farben *Sahara*, *Marsrot* und *Aschblau* vor (siehe Abbildung 19). Um einen Spielraum zu ermöglichen und Akzente setzen zu können wird für jede der drei Farben eine Variation erstellt. Außerdem soll der Farbumfang auch einen Weiß- und einen Schwarzton vorgeben. Die Farben werden auf der Palette zusätzlich zur visuellen Darstellung in ihrem RGB-, CMYK- und Hexadezimalwert beschrieben.

Sahara RGB 237 . 224 . 182 CMYK 00 . 05 . 30 . 10 Hex #ede0b6	Goldenrot RGB 189 . 142 . 0 CMYK 25 . 42 . 100 . 4 Hex #BD8E00	Weiß RGB 242 . 242 . 242 CMYK 3 . 3 . 3 . 0 Hex #F2F2F2
Marsrot RGB 159 . 41 . 20 CMYK 00 . 90 . 100 . 40 Hex #9f2914		Weinrot RGB 79 . 10 . 0 CMYK 41 . 89 . 86 . 65 Hex #4f0a00
Aschblau RGB 115 . 138 . 156 CMYK 35 . 10 . 00 . 45 Hex #738a9c	Kadettenblau RGB 48 . 67 . 85 CMYK 82 . 67 . 47 . 35 Hex #304355	Schwarz RGB 6 . 0 . 0 CMYK 70 . 70 . 70 . 90 Hex #060000

Abbildung 19: Farbpalette Smart Home Simulator

4.3.2. Logo

Für ein professionelles Auftreten nach außen sowie zur Erleichterung der Plakatierung und Werbung der digitalen Informationsplattform wird ein Logo entworfen und dem Projekt ein Name gegeben. Das Logo soll dabei möglichst einprägsam und einfach gestaltet werden. Außerdem soll mit diesem die Thematik Smart Home dargestellt werden, um bereits beim ersten Kontakt den Hintergrund des Projekts zu transportieren.

Zur Erstellung des Logos wird Affinity Photo genutzt. Dabei werden zunächst einfache Rechtecke genutzt, um ein Haus in der Seitenansicht zu zeichnen. Da bereits das Modell des Hauses für die künftige Nutzung vorliegt, kann zur Individualisierung des Logos der Querschnitt genau

dieses Hauses genutzt werden. Eine Besonderheit ist dabei die Anordnung der zwei Dachhälften, welche durch unterschiedliche Höhen und Neigungen versetzt sind. Die Smart Home Thematik wird, wie in Abbildung 20 dargestellt, innerhalb des Logos durch das gängige Zeichen für Funkverkehr mit eingebracht.

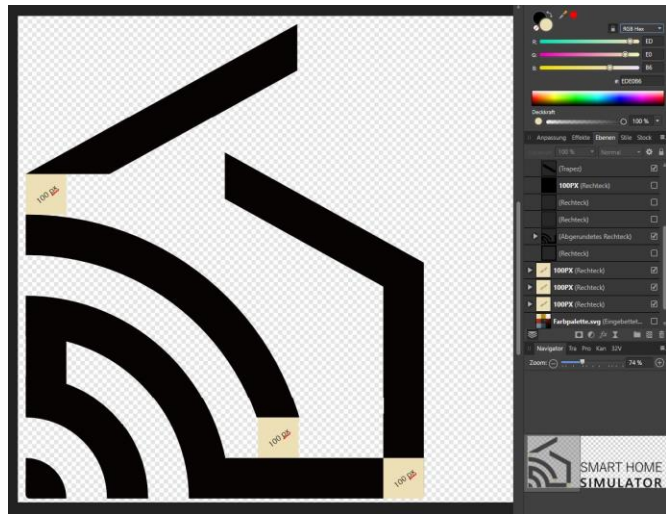


Abbildung 20: Erstellung Logo

Um den Nutzen der Anwendung kurz und präzise signalisieren zu können, wird die Bezeichnung Smart Home Simulator etabliert und als Schriftzug in das Logo integriert.

4.3.3. UI-Elemente

Für die Interaktion im Spiel werden UI-Elemente wie Buttons und Dialogfenster in Unity benötigt. Diese werden in Affinity Photo erstellt und als Bilder (PNG- oder JPEG-Dateien) in Unity eingebunden. Die User Interface-Elemente in Unity besitzen standardgemäß eine Image-Komponente, in welche ein Bild eingefügt werden kann.

Info-Buttons

Für das Bild der Info-Buttons werden zwei nicht ausgefüllte Ellipsen ineinander angelegt, wobei die äußere eine schmalere Kontur als die innere besitzt (siehe Abbildung 21). Gefüllt wird der daraus entstehende Kreis mit einer passend runden Fläche, welche eine Deckkraft von nur 50% besitzt, sodass der Button in der Szene hinter ihm liegende Objekte nicht vollständig bedeckt.



Abbildung 21: Erstellung Info-Button

Außerdem wird das weit verbreitete "i", welches für eine Informationsauskunft steht in der Mitte des Kreises angebracht. Nach demselben Prinzip werden die Navigationsbuttons in Affinity erstellt.

Info-Fenster

Um die wesentlichen Informationen der Anwendung zu integrieren, sollen Informationsfenster erstellt werden, welche beim Klick auf die Informationsbuttons geöffnet werden und die Funktionsweise der verschiedenen Smart Home Komponenten erläutern. Wie in Abbildung 22 dargestellt, setzen sich die Info-Fenster aus vier Teilen zusammen. In die obere linke Ecke wird ein Button integriert, welcher Aufschluss darüber gibt, zu welchem Typ die im Fenster erläuterte Smart Home Komponente gezählt werden kann. Im oberen mittleren Feld wird die Bezeichnung der Komponente aufgeführt. Dieses Feld dient daher als Überschrift. Das rote Feld in der oberen rechten Ecke dient zum Schließen des Fensters. Im darunterliegenden, größten Teil des Fensters werden die hauptsächlichen Informationen integriert.

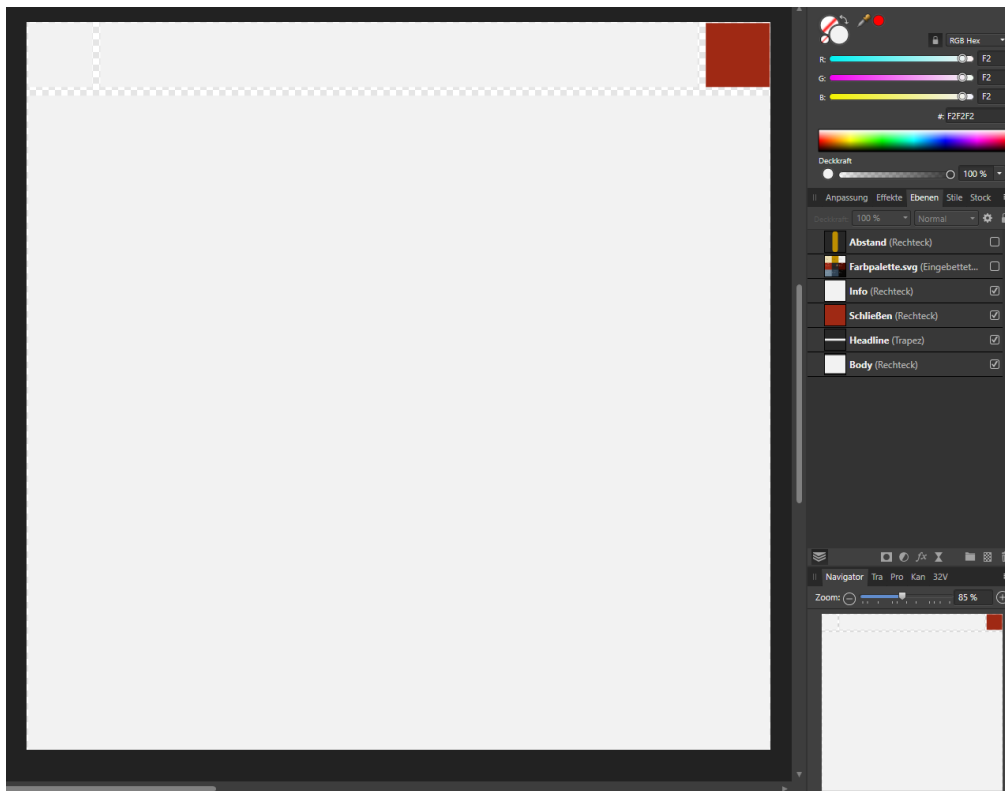


Abbildung 22: Info-Fenster

Hauptmenü

Damit im Spiel eine Übersicht über die Szenarien und die Steuerung gegeben werden kann sowie eigene Einstellungen getroffen werden können, wird ein Hauptmenü erstellt. Dieses setzt sich aus mehreren Teilen zusammen, wie in Abbildung 23 zu sehen. Im oberen Bereich befinden sich von links vier Flächen, in welche in Unity die Möglichkeit zu benutzerdefinierten Einstellung des Spiels implementiert werden soll. Beispielsweise das Aus- und Anschalten des Tons oder das Anzeigen bzw. Ausblenden der Navigationsbuttons. In der Mitte ist Platz für die Beschriftung „Menü“. Die rote Fläche dient zum Verlassen des Menüs.

