

Lernen mit Augmented Reality

Projektbericht

vorgelegt von:

Yannick Schiele

Kalter Markt 6

73525 Schwäbisch Gmünd

Studiengang:

Master Informatik

Matrikelnummer:

61802

Erstgutachter:

Dr. Marc Hermann

Ort und Datum:

Schwäbisch Gmünd, 31.08.2020

Abstract

Seit Jahrzehnten beschäftigt sich der Mensch mit tragbarer Hardware zur Darstellung von digitalen Inhalten bis hin zu virtuellen Welten, doch erst die neuesten Entwicklungen machen diese Technologien in der Praxis nutzbar. Neben der virtuellen Realität (VR) bietet auch die Augmented Reality (AR) eine lohnende Technologie für viele Anwendungsszenarien.

Die dafür notwendige Hardware umfasst dabei lediglich ein Smartphone mit dem Betriebssystem Android oder iOS. Die verhältnismäßig geringen Anschaffungskosten und die große Verbreitung dieser Geräte, machen AR zu einer interessanten Technologie in der Aus- und Weiterbildung. Die Möglichkeit, während einer Tätigkeit Informationen im Blickfeld darzustellen, bietet viele Anwendungsfälle.

Um eine Einschätzung in diesem Anwendungsbereich zu ermöglichen, werden in dieser Arbeit die Möglichkeiten von Augmented Reality im Bildungskontext zusammengefasst. Einer der prägendsten Vorteile ist die Erzielung höherer Lerneffekte durch virtuelles oder simuliertes "learning by doing" mithilfe der Visualisierungsmöglichkeiten von AR im Vergleich zu herkömmlichen Lehrmethoden.

Im Rahmen dieser Arbeit wurde eine solche Visualisierungsmöglichkeit zu einem virtuellen und simulierten "learning by doing" entwickelt. Mit dieser Lösung ist es Kindern möglich, Tierbilder mit einem Smartphone zu scannen und dadurch eine 3D Animation dieses Tieres und lehrreiche Informationen dazu via Augmented Reality dargestellt zu bekommen. Durch die Möglichkeit der Interaktion mit dem Tiermodell, wird das Lernen der dargestellten Informationen verbessert.

Inhaltsverzeichnis

Abstract.....	I
1 Einleitung	3
2 Projektmethodik	4
2.1 Projektziel	4
2.2 Vorgehen	4
3 Theoretische Grundlagen	5
3.1 Augmented Reality	5
3.2 AR im Bereich des Lernens	7
3.3 Apples ARKit.....	16
4 Beschreibung des Projektverlaufs	18
4.1 Konzeption	18
4.2 Realisierung.....	20
5 Projektergebnis	33
6 Fazit und Ausblick.....	35
Literatur.....	36

1 Einleitung

Die Digitalisierung ist inzwischen in fast allen Lebensbereichen angekommen und hat auch für die Weiterentwicklung des Schulunterrichts eine erhebliche Bedeutung. Augmented Reality ist dabei laut dem bekannten US-amerikanisches Marktforschungsunternehmen Gartner eine der zukunftsweisenden Technologien die derzeit viel Beachtung findet. [1] [2]

Augmented Reality (AR), ist eine Erweiterung der realen Welt durch zusätzliche, erweiterte Elemente, wie z.B. das Einfügen von Informationen oder Hilfestellungen. [3]

Die Kombination von innovativen Technologien, in diesem Fall Augmented Reality, und entsprechender didaktischer Umsetzung bietet eine große Chance. Daraus ist der Einsatz von AR für Bildung und Ausbildung in verschiedenen Arten und Sektoren für Bildungseinrichtungen und Unternehmen von zunehmendem Interesse. Es stellt sich bei diesem steigenden Interesse die Frage, wie die, mit einer Erweiterung der Realität verbundenen, Möglichkeiten sinnvoll und didaktisch fundiert für den Unterricht genutzt werden können. [4] [3]

In dieser Arbeit wird zunächst im folgenden Kapitel die genaue Projektmethodik anhand des Projektziels und das Vorgehen erläutert. Daraufhin werden die, für das Verständnis dieser Projektarbeit notwendigen, theoretischen Grundlagen, wie Augmented Reality und ihr Einsatz in der Bildung, näher erläutert. Kapitel 4 beschreibt den Verlauf der Projektarbeit aufgeteilt in Entwurfs- und Realisierungsphase. Abschließend wird das Ergebnis des Projekts dargestellt, sowie ein Fazit und ein Ausblick.

2 Projektmethodik

Dieses Kapitel dient der Beschreibung der Projektmethodik. Um einen Einblick in den Entwicklungsprozess der entwickelten Applikation zu geben, wird zunächst das Projektziel dargestellt und erläutert. Das nächste Unterkapitel gibt einen kurzen Überblick über die Vorgehensweise bei der Durchführung dieser Projektarbeit.

2.1 Projektziel

Das Ziel des Projekts ist die Entwicklung einer Augmented Reality fähigen iOS App basierend auf der zugehörigen Programmiersprache Swift. Die App soll nach dem erfolgreichen Scan-Vorgang gewisser Tierbilder, diese erkennen und ein 3D Modell sowie zusätzliche Bilder und Informationen zu diesem Tier anzeigen.

Diese dargestellten Informationen sollen dabei von dem Anwender selbständig durch entsprechende Interaktionen mit der AR-Darstellung gewechselt werden können. Auch das 3D Modell des Tieres soll einige Interaktionsfunktionalitäten beinhalten. Es muss sich durch die entsprechenden Aktionen bewegen, drehen und vergrößern bzw. verkleinern lassen.

Diese App soll auf allen iOS Geräten, also sowohl iPhones als auch iPads, ab der Version 12.1 genutzt werden können.

Am Ende des Projekts soll ein lauffähiges Produkt entwickelt und die erzielten Ergebnisse in Form eines Projektberichts dokumentiert werden.

2.2 Vorgehen

Die frühe Absprache, die bereits zu Beginn des Semesters stattfand, ermöglichte es, Interessensgebiete und Prioritäten dieser Projektarbeit zu diskutieren. Dies führte zu einer möglichst frühen Konkretisierung der Anforderungen und bestätigte das Projektziel.

Durch eine erste Recherche konnten die technischen Anforderungen an die Applikation abgeleitet werden. Gleichzeitig wurden die notwendigen Kompetenzen in der Entwicklung von iOS Apps erworben. Nach der Recherche konnte ein Entwurf der Komponenten und Technologien definiert werden. Schließlich wurde die Implementierung unter Berücksichtigung der Erkenntnisse aus der Ausbildungsphase, der Analyse- und Designphase in Angriff genommen.

3 Theoretische Grundlagen

In diesem Kapitel wird zunächst das Thema Augmented Reality erläutert, um ein Verständnis für die Technologie dieser Projektarbeit zu liefern (3.1). Daran schließen sich wichtige Theorien und Forschungen zu der Verwendung dieser Technologie im Bereich der Bildung und des Lernens (3.2). Zuletzt wird das von Apple bereitgestellte Framework ARKit und dessen Funktionalität, im Hinblick auf die entwickelte Applikation, erläutert (3.3).

3.1 Augmented Reality

Um die einzelnen Aspekte des didaktischen Konzepts von Augmented Reality (AR) zu verstehen, sind Grundkenntnisse dieser Technologie notwendig. Daher werden im Folgenden einige Grundlagen zu Augmented Reality dargelegt. Besonderes Augenmerk wird dabei auf die in dieser Ausarbeitung verwendete AR-Technologie gelegt.

Bei Augmented Reality wird die reale Umgebung mit virtuellen Objekten kombiniert. Es existiert eine Echtzeit-Interaktion mit den virtuellen Objekten, wobei die realen und virtuellen Objekte in einer dreidimensionalen Beziehung zueinanderstehen. [5] Abbildung 1 zeigt eine Beispielhafte AR-Applikation. Hierbei wird die reale Raumumgebung um einen Sessel als virtuelles Objekt erweitert.

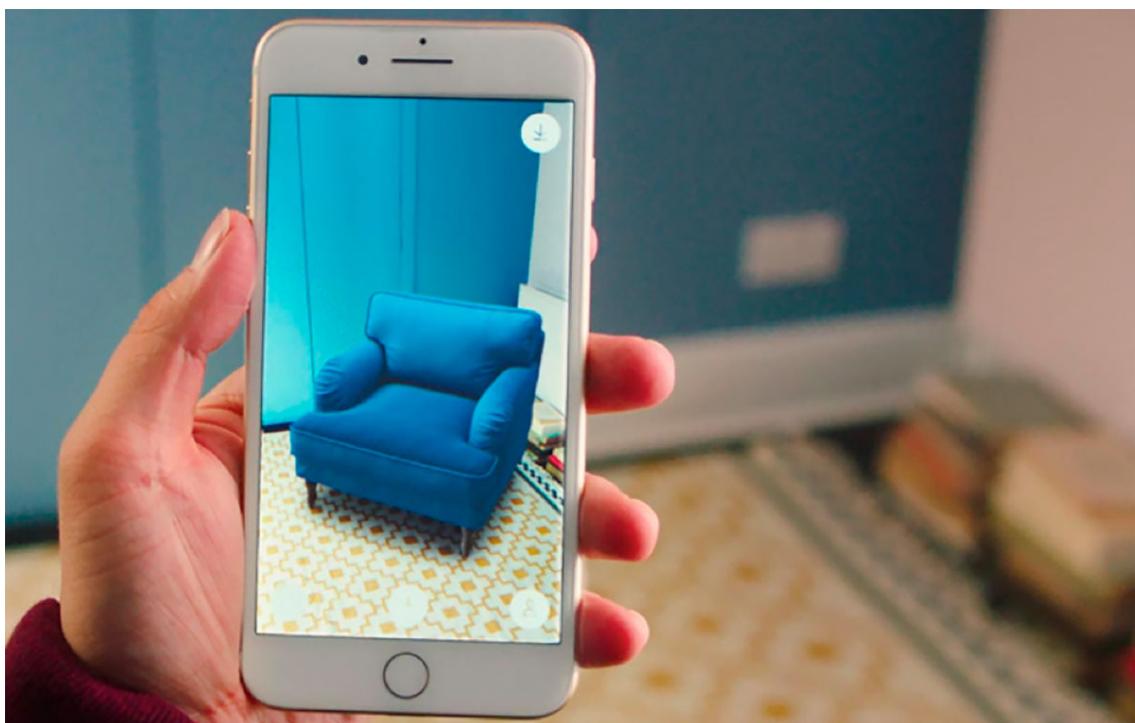


Abbildung 1: Beispielhafte AR-Applikation [6]

Es gibt drei grundsätzliche Varianten für die Darstellung von AR: projektionsbasierte AR, Video-See-Through und Optical-See-Through. [7]

Projektionsbasierte AR blendet die virtuellen Inhalte über Projektoren in die reale Welt ein. Das französische Start-up-Unternehmen "Hololamp" beispielsweise verwendet einen tragbaren Projektor, um die Gerichte eines Restaurants vor der Bestellung als AR-Objekt auf dem Esstisch anzuzeigen, siehe Abbildung 2.



Abbildung 2: AR Darstellung mit Hilfe eines Projektors [8]

Beim Video-See-Through wird das Bild einer Kamera, das die Benutzer betrachten, mit virtuellen Inhalten überlagert. Ein Beispiel für diese Möglichkeit ist AR mit Smartphones oder Tablets, siehe Abbildung 1. Diese Variante zur Darstellung von AR, wird auch in dieser Ausarbeitung verwendet, da hier keine weitere Hardware benötigt wird und die entwickelte Lösung somit auch auf einfache Art und Weise für eine breite Maße an Anwendern bereitgestellt werden kann.

Optical-See-Through lässt sich durch das eingesetzte transparente Display charakterisieren. Dadurch kann der Betrachter die reale Welt normal sehen, und die virtuellen Inhalte werden zusätzlich dargestellt. Diese Technologie wird z. B. bei der HoloLens-Brille von Microsoft eingesetzt, siehe Abbildung 3.



Abbildung 3: HoloLens-Brille von Microsoft [9]

Eine weitere Unterscheidung wird auch zwischen markerbasierter und markerloser Augmented Reality vorgenommen. Bei der ersten Variante sind künstliche Marker im Raum oder auf dem Körper notwendig, um die virtuellen Objekte zueinander und zum Raum in Beziehung zu setzen. Die markerlose AR erfordert keine zusätzlichen Marker im Raum. Diese Art der Augmented Reality ist mit der HoloLens gegeben. Wird die markerlose AR durch die Erkennung von dreidimensionalen Objekten in der Umgebung ergänzt, so dass virtuelle Objekte auch hinter realen Objekten verschwinden können, spricht man von raumbezogener Augmented Reality (R-AR). [4]

3.2 AR im Bereich des Lernens

Ein offensichtliches Anwendungsgebiet für die im vorigen Kapitel beschriebene neue Technologie im Rahmen der allgemeinen und beruflichen Bildung sind Schulen verschiedener Art. Während die meisten Schulen, unabhängig von der Art der Schule, bereits mit Computerräumen und digitalen Präsentationsmedien ausgestattet sind, nimmt zwar das Niveau der eingesetzten Informationstechnologie zu, allerdings werden Smartphones und Tablets weitestgehend nicht berücksichtigt. [3]

Abbildung 4 zeigt die konkrete Nutzung der digitalen Medien in den Schulen aus dem Jahre 2017.

