
DIPLOMARBEIT

Applied Augmented Reality in Education

Ausgeführt im Schuljahr 2034/24 von:

Recherche zu Varianten von Knapsack-Algorithmen und Umsetzung des Knapsack-Problems als AR-Anwendungsszenario inkl. Dokumentation || Erstellen/Auswerten eines Feedbackfragebogens zur Lernunterstützung

Moritz SKREPEK

5CHIF

Design und Umsetzung der 3D-Objekte zur AR-Abbildung || Analyse der Steuerungsmöglichkeiten (Menüführung, Gesten, ...) und Erstellen der Benutzeroberfläche für die AR-Applikation mit Fokus auf UX

Dustin LAMPEL

5CHIF

Erfassen realer Objekte und kontextgerechte Überlagerung der Realität mit AR-Device || Tagging v. realen Elementen mittels QR-Codes für Tracking || Unit-Tests für d. implementierten Knapsack-Algorithmus

Seref HAYLAZ

5CHIF

Evaluierung/Auswahl Laufzeit-/Entwicklungsumgebung für Umsetzung der Applikation und Integration mit AR-Device inkl. Recherche || Konzeption/Umsetzung des Anwendungsszenarios im Bereich Netzwerktechnik

Jonas SCHODITSCH

5CHIF

Betreuer / Betreuerin:

Mag. BEd. Reis Markus

Wiener Neustadt, am 18. Dezember 2023

Abgabevermerk:

Übernommen von:

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Wiener Neustadt, am 18. Dezember 2023

Verfasser / Verfasserinnen:

Moritz SKREPEK

Dustin LAMPEL

Seref HAYLAZ

Jonas SCHODITSCH

Inhaltsverzeichnis

Eidesstattliche Erklärung	i
Vorwort	iv
Diplomarbeit Dokumentation	v
Diploma Thesis Documentation	vii
Kurzfassung	ix
Abstract	x
1 Einleitung	1
1.1 Ausgangslage	1
1.2 Auslöser	1
1.3 Aufgabenstellung	1
1.4 Team	2
1.4.1 Aufteilung	2
2 Grundlagen	3
2.1 Vorgehensmodelle	3
2.1.1 Scrum	3
2.1.2 Begründung der Auswahl	5
2.2 Projektmanagement Tools	5
2.2.1 GitHub	5
2.2.2 Jira	6
2.3 Konzeption von Fragebögen	6
2.3.1 Planung der Fragebogenkonzeption	6
2.3.2 Abfassung der Fragen	6
2.3.3 Arten von Fragen	7
2.3.4 Struktur und Gliederung von Fragebögen	7
2.3.5 Mögliche Verfälschung des Resultats	7
2.3.6 Auswertung von Fragebögen	7
3 Produktspezifikationen	8
3.1 Anforderungen und Spezifikationen	8
3.1.1 Use Cases	8
3.2 Design	8
3.2.1 Abläufe	8
3.2.2 Mockups	8

3.3	Eingesetzte Technologien	8
3.3.1	Kriterien	8
3.3.2	Game Engine: Konzeption und Funktion	8
3.3.3	Unity foundation packages	10
3.3.4	Rendering Program	11
4	Feinkonzept und Realisierung	14
4.1	Entwicklungsumgebungen	14
4.1.1	Visual Studio 2022	14
4.1.2	Unity	14
4.1.3	Aufbau einer Unity-Applikation	15
4.1.4	Lebenszyklusmethoden in Unity	15
4.1.5	Managers	16
4.2	Hauptmenu	17
4.2.1	UI/UX	17
4.2.2	Laden der Level	17
4.3	Ping Level	17
4.3.1	Object Tracking	17
4.3.2	Kurvenberechnung	17
4.4	Knapsack Problem Level	17
4.4.1	Aufbau von dem Knapsack-Problem Level	18
4.4.2	Verwendung von QR-Codes	19
4.4.3	Platzieren des Inventars	21
4.4.4	Inventar-Controller	24
4.4.5	Knapsack-Algorithmus	24
4.4.6	Berechnen des eigenen Inventars	27
4.4.7	Darstellung der perfekten Lösung	29
4.4.8	Unit-Tests	29
4.5	Performance	29
5	Zusammenfassung und Abschluss	30
5.1	Ergebnis	30
5.2	Abnahme	30
5.3	Zukunft	30
A	Mockups	31
A.1	UI/UX	31
A.2	Hauptmenu Level Design	31
A.3	Ping-Paket Level Design	31
A.4	Knappsack-Level Design	31
B	Literatur	32

Vorwort

Die vorliegende Diplomarbeit wurde im Zuge der Reife- und Diplomsprüfung im Schuljahr 2023 / 24 an der Höheren Technischen Bundeslehr- und Versuchsanstalt Wiener Neustadt verfasst. Die Grundlegende zu dem arbeiten mit der Microsoft HoloLens2 lieferte uns unser Betreuer Mag. BEd. Markus Reis. Das Ergebniss dieser Diplomarbeit ist eine augmented reality Applikation für die Verwendung am Tag der offenen Tür.

Besonderer Dank gebührt unserem Betruer Mag. Markus Reis für sein unerschöpfliches Engagement und seine kompetente Unterstützung. Weiteres möchten wir uns bei unserem Abteilungsvorstand Mag. Nadja Trauner sowie unserem Jahrgangsvorstand MSc. Wolfgang Schermann bedanken, die uns die gesamte Zeit an dieser Schule unterstützt haben.

Diplomarbeit Dokumentation

Namen der Verfasser/innen	Skrepek Moritz Haylaz Seref Lampel Dustin Schoditsch Jonas
Jahrgang Schuljahr	5CHIF 2023 / 24
Thema der Diplomarbeit	Applied Augmented Reality in Education
Kooperationspartner	Land Niederösterreich, Abteilung Wissenschaft und Forschung

Aufgabenstellung	Darstellung von zwei ausgewählten IT-Grundprinzipien mittels der Microsoft HoloLens2.
------------------	---

Realisierung	Implementiert wurde eine Augmented Reality Applikation für die Microsoft HoloLens2. Um ein gutes Zusammenspiel zwischen Realität und Augmented Reality zu garantieren wurde Raumerkennung verwendet. Um mit den echten Objekten zu interagieren werden QR-Codes verwendet.
--------------	--

Ergebnisse	Planung, Design, Entwicklung und Test einer funktionsfähigen Augmented Reality-Applikation auf Basis des AR-Devices HoloLens2 von Microsoft, die es ermöglicht ausgewählte technische Themenstellungen im Bereich Informatik (Visualisierung eines Pings, Veranschaulichung Knapsack-Problem) für den Einsatz im Unterricht sowie beim Tag der offenen Tür visuell, interaktiv und spielerisch darzustellen.
------------	--

	HÖHERE TECHNISCHE BUNDES- LEHR- UND VERSUCHSANSTALT WIENER NEUSTADT
	Fachrichtung: Informatik Ausbildungsschwerpunkt: Softwareengineering

Typische Grafik, Foto etc. (mit Erläuterung)	Das vorliegende Bild stellt das Logo der AR-Applikation dar. <div data-bbox="742 922 1233 1142" data-label="Image"> </div>
---	---

Teilnahme an Wettbewerben, Auszeichnungen	
---	--

Möglichkeiten der Einsichtnahme in die Arbeit	HTBLuVA Wiener Neustadt Dr.-Eckener-Gasse 2 A 2700 Wiener Neustadt
---	--

Approbation (Datum, Unterschrift)	Prüfer Mag. Markus Reis	Abteilungsvorstand AV Mag. Nadja Trauner
--	--------------------------------	---

Diploma Thesis Documentation

Authors	Skrepek Moritz Haylaz Seref Lampel Dustin Schoditsch Jonas
Form	5CHIF
Academic Year	2023 / 24
Topic	Applied Augmented Reality in Education
Co-operation partners	Land Niederösterreich, Abteilung Wissenschaft und Forschung

Assignment of tasks	Representation of two selected basic IT principles using the Microsoft HoloLens2.
---------------------	---

Realization	An augmented reality application for the Microsoft HoloLens2 was implemented. In order to guarantee a good interaction between reality and augmented reality, spatial recognition was used. QR codes are used to interact with the real objects.
-------------	--

Results	Planning, design, development and testing of a functional augmented reality application based on the AR device HoloLens2 from Microsoft, which enables selected technical topics in the field of computer science (visualization of a ping, illustration of the Backpack problem) for use in lessons and on the day of open door visually, interactively and playfully.
---------	---

This image represents the logo of the AR application.



HTBLuVA Wiener Neustadt
Dr.-Eckener-Gasse 2
A 2700 Wiener Neustadt

AV Mag. Nadja Trauner

Kurzfassung

Diese Diplomschrift befasst sich mit der Konzeption einer Lernapplikation für die HTL Wiener Neustadt, sowie der Realisierung in Form von einer augmented reality Applikation auf der Microsoft HoloLens2.

Das Produkt setzt sich aus dem Hauptmenu, dem Ping Level und dem Knappsack-Problem Level in Form eines Unreal Engine 5 Programms zusammen.

In der Applikation können die Schüler am Tag der offenen Tür zwei wichtige Grundprinzipien der Informatik mit Hilfe von Augmented Reality interessant und spielerisch kennenlernen und dadurch erkennen, ob Sie sowas interessiert.

Abstract

This diploma thesis deals with the conception of a learning application for the HTL Wiener Neustadt, as well as the realization in the form of an augmented reality application on the Microsoft HoloLens2.

The product consists of the main menu, the ping level and the Knappsack problem level in the form of an Unreal Engine 5 program.

In the application, students can see two important things on the open day Basic principles of computer science with the help of augmented reality are interesting and Get to know each other in a playful way and thus see whether you are interested in something like that.

Kapitel 1

Einleitung

1.1 Ausgangslage

Um dem IT-Fachkräftemangel entgegenzuwirken, muss die Ausbildung im MINT-Bereich attraktiviert werden. Diese Diplomarbeit will hier, unterstützt durch das Förderprogramm "Wissenschaft trifft Schule" des Landes NÖ, einen wichtigen Beitrag leisten. Dazu sollen exemplarische Anwendungen im Bereich Augmented Reality für die Vermittlung von Informatik-Lehrinhalten evaluiert und umgesetzt werden.

1.2 Auslöser

Die Besucher des "Tag der offenen Tür" bekommen mit dieser Applikation die neusten Technologien vorgeführt und erkennen dadurch, dass die Schule sich auf einen sehr hohen Technologiestandard befindet. Dadurch kommt es zu einer deutlich erhöhten Nachfrage bei zukünftigen Bewerbungen für die Abteilung Informationstechnik. Weiters wird nach Außen hin der Ruf der Schule gestärkt und diese präsentiert sich damit als attraktiver Ausbildungsstandort für die zukünftigen Mitarbeiter vieler Unternehmen.