Im mittleren Teil des Menüs sollen die Informationen eingebunden werden. Beispielsweise eine Übersicht der möglichen Interaktionen, sowie die Information darüber, welche davon bereits ausgeführt wurden. Auch soll hier eine Erklärung der Steuerung integriert werden. Die Flächen in der unteren linken sowie rechten Ecke des Menü-Fensters werden in Unity als Buttons für Neustart und das Beenden des Spiels deklariert.

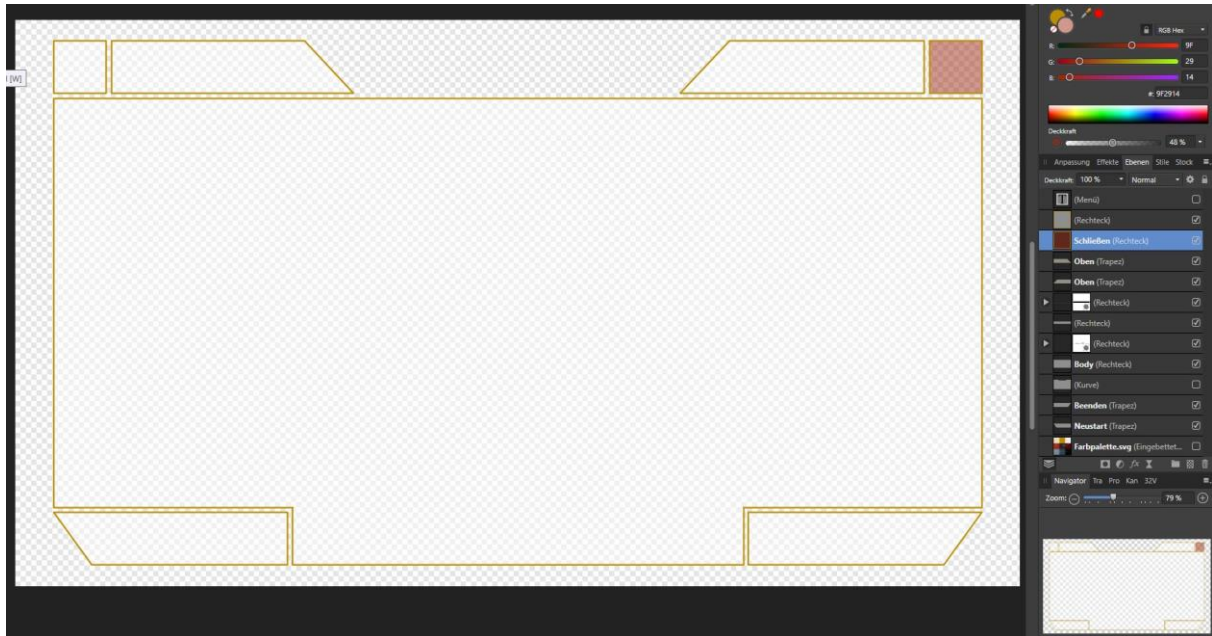


Abbildung 23: Erstellung Hauptmenü

Aufgaben-Übersicht

Um eine zusätzliche Übersicht über die Aufgaben bzw. Interaktionsmöglichkeiten des Simulators zu bieten, soll ein Übersichts-Fenster erstellt werden, das jederzeit eingesehen werden kann und in die normale Spielsicht eingebunden wird. Um eine nicht zu große Fläche des Bildschirms zu verdecken, soll standardmäßig nur ein Button zu sehen sein. Beim Klick auf diesen wird die gesamte Übersicht angezeigt. Somit besteht das Übersichtsfenster aus zwei Teilen. Zum einen der Button, auf der anderen Seite das Hauptelement, in welchem die Aufgaben aufgeführt werden. Da die Übersicht am oberen Bildschirmrand angebracht werden soll, befindet sich der Button unter dem Hauptelement. Dadurch kann der Button ständig angezeigt und das Element beim Klick lediglich in den Bildschirm, von oben nach unten bewegt werden. Damit die Sicht und damit darunterliegende Elemente nicht verdeckt werden, wird das Hauptelement halbtransparent angelegt. Der Button wird mit einem Verlauf von halbtransparent zu volltransparent gestaltet, um die Sichtbarkeit zu erhöhen. In Abbildung 24 ist die gesamte Aufgaben-Übersicht dargestellt.

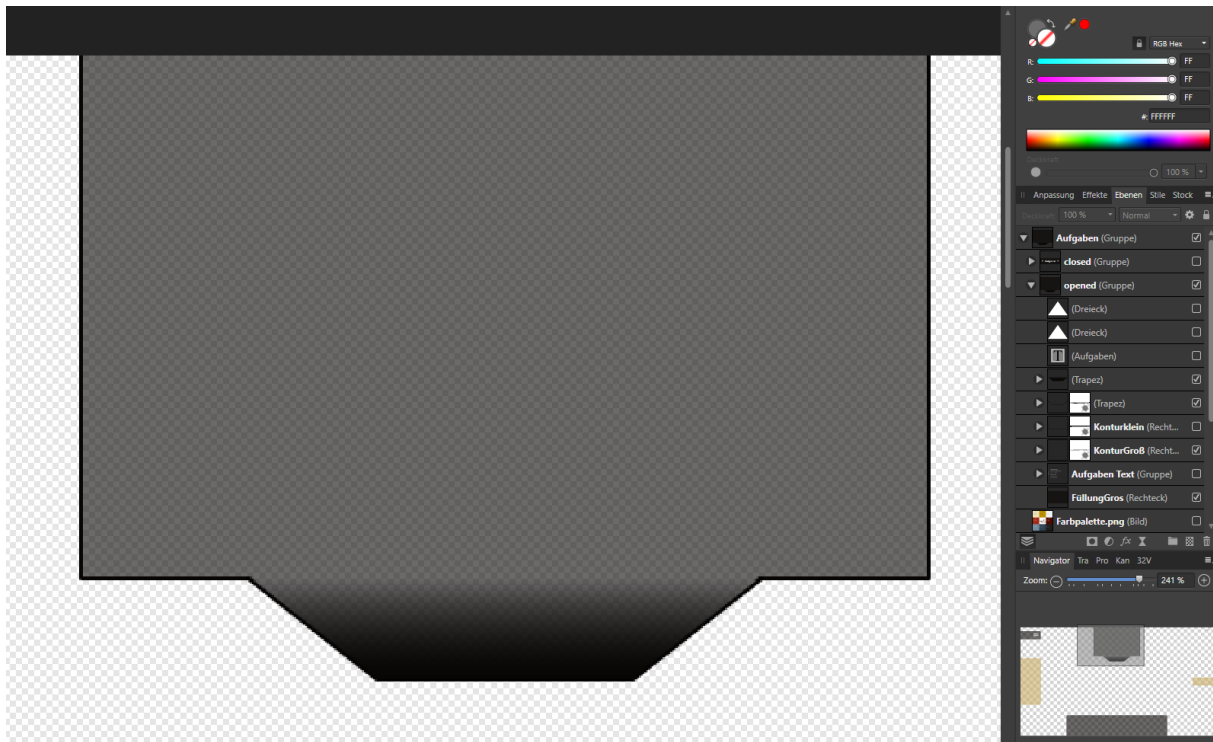


Abbildung 24: Erstellung Aufgaben-Übersicht

Um eine vollständige Bedienung des Simulators via Maus zu realisieren, werden Buttons erstellt, welche beim Herantreten an eine Türe angezeigt werden. Beim Klick soll die Türe geöffnet oder geschlossen werden. Um die Interaktionsmöglichkeit zu kennzeichnen, sowie das Erscheinungsbild des Buttons auffälliger zu gestalten, wird der *Goldenrot* Farbton der Farbpalette genutzt. Wie in Abbildung 24 dargestellt, wird auch hier eine Transparenz von 50% angewandt, um ein vollständiges Verdecken von dahinterliegenden Objekten zu verhindern.