Am häufigsten werden stationäre Computer eingesetzt (79 %), gut jeder zweite Schüler gibt an, dass zumindest selten am Whiteboard gearbeitet wird (52 %), etwas weniger (47 %) verwenden zumindest selten ein Smartphone im Unterricht. 37 Prozent verwenden ein Notebook. Tablets sind seltener im Einsatz, hier hat nur jeder Fünfte überhaupt schon mal Erfahrungen mit diesen Geräten in der Schule gesammelt. [10]

Ein genauerer Blick auf die Häufigkeit der Nutzung zeigt, dass mit einer regelmäßigen Nutzung (mind. mehrmals pro Woche) bislang nur das Whiteboard (31 %) und der Computer (22 %) nennenswert im Schulalltag angekommen sind. Smartphones (13 %), Laptops (9 %) oder gar Tablets (4 %) spielen noch immer keine große Rolle. Auch die weitere Betrachtung der Nutzung zumindest einmal im Monat bestätigt dieses Bild: Nur jeder zehnte Schüler nutzt im Zeitraum von vier Wochen einen Tablet-PC (11 %), jeder Vierte ein Notebook (25 %) und jeder Dritte ein Smartphone (31 %). Nur Whiteboards (43 %) und stationäre Computer (59 %) kommen relativ häufig in der Schule zum Einsatz. [10]

Diese digitalen Medien werden in der Schule genutzt

Nutzung digitaler Medien im Unterricht an deutschen Schulen 2017 (in %)

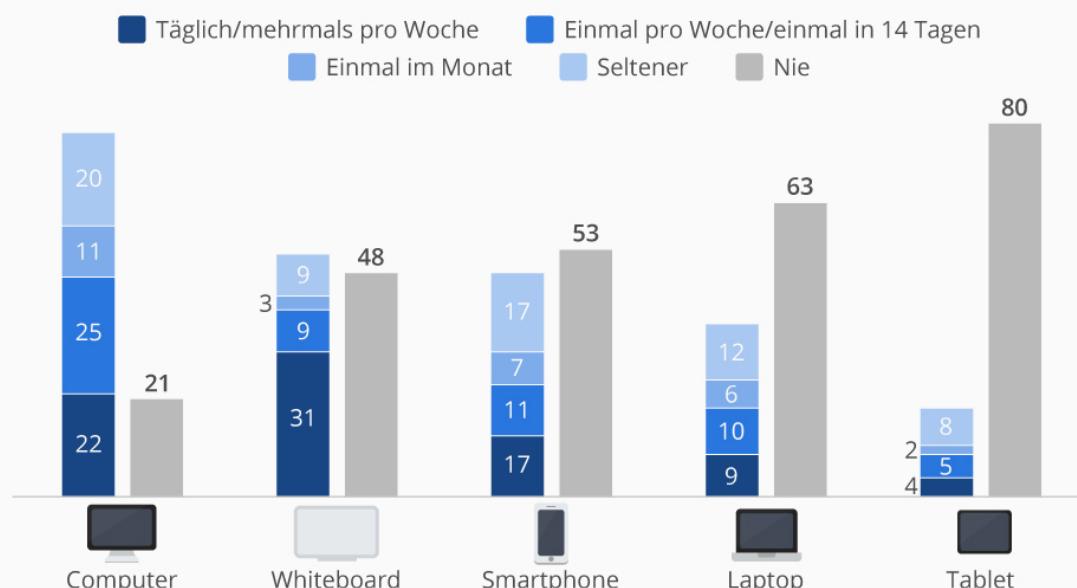


Abbildung 4: Nutzung der digitalen Medien in der Schule [10]

Dieser Statistik gegenüber steht die Besitzquote von Smartphones bei Schülern und Studenten. Ab einem Alter von 10 Jahren gibt die Statistik in Abbildung 5 an, dass bereits 75% der Kinder ein Smartphone besitzen. Ab 12 Jahren steigt die Besitzquote sogar auf über 90%.

Aus diesen Besitzquoten lässt sich ableiten, dass Schüler und Studierende es gewohnt sind, auf ihrem Smartphone nach Informationen zu suchen oder Anwendungen zu nutzen. Diese Art des in-formalen Lernens stellt heute einen nicht unerheblichen Teil des schulischen Wissenserwerbs dar.

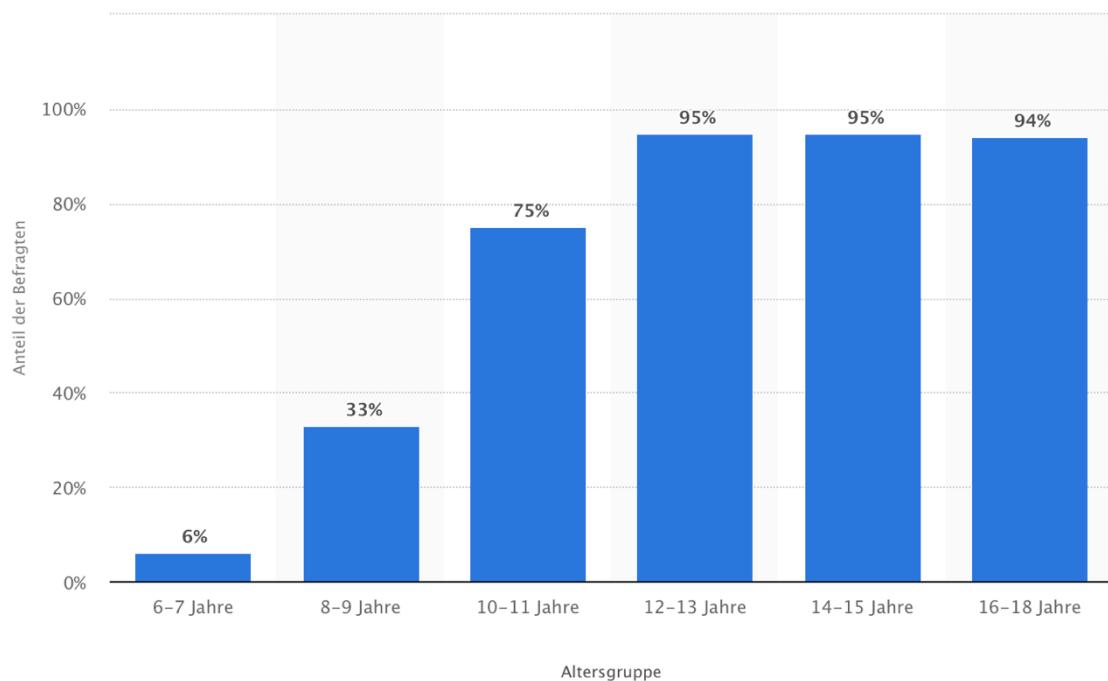


Abbildung 5: Smartphone-Besitz bei Kindern und Jugendlichen in Deutschland im Jahr 2019 nach Altersgruppe [11]

Um den Einsatz der AR-Technologie in Schulen und Universitäten voran zu treiben, muss das "BYOD"-Prinzip angewendet werden. "Bring Your Own Device" (dt. Bringen Sie Ihr eigenes Gerät mit) bedeutet, dass alle Schülerinnen und Schüler ihre eigenen Geräte in die Schule mitbringen und sie im Unterricht benutzen, falls sie eines besitzen.

Nach der Statistik sollte dies eine fast vollständige Abdeckung mit der benötigten Technik ermöglichen. Wenn im Einzelfall nicht jeder Schüler Student vor Ort über ein eigenes mobiles Gerät verfügt, sollte die Institution Leihgeräte für die jeweiligen Schüler oder Studenten zur Verfügung stellen. [12]

Durch den verstärkten Einsatz von mobilen Geräten wie Tablets und Smartphones in Verbindung mit AR-Anwendungen können Lehr- und Lernorte orts- und zeitunabhängig werden. Die konkreten Potenziale und Vorteile sind dabei vielfältig. [3]

Damit können theoretische Konzepte jedes Schulfachs jederzeit und überall angezeigt werden. Diese theoretische Darstellung kann durch die Visualisierung von konkreten Beispielen direkt angereichert und damit leichter verständlich gemacht werden. [13] So könnten beispielsweise verschiedene Inhalte im Informatikunterricht in Form von 3D-Modellen angereichert werden, wie z.B. die Konstruktion verschiedener Computerarchitekturen oder Komponenten. [3]

Insbesondere die flexiblen Möglichkeiten, von überall Zugang zu allen Lehrmaterialien zu haben, ermöglichen eine effektive Kommunikation und Zusammenarbeit auch mit geografisch verteilten Teams. [13] Auch für das Lehrpersonal lassen sich verschiedene Mehrwertfaktoren feststellen. Durch die Verwendung von Problemstellungen, Unterrichtsfächern oder Fragen, die von den Studierenden nur mit Hilfe von Virtualität bearbeitet werden können, kann die Motivation und das Engagement der Studierenden gefördert werden. [14] [15] Diese beiden relevanten Faktoren des Lernens werden im Folgenden noch einmal genauer in Bezug auf AR beschrieben.

Weitere positive Effekte sind die Förderung von Kreativität und Phantasie [16] sowie eine verbesserte Zusammenarbeit zwischen Lernenden und Lehrenden [17].

Um diese Vorteile und Potenziale nutzen zu können, sollten AR-Systeme sowohl einfach zu bedienen und zu verstehen sein als auch robust für den täglichen Gebrauch und in der Lage sein, den Schülern klare und präzise Informationen zu liefern. [3]

Interessant ist darüber hinaus das so genannte erweiterte Lehrbuch, in dem Modelle, Informationen oder Interaktionsmöglichkeiten virtuell mit normalen Lehrbüchern aufgerufen werden, wenn die entsprechenden Seiten geöffnet werden. [18] [19]

Solche Anwendungen und Umfragen im Schulbereich haben gezeigt, dass die Schulen an der Nutzung neuer Technologien in den Bereichen AR und Virtual Reality sehr interessiert sind. [3]

In dem folgenden Abschnitt werden einige besonders für die AR relevante didaktische Konzepte und Unterrichtsstrukturen vorgestellt.

Seit dem Bologna-Prozess ist deutlich geworden, dass die Studierenden als aktive Teilnehmer in einer Lehr-Lernsituation im Zentrum der Aufmerksamkeit stehen sollten. [20] Handlungsorientierung und aktivierende Methoden gehören daher zu den aktuellen Themen der Hochschuldidaktik. [21]

Viele Studien haben gezeigt, dass das Lernen nicht nach einem Sender-Empfänger-Prinzip funktioniert. Es beschränkt sich nicht auf die bloße Wahrnehmung eines Inputs. [22] Die Situation in einer Vorlesung ist jedoch sehr oft durch Frontalunterricht gekennzeichnet, der die Studierenden in eine eher passive "Empfängerrolle" versetzt. [4]

Es ist auch neurodidaktisch erwiesen, dass Learning by doing am erfolgreichsten ist. Studierende lernen also am besten durch aktive Tätigkeiten. [20]

In der Lerntheorie führen diese Aspekte zu einem Konstruktivismus, in dem die Lernenden eine aktive Rolle übernehmen und die Lehrenden als Berater und Unterstützer fungieren. Die Forderung an den didaktischen Entwurf im Konstruktivismus schließt eine Integration in Anwendungskontexte und -situationen ein. [23]

Diese wissenschaftlichen Erkenntnisse sollen in der im Rahmen dieser Projektarbeit entwickelten Anwendung beachtet und entsprechend umgesetzt werden. Indem die Anwender der Applikation nicht nur die fachlichen Inhalte eines Tieres anhand von Text dargestellt bekommen. Sondern die Interaktionen mit dem 3D Modell des Tieres, wie das Drehen und Bewegen betonen das „Tun“ und die Situierung des Lernens ganz besonders, weil dadurch auch Bewegungen im Raum gefordert sind.

Da, wie zuvor dargelegt, ein auf den Lernenden ausgerichteter Unterricht erforderlich ist, muss die Motivation der Studierenden umso stärker gefördert werden. Aus der Perspektive der Motivationsforschung und auf der Grundlage der Theorie der Selbstbestimmung von Deci und Ryan [24] gibt es drei wichtige Punkte, die die intrinsische Motivation der Lernenden beeinflussen:

- Zuwachs von Kompetenzen/Selbstwirksamkeit
- gewisses Maß an Autonomie und Selbstbestimmung
- soziale Eingebundenheit

Um aktiv in einer Gruppe arbeiten zu können, ist eine gemeinsame Wissensbasis und die Verwendung der gleichen Fachbegriffe notwendig. Zudem kann ein Kurs, der rein auf aktiver Teilnahme basiert, die Studierenden auf Dauer ermüden. Ein integratives Konzept sollte die Kombination von konstruktiven und lehrreichen Elementen im Unterricht ermöglichen. [25] [4] [20]

Abschließend sollen die Ergebnisse aus einer empirischen Studie von Jorge Martín-Gutiérrez und Manuel Contero [26] vorgestellt werden. In dieser Studie wurde ein um Augmented-Reality erweitertes Lehrbuch verwendet, um Maschinenbaustudenten auf ansprechende Art und Weise mechanische Standardelemente zu vermitteln.

25 Studenten nutzten dieses Lehrbuch, um die Darstellung und Bezeichnung von mechanischen Standardelementen zu studieren. Eine Kontrollgruppe von 22 Studierenden verwendete traditionelle Unterrichtsunterlagen mit statischen Bildern, um den gleichen Inhalt zu studieren. Die Ergebnisse wurden mit Hilfe eines Evaluationstests und einer Umfrage zur Benutzerfreundlichkeit analysiert. [26]



Abbildung 6: Beispiele für erweiterte virtuelle mechanische Standardelemente. [26]

Um die Auswirkungen von Bildungsinhalten auf die Studierenden zu analysieren, wird ein letzter, mit maximal 10 Punkten bewerteter Test durchgeführt, und die erzielten Ergebnisse erscheinen in Abbildung 7.