1.3 Aufgabenstellung

Erstellen des Levelinhalts mit der Verwendung von 2 realen Laptops. Mit Hilfe der HoloLens wird ein 3D modelliertes Ping Paket auf dem Kabel, dass die zwei Laptops verbindet dargestellt. Wenn der Benutzer auf der Tastatur auf die "ENTER" Taste drückt, wird ein Ping Befehl ausgeführt und die modellierten Pakete werden durch die HoloLens auf dem Netzkabel dargestellt. Dies veranschaulicht dem Benutzer den eigentlich nicht sichtbaren Ping von einem auf den anderen Laptop

Erstellen des zweiten Levels in dem der Benutzer das bekannte Rucksack oder auch Knapsack Problem lösen soll. Durch die HoloLens wird auf einem Tisch ein Spielartiges 2D-Inventar mit einer fix definierten Größe visuell dargestellt. Verwendet werden dabei typisch reale Gegenstände eines HTL Schülers die im täglichen Gebrauch sind. Z.B.: Laptop, Maus, Tastatur, Block, usw... Bei jedem Item können, wenn es in die Hand genommen wird über einen QR-Code der auf diesem Item befestigt ist, alle möglichen Informationen des Items angezeigt werden. Die Aufgabe des Benutzers ist es mit den gegebenen Items das Inventar bestmöglich zu befüllen und dadurch den bestmöglichen Wert pro Volumensprozent zu erreichen. Auf dem Tisch liegen verteilt viele Items, die aber nicht alle in das Inventar passen. Jedes einzelne Item kann der Benutzer aufheben und beliebig gedreht (Horizontal

und Vertikal) in das Inventar legen. Bei jedem neuzugelegtem Item, wird der aktuelle Inventarwert berechnet und angezeigt. Am Ende kann sich der User auch noch über einen Menüpunkt entscheiden, ob er die perfekte Lösung sehen will oder nicht. Wenn sich der User dazu entscheidet die perfekte Lösung anzuzeigen, wird Vertikal über dem vormalen Inventar noch ein Inventar projiziert, dass das normale Inventar widerspiegelt aber mit 3D-Modelierten Objekten.

1.4 Team

Das Diplomarbeitsteam besteht aus:

- Moritz SKREPEK
- Seref HAYLAZ
- Dustin LAMPEL
- Jonas SCHODITSCH

1.4.1 Aufteilung

Die Rolle des Projektleiters der Diplomarbeit nahm Moritz SKREPEK ein, da dieser die Grundidee für die Darstellung zweier IT-Grundprinzipien mittels der Microsoft HoloLens2 hatte. Das Entwickelte System lässt sich in das Hauptmenu, das Ping-Paket-Level und das Knappsack- Problem-Level gliedern. Die Implementierung des Hauptmenus übernahm Dustin LAMPEL, dabei verwendete er für die UI/UX das UX-Tools-Plug-Ins für Mixed Reality. Die Umsetzung des Ping-Paket-Levels übernahmen Seref HAYLAZ, Dustin LAMPEL und Jonas SCHODITSCH mittels Object-Tracking, Kurvenberechnungen und 3D-Objekten. Das Knappsack-Problem-Level übernahmen Moritz SKREPEK und Seref HAYLAZ mittels Verwendung von Spatial Anchors, Spatial Mapping, Qr-Code Tracking, Knappsack-Algorithmus, 3D-Objekte und Unittests.

Kapitel 2

Grundlagen

In diesem Kapitel werden das Vorgehensmodell und alle Tools, die für die erfolgreiche Abwicklung des Projekts nötig sind, erläutert.

2.1 Vorgehensmodelle

Im Vorfeld der Durchführung des Projekts wurden Informationen über diverse Vorgehensmodelle gesammelt. Für das Projektteam war schnell klar, dass ein agiles Modell gewählt werden sollte, da somit das Projekt dynamischer geplant und durchgeführt werden kann. Die Auswahl für Scrum stand direkt bei Projektbeginn fest. In dem folgenden Abschnitt wird dieses Vorgehensmodell genauer erklärt und unsere Entscheidung anschließend begründet.

2.1.1 Scrum

Scrum¹ ist ein agiles Projektmanagement-Framework zur effizienten Entwicklung von Produkten und Software. Es betont Zusammenarbeit, Anpassungsfähigkeit und Lieferung von Arbeitsfähigem in kurzen Zeitspannen (Sprints).

Die oben angeführte Definition gibt einen kurzen Einblick in das agile Vorgehensmodell Scrum. Zu den Hauptmerkmalen dieses Modells zählen folgende:

- Drei Rollen, welche nachfolgend erklärt werden
- Product Backlog, welches Anforderungen enthält
- Produktentwicklung erfolgt iterativ und in zeitlich definierten Zyklen
- Das Team arbeitet autonom
- Alle Mitglieder sind gleichberechtigt

Die drei Rollen in Scrum

- Product Owner²:
Die Pflege des Product Backlogs liegt in der Verantwortung dieser Rolle, die die fachliche Auftraggeberseite sowie sämtliche Stakeholders vertritt. Ein zentrales Anliegen ist die Priorisierung der Product Backlog Items in einer Weise, die den Business Value des Produkts maximal steigert und die Möglichkeit für frühzeitige Releases von

¹Scrum Alliance **WHAT-IS-SCRUM**

²Scrum-Rolle **Product-Owner**

Kernfunktionalitäten schafft. Dies ermöglicht einen schnellen Return on Investment. Um stets gut informiert zu sein, nimmt die Person nach Möglichkeit an den Daily Scrums teil, um auf passive Weise Einblicke zu gewinnen. Zudem steht sie dem Team für Rückfragen zur Verfügung, um einen reibungslosen Informationsaustausch zu gewährleisten.

- Scrum Master ³:

Der Scrum-Master übernimmt eine zentrale Rolle im Scrum-Prozess und ist für die korrekte Umsetzung desselben verantwortlich. Als Vermittler und Unterstützer fungiert er als Facilitator, der darauf abzielt, einen maximalen Nutzen zu erzielen und kontinuierliche Optimierung sicherzustellen. Ein zentrales Anliegen ist die Beseitigung von Hindernissen, um ein reibungsloses Voranschreiten des Teams zu gewährleisten. Der ScrumMaster sorgt für einen effizienten Informationsfluss zwischen dem Product Owner und dem Team, moderiert Scrum-Meetings und behält die Aktualität der Scrum-Artefakte wie Product Backlog, Sprint Backlog und Burndown Charts im Blick. Darüber hinaus liegt in seiner Verantwortung, das Team vor unberechtigten Eingriffen während des Sprints zu schützen. -Werte zu verstehen.

- Team ⁴:

Das Team, bestehend aus fünf bis zehn Personen (idealerweise sieben), zeichnet sich durch seine interdisziplinäre Zusammensetzung aus, die Entwickler, Architekten, Tester und technische Redakteure umfasst – eine Struktur, die in den meisten Fällen von Vorteil ist. Durch Selbstorganisation agiert das Team eigenständig und übernimmt die Verantwortung als sein eigener Manager. Es hat die Befugnis, autonom über die Aufteilung von Anforderungen in Aufgaben zu entscheiden und diese auf die einzelnen Mitglieder zu verteilen, wodurch der Sprint Backlog aus dem aktuellen Teil des Product Backlog entsteht.

Alle Anforderungen an das Produkt werden in sogenannten User Stories, die hauptsächlich vom Product Owner erstellt werden, im Product Backlog gesammelt. In einem Intervall, welcher als Sprint bezeichnet wird, werden die User Stories abgearbeitet. Die Projektentwicklung nach Scrum besteht dabei aus 5 Elementen:

- Sprint: Planning Meeting ⁵

Im Sprint Planning Meeting wird das Ziel des folgenden Sprints festgelegt. Hier werden jene Anforderungen im Product Backlog, die in diesem Sprint realisiert werden sollen, in einzelne Aufgaben zerteilt und anschließend im Sprint Backlog gesammelt.

- Sprint: ⁶

Ein Sprint ist eine Entwicklungsphase, in welcher eine vollfunktionsfähige und potentielle veröffentlichbare Software entsteht. Die Dauer eines solchen Sprints liegt zwischen 1 bis 4 Wochen.

- Daily Scrum: ⁷

Der Daily Scrum ist ein kurzes Teammeeting in dem Teammitglieder mitteilen, welche Aufgaben seit dem letzten Daily Scrum abgeschlossen wurden, woran bis zum nächsten Daily Scrum gearbeitet werden muss und wo momentan Probleme existieren. Somit sind alle Teammitglieder ständig up-to-date und kennen den aktuellen Stand wodurch anstehende Probleme leichter gelöst werden können.

³Scrum-Rolle **Scrum-Master**

⁴Scrum-Rolle **Team**

⁵Scrum-Meetings **Sprint-planing-meeting**

⁶Scrum-Meetings **Sprint**

⁷Scrum-Meetings **Daily-Scrum**

- Sprint Review: ⁸

In diesem Meeting präsentiert das Entwicklungsteam die in diesem Sprint abgeschlossenen Arbeitsergebnisse (z. B. fertige Produktinkremente) den Stakeholdern, dies können Produktbesitzer, Kunden, Führungskräfte und andere Interessengruppen sein.

- Sprint Retrospektive: ⁹

Ihr Hauptzweck besteht darin, dass das Scrum-Team (das Entwicklungsteam, der Scrum-Master und der Produktbesitzer) gemeinsam über den abgeschlossenen Sprint reflektiert und Möglichkeiten zur kontinuierlichen Verbesserung identifiziert.

Durch diese Elemente kann ein optimaler Projektablauf gewährleistet werden. Das Projekt ist jederzeit für Änderungen offen und durch eine enge Zusammenarbeit mit dem Kunden können Missverständnisse und Probleme früh behandelt und kommuniziert werden.

2.1.2 Begründung der Auswahl

Die Applied Augmented Reality in Education Applikation besteht aus 3 verschiedenen Level. Im Team welches aus vier Schülern bestand übernahm jede Person einen Teilbereich oder arbeiteten gemeinsam an einem dieser Level mit Unteraufgaben in diesem Level. Unterstützt wurde man von einem Lehrer, der stetz für Fragen bereitstand und oftmals in beratender Form vorhanden war. Als Vorgehensmodell wählte das Team das agile Modell Scrum. Die von Scrum gegebenen Richtlinien konnten leicht eingehalten werden, da das Team täglich in der Schule aufeinander traf als auch privat Kontakt hatten. Jederart Änderung, Problem oder Änderungen und anderartige Dinge konnten daher leicht kommuniziert und besprochen werden. Am Ende jedes Sprints wurden die erreichten Ergebnisse mit dem Betreuer besprochen, sowie die Neuerungen vorgestellt. In den Sprintreviews konnte somit Feedback zu den Ergebnissen gesammelt werden und von dem Betreuer konnten neue Ansichten und Denkweisen angebracht und integriert werden. Durch die Sprint Retrospektive konnten die Schüler einen größeren Mehrwert aus der Projektentwicklung schöpfen, da sie neben der Verwendung des Scrum-Prozesses auch ihre Fähigkeiten in den einzelnen Bereichen, durch das Besprechen der positiven und negativen Aspekte verbessern.

2.2 Projektmanagement Tools

Um einen positiven Verlauf des Projekts zu ermöglichen, benötigt man die unterstützenden Tolls beim Projektmanagement sowie die Verwaltung von Dateien.

2.2.1 GitHub

Als sogenanntes Repository für die Source Code Dateien wurde GitHub mit der dazugehörigen Webanwendung verwendet. Hier stand am Anfang des Projekt die Frage welche Technologie und welcher Anbieter gewählt werden soll. Andere namhafte Anbieter solche Verwaltungssystem sind:

- GitLab
- SourceForge

Ausschlaggebend für die Wahl von GitHub waren mehrere Punkte. Einerseits ist GitHub eine kostenlose Lösung. Das bedeutet, dass man gratis ein privates Projekt mit mehreren

⁸Scrum-Meetings **Sprint-Review**

⁹Scrum-Meetings **Sprint-Retrospektiv**

Mitgliedern anlegen kann. Manche Lösungen bieten hier beispielsweise nur eine begrenzte Anzahl von Mitgliedern an. Benötigt wurde lediglich ein Account zur Registration.

2.2.2 Jira

Als sogenanntes Verwaltungstool für die Vorgänge in dem Project wurde Jira mit der dazugehörigen Webanwendung verwendet. Hier stand am Anfang des Projekts ebenfalls die Frage welche Technologie und welcher Anbieter gewählt werden soll. Andere namhafte Anbieter solcher Tools sind:

- VivifyScrum
- KanBan

Ausschlaggebend für die Wahl von Jira waren mehrere Punkte. Einerseits ist Jira eine kostenlose Lösung. Das bedeutet, dass man ein SCRUM Board mit mehreren Mitgliedern gratis anlegen kann. Ein weiterer Punkt ist die direkte Verbindung zu dem GitHub Repository und die Möglichkeit, dass in Jira selbst neue Branches und Commits auf das Repository erstellt werden können.