4.4. Programmierung

Im Folgenden werden die Skripte, welche Funktionalität und Logik der Anwendung beschreiben, aufgeführt und erläutert.

4.4.1. Navigation

Bewegung

Das erste Skript, das der Szene hinzugefügt wird, soll die Steuerung und Navigation durch die Szene, anhand eines Spielerobjektes, ermöglichen. Dazu wird im ersten Schritt der Szene ein neues GameObject hinzugefügt. Diesem werden verschiedene Komponenten angehängt. Um später Kollisionen zu ermöglichen, wird dem Objekt eine Collider-Komponente hinzugefügt. Dieser Collider soll dabei die Form einer Kapsel (Capsule-Collider) haben. Zusätzlich bekommt das GameObject eine Character Controller Komponente zugewiesen. Anhand dieser Komponente kann beispielsweise der sogenannte Step Offset eingestellt werden, der die Höhe eines

Objektes vom Boden aus beschreibt, die das Spielerobjekt überschreiten kann. Außerdem wird dem Objekt die Hauptkamera der Szene untergeordnet. Dieses Vorgehen hat zur Folge, dass sich die Kamera automatisch mit dem Spielerobjekt bewegt, wenn dieses die Richtung in der Szene ändert.

Im erstellten Skript wird zunächst eine Variable vom Typ `CharacterController` erstellt. Dieser wird in der `Start()`-Funktion die `Character Controller` Komponente des `Playerobjekts` zugewiesen. Die Bewegung des Objektes wird anhand einer Funktion (`UpdateMovement()`) ermöglicht, die in der `Update()`-Funktion jeden Frame des Spiels aufgerufen wird.

In dieser werden zunächst die Tastatur-Inputs der Pfeiltasten bzw. W, A, S, D-Tasten als Horizontale (Pfeil links & Pfeil rechts, oder A & D) und Vertikale (Pfeil oben & Pfeil unten, oder W & S) abgefragt und als float in jeweils einer Variablen gespeichert. Der Rückgabewert ist dabei entweder 0 (sollte keine Taste gedrückt werden), 1 (sollte Pfeil oben bzw. Pfeil rechts gedrückt werden) oder -1 (sollte Pfeil unten bzw. Pfeil links gedrückt werden).

Die Bewegung erfolgt schließlich abhängig von der Richtung, in welche die Kamera zeigt (Rotation, siehe nächster Abschnitt) und von dem Wert der horizontalen und vertikalen Bewegungsvariable.

	0	1	-1
horizontal	keine	rechts	links
vertikal	keine	vor	zurück

Abbildung 25: Bewegungsrichtung in Abhängigkeit des Inputs

Um eine Alternative zur Navigation mit der Tastatur zu schaffen, werden außerdem Buttons in der Szene platziert. Diese verursachen beim Klick mit der linken Maustaste die automatische Navigation des Spielerobjekts zur entsprechenden Stelle des Buttons im Raum.

Rotation

Um dem Nutzer zusätzlich zur Navigation durch die Szene die Möglichkeit bieten zu können, die Sicht zu rotieren bzw. sich umzuschauen, wird ein weiteres Skript erstellt. Das Ziel hierbei ist, anhand der Computermausbewegung das Sichtfeld des Spielers zu rotieren.

Beeinflusst wird anhand der Computermaus-Eingabewerte nicht das Spielerobjekt, sondern die untergeordnete Hauptkamera. Grund dafür ist die vertikale Rotation, bei der das Spielerobjekt mit seiner länglichen Kapsel Form umfallen würde. Somit soll ausschließlich die Kamera rotiert und die Navigation an der Kamerarichtung in der x-z-Ebene ausgerichtet werden.

Da beabsichtigt wird, anhand von klickbaren Flächen Informationsfenster öffnen zu können, ist der Computermauszeiger für die Spielfunktion und -logik entscheidend und kann in Folge des Auslesens der Computermausbewegungen nicht verborgen werden. Aus diesem Grund wird die Rotation durch das Gedrückt-Halten der Maustaste und dem gleichzeitigen Bewegen ermöglicht.

Um zu ermöglichen, dass beim Klicken von klickbaren Flächen nicht rotiert, sondern die gewünschte Funktion ausgelöst wird, wird eine statische Variable des Typs Bool erstellt, die beim Überfahren einer klickbaren Fläche mit dem Mauszeiger auf *false* gesetzt wird. Der Funktion der Rotation wird eine Abfrage vorangestellt, welche den aktuellen Wert der erstellten Variablen überprüft. Nur, wenn die Variable den Wert *true* besitzt, wird der Input der Computermaus abgefragt und die Rotation ermöglicht.

4.4.2. Funktionalität

Cursor-Controller

Um dem Benutzer ein besseres Gefühl dafür geben zu können, welche Elemente die Möglichkeit zum Anklicken besitzen, soll der Cursor beim Überfahren von klickbaren Objekten geändert werden. Auch während der Rotation soll der Cursor geändert werden.

Dazu werden zunächst verschiedene Cursorbilder in das Unity Projekt geladen und in einem Skript (*CursorController()*) jeweils eine Funktion für jedes Bild erstellt.

In einem weiteren, für Buttons erstellten Skript wird aus der *onPointerEnter()*-Funktion die Funktion im *CursorController()*-Skript aufgerufen, welche das Bild des Cursors zu dem für klickbare Objekte ändert. Beim Verlassen der klickbaren Zone mit dem Curser wird aus der *onPointerExit()*-Funktion wieder der Standard-Cursur gesetzt.

Fenster- und Jalousien-Controller

Weitere zentrale Skripte, auf welche von außen zugegriffen werden kann, sind das Skript zur Steuerung der Fenster sowie das Skript zur Steuerung der Jalousien. Die Skripte sollen genutzt werden, um auf die Animationen der verschiedenen Fenster bzw. der verschiedenen Jalousien zugreifen zu können. Dazu wird im *FensterController()*-Skript für jedes Fenster eine Variable des Typs Animator angelegt, welcher die entsprechende Animator-Komponente des jeweiligen Fensters zugeordnet wird. In jedem Animator befinden sich eine Automation zum Öffnen, sowie eine Automation zum Schließen des jeweiligen Fensters. Durch angelegte Trigger, können über das Skript die Trigger aktiviert und somit die Fenster geöffnet oder geschlossen werden.

Informations-Buttons

Da in der Szene einige Informations-Buttons zu den verschiedenen Smart Home Komponenten in den verschiedenen Räumen platziert werden sollen, wird auch hierfür ein neues Skript erstellt. Dabei soll die Platzierung der Buttons in der dreidimensionalen Umgebung erfolgen, sowie das Anzeigen unter gewissen Voraussetzungen.

Um sogenannte User Interface (UI-) Elemente, wie Buttons, Bilder oder Textfelder, in eine Umgebung zu integrieren sind in Unity zwei Wege möglich. Die Elemente können in einem sogenannten Overlay-Canvas auf dem Bildschirm angezeigt werden, dabei wird das entsprechende Element auf einer zweidimensionalen Fläche auf dem Bildschirm, vor der 3D-Umgebung angezeigt. Eine weitere Möglichkeit ist das Anzeigen im dreidimensionalen Raum durch ein sogenanntes World-Space-Canvas. Dabei bekommt das UI-Element eine Position im Raum zugewiesen.

Für die Informations-Buttons wird die Möglichkeit des Overlay-Canvas gewählt, damit diese der Kamera immer zugewandt sind. Werden die Buttons in der Umgebung platziert und von der Seite betrachtet, können diese nicht gesehen werden, da ein Button lediglich aus einem zweidimensionalen Bild besteht. Dennoch sollen die Buttons an der richtigen Stelle im Raum erscheinen und sozusagen an die Objekte gepinnt werden, welche beschrieben werden sollen. Dazu wird jedem Button das erstellte Skript angehängt. In diesem wird eine Variable des Typs Transform angelegt, welchem jeweils das Zielobjekt zugewiesen wird. Die Position dieses Objekts wird mittels der Funktion *WorldToScreenPoint()* von einer dreidimensionalen World-Space-Position zu einer zweidimensionalen Screen-Space-Position transformiert. Diese kann der Position des Buttons, welcher ebenfalls im Screen-Space liegt, zugewiesen werden.

Angezeigt werden sollen die Buttons jedoch ausschließlich, wenn

- das Objekt, dem der Button zugeordnet ist, im Sichtfeld der Kamera ist,
- der Abstand zwischen dem Spielerobjekt und dem Objekt, dem der Button zugeordnet ist eine gewisse Größe unterschreitet,
- sich das Spielerobjekt im selben Raum befindet wie das Objekt, dem der Button zugeordnet ist und
- eine allgemein angelegte Variable des Typs Bool, mit welcher die Buttons von verschiedenen Skripten aus- und angeschaltet werden können, true ist.