	Marks (S.D.)	Std. Error
Exp. Group n = 25	5.84 (1.54)	0.31
Control Group n=22	4.5 (1.84)	0.39

Abbildung 7: Durchschnittliche Bewertung und Standardfehler [26]

Die Autoren legten dabei aus, dass der Test als bestanden gilt, wenn 5 Punkte erreicht sind. In der Gruppe, die mit erweitertem Lehrbuch studierte, gab es nur 5 Studenten (20%), die durchgefallen sind, während in der Gruppe der traditionellen Klassenarbeiten 11 Studenten (50%) den Test nicht bestanden haben. Es wurde ein t-Student Test für unabhängige Stichproben verwendet, um die Durchschnittswerte zu vergleichen, die sowohl von der Experimental- als auch von der Kontrollgruppe erhalten wurden. Die akademischen Ergebnisse sind statistisch signifikant ($t=2,708$, $p=0,009$). Die p-Werte liegen dabei deutlich unter dem kritischen Wert von 1%, was bedeutet, dass mittels erweitertem Lehrbuch statistisch signifikant bessere Ergebnisse erzielt werden können. [26]

Bei der Benutzerfreundlichkeit gibt es verschiedene quantitative Maße, die es ermöglichen, den Grad der Produktakzeptanz zu ermitteln. In dieser Studie wurde die Erfolgsrate des didaktischen Materials und der verwendeten Technologie ermitteln. Wie aus Abbildung 8 ersichtlich, sind Werte für Effektivität, Effizienz und Zufriedenheit dargestellt.

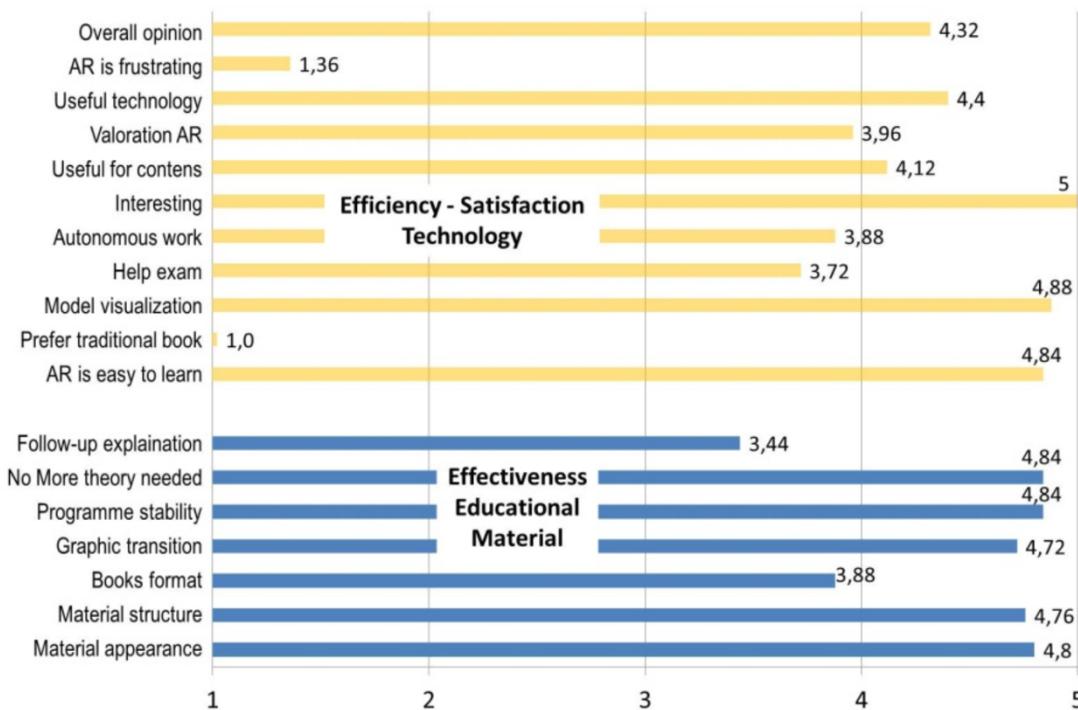


Abbildung 8: Wirksamkeit, Effizienz und Zufriedenheitsergebnisse [26]

Das erweiterte Lehrbuch wurde mit einem Effektivitätswert von 4,47 Punkten positiv bewertet und erreichte damit fast die Bestnote von 5 Punkten. Unter anderem bewerteten die Studierenden, dass das Buch visuell ansprechend ist, der Inhalt angemessen und gut strukturiert ist. Die Buchgröße eignet sich für die korrekte Ausführung notwendiger Gesten bei der Visualisierung erweiterter Informationen. Das erweiterte Lehrbuch wird als ziemlich vollständig in Bezug auf die gelieferten Informationen angesehen, da die Studierenden der Meinung sind, dass es nicht notwendig ist, während des Studiums weitere Dokumente durchzusehen. [26]

In Bezug auf Effizienz und Zufriedenheit werden alle Angelegenheiten positiv bewertet. Jeder Student (100%) ist der Meinung, dass die eingesetzte Technologie für ihn interessant ist, und die meisten von ihnen denken, dass diese Technologie ihnen helfen wird, eine bessere Prüfung abzulegen. Die allgemeine Einschätzung

ist, dass die AR-Technologie und die darin enthaltene erweiterte Lehrbuchbasis ein einfaches Lernen ermöglicht und für den Benutzer zu einem frustrationsfreien Hilfsmittel wird. [26]

Anhand der Daten in Abbildung 9 lässt sich belegen, dass die Versuchsgruppe motivierter ist als die Kontrollgruppe, die lediglich die Strategie verfolgte, die Prüfung zu bestehen. [26]

Tiefen-, Oberflächenmotivation und Oberflächenstrategie zeigen, dass die Versuchsgruppe motivierter ist als die Kontrollgruppe. Diese Gruppe führte die Studie mit mehr Anstrengung und Interesse durch. Diese Ergebnisse deuten darauf hin, dass sich die Versuchsgruppe auf das Lernen konzentriert hat, während die Kontrollgruppe die Datenspeicherung angewandt hat. [26]

Subscales	Mean (SD) Exp. Group (N=25)	Mean (SD) Ctr. Group (N=22)	T-test and P-value	
Deep Motive	3.94 (1.51)	1.76 (0.98)	12.92*	p=.0
Surface Motive	1.40 (0.75)	3.49 (1.28)	-15.45*	p=.0
Deep Strategy	3.09 (1.24)	3.03 (1.31)	.31	p=.75
Surface Strategy	2.55 (1.69)	3.44 (1.22)	-4.59*	p=.0

*, P<.001

Abbildung 9: Statistische Zusammenfassung (t-Test und p-Werte) für Motivation, Strategie und Vorgehensweise [26]

Diese Ergebnisse der Studie zeigen die, vorher theoretisch aufgeführten, Vorteile der Nutzung von Augmented Reality im Bereich des Lernens. Sie zeigen außerdem eine höhere Motivation beim Ausführen von Aufgaben und beim Lernen. Die Verwendung eines mit AR angereicherten Lehrbuchs ermutigt die Studenten zu einem tieferen Studienansatz und zu einer tieferen Veranlagung für das Lernen, wobei der Schwerpunkt auf der Kenntnis des Lehrstoffs und nicht nur auf dem Bestehen des Studienfachs liegt. [26]

3.3 Apples ARKit

Für die Entwicklung der Augmented-Reality-Anwendung wird im Rahmen dieser Projektarbeit das von Apple entwickelte Software Development Kit "ARKit" eingesetzt. Ein Software Development Kit ist ein Toolkit für Software-Entwickler, das einen einfachen Einstieg in ein bestimmtes Betriebssystem oder eine bestimmte Programmiersprache bietet. Ein SDK enthält alle Informationen und Dokumentationen sowie Werkzeuge, die die Softwareentwicklung erleichtern. [27]

Nachfolgend werden die für das Verständnis der Arbeit notwendigen theoretischen Grundlagen von ARKit erläutert.

ARKit wurde von Apple auf der Entwicklerkonferenz WWDC 2017 angekündigt und später als Teil des Betriebssystems iOS 11 veröffentlicht. ARKit ermöglicht es App-Entwicklern, AR-Funktionalitäten schnell und einfach in ihre Apps und Spiele zu integrieren. ARKit nutzt die Kamera, Prozessoren und Bewegungssensoren des iOS-Gerätes, um immersive Interaktionen zu erzeugen.

ARKit kann in drei Ebenen unterteilt werden, siehe Abbildung 10. Diese Ebenen arbeiten simultan miteinander.

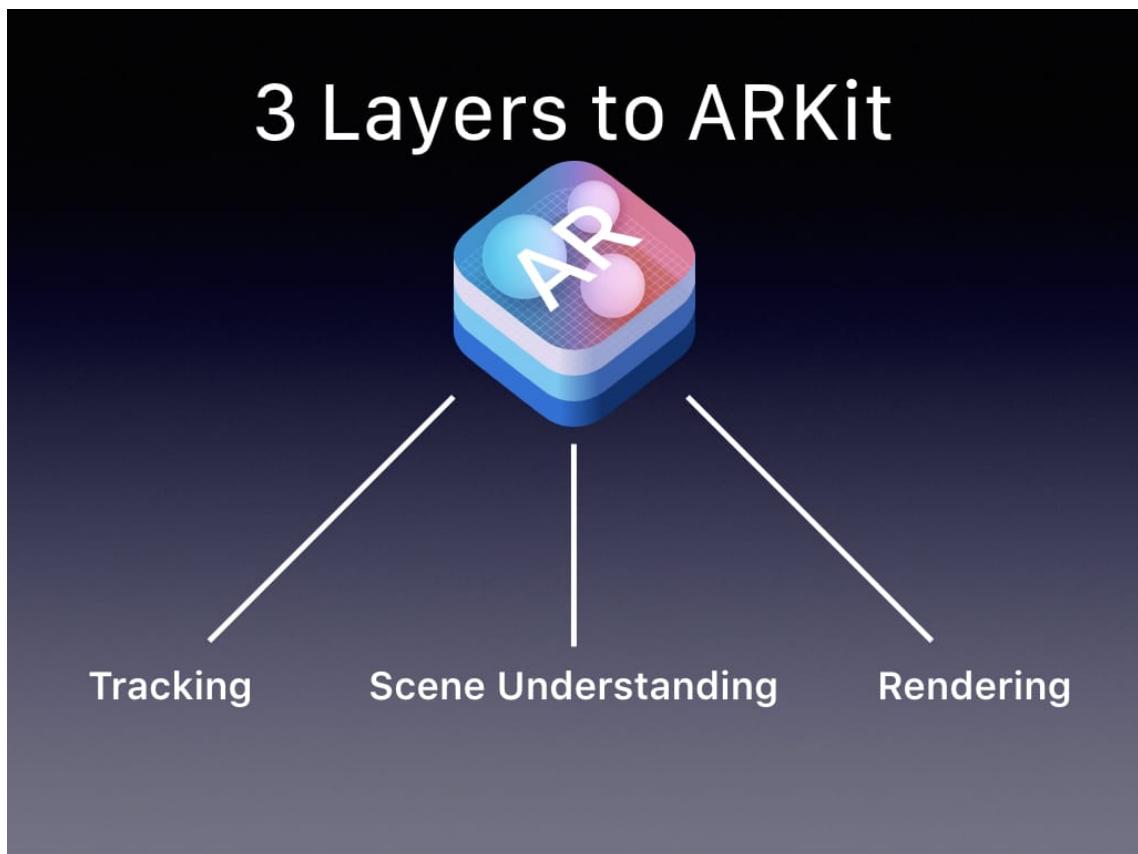


Abbildung 10: Drei Ebenen von ARKit [28]

Das Tracking ist die Schlüsselfunktion von ARKit. Es ermöglicht es, die Position, den Standort und die Ausrichtung eines Geräts in der realen Welt und live zu verfolgen. Das Verstehen der Szene bedeutet, dass ARKit die durch den Blick der Kamera dargestellte Umgebung analysiert und dann die Szene anpasst oder Informationen über sie liefert. Dies ermöglicht die Erkennung aller Oberflächen in der physischen Welt, wie z.B. des Bodens oder einer flachen Oberfläche. Dann erlaubt es uns, ein virtuelles Objekt darauf zu platzieren. Auch die Lichtschätzung kann integriert werden, um ein virtuelles Objekt zu beleuchten, das eine Lichtquelle in der physischen Welt simuliert. Das Rendering befasst sich mit der visuellen Ausgabe der erfassten Daten auf dem entsprechenden Bildschirm. In diesem Schritt werden die realen Objekte mit den virtuellen kombiniert und zu einer erweiterten Realität zusammengefügt. [28]

Dadurch ergibt sich die, in Abbildung 11, vereinfacht dargestellte Struktur der App.

