

HSR

HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz



NOSERENGINEERING
WE KNOW HOW

Author(en):

Jan Aeberli & Jürg Schleutermann

Betreuer:

Prof. Dr. Luc Bläser

Projektpartner:

Noser Engineering AG

Studienarbeit

Einkaufsgestaltung für die
Migros mit HoloLens

Ein HoloLens-Prototyp für den Filialleiter der Zukunft

Inhaltsverzeichnis

1	Abstract	2
2	Management Summary	3
3	Rechtliche Hinweise	4
4	Projektübersicht	5
4.1	Zweck und Ziel	5
4.2	Lieferumfang	6
4.3	Annahmen und Einschränkungen	6
5	Use Cases	7
5.1	Einleitung	7
5.2	Möglichkeiten	7
5.3	Prototyp	9
6	Experimente	12
6.1	Testaufbau	12
6.2	Experiment 1: Bilderkennung	14
6.3	Experiment 2: Räumliches Verständnis	22
6.4	Experiment 3: Benutzeroberfläche	27
7	Prototyp	34
7.1	Konzept	34
7.2	Architektur	40
7.3	Probleme & Lösungsansätze	48
8	Schlussfolgerungen	54
8.1	Technologie	54
8.2	Entwicklungsumgebung	55
8.3	Use Case	55
8.4	Ausblick	57
	Anhang	58
A	Lieferumfang	58
B	Installationsanleitung	60
C	Entwicklungsumgebung	66
D	Glossar	70
E	Verzeichnisse	73
E.1	Abbildungsverzeichnis	73
E.2	Tabellenverzeichnis	74
E.3	Quellenverzeichnis	74

1 Abstract

Der Migros-Genossenschafts-Bund will untersuchen, inwiefern neue Technologien im Bereich der Mixed Reality zukünftig eingesetzt werden könnten. Dabei sollen die verschiedenen Aspekte der Technologien beleuchtet und konkrete Anwendungsfälle analysiert werden. In Zusammenarbeit mit der Noser Engineering AG soll eine Machbarkeitsstudie durchgeführt werden, um zu evaluieren, wie mit Hilfe der Microsoft HoloLens Brille die Gestaltung und Verwaltung einer Einkaufs-Filiale unterstützt werden kann, so dass ein möglichst grosser Nutzen gegenüber den klassischen Spreadsheet-Tabellen entsteht. Zusätzlich soll eine Einschätzung der Zukunftschancen dieser Technologie erarbeitet werden, um für Folgeprojekte eine Entscheidungsgrundlage bieten zu können.

In einer ersten Phase wurden gemeinsam mit dem Migros-Genossenschafts-Bund ein konkreter Anwendungsfall ausgearbeitet und die grössten technischen Probleme durch unabhängige Experimente minimiert. Mithilfe der Resultate aus den Experimenten wurde in einer zweiten Phase ein Prototyp umgesetzt. Der Prototyp verwendet die Bilderkennungssoftware Vuforia um Produkte zu identifizieren und nutzt das räumliche Verständnis der HoloLens, um sich den Aufbau einer Filiale zu merken. In der letzten Phase wurde ein Bericht verfasst, welcher die gewonnen Erkenntnisse festhält und einen Ausblick in die Zukunft der HoloLens gibt.

Der resultierende Prototyp demonstriert, wie die Arbeit eines Filialleiters in Zukunft durch Mixed Reality Geräte wie der HoloLens erleichtert werden könnte. Über die Brille werden Informationen zum Sortiment direkt in der Filiale dargestellt. Der Filialleiter erhält so einen besseren Überblick und bemerkt Abweichungen schneller. Weitere Anwendungen in diesem Umfeld sind bereits in naher Zukunft denkbar. Insbesondere wenn man die rasante technische Entwicklung der letzten Jahre und die Pläne diverser Hersteller berücksichtigt.

2 Management Summary

Ausgangslage

Mit vermehrtem Aufkommen neuer Produkte im Bereich von Mixed Reality stellt sich die Frage nach Anwendungsfällen, in denen uns solche Technologien im Alltag sinnvoll unterstützen können. Microsoft hat 2016 mit der HoloLens-Brille ein Produkt auf den Markt gebracht, das einen vielversprechenden Ausblick in eine mögliche Zukunft gibt. Diese Machbarkeitsstudie beschäftigt sich mit der Frage, inwiefern HoloLens im Bereich des Detailhandels zum Einsatz kommen könnte.