Licht Controller

Um sämtliche sich in der Szene befindenden Lichter während der Laufzeit des Simulators ändern und steuern zu können, wird ein zentrales Skript erstellt, von welchem aus der Zugriff

auf alle Lichter und Emissive Materials möglich ist. Dazu wird zunächst eine Variable für jedes Emissive Material, für alle Point Lights sowie ein Array für jede Spot Lights-Gruppe angelegt.

Zu Beginn des Spiels wird zunächst die Emission aller Emissive Materials ausgeschaltet, da alle Lampen zu Beginn des Spiels deaktiviert sein sollen. Um Lampen anschalten zu können, werden Funktionen für die jeweiligen Lichter bzw. Lichtergruppen geschrieben. Das Anschalten der Spot Lights funktioniert dabei über eine for-Schleife, die den Array in welchem die einzelnen Spot Lights gespeichert sind, durchläuft und die Spot Lights somit nacheinander anschaltet. Außerdem wird die Emission des vorliegenden Materials aktiviert.

Die gesamte for-Schleife befindet sich innerhalb einer If-Abfrage, bei welcher die Bool-Variable des entsprechenden Lichtes ausgelesen wird, die angibt, ob das Licht an oder aus ist. Nur wenn das Licht aus ist, wird die beschriebene for-Schleife ausgeführt. Ist das Licht bereits angeschaltet und die Funktion wird ausgelöst, wird ebenfalls eine for-Schleife ausgeführt, welche jedoch zur Folge hat, dass alle Lichter im Array deaktiviert werden und die Emission des entsprechenden Materials deaktiviert wird.

4.4.3. UI-Elemente

Die bereits angelegten, in 4.3.3 beschriebenen User Interface Elemente wie Dialog- und Interaktionsfenster müssen ebenfalls durch Skripte in den Spielablauf eingebunden werden. Einige Dialogfenster werden im Rahmen von Szenarien (siehe Kapitel 4.4.4) angezeigt und aus entsprechenden Funktionen aufgerufen. Folgend werden verschiedene eigenständige UI-Elemente vorgestellt und beschrieben.

Hauptmenü

Um das Hauptmenü mit seinen enthaltenen Informationen während des Spiels aufrufen zu können, werden verschiedene Möglichkeiten implementiert. Das Menü soll durch einen Menübutton, der sich in der oberen linken Ecke des Bildschirms befindet, aufgerufen werden können. Außerdem soll die Möglichkeit bestehen, durch das Klicken von der ESC-Taste sowie der M-Taste auf der Tastatur das Menü öffnen und schließen zu können. Dazu werden drei Funktionen geschrieben. Die erste Funktion beinhaltet das einfache Öffnen des Menüfensters. Vorangestellt wird (jeden Frame) geprüft, ob entweder die ESC- oder die M-Taste gedrückt werden. Ist dies der Fall, wird der Status des Hauptmenüfensters bzw. dessen GameObjects überprüft. Ist das GameObject bereits aktiv, so wird es deaktiviert. Ist es nicht aktiv, wird es aktiviert.

Für das Implementieren des Öffnens und Schließens über Buttons werden zwei weitere Funktionen benötigt. Auf der einen Seite eine Funktion, welche beim Aufruf das Hauptmenüfenster aktiviert. Der Aufruf findet beim Klick auf den Menübutton statt. Auf der anderen Seite steht

eine einfache Funktion zum Schließen des Hauptmenüs, welche durch das Klicken des Schließen-Buttons, der sich im Hauptmenü selbst befindet, ausgeführt wird.

Aufgaben-Übersicht

Um eine Übersicht der möglichen Aufgaben, die in das digitale Smart Home integriert wurden zu geben, soll ein schnell verfügbares Fenster in den Standard Bildschirm integriert werden. Dazu wird das in Kapitel 4.3.3 angelegte und beschriebene Element genutzt.

Beim Klicken des Buttons zum Öffnen des Fensters wird ein Timer gestartet, welcher nach Ablauf der voreingestellten Zeit das Fenster automatisch wieder schließt. Auch wird das Fenster automatisch geschlossen, wenn der Spieler sich fortbewegt, das heißt eine der Tasten W, A, S, D oder der Pfeiltasten drückt.

Um das Fenster immer am oberen Rand des Bildschirms bewegen zu können und das unabhängig von der Auflösung des aktuellen Endgeräts des Benutzers, muss ebendiese ausgelesen sowie eine Berechnung durchgeführt werden.

Das Auslesen der aktuellen Bildschirmauflösung wird nicht nur zu Beginn des Spiels, sondern jeden Frame durchgeführt, da ein Nutzer im Browser zwischen der kleineren, in den Browser eingebetteten Auflösung und einem Vollbildschirm wechseln kann. Die aktuelle Auflösung wird in einer float-Variable für die Breite und in einer weiteren float-Variable für die Höhe gespeichert.

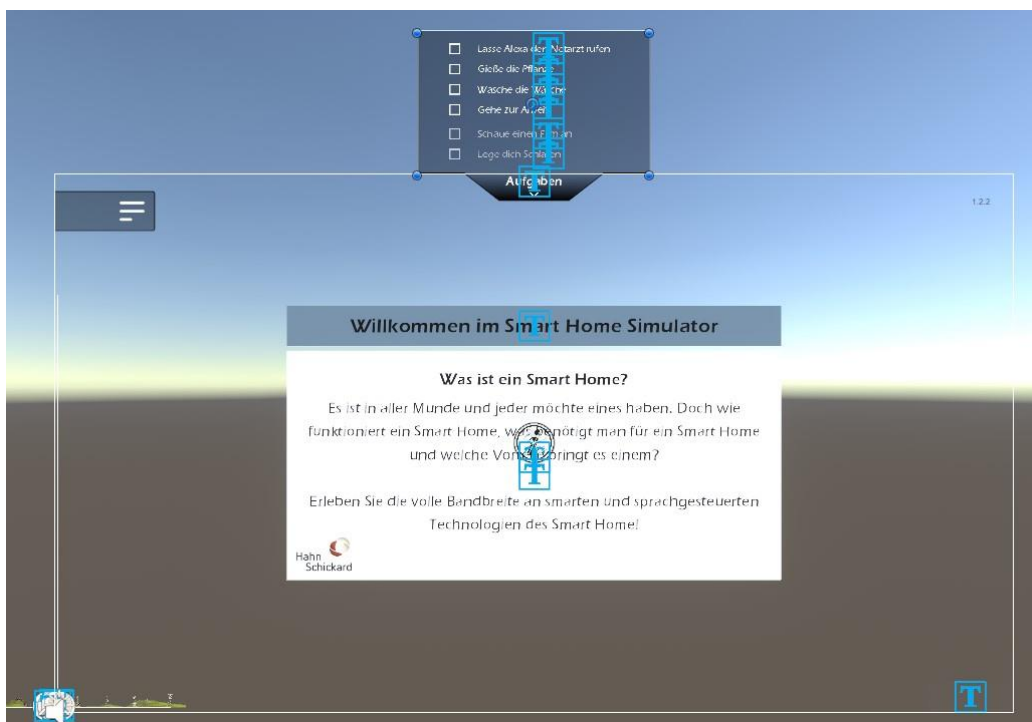


Abbildung 26: Aufgaben-Übersicht, geschlossen

Wie in Abbildung 26 zu sehen, befindet sich der Button zum Öffnen der Aufgaben-Übersicht außerhalb des Objekts. Das ermöglicht die einfache Orientierung am Bildschirmrand, sodass sich entweder das gesamte Objekt genau innerhalb oder außerhalb des Bildschirms befindet. In Unity wird die Bildschirmmitte als der Null-Punkt angesehen und auch die Mitte eines Objektes kennzeichnet der Nullpunkt dieses. Daher wird, um das gesamte Aufgaben-Übersicht-Objekt in den Bildschirm zu bewegen, die Position aus der Hälfte der gesamten Bildschirmhöhe abzüglich der Hälfte der Größe des Aufgaben-Übersicht-Objektes. Daraus ergibt sich folgende Rechnung:

$$(Screen.height/2) - (Object.height/2) = Position\ geöffnet$$

Die Position des geschlossenen Fensters ergibt sich ebenfalls aus der Hälfte der Bildschirmhöhe, jedoch addiert mit der Hälfte der Objekthöhe.

$$(Screen.height/2) + (Object.height/2) = Position\ geschlossen$$

4.4.4. Szenarien

Die in die Umgebung integrierten Szenarien, welche Smart Home Szenarien simulieren sollen, werden ebenfalls durch Skripte realisiert, die den Ablauf dieser beschreiben.