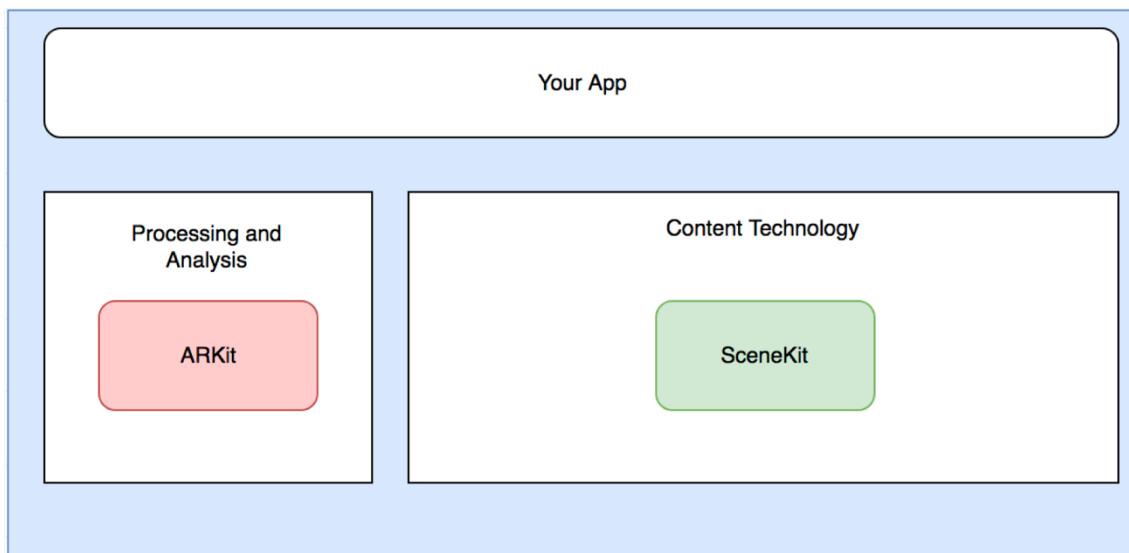


Abbildung 11: Vereinfachte Struktur der App

ARKit abstrahiert den gesamten komplexen Prozess der Erkennung von Elementen in der realen Welt innerhalb des Kamerafeeds und der relativen Positionierung der Benutzer, während ein anderes Framework namens SceneKit die Integration von 3D-Elementen übernimmt.

4 Beschreibung des Projektverlaufs

Für die Zielerreichung wurden drei verschiedene Projektstufen formuliert: Die erste Stufe der Konzeption, die zweite Stufe der Umsetzung und die dritte Stufe der Evaluation bzw. dem Transfer. Die Projektstufen charakterisieren Ziele, Ablauf und Methode der Projektdurchführung und sollen im Folgenden genauer erläutert werden.

4.1 Konzeption

Die Planung und Realisierung der Augmented Reality Anwendung im Bereich des Lernens begann im Februar 2020. Der hier folgend beschriebene Projektverlauf seit der ersten Evaluierungsphase kann in Konzeptions-, Umsetzungsphase, sowie Inbetriebnahme unterteilt werden.

Die erste Phase der Konzeption wurde zu Projektbeginn durchgeführt und diente der Erarbeitung eines Orientierungsrahmens. Sie wurde als allgemeine Leitidee vor der Umsetzung der Projektziele verstanden.

Die Konzeption enthält übergeordnete Ziele für die Strukturierung des Projekts und konzentrierte sich vor allem auf die Entwicklung eines Entwurfs für eine iOS-Anwendung, um Kindern das Lernen durch Augmented Reality zu ermöglichen, basierend auf den Ergebnissen der empirischen Studien von Jorge Martín-Gutiérrez und Manuel Contero [26].

Der Schwerpunkt lag auf der Entwicklung, Analyse und Bewertung durch vorhandene Technologien, die für den Einsatz von AR in diesem Bereich geeignet sind. Im Zentrum des Konzepts stand die Entwicklung eines Implementierungskonzepts für eine AR-Lernapplikation, die Kindern Kenntnisse über unterschiedliche Tierarten vermitteln soll.

Um diese Anwendung zu konkretisieren, wurden fünf wesentliche Schritte definiert, nach denen die Umsetzung erfolgen sollte:

1. Die Applikation soll definierte Bilder einer Tierart erkennen und zusätzliche Informationen zu dieser Tierart um das Bild herum in AR darstellen;
2. Die zusätzlichen Informationen sollen aus einem kurzen Text, einem zugehörigen Bild und einem 3D Modell des Tieres bestehen;

3. Mit dem 3D Modell des Tieres soll der Anwender in der erweiterten Realität interagieren können. Zu den geplanten Interaktionen gehören bewegen, drehen und vergrößern/verkleinern des Modells;
4. Die zusätzlichen Informationen sollen sich vom Anwender durch Interaktion mit der Darstellung ändern;
5. Die Umsetzung soll ausschließlich für das iOS Betriebssystem mit Hilfe der Programmiersprache Swift und dem Framework ARKit realisiert werden.

Abbildung 12 zeigt eine, zu Beginn der Konzeptionsphase entstandene, vereinfachte Skizze der Projektidee.

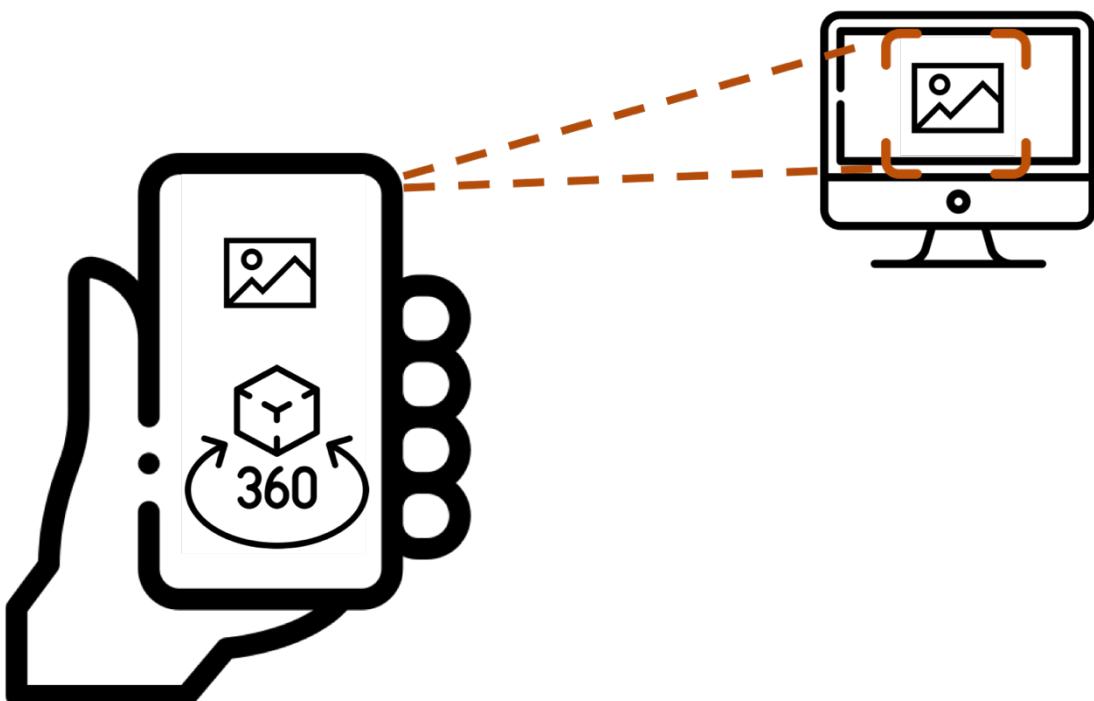


Abbildung 12: Skizze der Projektidee

Die Konzeption der Anwendung berücksichtigt die, in Kapitel 3.2 beschriebenen, Potenziale von Augmented Reality im Bereich des Lernens. So kann die Anwendung auf jedem mobilen Endgerät mit dem iOS Betriebssystem ausgeführt werden, was nach dem BYOD-Prinzip einer großen Anzahl an Schülern den Zugang ermöglicht. Außerdem ist der Zugang zu diesen Lernmitteln für die Schüler von überall aus möglich. Es spielt dabei keine Rolle, ob das Referenzbild, das die AR-Objekte erscheinen lässt, in analoger oder digitaler Form vorliegt.

Die Anwendung würde sich sehr einfach als ein erweitertes Lehrbuch einsetzen lassen. Damit könnten die theoretischen Darstellungen des Buches durch die 3D

Modelle in der erweiterten Realität angereichert und so für die Schüler leichter verständlich gemacht werden.

Die Motivation und das Engagement der Schüler wären durch so eine interessante und neuartige Technologie zusätzlich gefördert. Durch die aktive Tätigkeit dieser Art des Lernens, würde das erwiesenermaßen erfolgreiche Learning by Doing Prinzip zum Einsatz kommen und somit für einen besseren Lernerfolg der Schüler sorgen.

4.2 Realisierung

Die zweite Stufe der Umsetzung knüpft an die erste Stufe der Konzeption an und konkretisiert diese.

Die Umsetzung des Konzepts einer AR Anwendung im Bereich des Lernens orientierte sich an den fünf Schritten, die in Kapitel 4.1 erläutert werden. Vor der Programmierung des Projekts fand eine längere Recherche phase statt. In der Recherche Phase wurden die benötigten 3D Modelle eines Tieres und die zugehörigen Informationen über das Tier ermittelt.

Ein passendes 3D Modell eines Tieres zu finden, gestaltete sich schwieriger als erwartet. Zunächst war geplant, ein 3D Modell zu verwenden, das eine Animation ausführen kann. Allerdings wurde diese Planung nach langer Recherche verworfen, da solche Modelle ab einem Kaufpreis von 50 € beginnen. Aber auch nicht animierte 3D Modelle von Tieren sind kostenlos meistens nur in schlechter Qualität vorhanden. Ein weiteres Problem bei der Arbeit mit 3D Modellen ist die Vielzahl an Dateitypen. Die gängigsten Dateitypen sind unter anderem STL, SCN, OBJ, FBX, DAE, COLLADA, 3DS, IGES, STEP und VRML/X3D. Von den genannten Dateitypen unterstützt ARKit allerdings nur SCN, DAE und OBJ. Für die Verwendung des finalen 3D Modells, war zusätzlich die Installation der 3D-Erstellungs-Software „Blender“ notwendig, um die Datei in einen unterstützten Dateityp zu konvertieren.

Nachdem ein ansprechendes 3D Modell eines Elefanten gefunden und in das unterstützte Format SCN umgewandelt wurde, konnten weitere Informationen zu dem Tier recherchiert werden. Die gefundenen Informationen zu dem afrikanischen Elefanten werden in der App später neben dem 3D Modell als Infotafel dargestellt.

Die Programmierung der App wird im Folgenden anhand der wichtigsten Codeabschnitte und dem zugehörigen Ergebnis in der App erläutert.

Zu Beginn findet die Initialisierung der AR App statt, zu sehen in Abbildung 13. Das passiert bei der iOS App Entwicklung üblicherweise in der Funktion `viewWillAppear()`. Diese Funktion wird automatisch ausgeführt, sobald der entsprechende „View“ (oder auch Screen) dem Anwender angezeigt wird. Hier wird in Zeile 70 der Anwendung angegeben, welche Bilder als Referenzbilder verwendet werden sollen. Das funktioniert mit der Funktion `ARReferenceImage.referenceImages()`, die auf alle hinterlegten Referenzbilder in der angegebenen AR-Ressourcengruppe „AR Resources“ des Xcode-Projekts zugreift. Mit dem Aufruf in Zeile 80 `sceneView.session.run()` wird die AR-Session mit allen angegeben Konfigurationen gestartet.

```
66    // Notifies the view controller that its view is about to be added to a view hierarchy.
67    override func viewWillAppear(_ animated: Bool) {
68        super.viewWillAppear(animated)
69
70        guard let refImages = ARReferenceImage.referenceImages(inGroupNamed: "AR Resources", bundle: Bundle.main) else {
71            fatalError("Missing expected asset catalog resources.")
72        }
73
74        // Create a session configuration
75        let configuration = ARImageTrackingConfiguration()
76        configuration.trackingImages = refImages
77        configuration.maximumNumberOfTrackedImages = 1
78
79        // Run the view's session
80        sceneView.session.run(configuration, options: ARSession.RunOptions(arrayLiteral: [.resetTracking, .removeExistingAnchors]))
81    }
```

Abbildung 13: Code für die Initialisierung der AR-App

Abbildung 14 zeigt die Programmierung der Bilderkennung mit Hilfe von ARKit. Die Funktion `highlightDetection()` wird aufgerufen, wenn ein Bild aus dem AR Resources Ordner von der App erkannt wird. Nach erfolgreicher Bilderkennung der App wird eine Animation ausgeführt, die dem Anwender zeigen soll, dass das System das Bild erfolgreich erkannt wurde.

```

432     func highlightDetection(on rootNode: SCNNode, width: CGFloat, height: CGFloat, completionHandler block: @escaping (() -> Void)) {
433         let planeNode = SCNNode(geometry: SCNPlane(width: width, height: height))
434         planeNode.geometry?.firstMaterial?.diffuse.contents = UIColor.white
435         planeNode.position.z += 0.1
436         planeNode.opacity = 0
437
438         rootNode.addChildNode(planeNode)
439         planeNode.runAction(self.imageHighlightAction) {
440             block()
441         }
442     }
443
444     var imageHighlightAction: SCNAction {
445         return sequence([
446             .wait(duration: 0.25),
447             .fadeOpacity(to: 0.85, duration: 0.25),
448             .fadeOpacity(to: 0.15, duration: 0.25),
449             .fadeOpacity(to: 0.85, duration: 0.25),
450             .fadeOut(duration: 0.5),
451             .removeFromParentNode()
452         ])
453     }
454 }

```

Abbildung 14: Code für die Bilderkennung mit Hilfe von ARKit

Die Animation wird in Zeile 439 durch `planeNode.runAction(self.imageHighlightAction)` aufgerufen. Die Definition der Animation ist in Zeile 446 bis 451 definiert. Es wird ein weißes Feld über das Bild gelegt und die Sättigung variiert, so dass für den Anwender ein Blinken entsteht. Nach etwa 1,5 Sekunden verschwindet die Animation. Das Ergebnis der Animation ist in Abbildung 15 zu sehen.