2.3 Konzeption von Fragebögen

Bei jeder Umfrage werden Informationen von Personen oder Personengruppen zu der allgemeinen Umsetzung und dem Verständnis der Applikation gesammelt. Diese werden im Anschluss ausgewertet und interpretiert. Wichtig ist hier den Zweck jeder Umfrage genau zu definieren. Durch präzise und detaillierte Zielsetzungen ist es später dann möglich, den Erfolg der Umfrage zu garantieren.

2.3.1 Planung der Fragebogenkonzeption

Die Konzeption und Gestaltung eines Fragebogens ist der wichtigste Schritt bei der Planung. Eine gut überlegte Planungsphase führt zu besseren Ergebnissen und dadurch auch eine leichtere Evaluierung. Folgende Entscheidungen müssen daher schon im Vorfeld definiert und getroffen werden:

- Inhalt: evtl. bestehende Fragebögen verwenden oder anpassen.
- Umfang: Eher kurz halten (In Abhängigkeit von den Zielen).
- Ablauf und zeitlicher Rahmen: postalisch (längere Rücklaufzeit) oder elektronisch ¹⁰
- Zielgruppe: Vollbefragung oder Stichproben ¹¹

2.3.2 Abfassung der Fragen

Der Erfolg einer Umfrage benötigt eine genau Vorbereitung. Im Vorfeld muss klar sein, dass nur einzelne Ausschnitte eines Themengebietes behandelt werden können. Diese Ausschnitte müssen daher umso enger und genauer definiert werden. Hier ist daher vorallem die eindeutige Formulierung der Fragen wichtig.

im Vordergrund bei der Fragenformulierung stehen hier die Verständlichkeit bzw. die Unmissverständlichkeit. Folgende Regeln zur Formulierung sollen daher eingehalten werden:

¹⁰Vgl. Buehner S. -

¹¹Vgl. Mayer S. -

- Einfache Wörter: Wörter, keine Fachausdrücke, anderssprachige Wörter oder Fremdwörter
- Formulierung: Möglichst kurz
- Keine belastenden Wörter verwenden (z.B.: Ehrlichkeit, etc...)
- Keine hypothetischen Formulierungen
- Nur auf einen bestimmten Sachverhalt beziehen
- Keine Überforderung (Nicht zu viele Informationen auf einmal)
- Keine doppelten Verneinungen ¹²

Diese Kriterien gelten für eine schriftliche Befragung. Um das Resultat dieser Umfrage nicht zu verfälschen darf der Interviewer keine Extrafragen oder Umformulierungen an den gestellten Fragen tätigen.

2.3.3 Arten von Fragen

Je nach Anforderungsbedingungen wird zwischen einer offenen und geschlossenen Frage unterschieden ¹³

2.3.4 Struktur und Gliederung von Fragebögen

Hier wird verfasst wie die Allgemeine Struktur und Gliederung von Fragebögen aussehen soll. ¹⁴

2.3.5 Mögliche Verfälschung des Resultats

Welche Arten von Verfälschungen gibt es und diese Beschreiben Ursachen dafür beschreiben. ¹⁵

2.3.6 Auswertung von Fragebögen

Wie werten wir die Fragebögen aus? ¹⁶

¹²Vgl. Mayer S. -

¹³Vgl. Mayer S. -

¹⁴Vgl. Buehner S. -

¹⁵Vgl. Buehner S. -

¹⁶Vgl. Mayer S. -

Kapitel 3

Produktspezifikationen

Dieses Kapitel behandelt die Planung und Spezifikation des Projekts. Weiteres wird die verwendete Technologieauswahl begründet und mit Alternativlösungen verglichen.

3.1 Anforderungen und Spezifikationen

Hier steht der allgemeine Text für die Anforderungen und Spezifikationen

3.1.1 Use Cases

Hier steht der allgemeine Text für die Use Cases

3.2 Design

Hier steht der allgemeine Text für das Design

3.2.1 Abläufe

3.2.2 Mockups

Hier steht der allgemeine Text für die Mockups

3.3 Eingesetzte Technologien

3.3.1 Kriterien

Bei der Auswahl der eingesetzten Technologien war es besonders wichtig, dass diese möglichst zuverlässig und bereits etabliert sind. Die Technologien sollen ausfallsicher, leicht benutzbar und vorallem eine performant Verwendung der Applikation sicherstellen.

3.3.2 Game Engine: Konzeption und Funktion

Eine Game Engine stellt eine hochentwickelte und modulare Entwicklungsumgebung dar, die speziell für die Konzeption, Gestaltung und Implementierung von interaktiven digitalen Spielen entwickelt wurde. Als komplexe Softwarearchitektur bildet sie das Grundgerüst für die Realisierung von Spieleprojekten, wobei sie eine Vielzahl von Funktionen und Werkzeugen bereitstellt, um den Entwicklungsprozess zu erleichtern und zu optimieren.

Im Kern vereint eine Game Engine verschiedene Module, die für unterschiedliche Aspekte der Spieleentwicklung zuständig sind. Dazu gehören unter anderem die Grafik-Engine, die Physik-Engine, die Audio-Engine sowie Mechanismen für Kollisionserkennung, Animationen und Künstliche Intelligenz. Durch diese modulare Struktur ermöglicht die Game Engine eine effiziente und ressourcenschonende Entwicklung, indem sie Entwickler von der tiefen Implementierung grundlegender Funktionen entlastet.

Die Grafik-Engine ist dabei verantwortlich für die Darstellung visueller Elemente, von 2D-Grafiken bis hin zu komplexen 3D-Welten. Sie bietet Mechanismen zur Berechnung von Lichteffekten, Schatten, Texturen und Animationen. Die Physik-Engine simuliert realistische physikalische Interaktionen, um eine authentische Umgebungsgestaltung und realitätsnahe Bewegungen der Spielelemente zu gewährleisten. Die Audio-Engine ermöglicht die Integration von Klängen und Musik, um die Spielerfahrung zu vertiefen.

Ein entscheidendes Merkmal einer Game Engine ist auch die Integration von Programmiersprachen wie C++ oder C, die es Entwicklern erlauben, spezifische Spiellogik und Interaktionen zu implementieren. Dieser Aspekt ermöglicht die Flexibilität und Anpassbarkeit der Engine an die individuellen Anforderungen eines Spieleprojekts.

In der Gesamtheit fungiert die Game Engine als zentrale Schaltstelle für die kreative Entfaltung von Entwicklerteams, indem sie eine umfassende Plattform für das Design und die Umsetzung von Spieleideen bietet. Ihre Funktionen reichen von der effizienten Ressourcenverwaltung bis hin zur Bereitstellung von Werkzeugen für das Testen, Debuggen und Optimieren von Spielen.

Die Auswahl einer geeigneten Game Engine ist eine strategische Entscheidung und hängt von den spezifischen Anforderungen eines Projekts ab. In diesem Kontext wurden die beiden führenden Game Engines, Unity und Unreal Engine, evaluiert, wobei Unity aufgrund seiner Programmiersprache C, dem einfachen Einstieg für Anfänger, der exzellenten Dokumentation und der umfangreichen Tutorial-Ressourcen als präferierte Wahl für das vorliegende Projektteam hervorging.

Game Engine Auswahl und Wechsel im Projektverlauf

Zu Projektbeginn standen zwei der führenden Game Engines zur Auswahl – die Unreal Engine und Unity. Eine Game Engine fungiert als komplexe Softwareumgebung, speziell entwickelt für das Design und die Entwicklung von digitalen Spielen. Die Auswahl einer geeigneten Engine beeinflusst maßgeblich den Entwicklungsprozess und den Erfolg eines Projekts.

Nach einer gründlichen Recherche und Evaluierung der beiden Optionen entschied sich das Projektteam für Unity als präferierte Game Engine. Diese Entscheidung wurde durch mehrere Schlüsselfaktoren gestützt:

- **Programmiersprache: C** - Die Verwendung der Programmiersprache C erwies sich als entscheidend, da sie sich als äußerst effizient und benutzerfreundlich herausstellte.
- **Einfacher Einstieg für Anfänger** - Unity bietet einen leicht verständlichen Einstieg in die Spieleentwicklung, was besonders für Teammitglieder mit unterschiedlichem Erfahrungsniveau von Vorteil ist.
- **Sehr gute Dokumentation** - Die umfassende Dokumentation von Unity spielte eine zentrale Rolle für effiziente Entwicklung und Problemlösung im gesamten Projektverlauf.
- **Hohe Anzahl an Tutorials** - Unity überzeugte mit einer reichhaltigen Sammlung von Tutorials und Schulungsmaterialien, die eine kontinuierliche Weiterbildung und schnelle Lösung von Herausforderungen ermöglichten.

Ursprünglich war die Unreal Engine aufgrund ihres beliebten Blueprint-Scripting-Systems in Betracht gezogen worden. Jedoch traten im Verlauf der Entwicklungsarbeit spezifische Herausforderungen auf, die zu einer strategischen Entscheidung für den Wechsel zur Unity Game Engine führten. Die Herausforderungen umfassten:

1. **Mangelhafte Dokumentation für AR-Entwicklung in der Unreal Engine:** Die unzureichende Dokumentation für die Entwicklung von Augmented Reality (AR)-Anwendungen in der Unreal Engine erwies sich als erhebliche Hürde. Fehlende detaillierte Anleitungen und Referenzen für AR-spezifische Funktionen behinderten die effiziente Integration von AR-Elementen.
2. **Begrenzte Verfügbarkeit von AR-spezifischen Online-Tutorials:** Ein Mangel an Online-Tutorials, die sich speziell mit der Entwicklung von AR-Anwendungen in der Unreal Engine befassten, führte zu einer beträchtlichen Lernkurve für das Entwicklerteam und verzögerte den Implementierungsprozess von AR-spezifischen Features.
3. **Komplexität der AR-Entwicklung in der Unreal Engine:** Die Unreal Engine erwies sich als anspruchsvoller in Bezug auf die Umsetzung von AR-spezifischen Funktionen. Die Notwendigkeit, komplexe Skripte zu erstellen und vielfältige Einstellungen anzupassen, führte zu einem erhöhten Zeitaufwand für die Umsetzung von AR-Elementen.
4. **Mangelhafte Integration von AR-spezifischen Werkzeugen:** Schwächen in der Integration von AR-spezifischen Entwicklungswerkzeugen in der Unreal Engine erschwerten eine nahtlose Interaktion mit AR-Plattformen und die optimale Nutzung ihrer Funktionen.
5. **Eingeschränkte Community-Unterstützung für AR-Entwicklung:** Im Vergleich zu Unity war die Community-Unterstützung für die AR-Entwicklung in der Unreal Engine begrenzt. Die Verfügbarkeit von Ratschlägen und Lösungen für spezifische AR-Herausforderungen war eingeschränkt, was die Eigenständigkeit bei der Lösung von Problemen beeinträchtigte.

Diese Herausforderungen bildeten die Grundlage für die strategische Entscheidung des Projektteams, von der Unreal Engine zu Unity zu wechseln. Der Wechsel ermöglichte eine effizientere und zielführende Entwicklung der AR-Applikation, gestützt durch Unity's umfassende Unterstützung, detaillierte Dokumentation und breite Community-Ressourcen.

3.3.3 Unity foundation packages

In dem folgenden Abschnitt wird erklärt welche Packages in die Unity Applikation eingeführt werden müssen um die Entwicklung einer Augmented Reality Applikation ohne Problem ermöglichen zu können.