Sturzmatte

Bei einer Sturzmatte handelt es sich um eine mit Drucksensoren ausgestattete Matte. Diese sendet ein Signal, sobald sie belastet wird bzw. Druck auf sie ausgeübt wird (Fiessler Elektronik GmbH & Co. KG o.J.).

Simuliert wird die Funktion einer solchen Sturzmatte durch das Verwenden eines nicht sichtbaren Trigger-Objekts. In einem daran angefügten Skript wird anhand der *OnTriggerEnter()*-Funktion beim Eintreten in den Trigger einem Timer zunächst die Zahl null zugewiesen um anschließend nach jedem Frame zu prüfen ob der Timer größer sechs ist (siehe Abbildung 27). Wird diese Bedingung nicht erfüllt, wird dem Timer die vergangene Zeit hinzugerechnet und erneut abgefragt. Erreicht oder überschreitet der Wert des Timers die Zahl sechs, wird eine Audiodatei, in welcher Alexa zu hören ist, abgespielt. Außerdem wird das indirekte Licht im Schlafzimmer in roter Farbe angeschaltet sowie ein Dialog- bzw. Interaktionsfenster angezeigt. An dieser Stelle wird gefragt, ob der Notarzt gerufen werden soll. Entscheidet sich der Nutzer dafür den Notarzt nicht zu rufen und klickt dazu den entsprechenden Button, wird das Licht ausgeschaltet und das Szenario beendet. Dasselbe kann zu jeder Zeit durch das Verlassen des Triggers ausgelöst werden. Solange der Benutzer keine Interaktion vornimmt, läuft der Timer weiter und wird nach jedem Frame darauf geprüft, ob er größer 16 Sekunden ist. Erreicht er auch diesen Wert, wird, wie auch wenn der Benutzer den Button mit der Aufschrift

„Ja“ klickt, das angezeigte Interaktionsfenster geschlossen, eine nächste Audiodatei gestartet und das Licht beginnt zu blinken. Außerdem wird ein weiteres Dialogfenster angezeigt, in welchem die Funktionsweise einer Sturzmatte genauer erläutert wird. Wird dieses durch den Button geschlossen, wird das Licht ausgeschaltet und das Szenario beendet.

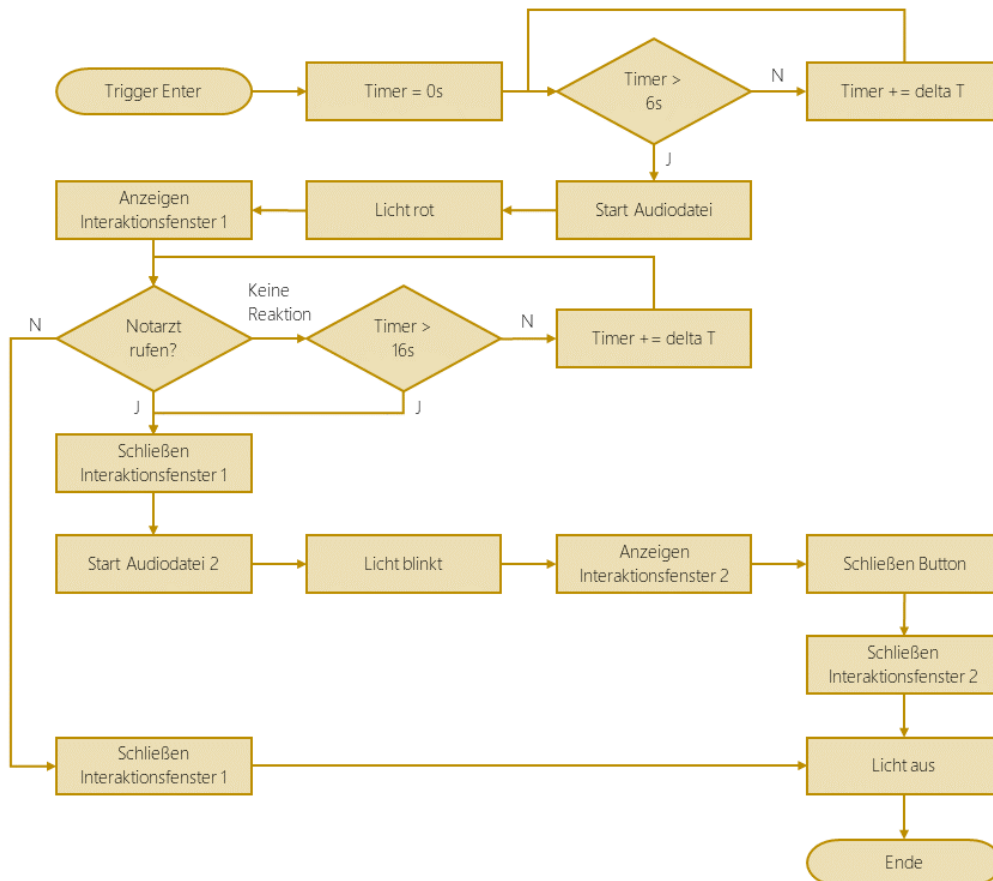


Abbildung 27: Flussdiagramm zum Ablauf "Sturzmatte-Szenario"

Pflanzensensor

Ein Pflanzensensor verfügt meist über mehrere Sensoren, die beispielsweise Feuchtigkeit des Bodens, Leitfähigkeit des Bodens, Helligkeit und Temperatur messen. Die gemessenen Werte werden anschließend mit den Normwerten der Pflanze verglichen (Neumann o.J.).

Um das Verhalten eines solchen Pflanzensensors zu simulieren, soll eine Abfrage über einen Sprachassistenten erstellt werden.

Die Möglichkeit des Zugriffs auf das Mikrofon und der Spracherkennung sind in Unity gegeben, und können mittels der Angabe von Schlüsselwörtern (Keywords) durchgeführt werden. Da die Spracherkennung von Unity jedoch nicht in einer WebGL-Version, die über einen Browser ausgeführt wird, funktioniert, müssen Spracheingaben anders simuliert werden.

Eine Möglichkeit bieten UI-Fenster, die in Textform auf eine Interaktion hinweisen. Das Szenario wird durch das Klicken eines Buttons gestartet.

Auf diese Weise wird auch das Szenario für die Abfrage des Pflanzensensors realisiert.

In der Umgebung wird um die Pflanze, für die die Abfrage eingerichtet werden soll, ein Trigger Objekt angelegt. Betritt das Spielerobjekt diesen Trigger während der Laufzeit des Spiels, wird ein Fenster geöffnet, in dem auf den Pflanzensensor und seine Funktionsweise hingewiesen wird. Außerdem befindet sich ein Button in dem Fenster, mit welchem die simulierte Abfrage des Pflanzensensors gestartet werden kann.

Die Interaktion mit dem Sprachassistenten wird durch einen erscheinenden Dialog simuliert. Dabei wird zunächst die Frage nach dem Zustand der Pflanze in einem Dialogfeld am unteren Rand des Bildschirms angezeigt. Darauf folgt das Abspielen einer Audio-Datei, in der eine Sprachassistenten-Stimme auf die Frage antwortet, sowie eine Textausgabe des gesprochenen Textes im Dialogfeld. Im Anschluss an die Ausgabe der Sensorwerte wird dem Nutzer die Möglichkeit geboten, mittels eines erscheinenden Buttons über der Gießkanne, die Pflanze zu gießen. Eine erneute Abfrage des Sensors ist möglich, wobei nach dem Gießen, nicht wie davor, der Wassergehalt in der Pflanzenerde als ausreichend beschrieben wird.