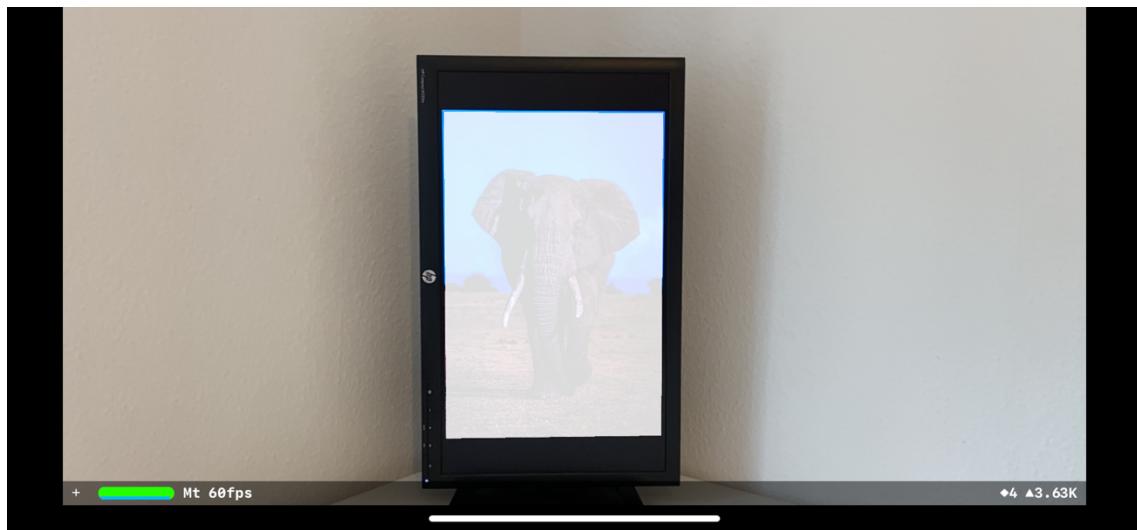


Abbildung 15: Ergebnis der Bilderkennung durch ARKit

Bei dieser, für die Anwendung essentiellen, Funktion traten im Laufe der Entwicklung Schwierigkeiten auf. Da das Referenzbild der Anwendung als Ankerpunkt für alle 3D Modelle der erweiterten Realität verwendet wird, hat die Position und die Größe des Bildes massiven Einfluss auf die Anwendung. Leider kommt es durch die Auswirkungen von Mikrobewegungen des Anwenders zu kleineren Ungenauigkeiten in der Bilderkennung. Das führt dazu, dass sich die Position des

Referenzbild häufig minimal ändert. Dadurch ändern sich auch die Positionen der 3D Modelle und sie stehen nicht ruhig im Raum, sondern wackeln leicht. Um diese Störung möglichst gering zu halten, wurden mehrere Schritte vorgenommen. Leider gibt es auf Seiten von Apple noch keinen Stabilisierungsalgorithmus, wie etwa bei der Fotografie. Eine weitere Gegenmaßnahme war die Verwendung eines sehr farbigen Referenzbildes, da somit ein starker Kontrast zur direkten Umgebung erzeugt werden konnte und sich das Bild dadurch genauer erkennen lässt.

Ende August stellte Apple eine Beta Version des im September veröffentlichten Betriebssystem iOS 14 für alle lizenzierten Apple Entwickler bereit. Aber auch mit dieser Betriebssystemversion wurde keine Möglichkeit gefunden das Objekt zu stabilisieren.

Das beste Ergebnis konnte schließlich damit erzielt werden, dass das Tierobjekt nicht das erkannte Referenzbild als Ankerpunkt verwendet. Dem Tierobjekt wird, anders wie bei den restlichen Objekten, ein eigener Referenzpunkt zur Ausrichtung zugewiesen. Dieser wird zwar auch über das Referenzbild ermittelt, allerdings besteht danach keine Verknüpfung. So haben die Bewegungen des Anwenders weniger Auswirkungen auf das Tiermodell. Ein Ergebnis wobei das Tier ohne kleine Verwacklungen im Raum liegt, wird allerdings nur mit einem Bildstabilisierungsalgorithmus zu erreichen sein.

Nachdem das Referenzbild von der Anwendung erkannt wird, beginnt diese mit der Darstellung von Objekten um das Bild. Beispielhaft wird hierfür die Funktion `displayDetailView()` aus Abbildung 16 aufgeführt, womit die Infotafel und der Pfeil aus Abbildung 17 und Abbildung 18 dargestellt werden.

```
350     func displayDetailView(on rootNode: SCNNode, xOffset: CGFloat) {
351         let detailPlane = SCNPlane(width: xOffset, height: xOffset * 1.4)
352         detailPlane.cornerRadius = 0.25
353
354         let detailNode = SCNNode(geometry: detailPlane)
355         detailNode.geometry?.firstMaterial?.diffuse.contents = SKScene(fileNamed: "Elefant_01")
356         detailNode.name = "detailView_left"
357
358         // Due to the origin of the iOS coordinate system, SCNMaterial's content appears upside down, so flip the y-axis.
359         detailNode.geometry?.firstMaterial?.diffuse.contentsTransform = SCNMatrix4Translate(SCNMatrix4MakeScale(1, -1, 1), 0, 1, 0)
360         detailNode.position.z -= 0.5
361         detailNode.opacity = 0
362
363         rootNode.addChildNode(detailNode)
364         detailNode.runAction(.sequence([
365             .wait(duration: 1.0),
366             .fadeOpacity(to: 1.0, duration: 1.5),
367             .moveBy(x: xOffset * -1.1, y: 0, z: -0.1, duration: 1.5),
368         ])
369     )
370
371     let arrowSceneLeft = SCNScene(named: "arrow_Left.dae")
372     guard let arrowNodeLeft = arrowSceneLeft?.rootNode.childNodes(withName: "arrowLeft_modal", recursively: true) else {
373         fatalError("ERROR")
374     }
375
376     arrowNodeLeft.opacity = 0
377     arrowNodeLeft.position.z -= 0.5
378     arrowNodeLeft.name = "left"
379
380     rootNode.addChildNode(arrowNodeLeft)
381     arrowNodeLeft.runAction(.sequence([
382         .wait(duration: 1.0),
383         .fadeOpacity(to: 1.0, duration: 1.5),
384         .moveBy(x: xOffset * -2.1, y: 0, z: -0.1, duration: 1.5),
385     ])
386     )
387 }
388 }
```

Abbildung 16: Code für die Darstellung der AR-Infotafel

Für die Infotafel wird eine `SCNPlane()` mit der Höhe und Breite des Referenzbildes in Zeile 351 initialisiert. Als Oberfläche wird die in Abbildung 18 „Elefant_01“ Datei definiert und mit der Funktion `runAction()` eine Animation für das Erscheinen der Tafel festgelegt.

Der Pfeil, der für das Hin- und Herwechseln zwischen den Tafeln verantwortlich ist, wird in Zeile 371 – 385 initialisiert. Er wird links von der Infotafel positioniert und synchron mit einer Animation eingeblendet.

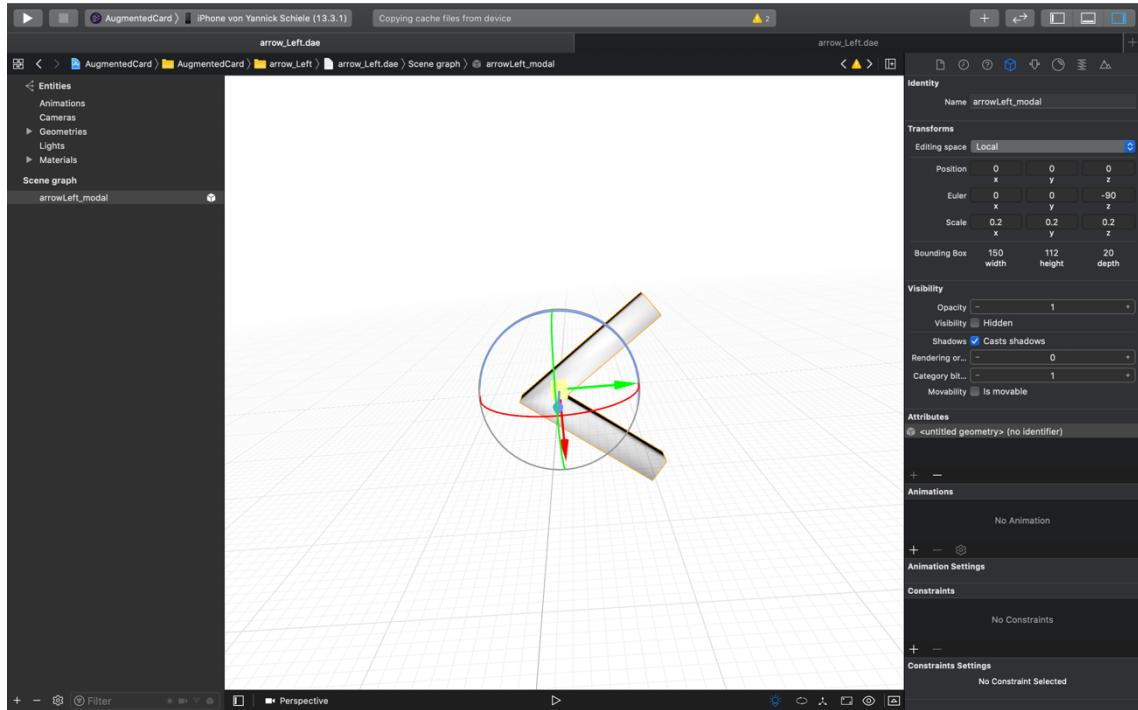


Abbildung 17: 3D-Pfeil für die Interaktion mit den Infotafeln

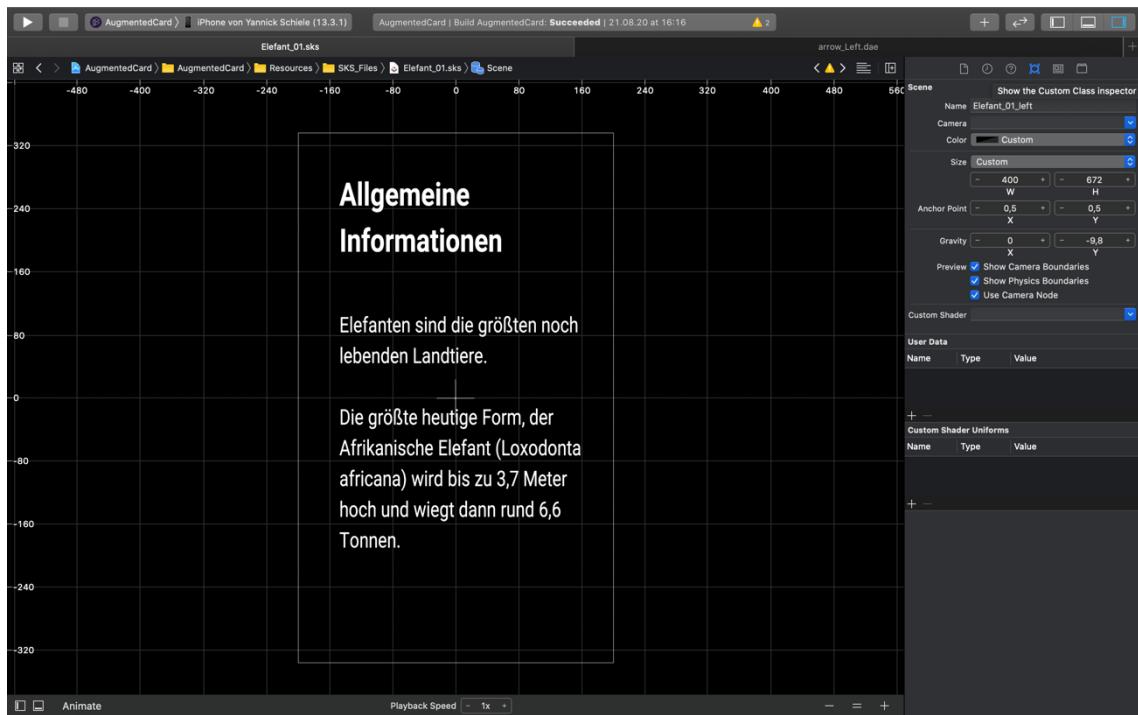


Abbildung 18: AR-Infotafel, die neben dem Tier Objekt dargestellt wird

Als nächstes folgt das Einsetzen des 3D-Tiermodells. Abbildung 19 zeigt das Modell im Editor der verwendeten Entwicklungsumgebung. Hier können verschiedene Eigenschaften des Modells festgelegt werden.