Vorgehen, Technologien

In Zusammenarbeit mit dem Migros-Genossenschafts-Bund wurden mögliche Anwendungsfälle für HoloLens diskutiert. Ein zukunftssträchtiges Einsatzgebiet wurde ausgewählt und im Zeitraum dieser Studienarbeit mit einem konkreten Prototyp umgesetzt. Die technische Machbarkeit der Idee wurde in einem ersten Schritt durch kleinere Experimente abgeklärt. Daraus resultierende Ergebnisse flossen im Anschluss in die eigentliche Realisierung des Prototyps mit ein. Gesammelte Erkenntnisse sind in der Dokumentation dieser Studienarbeit für mögliche Nachfolgeprojekte festgehalten.

Ergebnisse

Das Resultat dieser Studienarbeit ist ein Prototyp, mit dem die Verwaltung einer Filiale für das Personal vereinfacht wird. Das ausgearbeitete Konzept zeigt, wie täglich anfallende Aufgaben im Bereich der Lagerverwaltung oder Filialgestaltung zukünftig auf eine völlig neue Art und Weise ausgeführt werden könnten. Diese Arbeit belegt auch, dass es für ein Entwicklerteam mit wenig Erfahrung in diesem Bereich durchaus möglich ist, innerhalb von wenigen Wochen ein greifbares Konzept zu erarbeiten. Sie skizziert den Weg von der Idee auf Papier hin zum funktionierenden Prototyp, und die technischen sowie konzeptionellen Probleme, die dabei möglicherweise gelöst werden müssen.

Ausblick

Ähnlich wie die mobilen Geräte für die Hosentasche könnten solche optischen Helfer unseren Alltag komplett revolutionieren. Jeden Tag sammeln wir riesige Mengen an Daten, die uns beispielsweise im Arbeitsalltag unterstützen sollen. Doch was hilft diese Datenflut, wenn man sie danach nur auf einem Bildschirm betrachten kann? Viel interessanter wäre es doch, diese Informationen wieder zurück in die echte Welt zu bringen. Zurück in die Umgebung, für die wir Menschen eigentlich geschaffen sind. Eine Mischung aus Realität und kontextspezifischer Information erhöht unsere Effizienz und fordert weniger Vorstellungskraft. Technologien wie HoloLens sind ein wichtiger Schritt in die Zukunft, in der die Interaktion mit Computern natürlicher und intuitiver ablaufen wird.

3 Rechtliche Hinweise

Die Marken Migros, Farmer, Bon Chef, Migros Bio, Sun Queen sowie das orange M sind Eigentum des Migros-Genossenschafts-Bund. Die Verwendung erfolgte mit ausdrücklicher Zustimmung des Eigentümers der Markenrechte.

4 Projektübersicht



Quelle: Microsoft

Abbildung 1: Pressebild HoloLens

In den letzten Jahren hat sich die Technologie im Bereich Virtual Reality (VR) stark weiterentwickelt. Durch spezielle Hardware (meist in Form einer Brille) wird dem Anwender eine eigene - für ihn künstlich erschaffene Welt - präsentiert. Dabei ist vor allem wichtig, dass die Realität glaubwürdig erscheint und man regelrecht in die virtuelle Welt eintauchen kann. Diesen Effekt nennt man auch Immersion. Produkte wie “Oculus Rift” oder “HTC Vive” haben neben anderen zu einer weiten Verbreitung dieser Technologie beigetragen. Eine logische Weiterentwicklung der Virtual Reality stellt die Augmented Reality oder auch Mixed Reality dar. Diese hat nicht das Ziel, dem Anwender eine andere Realität zu präsentieren, sondern erweitert viel mehr die Wirklichkeit um kontextsensitive Informationen in Form von dreidimensionalen Objekten (Hologramme) oder durch textuelle Einblendungen. Augmented Reality (AR) könnte theoretisch in fast allen Bereichen des alltäglichen Lebens verwendet werden und so den Menschen in Zukunft bei unterschiedlichsten Aufgaben unterstützen. Medial stand vor allem Google mit der geplanten Entwicklung von “Google Glass” in den letzten Jahren stark im Fokus. Nun hat auch Microsoft eine eigene AR Brille veröffentlicht, mit welcher sich diese Studienarbeit im Detail auseinandersetzen wird.