Wassermelder

Um einen Wasserschaden in dem virtuellen Smart Home zu simulieren, wird eine Interaktion erstellt, bei welcher durch einen Klick auf einen sich in der Szene befindenden Button die Waschmaschine gestartet werden kann. Dabei wird auch hier auf eine Animation zurückgegriffen. Neben dem Auslösen der Animation wird zusätzlich ein Timer gestartet. Nach Ablauf der Zeit wird automatisch eine Audiodatei abgespielt, in der auf einen Wasserschaden hingewiesen wird. Zusätzlich wird die indirekte Beleuchtung in der Küche mit der Farbe Rot eingeschaltet und auf den Displays der smarten Wecker wird der Schriftzug Wasserschaden aktiviert. Auch das Abstellen der Stromzufuhr im Badezimmer wird durch das Abschalten aller Lichtquellen in diesem Zimmer simuliert. Nach einer kurzen Verzögerung von zwei Sekunden wird die Funktionsweise eines Wassermelders sowie des gesamten Wasserschaden-Szenarios erläutert. Die Informationen sind dabei auf mehrere Dialogfenster verteilt. Da innerhalb des Wasserschaden-Szenarios auch das automatische Kontaktieren eines Handwerksbetriebes demonstriert werden soll, wird passend zu den Informationen nach dem Anzeigen des dritten Dialogfensters der Sound einer Türklingel abgespielt. Durch das Klicken des Schließen-Buttons des letzten Fensters wird ein 3D-Modell einer Person vor der Haustüre angezeigt. Der Benutzer wird außerdem dazu aufgefordert, an die Gegensprechanlage zu gehen und den Handwerker eintreten zu lassen. An dieser Stelle wird zusätzlich eine Bool-Variable auf true gesetzt,

welche angibt, dass das Wasserschaden-Szenario aktiv ist. Diese Variable ermöglicht die folgende Abfrage beim Klicken des „Türe öffnen-Buttons“ in dem Steuerpanel. Ist die Variable auf true beim Klicken des Buttons, so wird eine Funktion ausgeführt. In dieser wird zunächst eine Animation des Handwerker-3D-Modells gestartet, sowie die Haustüre geöffnet. Außerdem werden erneut Dialogfenster eingeblendet. Beim Schließen des letzten Dialogfensters wird das Szenario durch das Abschalten des Lichtes und der Anzeigen auf den Digitalweckern beendet.

TV

Das TV-Szenario soll die Vernetzung verschiedenster Smart Home Komponenten sowie deren automatische Steuerung darstellen. Integriert sind dabei der Sprachassistent, die Lautsprecher, verschiedene Lichter und Jalousien sowie der Fernseher. Das Auslösen des Szenarios ist nur bei Nacht möglich, das heißt, wenn die zentrale Bool-Variable „Day“ auf false ist. Dazu wird ein Trigger vor dem Sofa aktiviert. Beim Eintreten in diesen wird ein Interaktionsfenster angezeigt, welches die Möglichkeit erläutert über den Sprachassistenten den Fernseher zu starten. Klickt man auf den im Fenster angezeigten Button, wird eine Funktion gestartet, welche das Spielerobjekt in der Größe in Y-Richtung über eine Zeitdauer von zwei Sekunden kleiner macht sowie die Position in X-Richtung verändert. Dabei entsteht der Eindruck, dass man sich auf das Sofa setzt. Währenddessen wird ein Dialogfenster angezeigt, indem in Textform Alexa angesprochen wird. Die Antwort, dass der Fernseher gestartet wird, wird ebenfalls in Text, sowie auch als Ton wiedergegeben. Im Anschluss werden über das zentrale Jalousien-Script die Jalousien in Wohnzimmer und Küche heruntergefahren. Dies geschieht in Abhängigkeit davon, ob sie bereits geschlossen sind oder nicht. Auch das Licht im Wohnzimmer wird über das zentrale Licht-Script in seiner Farbe sowie in der Intensität verändert. Außerdem wird der Fernseher gestartet, indem ein Video in der Größe des Fernsehbildschirms abgespielt wird.

Während des Szenarios ist das Bewegen in der Szene nicht möglich. Auch die Navigationsbuttons und die Anzeige der Aufgaben sind deaktiviert. Um das Szenario zu beenden, wird ein einfacher Button im mittleren unteren Teil des Screens eingeblendet. Wird dieser gedrückt, wird das Spielerobjekt wieder zur ursprünglichen Größe und Position gebracht und das Navigieren ist wieder möglich.

4.5. Durchführung von Tests

Um die implementierten Szenarien und Funktionalitäten anhand von zielgruppenrelevanten Personen testen und bewerten zu lassen und somit die zielgruppenorientierte Entwicklung zu ermöglichen, wurden während der Umsetzung verschiedene Tests durchgeführt.

Zum einen ist bei der iterativen Entwicklung einer Anwendung das Testen des Entwicklers selbst notwendig, um die teilweise in Code verfasste Funktionalität zu testen. Jedoch ist auch die Durchführung von Tests mit Probanden sehr wichtig für eine nutzerfreundliche Anwendung. Im Rahmen dieser Arbeit wurden im wesentlichen drei Testläufe durchgeführt. Grundlage für Rückmeldungen und Feedback bot ein Fragebogen (siehe Anhang 1).

Der erste Test fand innerhalb der Abteilung Software Solutions statt, in welcher die Arbeit durchgeführt wird. Da sich die Personen dieser Testgruppe allein durch ihre Arbeit sehr gut mit Computern und Software auskennen, spiegeln sie nur bedingt die Zielgruppe, welche auch ältere und nicht Computer-affine Personen umfasst, wider. Jedoch konnte durch das Verständnis von Software und dadurch den Entwicklungsprozess von Anwendungen hilfreiche und fachspezifische Rückmeldung eingeholt werden.

Ein weiterer Test wurde auf einer Fachtagungsmesse zu den Themenfeldern Smart Living, Smart Home und Smart Building durchgeführt. Dabei wurde die Anwendung innerhalb eines eigenen Messestandes präsentiert und konnte an einem PC von Besuchern und Teilnehmern der Messe selbst bedient werden. Die am Test teilnehmenden Personen spiegeln die Zielgruppe deutlich besser wider als beim ersten Testdurchlauf. Die Probanden zeichnen sich aus durch ein Interesse an Smart Home Anwendungen und erstrecken sich bezogen auf das Alter von 25-30 Jahren bis hin zu 65 Jahren.

Ein letzter Test wurde innerhalb der Hahn-Schickard-Gesellschaft für angewandte Forschung durchgeführt. Dazu wurde ebenfalls ein Stand im Gebäude des Instituts aufgebaut und die Anwendung anhand eines PCs zur Nutzung bereitgestellt. Auch hier wird eine vielseitige Probandengruppe erreicht.

Anhand der verschiedenen Testläufe und dem auf den Fragebögen hinterlassenen Einschätzungen und Feedbacks konnten einige Erkenntnisse gewonnen werden. Durch die freiwillige Angabe des Alters, die in allen abgegebenen Feedbackbögen ausgefüllt wurde, konnten altersabhängige Probleme identifiziert werden. So gaben vorwiegend Personen über 50 Jahren an, Probleme mit der Steuerung zu haben. Abgeleitet aus diesem Problem wurde die Möglichkeit implementiert, über Navigationsbuttons durch die Szene zu navigieren.

Die Nützlichkeit des Simulators wurde von allen Probanden als nützlich bzw. sehr nützlich eingestuft.

5. Schlussbetrachtung und Ausblick

Für die Erstellung einer digitalen Informationsplattform zur Darstellung von Smart Home Anwendungen für die Hahn-Schickard-Gesellschaft wurde als erstes der Stand der Technik untersucht. Dabei wurden verschiedene Projekte unterschiedlichster Anwendungsbereiche gesichtet und verglichen. Dabei fiel auf, dass die Möglichkeit, Informationen interaktiv darzustellen bereits häufig Anwendung findet. Vor allem werden digitale und interaktive Informationsplattformen in den verschiedensten Kontexten verwendet, die Anwendung reicht dabei von Live-Messen über Produktdarstellungen und Musterhausführungen bis hin zu Campusrundgängen. Durch das Anlegen von drei Konzepten zur Umsetzung der Informationsplattform konnte der passendste Weg zur Erstellung gewählt werden. Hauptkriterium war dabei die möglichst hohe Interaktivität in der Darstellung von Smart Home Szenarien. Anschließend konnte ein 3D-Modell eines Hauses erstellt werden, in welches die Smart Home Elemente integriert werden sollten. Auch weitere 3D-Modelle, die zur Darstellung von Smart Home Komponenten dienen oder als Dekoration der Szene zu verstehen sind, wurden erstellt und innerhalb von Unity zu einer Szene zusammengefügt. Im Anschluss konnten User Interface Elemente erstellt werden sowie Designvorlagen für diese. In Unity wurden die erstellten 3D-Modelle zusammengeführt und anhand von Animationen und Skripten in C# um eine Funktionalität erweitert.