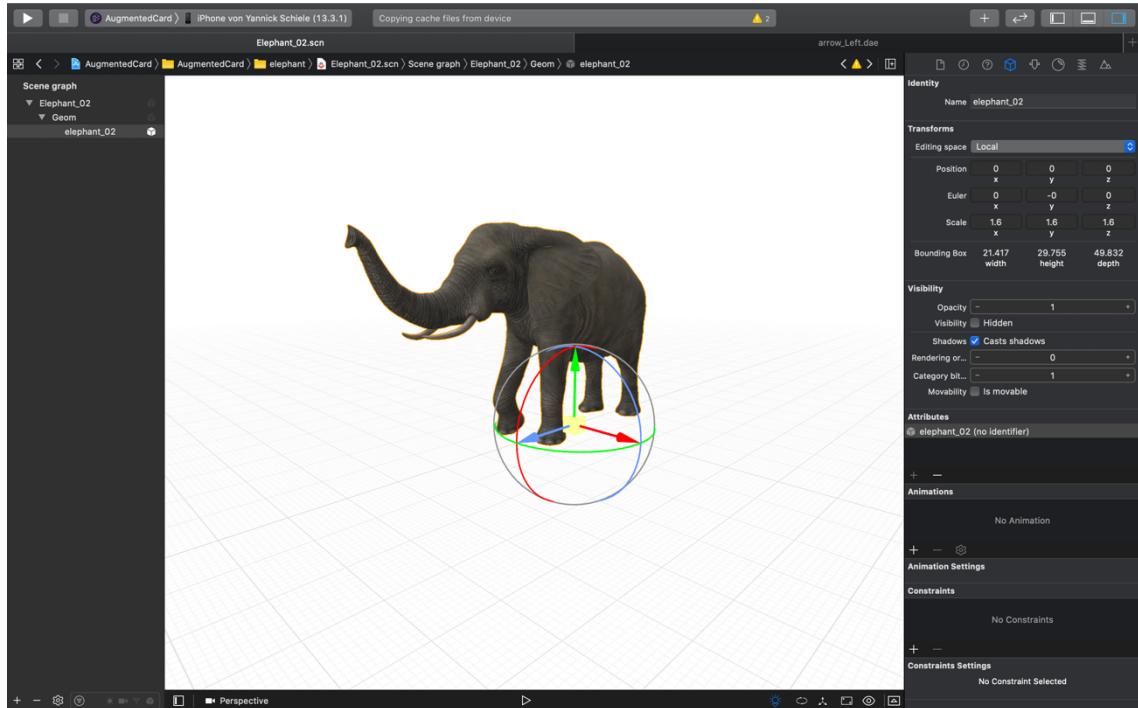


Abbildung 19: 3D-Modell des Tieres im xCode Editor

Die Größe und Positionierung werden allerdings, wie bei allen Objekten, im Code ausgeführt, siehe Abbildung 20. In Zeile 331 bis 333 wird die SCN Datei aus Abbildung 19 an die Variable `elephantNode` übergeben. Dadurch kann mit dem Aufruf `elephantNode.position()` die genaue Position des Objektes festgelegt werden. Auch das Tierobjekt wird durch eine Animation eingeblendet und dadurch ergibt sich die Gesamterscheinung aus Abbildung 21.

```

328 func showAnimalModal(on rootNode: SCNNODE, xOffset: CGFLOAT) {
329
330    //1. Modell
331    let elephantScene = SCNScene(named: "Elephant_02.scn")
332    guard let elephantNode = elephantScene?.rootNode.childNode(withName: "elephant_02", recursively: true) else {
333        fatalError("ERROR")
334    }
335
336    elephantNode.position = SCNVector3(rootNode.position.x , rootNode.position.y - 30 , rootNode.position.z + 50)
337    //elephantNode.opacity = 0
338    rootNode.addChildNode(elephantNode)
339    elephantNode.runAction(.sequence([
340        .wait(duration: 1.0),
341        .fadeOpacity(to: 1.0, duration: 2),
342        //.move(to: SCNVector3(rootNode.position.x , rootNode.position.y, rootNode.position.z + 8), duration: 1.7)
343    ])
344    )
345
346    showSlider()
347    elephant = elephantNode
348 }
```

Abbildung 20: Code für die Darstellung des 3D Tiermodells

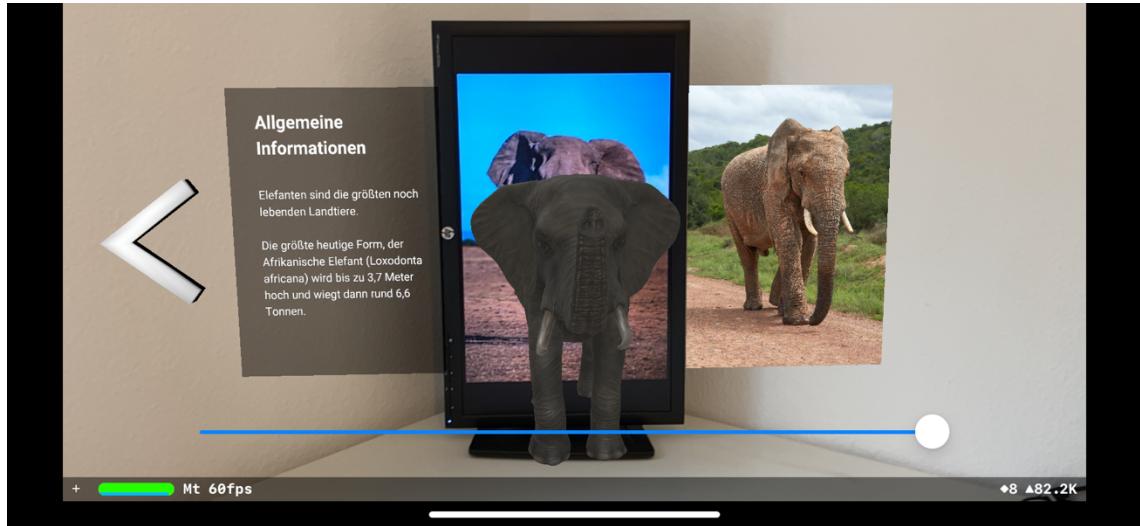


Abbildung 21: Gesamterscheinung der Anwendung

Hier ist das gesamte User Interface der finalen iOS App zu sehen. Links und rechts des Referenzbildes sind Tafeln positioniert. Die linke Tafel gibt in Textform Informationen zum Tier wieder, während die rechte Tafel ein Bild zu den Informationen repräsentiert. Hier ist bereits zu erkennen, dass die Pfeile nur dargestellt werden, wenn in diese Richtung mit den Informationen interagiert werden kann. Ansonsten sind die Pfeile, wie hier bspw. zu sehen der rechte Pfeil, ausgeblendet. Die Funktionalität des Reglers wird später in diesem Kapitel erläutert.

Eine wichtige Funktionalität, um den Learning by doing Ansatz zu ermöglichen ist die Interaktion des Anwenders mit dem Tiermodell. In Abbildung 22 werden die verschiedenen Interaktionsmöglichkeiten mit der Funktion `setupGestures()` initialisiert.

```

92     private func setupGestures() {
93
94         let panGesture = UIPanGestureRecognizer(target: self, action: #selector(rotateObject(_:)))
95         sceneView.addGestureRecognizer(panGesture)
96
97         let tapGesture = UITapGestureRecognizer(target: self, action: #selector(didTap(_:)))
98         tapGesture.numberOfTapsRequired = 1
99         sceneView.addGestureRecognizer(tapGesture)
100
101        let pinchGestureRecognizer = UIPinchGestureRecognizer(target: self, action: #selector(handlePinch(_:)))
102        sceneView.addGestureRecognizer(pinchGestureRecognizer)
103
104    }

```

Abbildung 22: Code für die Initialisierung der Interaktionsmöglichkeiten

Die Anwendung bietet die Möglichkeit zur Interaktion, durch ziehende (`UIPanGestureRecognizer`), berührende (`UITapGestureRecognizer`) und zoomenden (`UIPinchGestureRecognizer`) Gesten. Abbildung 23 gibt eine Übersicht zu den verwendeten Gesten.

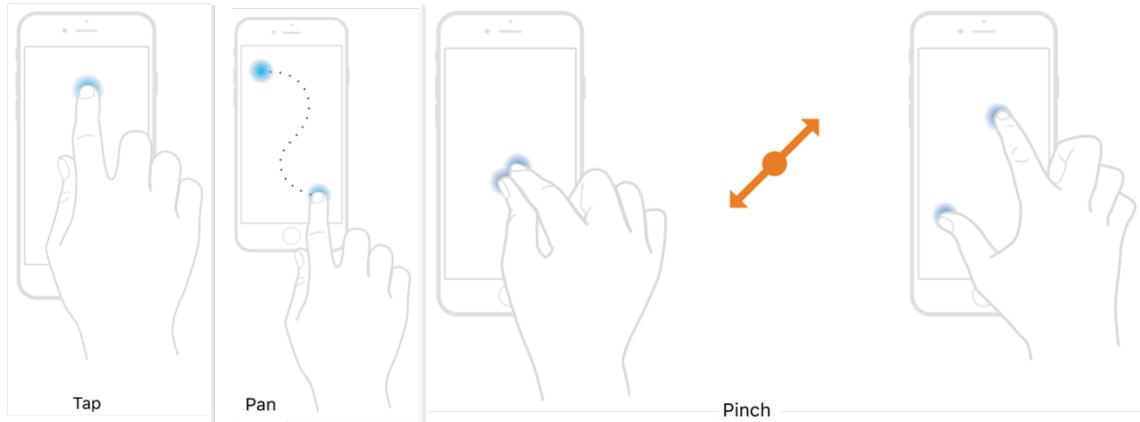


Abbildung 23: Übersicht der Interaktionsmöglichkeiten [29]

Die ziehende Geste führt die Funktion `rotateObject()` aus. Die in Abbildung 24 dargestellte Funktion unterscheidet mit wie vielen Fingern der Anwender den Screen berührt.

```

205    @objc func rotateObject(_ gesture: UIPanGestureRecognizer) {
206
207        guard let nodeToRotate = elephant else {
208            return
209        }
210
211        if gesture.numberOfTouches == 2 {          //Rotation
212            let translation = gesture.translation(in: gesture.view!)
213            var newAngleY = (Float)(translation.x)*(Float)(Double.pi)/180.0
214            newAngleY += currentAngleY
215            nodeToRotate.eulerAngles.y = newAngleY
216            if(gesture.state == .ended) {
217                currentAngleY = newAngleY
218            }
219        } else if gesture.numberOfTouches == 1 {      //Bewegen
220
221            guard let view = sceneView else { return }
222            let location = gesture.location(in: self.sceneView)
223            switch gesture.state {
224            case .began:
225                guard let hitNodeResult = view.hitTest(location, options: nil).first else { return }
226                lastPanLocation = hitNodeResult.worldCoordinates
227                panStartZ = CGFloat(view.projectPoint(lastPanLocation!).z)
228
229            case .changed:
230                guard panStartZ != nil else { return }
231
232                let worldTouchPosition = view.unprojectPoint(SCNVector3(location.x, location.y, panStartZ!))
233
234                let movementVector = SCNVector3(worldTouchPosition.x - lastPanLocation!.x,
235                                                worldTouchPosition.y - lastPanLocation!.y,
236                                                worldTouchPosition.z - lastPanLocation!.z)
237                nodeToRotate.localTranslate(by: movementVector)
238
239                self.lastPanLocation = worldTouchPosition
240            case .ended:
241                panStartZ = nil
242
243            default:
244                break
245        }

```

Abbildung 24: Code für die Dreh-Interaktion des 3D Modells

Verwendet der Anwender zwei Finger, dann rotiert die App das Tierobjekt je nachdem wie der Anwender seine Finger bewegt (Zeile 212 – 218). Abbildung 26 zeigt das Ergebnis einer Dreh Interaktion.

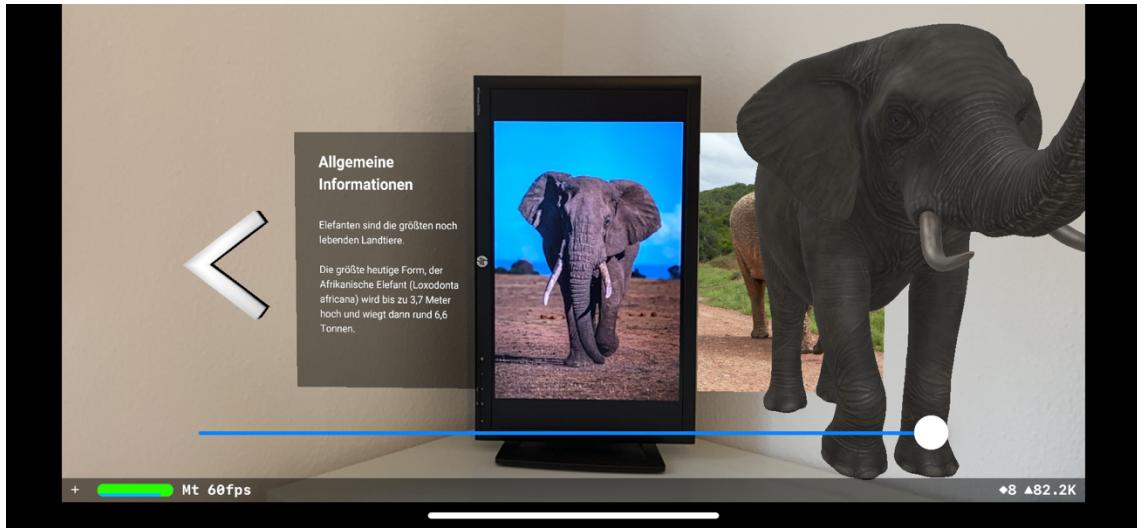


Abbildung 25: Ergebnis der Beweg-Interaktion

Setzt der Anwender nur einen Finger ein, so kann er das Tier mit seinem Finger in alle Richtungen bewegen. Der dafür benötigte Code ist in Abbildung 24 von Zeile 219 bis 244 zu finden. Hier müssen weitere Unterscheidungen stattfinden. Da nur das Tiermodell bewegt werden soll, darf bei der Berührung eines anderen Objektes keinerlei Veränderung auftreten.

Auch in welcher Phase der Interaktion sich der Nutzer befindet, ist eine notwendige Information für die Anwendung, da sich das Modell nur bewegen soll, so lange die Interaktion stattfindet. Vor und nach dem Bewegen durch den Nutzer soll das Tier so ruhig wie möglich an der entsprechenden Position stehen.