MRTK3

Das Mixed Reality Toolkit (MRTK) ¹ ist eine Sammlung von Tools, Skripten und Ressourcen, die speziell für die Entwicklung von Mixed-Reality-Anwendungen, einschließlich Augmented Reality, in Unity entwickelt wurden. MRTK3 ist eine Weiterentwicklung der vorherigen Versionen und bietet viele Vorteile für AR-Anwendungen:

- Interaktions- und Benutzerführung:
MRTK3 stellt eine Reihe von Interaktionskomponenten und -systemen zur Verfügung, die es Entwicklern ermöglichen, intuitivere Benutzererfahrungen in AR-Anwendungen

¹Microsoft **MRTK3**

zu gestalten. Dies umfasst Dinge wie das Platzieren von Objekten in der realen Welt, die Verfolgung von Handgesten und die Unterstützung von Blickverfolgung.

- **Standardisierte APIs:**
Durch die Verwendung von MRTK3 kannst du auf standardisierte APIs und Komponenten zugreifen, die speziell für AR-Anwendungen entwickelt wurden. Dies erleichtert die Implementierung von Funktionen wie Handgesten, Sprachsteuerung und Objektplatzierung.
- **Einfache Konfiguration und Anpassung:**
MRTK3 bietet eine einfache Konfiguration und Anpassung über die Unity-Oberfläche. Dies erleichtert die Anpassung deiner AR-Anwendung an spezifische Anforderungen und Use Cases.

Microsoft OpenXR Plugin

Das Microsoft OpenXR Plugin ² ist eine Sammlung von Tools ist ein wichtiges Plugin für Unity, das die Integration von OpenXR-Unterstützung in die AR-Anwendung ermöglicht. OpenXR ist ein offener Industriestandard, der die Entwicklung von XR (Extended Reality)-Anwendungen, einschließlich Augmented Reality, erleichtert. Anschließend ein paar Punkte wieso dieses Plugin so wichtig ist:

- **Geräteunabhängigkeit:**
Durch die Verwendung von OpenXR und dem Microsoft OpenXR Plugin kann die AR-Anwendung auf verschiedenen XR-Geräten ausgeführt werden, ohne die Kernfunktionalität für jedes einzelne Gerät neu entwickeln zu müssen. Dies gewährleistet eine reibungslose Interaktion mit der HoloLens 2 und anderen XR-Geräten.
- **Leistungssteigerung und Stabilität:**
Die Nutzung von OpenXR und des Microsoft OpenXR Plugins kann die Leistung und Stabilität der AR-Anwendung erheblich verbessern. Sie gewährleisten eine reibungslose Ausführung der Anwendung auf dem Zielsystem und bieten eine optimale Benutzererfahrung. itemize]

3.3.4 Rendering Program

Das Rendering Program wird benötigt um die eingesetzten 3D-Modellen für die zwei Level zu erstellen. Die Auswahl dieses Rendering Programs Blender war bereits bei Projektstart klar.

Diese Entscheidung ist begründet durch folgende Punkte:

- **Kostenfrei und Opensource**
Blender ist kostenfrei und quelloffen, was bedeutet, dass Sie es ohne Lizenzkosten nutzen können. Das kann bei der Entwicklung von AR-Anwendungen mit begrenztem Budget besonders attraktiv sein.
- **Echtzeit-Rendering**
Blender verfügt über einen Echtzeit-Renderer namens Eevee, der schnelle Vorschauen und Renderings ermöglicht. Dies kann nützlich sein, um AR-Inhalte in Echtzeit anzuzeigen und zu überprüfen.
- **Integration mit AR-Frameworks**
Obwohl Blender nicht direkt AR-Funktionen unterstützt, können Sie die erstellten 3D-Modelle und Animationen in AR-Entwicklungsumgebungen wie Unity oder Unreal Engine importieren und dort AR-spezifische Funktionalitäten hinzufügen.

²Khronos **OpenXR**

Wie funktioniert Blender im Allgemeinen? Die folgende Beschreibung hebt die Schlüsselaspekte, sowie die Funktionalität von Blender hervor.

- **Benutzeroberfläche und Interaktion**
Blender verfügt über eine komplex gestaltete Benutzeroberfläche, die sich jedoch durch eine hohe Anpassbarkeit auszeichnet. Die Hauptansicht enthält 3D-Modelle, Ansichten, Fenster und Panels. Benutzer interagieren mit den Objekten und Werkzeugen über Maus- und Tastaturbefehle. Erfahrene Nutzer sind in der Lage einige Zeit zu sparen, in dem sie sogenannte Hotkeys oder Shortcuts verwenden. Wenn man zum Beispiel die Tasten **STRG** und **A** gleichzeitig drückt, kann man alle Objekte auswählen, ohne diese mit der Maus einzeln anzuklicken und auszuwählen.
- **3D-Modellierung**
Blender ermöglicht die Erstellung von 3D-Modellen mithilfe von Primitiven wie Würfeln, Kugeln, Flächen und Kurven. Diese können dann bearbeitet und modifiziert werden, um komplexe Formen zu erstellen. Zu den Modellierungswerkzeugen gehören Extrusion, Verschiebung, Skalierung und Rotation.
- **Animation**
Blender unterstützt die Erstellung von Animationen durch das Setzen von Schlüsselbildern (Keyframes) und die Interpolation von Position, Rotation und Skalierung zwischen diesen Schlüsselbildern. Es bietet auch fortschrittliche Animationstools für Skelettanimation, Constraints und Pfadbewegungen.
- **Materialien und Texturen**
Für die Erzeugung realistischer Oberflächen können Materialien erstellt und Texturen auf Objekte angewendet werden. Blender ermöglicht die Feinanpassung von Materialeigenschaften wie Diffusreflexion, Glanz, Transparenz und Emission.
- **Rendering**
Blender verfügt über einen integrierten Renderer, der hochwertige Bilder und Animationen erstellen kann. Benutzer können Render-Einstellungen anpassen und Bilder in verschiedenen Formaten exportieren.
- **Skripting und Add-Ons**
Fortgeschrittene Benutzer können Blender durch Python-Skripting anpassen und erweitern, um benutzerdefinierte Tools und Automatisierungen zu erstellen. Darüber hinaus gibt es eine aktive Entwicklergemeinschaft, die Add-Ons entwickelt, um die Funktionalität von Blender zu erweitern.
- **Gemeinschaft und Ressourcen**
Blender verfügt über eine engagierte Benutzergemeinschaft, die eine umfangreiche Dokumentation, Tutorials und Foren bereitstellt. Diese Ressourcen erleichtern die Einarbeitung und die Lösung von Problemen.

In unserer Diplomarbeit kommt Blender in beiden Leveln zum Einsatz. Die Hauptanwendung des Programms findet in Level 2 statt. Dabei wird Blender für die digitale Modellierung der wichtigen täglichen Gegenstände von Schülern eingesetzt. Das Ziel am Ende ist es, ein großes Pool aus einigen Objekten zu haben, um den Benutzern eine gute Auswahl zu geben.

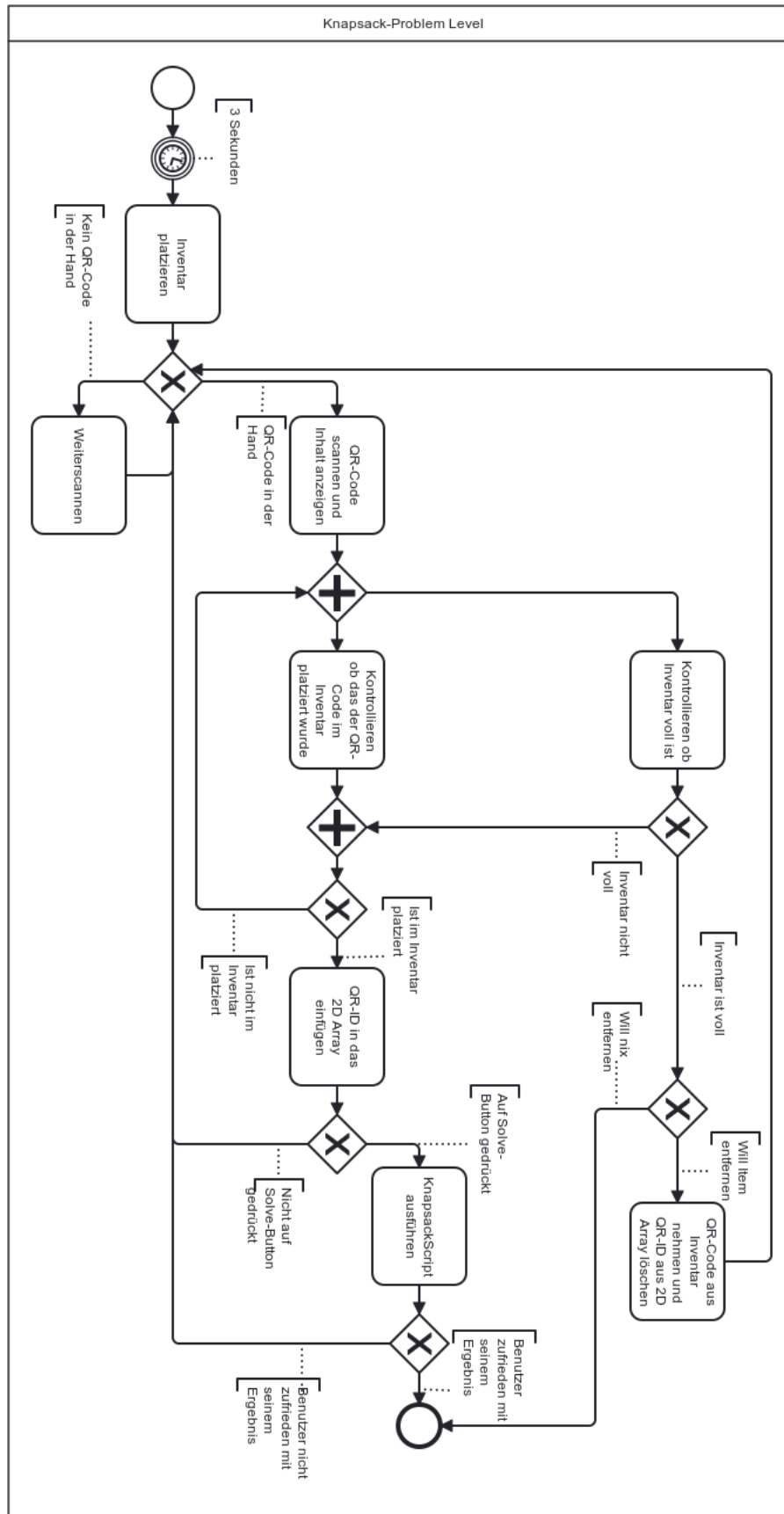


Abbildung 3.1: Ablaufdiagramm des Knapsack-Problem Levels

Kapitel 4

Feinkonzept und Realisierung

4.1 Entwicklungsumgebungen

4.1.1 Visual Studio 2022

Visual Studio 2022 ist eine integrierte Entwicklungsumgebung (IDE) von Microsoft, die speziell für die Entwicklung von Softwareanwendungen, Webanwendungen und Desktop-Anwendungen konzipiert ist. Es handelt sich um eine umfangreiche Entwicklungsumgebung, die von Entwicklern weltweit für eine breite Palette von Anwendungsfällen eingesetzt wird.

4.1.2 Unity

Der Unity-Editor, entwickelt von Unity Technologies, fungiert als umfassende integrierte Entwicklungsumgebung (IDE) und zentrale Arbeitsumgebung für die Konzeption und Umsetzung von 2D-, 3D-, Augmented Reality (AR) und Virtual Reality (VR) Anwendungen und Spielen. Als Kernelement der Unity-Plattform spielt der Editor eine entscheidende Rolle in der Entwicklung von Projekten, die auf Unity-Technologien basieren.