Abschließend ist zu erwähnen, dass die gewählte Art der Informationsplattform den Zweck der Darstellung und Demonstration von Smart Home Szenarien am besten erfüllt. Im Vergleich zu den erwähnten, eher statischen Konzepten, schafft eine Simulation, durch während der Laufzeit veränderbare Zustände, einen weitaus höheren Mehrwert und tiefere Einblicke in die Smart Home Funktionalität. Auch der Lerneffekt wird durch das einfache Demonstrieren von Abläufen deutlich gesteigert. Dennoch sind auch Informationen in Textform, zur genauen Erläuterung von Technik und Zusammenhängen notwendig, um die gewünschte Informationstiefe zu erreichen. Somit ist das Zusammenspiel von Demonstrationen und gezielt platzierten Informationstexten der zweckdienlichste Ansatz zur Umsetzung einer digitalen Informationsplattform zum Thema Smart Home.

Um ein zielgruppenorientierteres Ergebnis zu erhalten, welches die Bedürfnisse des Abnehmerkreises besser erfüllt, sind weitere Testläufe mit dem Produkt notwendig. Da für die Darstellung des Sinns und Zwecks von Smart Home Anwendungen unter anderem auch der häuslichen Pflege eine große Bedeutung zugeschrieben wird, ist die geeignete Auslegung der Plattform für an diesem Themenfeld Interessierte wichtig. Ein wichtiger Bestandteil der Zielgruppe sind daher ältere Personen, welche durch ihr Alter oder Krankheiten mit einer abnehmenden

Selbstständigkeit konfrontiert werden. Auch wenn bislang ein großer Teil dieser Menschen selbst nicht vertraut mit moderner Computertechnologie ist und das Internet nicht oder nur wenig nutzt, gelten sie als wichtiger Bestandteil der Zielgruppe. Denn Menschen, welche das Internet nicht oder nur in geringem Maße nutzen, haben meist auch zwangsläufig kaum Kenntnis von Smart Home Technologien. Jedoch kümmern sich häufig die jüngeren Verwandten um Pflegefälle in der Familie und sorgen sich daher auch um ggf. technische Ausstattung und Hilfen im Haushalt. Aufgrund der bereits heute sehr hohen täglichen Nutzung des Internets aller Altersgruppen, ist die Anwendung durchaus relevant für Personen, welche aus Gründen der häuslichen Pflege interessiert an Unterstützung durch Smart Home Anwendungen sind.

Erweiterungen der digitalen Informationsplattform sind in verschiedenster Hinsicht möglich. So kann die Plattform um zusätzliche Informationen in Form von Informationsfenstern oder gar Szenarien und Aufgaben bzw. Interaktionsmöglichkeiten erweitert werden. Auch ist eine auf die mobile Nutzung ausgelegte Version möglich, da für die aktuelle Nutzung ein PC notwendig ist.

Für die Hahn-Schickard-Gesellschaft für angewandte Forschung wäre eine zusätzliche VR-Version der Anwendung interessant. Damit ließe sich die Interaktivität steigern und für Demonstrationszwecke wie auf Messen und sonstigen Veranstaltungen wäre diese Möglichkeit ein besonderes Ereignis für die Nutzer.

Der Einsatz der webbasierten Lösung ist durch die einfache Integration als Canvas-Element in einer HTML-Seite sehr flexibel gestaltbar. Dadurch ergibt sich zum einen die Möglichkeit, die Anwendung in eine bereits bestehende Seite zu implementieren. Ein weiterer Ansatz wäre das Erstellen einer eigenen Seite. Dabei können auch Informationen oder sogar Medieninhalte außerhalb des Canvas und damit unabhängig von der Anwendung in die Seite integriert werden.

Aufgrund des zunehmenden Fachkräftemangels in Deutschland und der steigenden Nachfrage von Pflegepersonal werden vernetzte Technologien in Wohnhäusern in Zukunft wichtige Rollen im alltäglichen Leben spielen. Um über aktuelle und moderne Technologien aufzuklären, Informationen zu vermitteln und eine Simulation bzw. eine Vorführung von möglichen Szenarien durchzuführen, bieten digitale, interaktive 3D-Touren eine exzellente Möglichkeit. So kann auch die Hahn-Schickard-Gesellschaft in Zukunft von solchen 3D-Visualisierungen und Simulationen profitieren.

Allgemein ist zu erwarten, dass die Zahl von Online-Touren sowie Online-Demonstrationen und Veranstaltungen in Zukunft stark zunehmen wird.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eidesstatt, dass ich die Ihnen vorliegende Bachelorarbeit selbstständig und ohne Nutzung anderer als der aufgeführten Hilfsmittel erstellt habe. Die direkte oder indirekte Übernahme von Gedanken aus fremden Quellen sind als solche kenntlich gemacht.

Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen und wurde auch noch nicht veröffentlicht.

Villingen-Schwenningen, am 19. 08. 2022

(Timo Mattes)

Literaturverzeichnis

Ambient Assisted Living Deutschland (o.J.): Technik die unser Leben vereinfacht. Online verfügbar unter <http://www.aal-deutschland.de/>, zuletzt geprüft am 25.05.2022.

ARD/ZDF-Forschungskommission (2022): Pressemitteilung | ARD/ZDF-Forschungskommission. 25 Jahre ARD/ZDF-Onlinestudie: Nach Corona-Tief steigt die Unterwegsnutzung wieder, Streaming und die Mediatheken sorgen weiter für mehr Mediennutzung im Internet. Online verfügbar unter <https://www.ard-zdf-onlinestudie.de/ardzdf-onlinestudie/pressemitteilung/>, zuletzt aktualisiert am 15.08.2022, zuletzt geprüft am 15.08.2022.

Autodesk, Inc. (o.J.): 3ds Max: Erstellen Sie umfangreiche Welten und hochwertige Entwürfe. Modellieren und rendern Sie in atemberaubendem 3D. Online verfügbar unter <https://www.autodesk.de/products/3ds-max/overview?term=1-YEAR&tab=subscription>, zuletzt geprüft am 27.05.2022.

Barth, Timo (2017): 3D Modelle und 3D Renderings - Darstellung von Systemen zur Akustikdämmung. Online verfügbar unter <https://www.virtualemotion.de/digitale-welt/low-poly-high-poly-modellen>, zuletzt aktualisiert am 26.09.2017, zuletzt geprüft am 01.07.2022.

Bender, Michael; Brill, Manfred (2006): Computergrafik. Ein anwendungsorientiertes Lehrbuch. 2., überarb. Aufl. München, Wien: Hanser. Online verfügbar unter https://bilder.buecher.de/zusatz/12/12262/12262366 lese_1.pdf, zuletzt geprüft am 01.07.2022.

Bibliographisches Institut GmbH (o.J.a): Internetplattform. Online verfügbar unter <https://www.duden.de/rechtschreibung/Internetplattform>, zuletzt aktualisiert am 27.04.2018, zuletzt geprüft am 30.06.2022.

Bibliographisches Institut GmbH (o.J.b): Plattform. Online verfügbar unter <https://www.duden.de/rechtschreibung/Plattform>, zuletzt aktualisiert am 27.04.2018, zuletzt geprüft am 30.06.2022.

BITKOM (2014): Vor dem Boom - Marktaussichten für Smart Home. Ergebnisdokument der Fokusgruppe Connected Home. Unter Mitarbeit von Stephan Schneider, Tobias Arns, Sven Brandt, Ralf Kirchner, Bernd Dechert, Esther Hild et al. Nationaler IT-Gipfel. Berlin, 23.10, zuletzt geprüft am 23.05.2022.

Fiessler Elektronik GmbH & Co. KG (o.J.): Sicherheits-Trittmatten. Online verfügbar unter <https://www.fiessler.de/de/startseite/angebotsspektrum/sicherheitsprodukte/sicherheits-trittmatten>, zuletzt geprüft am 11.07.2022.

Garthe, Bernhard (o.J.): 3D Modellierung. Ausarbeitung zum Schwerpunktseminar. Online verfügbar unter <https://homepages.thm.de/~hg10013/Lehre/MMS/SS02/Garthe/text.htm>, zuletzt aktualisiert am 10.10.2002, zuletzt geprüft am 01.07.2022.

homeandsmart GmbH (o.J.): Smart Home Sicherheit - wie sicher ist das smarte zu Hause? Smart Home: Haus- und Datensicherheit sowie Schutz vor Hackern. Online verfügbar unter <https://www.homeandsmart.de/sicherheit>, zuletzt geprüft am 23.05.2022.

Infraserv GmbH & Co. Höchst KG (o.J.): Status quo der Gebäudeleittechnik. Online verfügbar unter <https://www.infraserv.com/de/leistungen/facility-management/gebäudeautomation/entwicklung-gebäudeleittechnik/>, zuletzt geprüft am 25.05.2022.

Karolin Wisser; Wisser, Karolin (2018): Gebäudeautomation in Wohngebäuden (Smart Home). Eine Analyse der Akzeptanz. Wiesbaden: Vieweg.