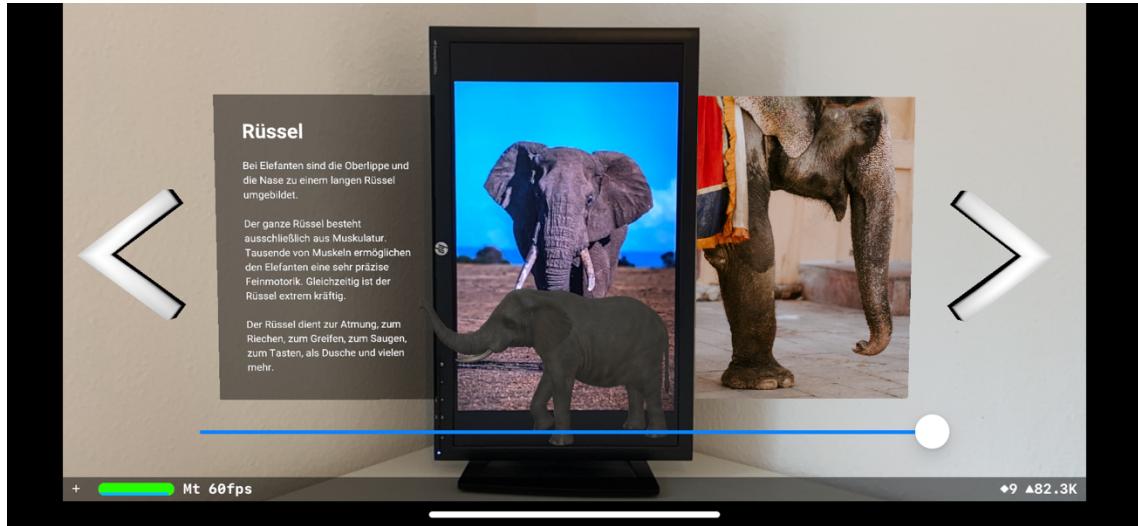


Abbildung 26: Ergebnis der Dreh-Interaktion

Als dritte Interaktionsmöglichkeit kann das Modell gezoomt, also vergrößert oder verkleinert werden. Die Funktion `handlePinch()` aus Abbildung 27 verwendet `UIPanGestureRecognizer` um die Größe des Modells der Geste des Anwenders entsprechend anzupassen.

```

252     // Zoom Gesture
253     @objc func handlePinch(_ gesture: UIPinchGestureRecognizer){
254         guard let nodeToRotate = elephant else { return }
255
256         if (gesture.state == .changed) {
257             let pinchScaleX = Float(gesture.scale) * nodeToRotate.scale.x
258             let pinchScaleY = Float(gesture.scale) * nodeToRotate.scale.y
259             let pinchScaleZ = Float(gesture.scale) * nodeToRotate.scale.z
260             nodeToRotate.scale = SCNVector3(pinchScaleX, pinchScaleY, pinchScaleZ)
261             gesture.scale=1
262         }
263     }
264 }
```

Abbildung 27: Code für die Zoom Interaktion

Abbildung 28 zeigt das Ergebnis, nachdem der Anwender das Tier Objekt vergrößert hat.

All diese Interaktionen lassen sich auch nacheinander ausführen. So kann der Anwender Größe, Ausrichtung und Position des Modells nach seinen Wünschen anpassen.

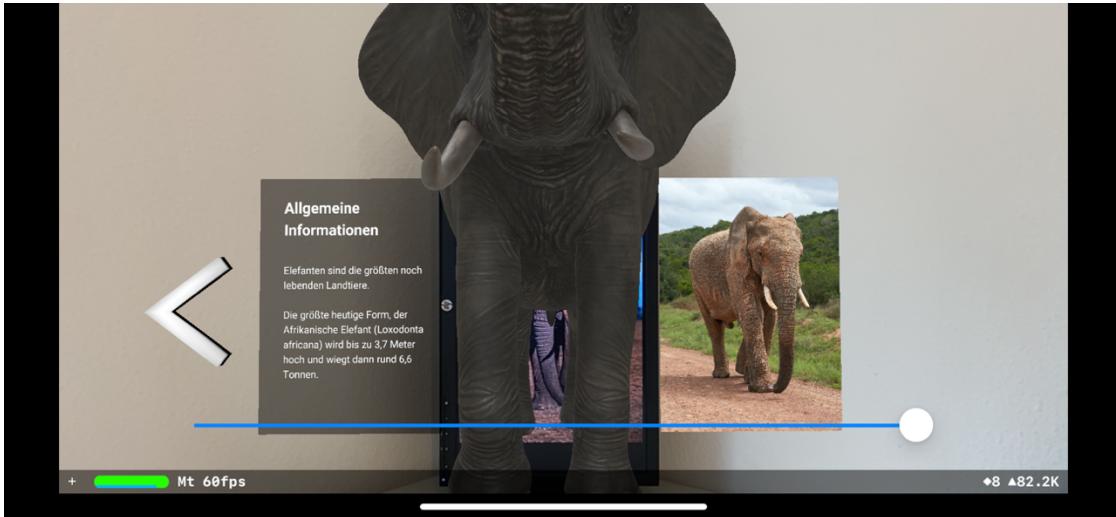


Abbildung 28: Ergebnis der Zoom Interaktion

Zusätzlich zu der Interaktion mit dem Tier Modell ermöglicht die App auch Interaktionen mit den Informations-Tafeln. Die Funktion `didTap()` in Abbildung 29 nutzt `UITapGestureRecognizer` um eine Berührung der Pfeile durch die Anwender zu ermitteln.

```

111    @objc func didTap(_ gesture: UITapGestureRecognizer){
112
113        //1. Get The Current Touch Location In The View
114        let currentTouchLocation = gesture.location(in: self.sceneView)
115
116        //2. Perform An SCNHitTest To Determine If We Have Hit An SCNNode
117        guard let hitTestNode = self.sceneView.hitTest(currentTouchLocation, options: nil).first?.node else { return }
118
119
120        if hitTestNode.name == "left"{ //TODO: Feedback
121
122            if counter < (arrayImagesLeft.count - 1) {
123                SCNTransaction.begin()
124                SCNTransaction.commit()
125                let actionSmall = SCNAction.scale(to: 0.1, duration: 0.2)
126                let actionBig = SCNAction.scale(to: 0.4, duration: 0.4)
127                let actionNormal = SCNAction.scale(to: 0.2, duration: 0.2)
128                let action = SCNAction.sequence([actionSmall,actionBig, actionNormal])
129                hitTestNode.runAction(action)
130
131                let detailView = self.sceneView.scene.rootNode.childNode(withName: "detailView_left", recursively: true)
132                detailView!.geometry?.firstMaterial?.diffuse.contents = SKScene(fileNamed: arrayImagesLeft[counter + 1])
133
134                let detailView_right = self.sceneView.scene.rootNode.childNode(withName: "detailView_right", recursively: true)
135                detailView_right!.geometry?.firstMaterial?.diffuse.contents = SKScene(fileNamed: arrayImagesRight[counter + 1])
136
137                counter = counter + 1
138
139        }

```

Abbildung 29: Code für die Interaktion mit den Pfeil Objekten

Berührt der Anwender eins der Pfeil Objekte, so ändert sich sowohl der Text links als auch das Bild rechts des Tier Objektes. Außerdem wird eine Animation ausgeführt, um dem Anwender ein Feedback zu geben, dass die Interaktion erfolgreich ausgeführt wurde. Abbildung 30 zeigt einen Ausschnitt dieser Animation, bei der sich die Größe des Pfeils ändert.

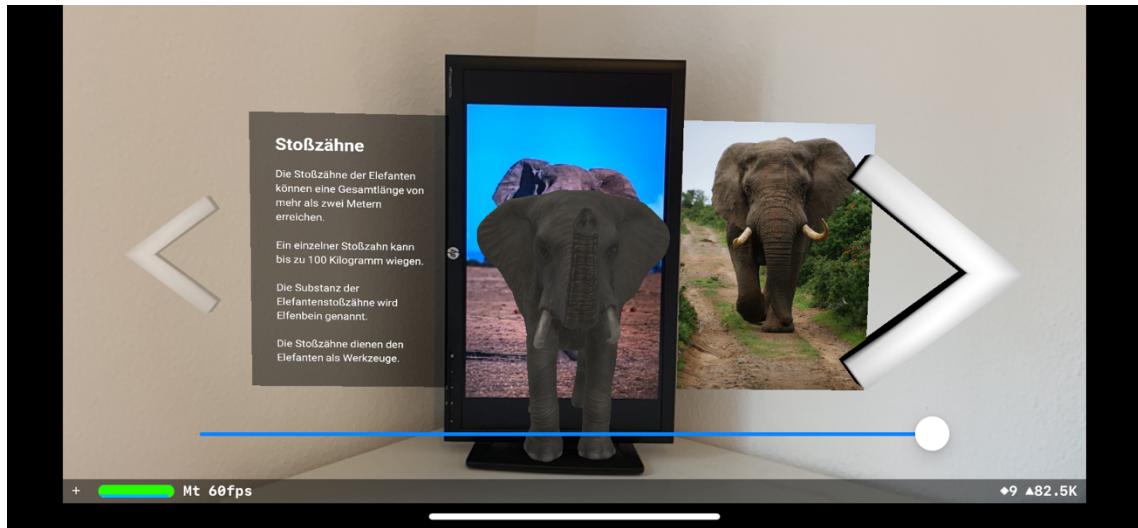


Abbildung 30: Ergebnis der Interaktion mit den Pfeil Objekten

Als letzte wichtige Funktionalität ist die Möglichkeit zu Änderung der Transparenz des vorhanden Tiermodells. Dieser Wert kann mit Hilfe des Reglers durch den Anwender eingestellt werden. Die Funktion `sliderForAlpha()` nimmt dafür den aktuellen Wert des Reglers und verwendet diesen als Transparenzfaktor des Modells, siehe Abbildung 31. Abbildung 32 zeigt das veränderte Tiermodell mit einem Transparenzfaktor von 50%.

```
○      @IBAction func sliderForAlpha(_ sender: UISlider) {
62          elephant!.opacity = CGFloat(sender.value)
63      }
64
```

Abbildung 31: Code für die Änderung der Transparenz des Modells

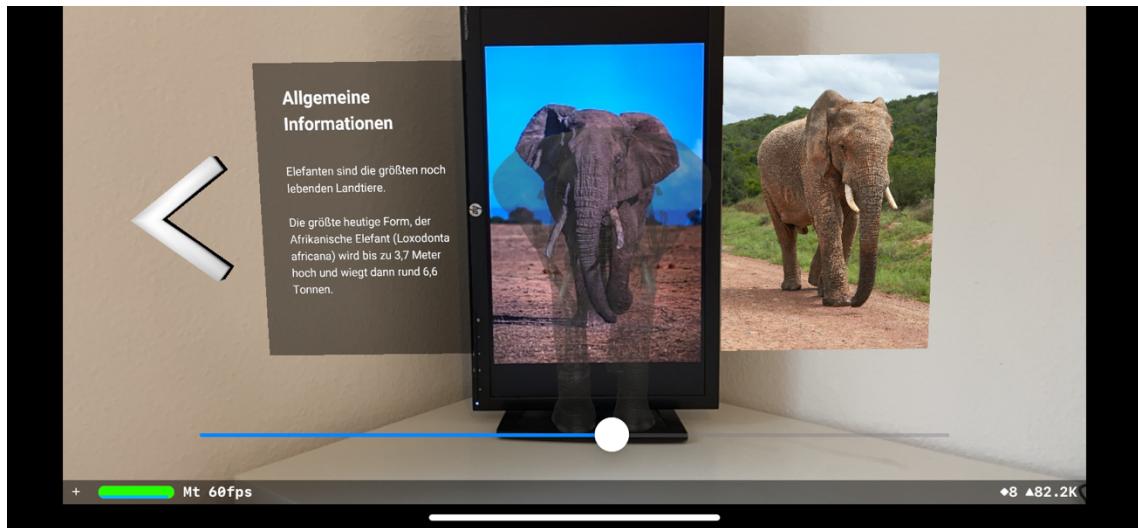


Abbildung 32: Ergebnis der Änderung der Transparenz des Tiermodells

5 Projektergebnis

In diesem Kapitel soll das erreichte Ergebnis der Projektarbeit vorgestellt werden. Im vorherigen Kapitel wurden bereits die Implementierung und die jeweiligen Ergebnisse der wichtigsten Funktionen der App vorgestellt. In diesem Kapitel sollen das Gesamtergebnis und ein Überblick präsentiert werden.

Um die Architektur der Applikation besser zu verstehen, zeigt Abbildung 33 ein Klassendiagramm. Ein Klassendiagramm ist ein Strukturdiagramm der Unified Modeling Language (UML) zur grafischen Darstellung von Klassen, Schnittstellen und ihren Beziehungen.