Die Funktionalität des Unity-Editors erstreckt sich über verschiedene Aspekte der Softwareentwicklung, angefangen bei der visuellen Gestaltung von Szenen und Spielwelten bis hin zur Implementierung komplexer Logik und Interaktionen. Die folgenden Abschnitte vertiefen die Schlüsselmerkmale und Funktionen des Unity-Editors, die ihn zu einem essenziellen Werkzeug für Entwickler machen.

Multidisziplinäre Unterstützung und Integration

Der Unity-Editor zeichnet sich durch seine multidisziplinäre Unterstützung aus, die Entwicklern ermöglicht, kollaborativ an Projekten zu arbeiten. Künstler, Entwickler und Designer können innerhalb derselben Umgebung zusammenarbeiten, wodurch ein nahtloser Austausch von Assets, Szenen und Ressourcen ermöglicht wird. Die Integration von Grafik-, Physik- und Audio-Engines erleichtert die Schaffung immersiver und ansprechender digitaler Umgebungen.

Szenengestaltung und Asset-Management

Ein zentrales Merkmal des Unity-Editors ist die intuitive Szenengestaltung, die es Entwicklern ermöglicht, 2D- und 3D-Szenen durch Drag-and-Drop-Operationen zu erstellen und anzupassen. Das Asset-Management ermöglicht eine effiziente Organisation von Ressourcen

wie Modelle, Texturen und Audio-Dateien. Hierbei kommt dem Editor eine Schlüsselrolle in der Strukturierung und Verwaltung umfangreicher Projekte zu.

Programmierung und Skripterstellung

Der Unity-Editor integriert leistungsstarke Programmierfunktionen, die Entwicklern erlauben, Skripte in C oder JavaScript zu verfassen. Die Implementierung von Logik, Interaktionen und Funktionalitäten erfolgt durch die Integration von Skripten in GameObjects und Szenen. Die Echtzeitansicht von Codeänderungen unterstützt einen iterativen Entwicklungsprozess.

Unterstützung für Augmented Reality (AR) und Virtual Reality (VR)

Der Unity-Editor ist essenziell für die Entwicklung von AR- und VR-Anwendungen. Durch die Integration von AR Foundation und XR Interaction Toolkit bietet der Editor leistungsstarke Werkzeuge zur Erstellung immersiver Erlebnisse. Die Möglichkeit, Szenen in Echtzeit in AR- und VR-Geräten zu überprüfen, unterstützt Entwickler bei der Feinabstimmung und Optimierung ihrer Projekte.

Erweiterte Debugging- und Profiling-Werkzeuge

Der Unity-Editor stellt umfassende Debugging- und Profiling-Werkzeuge zur Verfügung, um die Leistung und Funktionalität von Anwendungen zu optimieren. Durch Echtzeit-Inspektion, Fehlerverfolgung und Ressourcenüberwachung unterstützt der Editor Entwickler bei der Identifizierung und Behebung von Problemen, um eine reibungslose Ausführung der Anwendungen sicherzustellen.

4.1.3 Aufbau einer Unity-Applikation

Die Struktur einer Unity-Applikation ist entscheidend für eine effektive Entwicklung und Organisation von 3D-Anwendungen und Spielen. Eine typische Unity-Anwendung besteht aus verschiedenen Schlüsselementen, darunter Szenen, GameObjects, Komponenten, Skripte und Assets. Diese werden koordiniert durch die Hauptkomponente der Anwendung, die sogenannte "GameManager" oder "MainScene". In diesem Abschnitt werden die grundlegenden Bausteine einer Unity-Anwendung sowie bewährte Praktiken für die Strukturierung und Verwaltung dieser Elemente beleuchtet.

4.1.4 Lebenszyklusmethoden in Unity

Die Entwicklung von Augmented Reality (AR)-Applikationen in Unity erfordert ein tiefgreifendes Verständnis der Lebenszyklusmethoden, die in MonoBehaviour-Klassen implementiert werden können. Diese Methoden regeln den Fluss der Programmlogik und ermöglichen Entwicklern, spezifische Aktionen zu bestimmten Zeitpunkten im Lebenszyklus einer Anwendung auszuführen.

- **Awake():** Die **Awake()**-Methode wird aufgerufen, wenn das Skript erstellt wird. Dies geschieht vor anderen Initialisierungsmethoden wie **Start()**. Sie eignet sich für die Durchführung von Initialisierungen, bei denen auf andere Skriptkomponenten oder Ressourcen zugegriffen werden soll. Der Hauptzweck besteht darin, die Ressourcen für das Skript vorzubereiten.
- **Start():** Die **Start()**-Methode wird vor dem ersten Frame aufgerufen und bietet die Möglichkeit, Initialisierungsaufgaben durchzuführen. Im Gegensatz zu **Awake()**

garantiert **Start()** die vollständige Initialisierung aller GameObjects in der Szene. Entwickler nutzen diese Methode oft für Konfigurationen und Vorbereitungen, die spezifisch für die Startphase der Anwendung sind.

- **Update():** Die **Update()**-Methode ist von entscheidender Bedeutung, da sie in jedem Frame aufgerufen wird. Hier kann kontinuierliche Logik ausgeführt werden, wie etwa die Aktualisierung von Animationen, die Verarbeitung von Benutzereingaben oder die Anpassung von Positionen basierend auf der Zeit. Es ist wichtig zu beachten, dass **Update()** häufig aufgerufen wird und daher effizient implementiert werden sollte.
- **LateUpdate():** Ähnlich wie **Update()**, wird aber nachdem alle **Update()**-Methoden aufgerufen wurden. Dies ist besonders nützlich, wenn Anpassungen oder Berechnungen vorgenommen werden müssen, nachdem andere GameObjects und Skripte bereits ihre **Update()**-Logik abgeschlossen haben. Beispielsweise eignet sich **LateUpdate()** gut für Kamera-Anpassungen, bei denen die Position anderer GameObjects bereits aktualisiert wurde.
- **OnEnable() und OnDisable():** Die **OnEnable()**-Methode wird aufgerufen, wenn ein Skript aktiviert wird, während **OnDisable()** aufgerufen wird, wenn es deaktiviert wird. Diese Methoden bieten die Möglichkeit, spezifische Aktionen auszuführen, wenn ein Skript seine Ausführung aufnimmt oder beendet. Entwickler können diese nutzen, um Ressourcen zu laden oder freizugeben, Abonnements auf Ereignisse einzurichten oder abzubreaken, oder um andere vorbereitende oder aufräumende Maßnahmen durchzuführen.

4.1.5 Managers

In diesem Level werden mehrere von Unity und dem Mixed Reality Toolkit 3 bereitgestellten Manager ¹ verwendet. Unter einer Manager versteht man eine Komponente die einer Unity-Szene hinzugefügt wird die dazu dient, bestimmte Aspekte oder Funktionen der Anwendung zu verwalten und zu steuern. Diese Manager spielen eine wichtige Rolle in der Organisation und Kontrolle verschiedener Teile der Unity-Anwendung. In dem "Knapsack Problem Level" werden folgende Manager verwendet:

- **ARPlaneManager²:**
Dieser Manager wird verwendet um in der Umgebung des Benutzers alle Horizontalen Flächen zu erkennen und zu tracken. Außerdem erleichtert er das platzieren von Objekten in der echten Welt. Diese Flächen werden anschließend mit einer Textur markiert. Wenn der User für die vorgeschriebene Zeit auf eine dieser Flächen schaut wird in der Mitte dieser Fläche das Inventar als 3D Objekt dargestellt. An dieses 3D Objekt wird anschließend auch ein Spatial Anchor angehängt und in dem ARAnchor-Manager verwaltet.
- **ARRaycastManager³:**
Dieser Manager wird verwendet um aus einem Origin Punkt also in diesem Fall die Kamera der HoloLens2, raycasting durchzuführen. Diese Raycasts treffen dann auf bereits markierte und getrackte Planes. Wenn dies der Fall ist, ist bekannt, dass der Benutzer auf dieses Plane sieht. Dies ermöglicht dann eine akkurate Platzierung eines 3D Objekts in der realen Welt.

¹Medium Managers

²Unity **PlaneManager**

³Unity **RaycastManager**

4.2 Hauptmenu

Das Hauptmenu dient dazu um das Basic UI/UX System zu implementieren. Hier kann der Benutzer dann diverse Einstellungen tätigen als auch das gewünschte Level auswählen und starten

4.2.1 UI/UX

Mittels verwendung des UX-Tools-Plug-Ins für Mixed Reality wird mit bereitgestellten Knöpfen, Oberflächen, Comboboxen, etc... die Benutzeroberfläche erstellt.

4.2.2 Laden der Level

Durch einen Knopfdruck wird dann in Unreal Engine das der ausgewählte Level geladen.

4.3 Ping Level

In diesem Level wird das IT-Grundprinzip eines Pings zwischen zweier PCs dargestellt. Das Kabel zwischen den zwei PCs wird von der HoloLens getrackt und mittels Kurvenberechnung wird dann eine unsichtbare Kurve über dieses Kabel gezeichnet. Wenn dann der Benutzer auf die Enter Taste auf einem PC drückt wird ein Ping-Paket simuliert und auf dieser Kurve von einem PC zu dem anderen geschickt.

4.3.1 Object Tracking

Durch verwendung von bereitgestellten Technologien der HoloLens2 werden die zwei PCs und das Kabel getrackt.

4.3.2 Kurvenberechnung

Durch Berechnung der Kurve wird das Kabel als Kurve gespeichert und dadurch wird es ermöglicht, dass das 3D-Ping-Paket über diese Kurve von einem PC zum anderen läuft.

4.4 Knapsack Problem Level

Im zweiten Level dieses Projekts steht das Knapsack-Problem im Fokus. Ziel ist es, bbgbi-nen Programmieralgorithmus mithilfe von Augmented Reality (AR) darzustellen. Dieser Algorithmus wird nicht nur in der Höheren Technischen Lehranstalt (HTL) vermittelt, sondern die Benutzer sollen ihn auch selbst programmieren können.

Der Level beginnt damit, dass der Benutzer aufgefordert wird, auf eine horizontale, flache Oberfläche zu schauen. Diese Oberfläche kann ein Tisch, der Boden oder ähnliches sein. Der Benutzer wird dann gebeten, für eine bestimmte vorgegebene Zeit auf diese Oberfläche zu schauen. Nach dieser Zeit werden das Inventar, der Solve-Button und drei Informationslabels auf dieser Oberfläche platziert.

Das Inventar wird durch ein 3x3 zweidimensionales Gitter repräsentiert, ähnlich wie das Inventar in einem Spiel. Zusätzlich befinden sich auf der Oberfläche 11 Bauklötze, die mit QR-Codes versehen sind. Diese Bauklötze repräsentieren die Items, die der Benutzer in das Inventar legen kann. Durch Aufheben und Nahheranhalten an die HoloLens wird der QR-Code gescannt. Dadurch werden das dazugehörige 3D-Modell, der Wert und das

Gewicht des Items angezeigt. Diese Informationen sind für den Benutzer wichtig, um das Gewicht des Items und seinen Einfluss auf das Inventar zu verstehen.

Nach dem Scannen kann der Benutzer einen beliebigen Bauklotz in das Inventar platzieren. Wenn der Benutzer wissen möchte, welchen Wert seine Lösung hat, kann er auf den Solve-Button klicken. Dies löst den Knapsack-Algorithmus aus, der den Wert des eigenen Inventars berechnet. Zusätzlich wird die optimale Lösung für das Problem ermittelt.

Insgesamt bietet dieses Unity Level für die HoloLens 2 eine interaktive und visuelle Erfahrung, bei der die Benutzer das Knapsack-Problem nicht nur verstehen, sondern auch praktisch anwenden können.