Luber, Stefan; Litzel, Nico (2019): Definition - Was ist Smart Home. Vogel Communications Group GmbH & Co.KG. Online verfügbar unter <https://www.bigdata-insider.de/was-ist-smart-home-a-809018/>, zuletzt geprüft am 25.05.2022.

Maier, Dominik (2021): Unity Software – die große Aktienanalyse. Online verfügbar unter <https://goldesel.de/Artikel/unity-software-die-grosse-aktienanalyse>, zuletzt aktualisiert am 18.10.2021, zuletzt geprüft am 22.07.2022.

Michael (2021): Alles über Polygone, Triangles und Vertices in Unity. In: *NOBREAKPOINTS*, 19.07.2021. Online verfügbar unter <https://blog.nobreakpoints.com/alles-ueber-polygone-triangles-und-vertices-in-unity/>, zuletzt geprüft am 01.07.2022.

Neumann, Volkmar (o.J.): Pflanzen-Sensoren – High-Tech im Garten. Online verfügbar unter https://www.garten-pflanzen.info/pflanzen-sensoren-high-tech-im-garten/#Wie_funktionieren_die_Pflanzen-Sensoren, zuletzt geprüft am 11.07.2022.

Oberste-Ufer, Kai (2020): Was die beste BIM-Software im Jahr 2020 auszeichnet. dormakaba Holding AG. Online verfügbar unter <https://blog.dormakaba.com/de/was-die-beste-bim-software-im-jahr-2020-auszeichnet/>, zuletzt geprüft am 27.05.2022.

Parisi, Tony (2014): Programming 3D applications with HTML5 and WebGL. [3D animation and visualization for web pages]. 1. ed. Beijing [u.a.]: O'Reilly, zuletzt geprüft am 09.05.2022.

Petty, Josh (o.J.): What is 3D Modeling & What's It Used For? Online verfügbar unter <https://conceptartempire.com/what-is-3d-modeling/>, zuletzt geprüft am 01.07.2022.

Phillip (2020): Was ist ein Modell? (Wissenschaftstheorie einfach erklärt). Online verfügbar unter <https://shribe.de/was-ist-ein-modell/>, zuletzt aktualisiert am 14.06.2022, zuletzt geprüft am 01.07.2022.

Preim, Bernhard; Dachsel, Raimund (2015): Interaktive Systeme. Berlin, Heidelberg: Springer Berlin Heidelberg.

Prof. em. Dr. Schlottke, Peter F. (o.J.): Modell. Hg. v. Markus Antonius Wirtz. Online verfügbar unter <https://dorsch.hogrefe.com/stichwort/modell>, zuletzt aktualisiert am 10.03.2021, zuletzt geprüft am 01.07.2022.

Red Hat, Inc. (2019): Was ist eine IDE? Online verfügbar unter <https://www.red-hat.com/de/topics/middleware/what-is-ide>, zuletzt geprüft am 22.07.2022.

Serlo Education e. V. (o.J.): Modellierung (Informatik). Online verfügbar unter <https://de.serlo.org/informatik/176590/modellierung-informatik>, zuletzt geprüft am 01.07.2022.

Stadtwerke Düsseldorf AG (o.J.): Smart Home: Vernetztes Zuhause. Vorurteile rund um den Trend aus dem Weg geräumt. Online verfügbar unter <https://www.swd-ag.de/magazin/smart-home/>, zuletzt geprüft am 25.05.2022.

Struck, Ann-Marie (2021): Das vernetzte Zuhause - Immer mehr Deutsche nutzen Smart Home. Vogel Communications Group. Online verfügbar unter <https://www.it-business.de/immer-mehr-deutsche-nutzen-smart-home-a-1050566/>, zuletzt geprüft am 25.05.2022.

Unity Technologies (Hg.) (o.J.): Echtzeit-3D-Technologie in Architektur-, Ingenieurs- und Bauwesen erklärt. Online verfügbar unter <https://unity.com/de/solutions/architecture-engineering-construction/rt3d-explained>, zuletzt geprüft am 09.05.

Unity Technologies (2022): Scripting. Online verfügbar unter <https://docs.unity3d.com/Manual/ScriptingSection.html>, zuletzt aktualisiert am 17.07.2022, zuletzt geprüft am 22.07.2022.

Vertex – DGL Wiki (2014). Online verfügbar unter <https://wiki.delphigl.com/index.php/Vertex>, zuletzt aktualisiert am 04.01.2014, zuletzt geprüft am 01.07.2022.

VisCircle GmbH (o.J.a): Einsteigerguide zum UV-Mapping: Grundlagen und ein wenig mehr. Online verfügbar unter <https://viscircle.de/einsteigerguide-zum-uv-mapping-grundlagen-und-anwendungsvorschlaege/>, zuletzt aktualisiert am 15.11.2017, zuletzt geprüft am 04.07.2022.

VisCircle GmbH (o.J.b): Einsteigerguide zum UV-Mapping: Grundlagen und ein wenig mehr. Online verfügbar unter <https://viscircle.de/einsteigerguide-zum-uv-mapping-grundlagen->

und-anwendungsvorschlaege/, zuletzt aktualisiert am 15.11.2017, zuletzt geprüft am 04.07.2022.

VNR Verlag für die Deutsche Wirtschaft AG, Computerwissen (2021): Smart Home Klimaanlage. Intelligent temperieren in den eigenen vier Wänden. Online verfügbar unter <https://www.computerwissen.de/internet-und-netzwerk/smart-home/smart-home-moeglichkeiten/smart-home-klimaanlagen/>, zuletzt geprüft am 23.05.2022.


Wenke, Henning (2012): Computergrafik. Universität Osnabrück. Online verfügbar unter [http://www-lehre.inf.uos.de/~cg/2012/PDF/2012-05-14%20Transformationen%20mit%20OpenGL%20\(Web\).pdf](http://www-lehre.inf.uos.de/~cg/2012/PDF/2012-05-14%20Transformationen%20mit%20OpenGL%20(Web).pdf), zuletzt aktualisiert am 14.05.2012, zuletzt geprüft am 01.07.2022.

Wolfangel, Eva (2019): Smart Homes - Intelligent, aber unbeliebt. Mit einem Smart Home ließe sich Energie und Geld sparen sowie komfortabler leben. Trotzdem kommt die Technik den meisten Deutschen nicht ins Haus. Spektrum der Wissenschaft Verlagsgesellschaft mbH. Online verfügbar unter <https://www.spektrum.de/news/das-smart-home-leidet-unter-akzeptanzproblemen/1681066>, zuletzt geprüft am 25.05.2022.

Anhang 1 – Fragebogen zur Durchführung von Tests

Smart Home Simulator

Fragebogen



Hahn
Schickard

Freiwillige Angaben: Ihr Alter *: _____ Ihr Geschlecht *: _____

1. Kennen Sie sich bereits mit Smart Home aus?

Ich kenne mich nicht aus Ich nutze selbst folgende smarte Gadgets zu Hause:
 Ich kenne mich ein wenig aus _____
 Ich kenne mich gut aus _____

2. Für wie nützlich halten Sie den Simulator?

Sehr nützlich Nützlich Kaum nützlich Nicht nützlich

Der Simulator hat mir keine neuen Erkenntnisse gebracht Ich konnte Neues über Smart Home durch den Simulator lernen

3. Folgende Probleme sind bei Ihnen aufgetreten:

4. Wie intuitiv war die Steuerung für Sie?

Sehr intuitiv Ich bin klargekommen Umständlich

Wie würden Sie sich die Steuerung vorstellen?

5. Umfang des Simulators

Ich habe alle Aufgaben erledigt und mir alles angesehen

Ich habe alle Aufgaben erledigt

Ich konnte nicht alle Aufgaben finden

Ich habe nur kurz reingeschaut

Was haben Sie durchgeführt?


Der Pflanzensensor wurde abgefragt

Der Notarzt wurde gerufen


Der Wassermelder hat ausgelöst

Ich habe Jalousien über den Steuerbildschirm gesteuert

6. Welche Smart Home Komponenten (Aktoren & Sensoren) sollten noch in den Simulator eingebunden werden?



SMART HOME
SIMULATOR



Smart Home **Simulator**

Fragebogen



7. Welche Smart Home Szenarien könnten Sie sich im Simulator vorstellen?

8. Sonstige Anmerkungen / Ideen

----- Vielen Dank für Ihre Teilnahme! -----

* Die Angaben sind anonym und werden nicht mit Ihnen in Verbindung gebracht. Die Daten werden ausschließlich zur Verbesserung des „Smart Home Simulators“ verwendet.