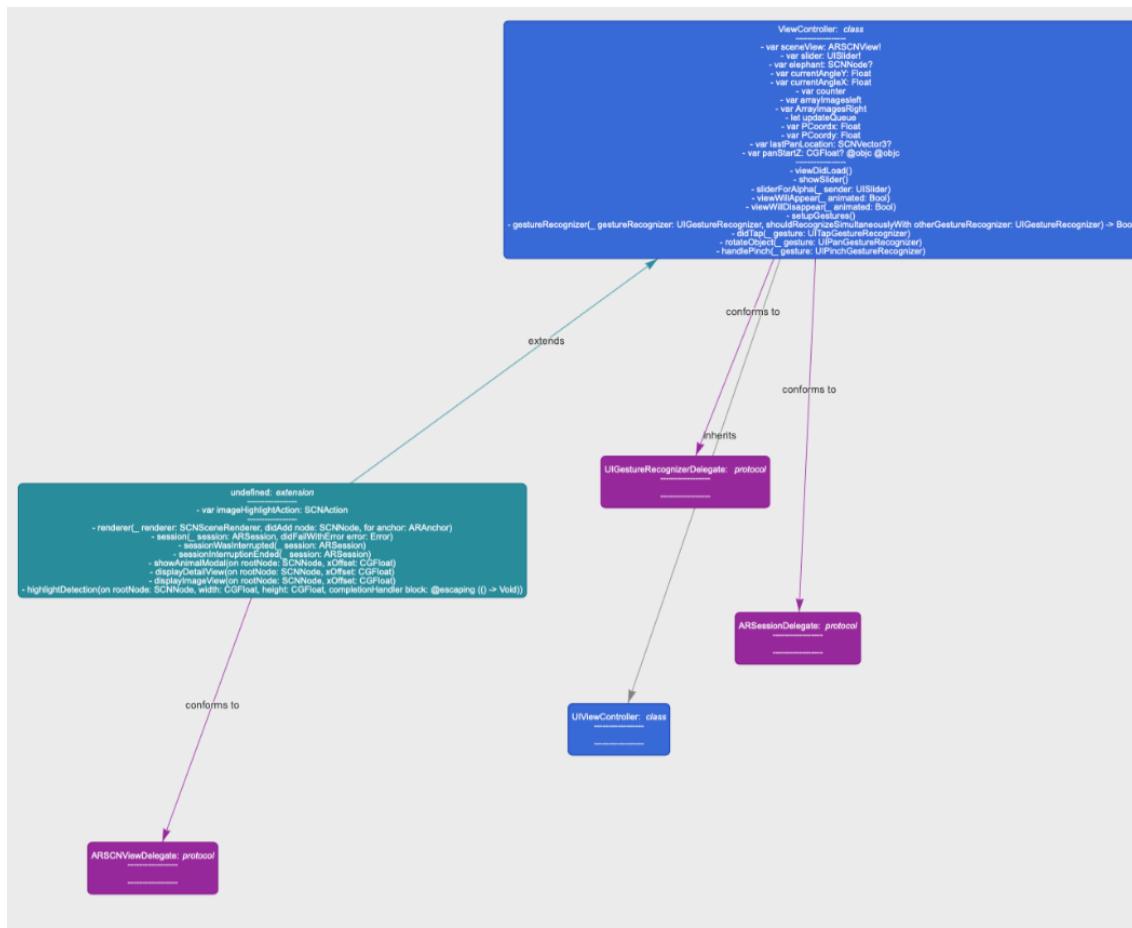


Abbildung 33: UML Diagramm der App

Es ist zu erkennen, dass die App hauptsächliche aus einer Klasse besteht. Bei der Entwicklung von iOS Applikationen ist es üblich, pro Screen der App eine Klasse zu erstellen. Da in dieser Applikation alles in einem Screen stattfindet, ist auch nur eine Klasse notwendig, in diesem Fall die Klasse [ViewController](#).

```

ViewController: class
-----
- var sceneView: ARSCNView!
  - var slider: UISlider!
  - var elephant: SCNNode?
  - var currentAngleY: Float
  - var currentAngleX: Float
  - var counter
  - var arrayImagesLeft
  - var ArrayImagesRight
  - let updateQueue
  - var PCoordx: Float
  - var PCoordy: Float
  - var lastPanLocation: SCNVector3?
  - var panStartZ: CGFloat? @objc @objc
  - viewDidLoad()
  - showSlider()
  - sliderForAlpha(_ sender: UISlider)
  - viewWillAppear(_ animated: Bool)
  - viewWillDisappear(_ animated: Bool)
  - setupGestures()

- gestureRecognizer(_ gestureRecognizer: UIGestureRecognizer, shouldRecognizeSimultaneouslyWith otherGestureRecognizer: UIGestureRecognizer) -> Bool
  - didTap(_ gesture: UITapGestureRecognizer)
  - rotateObject(_ gesture: UIPanGestureRecognizer)
  - handlePinch(_ gesture: UIPinchGestureRecognizer)

```

Abbildung 34: Hauptklasse ViewController

Abbildung 34 zeigt eine Übersicht über alle in dieser Klasse enthaltenen Variablen und Funktionen. Die wichtigsten wurden ausführlich im vorherigen Kapitel erläutert. Um alle Funktionalitäten zu ermöglichen, war eine Erweiterung der Klasse um AR spezifische Funktionen notwendig. Diese Erweiterung ist in Abbildung 35 dargestellt. Dort ist beispielweise definiert, was passiert, wenn die **ARSession** unterbrochen oder beendet wird.

```

undefined: extension
-----
- var imageHighlightAction: SCNAccelerationAction

- renderer(_ renderer: SCNSceneRenderer, didAdd node: SCNNode, for anchor: ARAnchor)
  - session(_ session: ARSession, didFailWithError error: Error)
    - sessionWasInterrupted(_ session: ARSession)
    - sessionInterruptionEnded(_ session: ARSession)
  - showAnimalModal(on rootNode: SCNNode, xOffset: CGFloat)
  - displayDetailView(on rootNode: SCNNode, xOffset: CGFloat)
  - displayImageview(on rootNode: SCNNode, xOffset: CGFloat)

- highlightDetection(on rootNode: SCNNode, width: CGFloat, height: CGFloat, completionHandler block: @escaping () -> Void)

```

Abbildung 35: Erweiterung des ViewControllers

Der gesamte Code der App ist in folgendem öffentlichen Github Repository zu finden: <https://github.com/yschiele/ARLearning>

Dort findet sich auch ein Ordner, in dem einige Videos der App enthalten sind. In diesen Videos ist die Funktionsweise der App noch besser zu erkennen. Mit Hilfe des bereitgestellten Codes kann die App über die Entwicklungsumgebung Xcode auf iOS Geräten installiert werden.

6 Fazit und Ausblick

Das Ziel des Projekts, eine App die das Lernen durch Augmented Reality unterstützt zu realisieren, konnte umgesetzt werden. Diese Applikation erkennt vorgegebene Bilder, egal ob in analoger oder digitaler Form. Um dieses Bild herum wird eine erweiterte Realität erschaffen. Diese beinhaltet verschiedene 3D Modelle, die zur Interaktion mit dem Anwender vorgesehen sind. Der Anwender kann mit Hilfe der 3D Modelle und der Informationstafeln sein Wissen durch das „Learning by doing Prinzip“ aktiv erweitern.

Diese Art von Applikation bietet sehr viele Einsatzmöglichkeiten. Das, in Kapitel 3.2 erwähnte, erweiterte Lehrbuch wäre auf Basis dieses Projekts schnell umsetzbar. Dafür müsste die bisherige App nur um weitere Referenzbilder, zugehörige Modelle und Informationen erweitert werden.

Allerdings konnte die entwickelte Applikation auf Grund der aktuellen Corona Situation nicht von der entsprechenden Zielgruppe getestet werden. Bevor diese App weiterentwickelt werden würde, ist es essentiell ihre Usability zu überprüfen. Vor allem die unterschiedlichen Interaktionen mit dem 3D-Tiermodell, könnten für Anwender mit wenig Erfahrung im AR Bereich eine Herausforderung darstellen. Deshalb müsste dahingehen überprüft werden, ob die Interaktion mit dem 3D Modell über verschiedene Buttons zu einer einfacheren Bedienung führen würde. Außerdem wäre durch einen Bildstabilisierung eine noch ruhigere Darstellung des 3D Modells möglich.

Als Fazit lässt sich sagen, dass Anwendungen mit Augmented-Reality-Inhalten das Potenzial haben, die Art und Weise, wie Menschen lernen, zu verändern beziehungsweise zu erweitern. Die Implementierung dieser Anwendung wird durch die Bereitstellung verschiedener Frameworks immer weiter vereinfacht. Zusätzlich dazu führt die hohe Besitzquote der benötigten Hardware dazu, dass diese Technologie immer mehr Menschen zur Verfügung steht.

Literatur

- [1] D. U. Schmid, D. L. Goertz , S. Radomski , S. Thom und D. J. Behrens, „Monitor Digitale Bildung: Die Hochschulen im digitalen Zeitalter,“ Bertelsmann Stiftung, Gütersloh, 2017.
- [2] K. Panetta, „Gartner Top 10 Strategic Technology Trends for 2019,“ Gartner, Inc., 15 Oktober 2018. [Online]. Available: <https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019/>. [Zugriff am 6 Juni 2020].
- [3] C. de Witt und C. Gloerfeld, Handbuch Mobile Learning, Hagen: Springer Fachmedien Wiesbaden, 2018.
- [4] T. Barton, C. Müller und C. Seel, Hochschulen in Zeiten der Digitalisierung, Worms: Springer Fachmedien Wiesbaden GmbH, 2019.
- [5] R. T. Azuma, „A Survey of Augmented Reality,“ Malibu, CA, 1997.
- [6] J. Tißler, „Augmented Reality im Marketing: 8 aktuelle Beispiele,“ 06 Juni 2018. [Online]. Available: <https://dmexco.com/de/stories/augmented-reality-im-marketing-8-aktuelle-beispiele/>. [Zugriff am 20 August 2020].
- [7] D. Schmalstieg und T. Höllerer, Augmented Reality: Principles and Practice, Boston: Addison-Wesley, 2016.
- [8] T. Bezmalinovic, „Menükarte adé: Projektor bietet Augmented-Reality-Vorschau auf Gerichte,“ 05 Februar 2018. [Online]. Available: <https://mixed.de/menuekarte-ade-projektor-bietet-augmented-reality-vorschau-auf-gerichte/>. [Zugriff am 20 August 2020].
- [9] Microsoft, „Entwickler aufgepasst: Spannende Erlebnisse für eine neue Realität,“ Microsoft, 2020. [Online]. Available: <https://www.microsoft.com/de-de/hololens/developers>. [Zugriff am 20 August 2020].
- [10] M. Brandt, „Diese digitalen Medien werden in der Schule genutzt,“ Statista, 19 Juli 2017. [Online]. Available:

- <https://de.statista.com/infografik/14215/in-deutschen-schulen-genutzte-digitale-medien/>. [Zugriff am 30 Mai 2020].
- [11] . F. Tenzer, „Smartphone-Besitz bei Kindern und Jugendlichen in Deutschland im Jahr 2019 nach Altersgruppe,“ Statista, 7 April 2020. [Online]. Available: <https://de.statista.com/statistik/daten/studie/1106/umfrage/handybesitz-bei-jugendlichen-nach-altersgruppen/#professional>. [Zugriff am 30 Mai 2020].
- [12] D. Steppuhn, SmartSchool – Die Schule von morgen, Köln: Springer Fachmedien Wiesbaden GmbH, 2019.
- [13] F. Liarokapis und E. Anderson, „Using Augmented Reality as a Medium to Assist Teaching in Higher Education,“ The Eurographics Association, 2010.
- [14] L. Kerawalla , R. Luckin, S. Seljeflot und A. Woolard, „Making it real: exploring the potential of Augmented Reality for teaching primary school science.,“ Sussex, 2006.
- [15] B. Shelton und N. Hedley, „Using augmented reality for teaching Earth-Sun relationships to undergraduate geography students.,“ in *IEEE international augmented reality toolkit workshop*, Darmstadt, 2002.
- [16] E. Klopfer und S. Yoon, „Developing games and simulations for today and tomorrow's tech savvy youth.,“ *Techtrends*, Nr. 49, pp. 33-41, 2004.
- [17] M. Billinghurst und H. Kato, „Collaborative augmented reality,“ *Association for Computing Machinery*, pp. 64-70, Juli 2002.
- [18] M. Billinghurst und H. Kato, „The MagicBook — Moving Seamlessly between Reality and Virtuality,“ *IEEE Computer Graphics and Applications*, pp. 6-8, 2001.
- [19] S. Moser und J. Zumbach, „Augmented Reality - erweiterte multimediale Lernerfahrungen.,“ in *Zukunft des Lernens. Wie digitale Medien Schule, Aus- und Weiterbildung verändern.* , Glückstadt, 2012, pp. 145-164.

- [20] F. Waldherr und C. Walter, didaktisch und praktisch: Ideen und Methoden für die Hochschullehre, Stuttgart: Schäffer-Poeschel Verlag, 2014.
- [21] Deutsche Gesellschaft für Hochschuldidaktik, „Qualitätsstandards für die Anerkennung von Leistungen in der hochschuldidaktischen Weiterbildung,“ Göttingen, 2013.
- [22] A. Böss-Ostendorf und H. Senft, Einführung in die Hochschul-Lehre: Ein Didaktik-Coach, Opladen: Utb GmbH, 2010.
- [23] M. Kerres, Mediendidaktik: Konzeption und Entwicklung mediengestützter Lernangebote, Oldenbourg: Oldenbourg Wissenschaftsverlag GmbH, 2013.
- [24] R. Richard und D. Edward, „Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being,“ University of Rochester, Rochester, 2000.
- [25] H. Mandl und G. Reinmann-Rothmeier, „Unterrichten und Lernumgebungen gestalten.,“ LMU München, Weinheim, 2001.
- [26] J. Martín-Gutiérrez und M. Contero, „Improving Academic Performance and Motivation in Engineering Education with Augmented Reality,“ in *HCI International 2011 – Posters’ Extended Abstracts*, Orlando, FL, USA, 2011.
- [27] VisCircle GmbH , „Apple ARKit: Definition, Voraussetzungen und Anwendungsbeispiele.,“ VisCircle GmbH , 2020. [Online]. Available: <https://viscircle.de/apple-arkit-definition-voraussetzungen-und-anwendungsbeispiele/>. [Zugriff am 23 August 2020].
- [28] D. To, „Introduction to ARKit,“ Shadowness Inc, 2018. [Online]. Available: <https://designcode.io/arkit-intro>. [Zugriff am 23 August 2020].
- [29] Apple, „Handling Pinch Gestures,“ Apple, 2020. [Online]. Available: https://developer.apple.com/documentation/uikit/touches_presses_and_gestures/handling_uikit_gestures/handling_pinch_gestures?language=objc. [Zugriff am 24 August 2020].