4.4.1 Aufbau von dem Knapsack-Problem Level

In diesem Abschnitt wird darauf eingegangen wie eine Szene im Unity Editor aufgebaut ist und wie diese Grundsätzlich funktionieren.

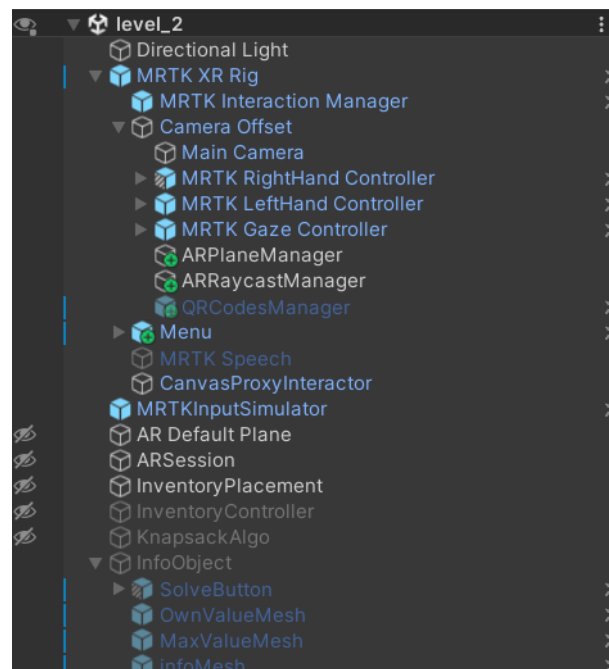


Abbildung 4.1: Hierarchie des Knapsack-Problem Levels im Unity Editor.

In dieser Abbildung ist der Aufbau und Inhalt der 2. Szene⁴ also dem Knapsack-Problem Level zu sehen. Wie zu sehen ist besteht die Szene für dieses Level aus mehreren wichtigen Unity Game Objekten. Darunter sind die folgenden:

- **level-2:** Die Scene in der Alle Unity-Game-Objekte enthalten sind.
- **MRTK XR Rig:** Das Objekt in dem die Hauptkomponenten wie z.B.: *Main Camera*, *MRTK Interaction Manager*, *Alle Manager*, etc... für die AR-Applikation enthalten sind.
- **AR Default Plane:** In der Augmented Reality (AR)-Entwicklung bezieht sich ein Plane⁵ normalerweise auf eine erkannte horizontale Fläche in der realen Welt, auf der virtuelle Objekte platziert werden können. Diese Flächen können zum Beispiel Tische,

⁴Unity **Scene**

⁵Unity **Plane**

Böden oder andere ebene Oberflächen sein. Das Erkennen und Tracking von Planes ist entscheidend, um AR-Objekte realistisch in die Umgebung zu integrieren.

- **ARSession:** In Unity und AR Foundation bezieht sich die ARSession im Allgemeinen auf die Hauptkomponente, die die AR-Funktionalitäten steuert und koordiniert.
- **InventoryPlacement:** Ist ein Game Objekt, dem das *Knapsackscript.cs* Script zugewiesen ist und bei Szenen-Start aktiv ist. Dieses Game Objekt kümmert sich darum, dass zu Beginn der Szene für drei Sekunden lang nach Planes gescanned wird. Nach Abschluss dieser 3 Sekunden wird anschließend dann das Inventar auf dem Plane auf das der User schaut platziert.
- **InventoryController:** Das Game Objekt, dem das *InventoryController.cs* Script zugewiesen ist und nach Abschluss des InventoryPlacement Scripts aktiviert wird. Dieses Game Objekt wird jeden Frame ausgelöst und überprüft ob ein mit QR-Code versehenes Item in dem 3D Inventar platziert wurde.
- **KnapsackAlgo:** Ist das Game Objekt, dem das *KnapsackAlgo.cs* Script zugewiesen ist und bei drücken des *SolveButtons* aktiviert. Bei aktivierung dieses Scripts wird der Knapsack Algorithmus ausgeführt und somit der maximale Wert, der Wert des selbst zusammengestellten Inventars und die Perfekte Lösung errechnet.
- **InfoObject:** Dieses Game Objekt ist eine Sammlung aus mehreren Unity Game Objekten. Dieses Objekt wird bei Ausführung des InventoryPlacement Scripts neben dem Inventar platziert. Es beinhaltet einen Button, der bei Knopfdruck das *KnapsackAlgo* Game Objekt aktiviert, und 3 verschiedene TextMeshPros⁶ die für die Ausgabe von Infos zuständig sind. Außerdem ist bei diesem Game Objekt gut die Strukturierung von Game Objekten in Unity zu sehen. Die drei TextMeshPros und der Button sind dem *InfoObject* untergeordnet was bedeutet, dass diese vier Game Objekte *Children* von dem *InfoObject* sind. Das übergeordnete Game Objekt wird hier dann als *Parent* bezeichnet.

In der Abbildung ist zu sehen, dass ein Paar Game Objekte ausgegraut und nicht ausgegraut sind und, dass neben ein paar Game Objekten ein durchgestrichenes Auge zu sehen ist. Wenn ein Game Objekt im Unity Editor ausgegraut ist bedeutet das, dass dieses Game Objekt und somit alle angehängten Scripts von diesem Game Objekt deaktiviert sind. Das bedeutet, dass dieses Game Objekt samt allen Scripts zu Szenenbeginn nicht aufgerufen und somit auch nicht ausgeführt werden. Nicht ausgegraute Game Objekte wiederum sind daher genau das Gegenteil. Das bedeutet, dass das Game Objekt selbst samt allen angehängten Scripts alle aktiviert sind und somit zu Szenenbeginn aufgerufen und ausgeführt werden.

Wenn neben einem Game Objekt das durchgestrichene Auge zu sehen ist bedeutet das nur, dass dieses Game Objekt im Unity Editor nicht zu sehen ist. Andererseits, wenn kein Zeichen neben dem Game Objekt zu sehen ist, ist dieses Objekt im Unity Editor sichtbar. Dies dient dazu, dass falls in der Unity Szene viele Game Objekte vorhanden sind, dass man diejenige ausblendet die nicht im Editor sichtbar sein müssen wie zum Beispiel Tesh Meshes oder Lables.

4.4.2 Verwendung von QR-Codes

Im vorangegangenen Abschnitt wurde bereits darauf hingewiesen, dass QR-Codes in diesem Level verwendet werden, um die verschiedenen Items zu repräsentieren. Diese QR-Codes spielen eine entscheidende Rolle, indem sie dazu dienen, vielfältige Informationen

⁶Unity **TextMesh**

zu den einzelnen Objekten zu speichern und sie anschließend in einer virtuellen Umgebung abzubilden. Im folgenden Abschnitt möchten wir näher darauf eingehen, wie genau diese QR-Codes generiert werden und welchen Zweck sie innerhalb der Augmented Reality (AR)-Applikation erfüllen. Hierbei wird insbesondere betrachtet, wie die Generierung der Codes erfolgt und auf welche Weise sie innerhalb der Anwendung zur Interaktion mit den realen Objekten verwendet werden.

Inhalt der QR-Codes

Die Informationen, die in einem QR-Code gespeichert werden, sind begrenzt. In unserem Anwendungsfall wird lediglich eine einzelne Zahl im Bereich von 1 bis 11 abgespeichert. Diese Zahlen repräsentieren die 11 verschiedenen Modelle, die wir unterscheiden möchten. Da nur eine Zahl gespeichert wird, genügt ein QR-Code der Größe 21x21 Module (Version 1). Die geringe Anzahl von Modulen ermöglicht eine schnellere Erkennung auch über größere Distanzen.

Die zugehörigen Zahlen erhalten in der Software, genauer gesagt in der Klasse *QRItem.cs*, einen Kontext. Der folgende Codeausschnitt zeigt dies:

```
1 public class QRItem
2 {
3     public struct QRData
4     {
5         public int id;
6         public string name;
7         public Vector3 position;
8         public int weight;
9         public int value;
10    }
11
12    public QRData qrData;
13
14    public Dictionary<int, QRData> items = new Dictionary<int, QRData>()
15    {
16        {1, new QRData { id = 1, name = "Laptop", weight = 70, value = 100 }},
17        {2, new QRData { id = 2, name = "Router", weight = 25, value = 50 }},
18        {3, new QRData { id = 3, name = "Maus", weight = 20, value = 30 }},
19        // ...
20        {11, new QRData { id = 11, name = "Handy", weight = 30, value = 100 }}
21    };
22
23    public QRItem(int id)
24    {
25        items.TryGetValue(id, out qrData);
26    }
27 }
```

In dieser Klasse wird ein Dictionary verwendet, das den Zahlen die folgenden Informationen zuordnet:

- **Item Id:** Die numerische Kennung im QR-Code.
- **Item Name:** Die Bezeichnung des Items, das dieser QR-Code repräsentiert.
- **Item Position:** Die Position des Items in der virtuellen Umgebung.
- **Item Weight:** Das Gewicht des Items.
- **Item Value:** Der Wert des Items.

Diese Informationen spielen eine wesentliche Rolle in der weiteren Berechnung des Knapsack-Algorithmus.

QR-Code Tracking

Das Tracking der QR-Codes erfolgt mithilfe des *QRCodeManager.cs* Skripts. Diese Klasse ist ein Singleton, das die Erkennung und Verfolgung der QR-Codes steuert. Das folgende UML-Diagramm zeigt die wichtigsten Methoden und zusammenhängenden Klassen: Nach der Erkennung eines QR-Codes erfolgen eine Reihe von Schritten, um diese Informationen zu speichern, verarbeiten und zuletzt darzustellen. Hier eine kurze Übersicht:

4.4.3 Platzieren des Inventars

In diesem Abschnitt wird die *PlaceObjectOnLookedAtDesk* Klasse der AARiE Applikation beschrieben. Diese Klasse implementiert das Platzieren des 3D Objektes des Inventars auf einem Tisch und kümmert sich zusätzlich um das aktivieren und deaktivieren sämtlicher Game Objekte um den weiteren Verlauf des Spiels zu handeln.

Frame-Aktualisierung um richtiges Plane zu finden

```

1 void Update()
2 {
3     if (!objectPlaced && canStartScript)
4     {
5         List<ARRaycastHit> hits = new List<ARRaycastHit>();
6         // Use Camera.main.transform.forward as the ray direction
7         if (raycastManager.Raycast(new Ray(Camera.main.transform.position,
8             Camera.main.transform.forward), hits, TrackableType.Planes))
9         {
10             ARPlane closestPlane = FindClosestPlane(hits);
11             if (closestPlane != null)
12             {
13                 if (selectedDeskPlane == null || selectedDeskPlane !=
14                     closestPlane)
15                 {
16                     selectedDeskPlane = closestPlane;
17                     lookStartTime = Time.time; // Start the timer when a new
18                     plane is selected.
19                 }
20                 float timeLookedAtPlane = Time.time - lookStartTime;
21                 if (timeLookedAtPlane >= requiredLookTime)
22                 {
23                     PlaceObjectOnDesk(selectedDeskPlane);
24                     objectPlaced = true;
25                 }
26             }
27             else
28             {
29                 selectedDeskPlane = null;
30             }
31         }
32     }
33     else
34     {
35         selectedDeskPlane = null;
36     }
37 }

```

```

32     }
33 }
34 }

```

Erklärung:

1. **Bedingte Aktualisierung:** Die Update-Funktion wird nur ausgeführt, wenn das Objekt noch nicht platziert wurde und das Skript gestartet werden kann. Wenn die Funktion dann gestartet wird, wird diese jeden Frame 1 mal ausgeführt.
2. **Raycasting auf Planes:** Ein Raycast wird durchgeführt, um festzustellen, ob die Kamera auf ein AR-Plane blickt.
3. **Suche nach dem nächsten AR-Plane:** Die Funktion `FindClosestPlane` wird aufgerufen, um das nächste AR-Plane zu finden.
4. **Zeiterfassung für Blickzeit:** Die Zeit, die auf das ausgewählte AR-Plane geschaut wurde, wird gemessen.
5. **Platzierung des Objekts bei ausreichender Blickzeit:** Wenn die benötigte Blickzeit überschritten wurde, wird die Funktion `PlaceObjectOnDesk` aufgerufen und das Objekt wird platziert.