4.1 Zweck und Ziel

Die Studienarbeit wird in Zusammenarbeit mit dem Industriepartner Noser Engineering AG durchgeführt. Im Rahmen einer Machbarkeitsstudie zusammen mit dem Migros-Genossenschaftsbund soll untersucht werden, inwiefern die neue Microsoft HoloLens Brille den Detailhandel bei der Gestaltung und Verwaltung von Filialen möglichst sinnvoll unterstützen kann. Es soll abgeklärt werden, was heute aus technischer Sicht bereits möglich ist und wie die Zukunft dieser Technologie aussehen könnte. Diese Arbeit soll als Basis für weiterführende Projekte im Bereich Mixed Reality verwendet werden können.

4.2 Lieferumfang

Der Lieferumfang dieses Projekts umfasst:

- Abstract und Management Summary
- Quellcode für Prototyp und Experimente
- Installationsanleitung
- Technischer Bericht inklusive Ausblick
- Projektjournal

Der Bericht wird dem betreuenden Dozenten in gedruckter Form abgegeben. Alle zusätzlichen Abgaben werden via CD/DVD/USB dem Ausdruck beigelegt. Auf Wunsch werden weitere Kopien an die beteiligten Personen dieses Projekts verteilt. Details zum Lieferumfang können Anhang A entnommen werden.

4.3 Annahmen und Einschränkungen

Das Ziel dieser Arbeit besteht nicht darin, eine marktreife Applikation zu produzieren. Es geht viel mehr darum, die Machbarkeit eines zu definierenden Anwendungsfalls mithilfe eines Prototyps zu belegen. Zudem hat keiner der Autoren Erfahrungen in der Entwicklung solcher Applikationen, wodurch das ganze Wissen im Laufe der Arbeit zuerst erarbeitet werden muss. Die festgehaltenen Erkenntnisse bzgl. der Machbarkeit sind somit nicht absolut und könnten unter Umständen durch jemanden mit mehr Erfahrung in Technologien dieser Art widerlegt werden. Ausserhalb der Systemgrenze und damit auch ausserhalb der Verantwortung dieses Projekts sind ausserdem:

- Schulungen von Mitarbeitern
- Beschaffung der Gerätschaften (HoloLens)
- Einrichten von Infrastrukturen für die Verwendung des Prototyps

5 Use Cases

5.1 Einleitung

Im ersten Moment schiessen einem sicherlich viele Ideen durch den Kopf, die sich mit Technologien wie HoloLens realisieren lassen würden. Viele davon beschreiben Funktionalitäten, die dem Träger der Brille in bestimmten Situationen des Alltags zusätzliche Informationen einblenden. Man muss allerdings beachten, dass man bei aller Euphorie die Sinnhaftigkeit in den Augen behält. Aus technischer Sicht ist heute bereits sehr vieles machbar. Ein grosser Teil der technische Spielereien, sind zwar im ersten Moment beeindruckend, stellen sich dann aber über einen längeren Zeitraum betrachtet doch eher als unpraktisch oder nicht wirklich hilfreich heraus.

Im Rahmen einer Vorabklärung hat man sich mit verantwortlichen Personen von Migros und Noser Engineering zusammengesetzt, um mögliche Anwendungsszenarien (nachfolgend als Use Case bezeichnet) im Kontext des Detailhandels zu diskutieren. Dabei hat sich relativ schnell herauskristallisiert, dass zum jetzigen Zeitpunkt eine Anwendung für Endkunden keinen Mehrwert bietet. Die HoloLens ist noch zu teuer, und man zieht damit als Träger auf eine unangenehme Art und Weise die Aufmerksamkeit auf sich, wenn man damit in die Öffentlichkeit tritt. Ähnlich wie bei Smartphones kann diese Situation allerdings schon in ein paar Jahren ganz anders aussehen.

Deshalb sollen im nachfolgenden Kapitel auch die Ideen kurz umschrieben werden, welche im Rahmen dieser Studienarbeit nicht weiterverfolgt wurden. So kann man in Zukunft darauf zurückgreifen, wenn sich Umstände und Technologie weiterentwickelt haben. Für den weiteren Verlauf der Arbeit liegt der Fokus dann eher auf **Anwendungen im professionellen Umfeld**, die weniger stark in der Öffentlichkeit stattfinden (beispielsweise ausserhalb der offiziellen Öffnungszeiten einer Filiale).

5.2 Möglichkeiten

Use Case 1: 3D Visualisierung des Einrichtungsplans

Die Planung einer neuen Migros Filiale steht an. Designer und Management möchten die zukünftige Filiale als Modell, mithilfe von HoloLens, gemeinsam auf einem Tisch platzieren und begutachten können. Das gibt ihnen vor den Baumassnahmen schon eine erste Idee, wie die Filiale später aussehen wird. Ähnlich wie das Modell bei einem Architekten. Sie können das Modell direkt verändern und individuelle Wünsche einbringen.