Das am kürzesten entfernte Plane⁷ finden

```

1  ARPlane FindClosestPlane(List<ARRaycastHit> hits)
2  {
3      ARPlane closestPlane = null;
4      float closestDistance = float.MaxValue;
5      foreach (var hit in hits)
6      {
7          ARPlane plane = planeManager.GetPlane(hit.trackableId);
8          if (plane != null)
9          {
10             float distanceToPlane = Vector3.Distance(Camera.main.transform.
position, hit.pose.position);
11             if (distanceToPlane < closestDistance)
12             {
13                 closestPlane = plane;
14                 closestDistance = distanceToPlane;
15             }
16         }
17     }
18     return closestPlane;
19 }

```

Erklärung:

1. **Suche nach der nächsten AR-Plane:** Die Funktion durchläuft alle getroffenen AR-Planes und findet diejenige mit der geringsten Distanz zur Kamera.
2. **Berechnung der Distanz:** Die Distanz zwischen der Kameraposition und der Position der getroffenen AR-Plane wird berechnet.
3. **Vergleich der Distanzen:** Die Distanz zur aktuellen AR-Plane wird mit der bisher geringsten Distanz verglichen. Wenn sie kleiner ist, wird die aktuelle AR-Plane als

⁷Unity **Plane**

nächste ausgewählt.

Berechnung der Position und aktivierung/deaktivierung sämtlicher Objekte

```
1 void PlaceObjectOnDesk(ARPlane deskPlane)
2 {
3     qrCodesManager.SetActive(true);
4     // Calculate the object's position above the center of the plane.
5     Vector3 objectPosition = deskPlane.center + Vector3.up * heightOffset;
6     // Calculate the rotation to rotate the object -90 degrees around the x-
7     // axis.
8     Quaternion objectRotation = Quaternion.Euler(-90f, 0f, 0f);
9     // Instantiate the object with rotation.
10    GameObject instantiatedObject = Instantiate(inventoryObject,
11    objectPosition, objectRotation);
12    // Set the scale of the instantiated object.
13    instantiatedObject.transform.localScale = new Vector3(20f, 20f, 20f);
14    // Spawn infoGameObject (Two TextMeshes and button for Knapsack Algorithm)
15    Vector3 infoObjectPosition = objectPosition - Vector3.forward * 4.415f +
16    Vector3.right * 0.4f;
17    infoObject.transform.position = infoObjectPosition;
18    infoObject.SetActive(true);
19    // Set the inventoryObject in the InventoryController
20    inventoryController.SetInventoryObject(instantiatedObject);
21    // Enable the InventoryController
22    inventoryController.gameObject.SetActive(true);
23    // Set the visibility of the planes.
24    planeManager.planePrefab.SetActive(false);
25    // Disable this script so it won't run again.
26    gameObject.SetActive(false);
27 }
```

Erklärung:

1. **Aktivierung von QR-Codes und Objektplatzierung:** Die QR-Code-Manager werden aktiviert, und die Position des zu platzierenden Objekts wird berechnet.
2. **Berechnung der Objektposition und Rotation:** Die Position des Objekts wird über dem Zentrum der ausgewählten AR-Plane berechnet, und die Rotation wird auf -90 Grad um die x-Achse festgelegt.
3. **Instantiierung des Objekts:** Das Objekt wird instanziiert und skaliert.
4. **Platzierung des infoGameObject:** Die Position des Info-Objekts (TextMeshes und Button für den Knapsack-Algorithmus) wird festgelegt und aktiviert.
5. **Setzen des Inventarobjekts im InventoryController:** Das platzierte Objekt wird im InventoryController festgelegt.
6. **Aktivierung des InventoryController:** Der InventoryController wird aktiviert.
7. **Sichtbarkeit der Planes ausschalten:** Die Sichtbarkeit der AR-Planes wird ausgeschaltet.
8. **Deaktivierung des Skripts:** Das aktuelle Skript wird deaktiviert, damit es nicht erneut ausgeführt wird.

4.4.4 Inventar-Controller

In diesem Abschnitt wird genauer auf das InventarController Script eingegangen und erklärt wie dieses funktioniert.

4.4.5 Knapsack-Algorithmus

Im folgenden Abschnitt wird der Knapsack-Algorithmus allgemein erklärt, die Problemstellung verdeutlicht und die Unterschiede der beiden möglichen Algorithmen erläutert. Außerdem wird darauf eingegangen, warum er so wichtig ist und in welchen Anwendungen er zum Einsatz kommt. Zudem wird das *KnapsackScript.cs* besprochen, welches diesen Algorithmus implementiert, um den maximal erreichbaren Wert und die optimale Lösung zu finden.

Der Knapsack-Algorithmus ist ein häufig verwendetes Werkzeug in der Informatik, um das Problem der optimalen Ressourcenallokation zu lösen. Dieses Problem tritt auf, wenn eine begrenzte Menge an Ressourcen so effizient wie möglich genutzt werden soll, um einen bestimmten Nutzen oder Gewinn zu maximieren. Der Begriff "Knapsack" leitet sich von der Idee ab, dass man versucht, einen Rucksack mit begrenztem Fassungsvermögen mit Gegenständen zu füllen, die unterschiedliche Werte und Gewichte haben.

Problemstellung

Gegeben sei ein Rucksack mit begrenzter Kapazität und eine Menge von Gegenständen, von denen jeder einen bestimmten Wert und ein bestimmtes Gewicht hat. Das Ziel besteht darin, die Gegenstände auszuwählen, die in den Rucksack passen und den Gesamtwert maximieren.

Algorithmus

Der Knapsack-Algorithmus kann in verschiedenen Varianten implementiert werden, darunter der dynamische Programmieransatz und der Greedy-Ansatz. Im dynamischen Programmieransatz wird eine Tabelle erstellt, um Teilprobleme zu lösen und die optimale Lösung zu berechnen. Der Greedy-Ansatz hingegen wählt Gegenstände basierend auf bestimmten Kriterien aus, um eine lokale Optimierung zu erreichen.

In unserem Kontext wurde der dynamische Programmieransatz implementiert, um den Knapsack-Algorithmus umzusetzen. Der Code dafür ist in der Datei *KnapsackScript.cs* zu finden, welche den maximal erreichbaren Wert und die optimale Lösung berechnet.

Anwendungen

Der Knapsack-Algorithmus findet Anwendung in verschiedenen Bereichen, darunter Logistik, Finanzplanung, Ressourcenmanagement und Netzwerkoptimierung. Beispielsweise kann er verwendet werden, um den effizientesten Transport von Gütern mit begrenzten Kapazitäten zu planen oder in der Finanzplanung, um das Portfolio von Investitionen zu optimieren.

Funktionsaufruf

```
1 int maxValue = KnapsackMaxValue(out usedItems, out int coveredCapacity);  
2 int inventoryValue = -1;  
3 maxMesh.text = "Maximal erreichbarer Wert: " + maxValue.ToString();
```

Erklärung: Hier wird die Funktion aufgerufen und dem textMesh **maxMesh** wird dieser Wert zugewiesen um dem User diesen Wert darzustellen.

Variablen Initialisierung

```
1 int n = items.Count;
2 int[,] dp = new int[n + 1, capacity + 1];
3 bool[,] selected = new bool[n + 1, capacity + 1];
```

Erklärung: Die Initialisierung erstellt die notwendigen Arrays für die dynamische Programmierung.

Knapsack-Algorithmus

```
1 for (int i = 0; i <= n; i++)
2 {
3     for (int w = 0; w <= capacity; w++)
4     {
5         % Initialisierung der DP-Tabelle
6         if (i == 0 || w == 0)
7             dp[i, w] = 0;
8         else if (items[i].weight <= w && i <= maxItems)
9         {
10            % Berechnung des neuen Werts, wenn das Item ausgewählt wird
11            int newValue = items[i].value + dp[i - 1, w - items[i].weight];
12            if (newValue > dp[i - 1, w])
13            {
14                dp[i, w] = newValue;
15                selected[i, w] = true;
16            }
17            else
18            {
19                dp[i, w] = dp[i - 1, w];
20                selected[i, w] = false;
21            }
22        }
23        else
24        {
25            dp[i, w] = dp[i - 1, w];
26            selected[i, w] = false;
27        }
28    }
29 }
```

Erklärung:

1. **Iterationen durch die Items und Kapazität:** Die äußere Schleife durchläuft jedes Item (*i*) von 0 bis zur Gesamtanzahl der Items (*n*). Die innere Schleife durchläuft jede mögliche Kapazität (*w*) von 0 bis zur maximalen Kapazität des Rucksacks.
2. **Initialisierung der DP-Tabelle:** Wenn *i* oder *w* 0 ist, wird der Wert in der DP-Tabelle auf 0 gesetzt. Dies liegt daran, dass es keine Items gibt oder die Kapazität des Rucksacks 0 ist.

3. **Überprüfung, ob das Item ausgewählt werden kann:** Wenn das aktuelle Item ausgewählt werden kann (basierend auf Gewicht und maximaler Anzahl von Items), wird der neue Wert berechnet, wenn das Item ausgewählt wird.
4. **Vergleich und Auswahl:** Der neue Wert wird mit dem Wert verglichen, wenn das Item nicht ausgewählt wird. Wenn der neue Wert größer ist, wird das Item ausgewählt (`selected[i, w] = true`), andernfalls wird das vorherige Ergebnis beibehalten (`selected[i, w] = false`).
5. **Falls Item nicht ausgewählt wird:** Wenn das aktuelle Item nicht ausgewählt wird, wird der Wert in der DP-Tabelle auf den Wert gesetzt, den das Rucksack ohne das aktuelle Item in der vorherigen Iteration erreicht hat.

Bedeutung der DP-Tabelle: Die DP-Tabelle (dp) dient dazu, Teilprobleme zu lösen und die besten Ergebnisse für verschiedene Kombinationen von Items und Kapazitäten zu speichern. Jeder Eintrag in der Tabelle repräsentiert den maximal erreichbaren Wert für eine bestimmte Menge von ausgewählten Items und eine bestimmte Kapazität des Rucksacks.

Insgesamt ermöglicht dieser Algorithmus, die optimale Auswahl von Gegenständen zu finden, die in den Rucksack passt, um den Gesamtwert zu maximieren.

Backtracking, um ausgewählte Items und perfekt Lösung zu finden

```

1 List<List<int>> tempUsedItems = new List<List<int>>();
2 int row = n;
3 int col = capacity;
4 while (row > 0 && col > 0)
5 {
6     List<int> group = new List<int>();
7     while (row > 0 && col > 0 && selected[row, col])
8     {
9         // Item zu ausgewählter Gruppe hinzufügen
10        group.Add(items[row].id);
11        col -= items[row].weight;
12        row--;
13    }
14
15    // Hinzufügen der Gruppe zur Liste
16    if (group.Count > 0)
17    {
18        tempUsedItems.Add(group);
19    }
20
21    // Fortfahren mit dem Backtracking
22    if (row > 0 && col > 0)
23    {
24        row--;
25    }
26 }

```

Erklärung:

1. **Initialisierung von Zeilen- und Spaltenindizes:** Die Variablen `row` und `col` werden auf die letzte Zeile und Spalte der DP-Tabelle gesetzt.

2. **Hauptschleife für Backtracking:** Die äußere Schleife wird durchlaufen, solange die Zeile und die Spalte größer als 0 sind.
3. **Innere Schleife für Gruppenbildung:** Solange die Zeile und die Spalte größer als 0 sind und das Item in der DP-Tabelle ausgewählt wurde (`selected[row, col]`), wird das Item zur aktuellen Gruppe hinzugefügt, das Gewicht von der aktuellen Kapazität abgezogen, und die Zeile wird dekrementiert.
4. **Hinzufügen der Gruppe zur Liste:** Wenn die Gruppe mindestens ein Item enthält, wird die Gruppe zur temporären Liste `tempUsedItems` hinzugefügt.
5. **Fortfahren mit dem Backtracking:** Die Zeile wird dekrementiert, um das nächste ausgewählte Item zu finden.

Diese Schritte werden wiederholt, bis alle ausgewählten Items gefunden sind und die temporäre Liste `tempUsedItems` alle ausgewählten Gruppen enthält.

Gruppengröße berechnen

```

1 int maxGroupSize = 0;
2 foreach (var group in tempUsedItems)
3 {
4     if (group.Count > maxGroupSize)
5         maxGroupSize = group.Count;
6 }

```

Erklärung: Findet die maximale Gruppengröße.

Temporäres Array umwandeln und das usedItems Array befüllen

```

1 usedItems = new int[tempUsedItems.Count, maxGroupSize];
2 for (int i = 0; i < tempUsedItems.Count; i++)
3 {
4     for (int j = 0; j < tempUsedItems[i].Count; j++)
5     {
6         usedItems[i, j] = tempUsedItems[i][j];
7     }
8 }

```

Erklärung: Konvertiert die temporäre Liste von Gruppen in ein 2D-Array, um die ausgewählten Items also die perfekt Lösung zu speichern.

Rückgabewert

```

1 return dp[n, capacity];

```

Erklärung: Gibt den Gesamtwert des besten Rucksacks zurück.

4.4.6 Berechnen des eigenen Inventars

In diesem Abschnitt wird auf die Funktion *KnapsackInventoryValue* des *KnapsackScript.cs* eingegangen. Diese Funktion berechnet das Inventar Value des selbst gebauten Inventars.

Funktionsaufruf

```
1 inventoryValue = KnapsackInventoryValue(inventory);
2 if (maxValue == inventoryValue)
3 {
4     infoMesh.color = Color.green;
5     infoMesh.text = "Maximale Punktzahl erreicht";
6 }
7 else
8 {
9     infoMesh.text = "";
10 }
11 ownMesh.text = "Erreichter Wert: " + inventoryValue.ToString();
```

Erklärung:

1. **Inventar Wert berechnen:** Die Funktion KnapsackInventoryValue wird mit dem übergebenem inventory aufgerufen und der Inventar Wert berechnet.
2. **Ergebnis Handling:** Wenn der eigene Wert gleich dem Maximalen errechnetem Wert entspricht hat der Benutzer gewonnen. Anderenfalls wird dem infoMesh der aktuelle Wert zugewiesen.

Code für die Berechnung des eigenen Inventars

```
1 int KnapsackInventoryValue(int[,] inventory)
2 {
3     if(inventory == null)
4     {
5         throw new System.Exception("Inventory is null");
6     }
7
8     int totalValue = 0;
9
10    foreach (var item in items.Values)
11    {
12        int itemId = item.id;
13        int itemValue = item.value;
14
15        for (int j = 0; j < inventory.GetLength(0); j++)
16        {
17            for (int k = 0; k < inventory.GetLength(1); k++)
18            {
19                if (inventory[j, k] == itemId)
20                {
21                    totalValue += itemValue;
22                }
23            }
24        }
25    }
26
27    return totalValue;
28 }
```

Erklärung:

1. **Überprüfung auf Null:** Die Funktion überprüft, ob das übergebene Inventar null ist. Falls ja, wird eine Ausnahme ausgelöst.
2. **Initialisierung des Gesamtwerts:** Die Variable `totalValue` wird auf 0 initialisiert. Hier wird der Gesamtwert des Rucksackinventars gespeichert.
3. **Durchlauf aller Gegenstände:** Die Funktion durchläuft alle Gegenstände im globalen `items`-Dictionary.
4. **Extrahierung von Informationen:** Für jeden Gegenstand werden die ID (`itemId`) und der Wert (`itemValue`) extrahiert.
5. **Durchlauf des Inventars:** Die doppelte Schleife durchläuft das gesamte Inventar (`inventory`) und prüft, ob der aktuelle Gegenstand vorhanden ist.
6. **Berechnung des Gesamtwerts:** Wenn der Gegenstand im Inventar gefunden wird, wird sein Wert zum Gesamtwert (`totalValue`) addiert.
7. **Rückgabe des Gesamtwerts:** Am Ende wird der Gesamtwert des Rucksackinventars zurückgegeben.

4.4.7 Darstellung der perfekten Lösung

Um die perfekt Lösung nach der Berechnung erfolgreich darzustellen wird in diesem Abschnitt auf die `PLACEHOLDES` Funktion des *KnapsackScript.cs* eingegangen um die errechnete Lösung nach dem drücken des *SolveButton* anzuzeigen

4.4.8 Unit-Tests

Durch Hilfe von Unit-Tests wird versichert, dass der implementierte Knappsack-Algorithmus richtig und performant funktioniert.

4.5 Performance

Performance-Messung

Kapitel 5

Zusammenfassung und Abschluss

5.1 Ergebnis

Hier steht der allgemeine Text für das Ergebnis

5.2 Abnahme

Hier steht der allgemeine Text für das Abnahme

5.3 Zukunft

Hier steht der allgemeine Text für die Zukunft

Anhang A

Mockups

A.1 UI/UX

A.2 Hauptmenu Level Design

A.3 Ping-Paket Level Design

A.4 Knappsack-Level Design

Anhang B

Literatur

- Blender. URL: <https://www.blender.org/about/> (besucht am 06.10.2023).
- Scrum Alliance Inc. WHAT IS SCRUM? URL: <https://www.scrumalliance.org/about-scrum#> (besucht am 06.10.2023).
- Scrum-Master.de Scrum-Rollen - Product Owner. URL: https://scrum-master.de/Scrum-Rollen/Scrum-Rollen_Product_Owner (besucht am 10.11.2023).
- Scrum-Master.de Scrum-Rollen - Scrum Master. URL: https://scrum-master.de/Scrum-Rollen/Scrum-Rollen_ScrumMaster (besucht am 10.11.2023).
- Scrum-Master.de Scrum-Rollen - Team. URL: https://scrum-master.de/Scrum-Rollen/Scrum-Rollen_Team (besucht am 10.11.2023).
- Scrum-Master.de Scrum-Meetings - Sprint - Team. URL: <https://scrum-master.de/Scrum-Meetings/Sprint> (besucht am 10.11.2023).
- Scrum-Master.de Scrum-Meetings - Sprint Planing Meeting - Team. URL: https://scrum-master.de/Scrum-Meetings/Sprint_Planning_Meeting (besucht am 10.11.2023).
- Scrum-Master.de Scrum-Meetings - Daily Scrum Meeting - Team. URL: <https://scrum-master.de/Scrum-Meetings/Sprint> (besucht am 10.11.2023).
- Scrum-Master.de Scrum-Meetings - Sprint Review Meeting - Team. URL: https://scrum-master.de/Scrum-Meetings/Sprint_Review_Meeting (besucht am 10.11.2023).
- Scrum-Master.de Scrum-Meetings - Sprint Retroperspektiv Meeting - Team. URL: https://scrum-master.de/Scrum-Meetings/Sprint_Review_Meeting (besucht am 10.11.2023).
- Microsoft. MIXED REALITY TOOLKIT 3. URL: <https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-overview/> (besucht am 05.11.2023).
- Khronos Group. OPENXR. URL: <https://www.khronos.org/openxr/> (besucht am 05.11.2023).
- Medium. CREATING MANAGER CLASSES IN UNITY. URL: <https://sneakydaggersgames.medium.com/creating-manager-classes-in-unity-a77cf7edcba5> (besucht am 05.11.2023).
- Unity. AR plane manager. URL: <https://docs.unity.cn/Packages/com.unity.xr.arfoundation@4.1/manual/plane-manager.html> (besucht am 05.11.2023).
- Unity. AR raycast manager. URL: <https://docs.unity.cn/Packages/com.unity.xr.arfoundation@5.0/api/UnityEngine.XR.ARFoundation.ARRaycastManager.html> (besucht am 05.11.2023).
- Unity. Plane. URL: <https://docs.unity3d.com/ScriptReference/Plane.html> (besucht am 13.12.2023).
- Unity. Scenes. URL: <https://docs.unity3d.com/Manual/CreatingScenes.html> (besucht am 13.12.2023).
- Color Doku, URL: <https://docs.unity3d.com/ScriptReference/Color.html%7D> besucht am 4.11.2023).

- Trackable Doku, URL: <https://docs.unity3d.com/2019.2/Documentation/ScriptReference/Experimental.XR.TrackableType.html%7D> (besucht am 18.11.2023).
- ARRaycastHit Doku, URL: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/api/UnityEngine.XR.ARFoundation.ARRaycastHit.html%7D> (besucht am 18.11.2023).
- PhotoCaputre, URL: <https://docs.unity3d.com/ScriptReference/Windows.WebCam.PhotoCapture.html%7D> (besucht am 2.11.2023).
- Unity. TextMeshPro. URL: <https://docs.unity3d.com/Manual/com.unity.textmeshpro.html> (besucht am 15.12.2023).
- Foto-/Videokamera in Unity, URL: <https://learn.microsoft.com/de-de/windows/mixed-reality/develop/unity/locatable-camera-in-unity%7D> (besucht am 2.11.2023)
- Microsoft. Buttons — MRTK2. URL: <https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/ux-building-blocks/button?view=mrtkunity-2022-05> (besucht am 7.11.2023).
- Microsoft. Menü "Nahe"— MRTK2. URL: <https://learn.microsoft.com/de-de/windows/mixed-reality/mrtk-unity/mrtk2/features/ux-building-blocks/near-menu?view=mrtkunity-2022-05> (besucht am 8.11.2023).
- Unity. Load scene on button press. URL: https://blog.insane.engineer/post/unity_button_load_scene/ (besucht am 8.11.2023).
- Blender. Can't see added cube on my scene collection [duplicate]. URL: <https://blender.stackexchange.com/questions/162424/cant-see-added-cube-on-my-scene-collection> (besucht am 15.11.2023).
- Blender. How do I Inset a face equally? URL: <https://blender.stackexchange.com/questions/50876/how-do-i-inset-a-face-equally> (besucht am 17.11.2023).
- Blender. Loop Tools. URL: <https://docs.blender.org/manual/en/latest/addons/mesh/looptools.html> (besucht am 20.11.2023).
- Unity. QR-Code Tracking. URL: <https://learn.microsoft.com/en-us/samples/microsoft/mixedreality-qr-code-sample/qr-code-tracking-in-unity/> (besucht am 2.11.2023).
- Unity. QR-Code Tracking Overview. URL: <https://learn.microsoft.com/de-de/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview> (besucht am 30.10.2023).
- Unity. SpacialGraphNode Class. URL: <https://learn.microsoft.com/de-de/dotnet/api/microsoft.mixedreality.openxr.spatialgraphnode?view=mixedreality-openxr-plugin-1.9> (besucht am 2.11.2023).
- Scholl, Armin. *Die Befragung* 3. Aufl. Stuttgart: utb GmbH, 2014.
- Mayer, Horst. *Interview und schriftliche Befragung. Entwicklung, Durchführung und Auswertung* 3. Aufl. München: R. Oldenbourg, 2008.
- Bühner, Markus. *Einführung in die Test- und Fragebogenkonstruktion*. 3. Aufl. München: Pearson Studium, 2021.