Use Case 2: 3D Visualisierung einer Ausstellungsfläche

Zu bestimmten Zeiten im Jahr (z.B. an Weihnachten) müssen Teile einer Filiale jeweils umgestaltet werden. Der Innenarchitekt muss eine Möglichkeit haben, diese speziellen Ausstellungsflächen (Kojen) zu visualisieren, um eine gemeinsame Planung zu ermöglichen. Mit Unterstützung von HoloLens werden die Kojen als holografisches Modell dargestellt.

Über die Planungssoftware des Architekten kann eine Koje während der Besprechung den Wünschen angepasst werden. Die Änderungen sind über die HoloLens für alle Beteiligten unmittelbar sichtbar.

Use Case 3: 3D Visualisierung eines Regals

Ein Filialleiter möchte zum Angebot stehende Produkte in die diversen Regale der Filiale verteilen können. Um sich die spätere Verteilung besser vorstellen zu können, verwendet er die HoloLens Brille, welche ihm ein holografisches Modell der Regale mit den darin angeordneten Produkten zeigt. Die Produkte können verschoben und umsortiert werden.

Use Case 4: Unterstützung bei der Bewertung einer Filiale auf dem Tisch

Ein Filialleiter möchte, dass seine Filiale optimal läuft. Produkte die schlecht laufen sollen aussortiert, spezielle Angebote prominent platziert und Besucherströme effizient und umsatzfördernd gelenkt werden. Um solche Optimierungen zu ermöglichen, verwendet der Filialleiter die HoloLens, welche ihm ein holografisches Modell der Filiale mit aktuellen Kennzahlen darstellt. Über dieses Modell kann der Filialleiter anschliessend seine Veränderungen planen und umsetzen.

Use Case 5: Unterstützung bei der Bewertung einer Filiale vor Ort

Ein Filialleiter möchte, dass seine Filiale optimal läuft. Produkte die schlecht laufen sollen aussortiert, spezielle Angebote prominent platziert und Besucherströme effizient und umsatzfördernd gelenkt werden. Er verwendet die HoloLens um sich direkt in der Filiale zu den Produkten bestimmte Zusatzinformationen wie z.B. Lagerbestand, Umsatz oder Vergleiche mit anderen Filialen darstellen zu lassen. Die Informationen werden direkt bei den Produkten dargestellt und ermöglichen ihm so eine direkte Analyse vor Ort.

Use Case 6: Visualisierung für Verpackungs-/Produktgestaltung

Produktentwickler erstellen das Design eines neuen Produkts in ihrer 3D Software. Über eine Integration können sie sich ihre Arbeit mit der HoloLens direkt als Hologramm in der echten Welt anschauen. Die Ansicht wird über mehrere Brillen hinweg synchronisiert, um das neue Design mehreren Personen in einer Präsentation oder Besprechung vorstellen zu können. Änderungsvorschläge werden in der 3D Applikation umgesetzt und sind anschliessend für alle Beteiligten gleich sichtbar.

Use Case 7: Der Möbelkonfigurator 2.0

Der Kunde kann auf der Webseite von Interio oder Micasa bestimmte Möbel selber mit wenigen Mausklicks konfigurieren und seinen Wünschen anpassen. Mit HoloLens kann er sich sein konfiguriertes Möbel in Echtgrösse innerhalb der eigenen vier Wände platzieren und es von allen Seiten begutachten. Die HoloLens aktualisiert die Ansicht, sobald

der Kunde im Konfigurator weitere Anpassungen vornimmt. Kleinere Anpassungen wie Farben oder Dekore können direkt mit der Brille vorgenommen werden.

Use Case 8: Einkaufshilfe vor Ort

Der Kunde kann in der HoloLens-Anwendung Angaben zu seinem Essverhalten sowie mögliche Allergien festhalten. Sobald er mit der Brille eine Filiale betritt, werden ihm beim Blick auf ein Regal bestimmte Produkte basierend auf seinen Einstellungen eingefärbt. Ein Allergiker erkennt dadurch sofort Produkte, die für ihn gefährlich sind. Auch für Vegetarier oder Veganer wird der Einkauf dadurch einfacher.

Use Case 9: Navigation durch die Filiale

Der Kunde synchronisiert seine Einkaufsliste mit der HoloLens. Beim Betreten einer Filiale wird für die aufgelisteten Produkte automatisch eine Route durch den Laden berechnet, die ihn zusätzlich an bestimmten Aktionen oder Angeboten vorbeiführt (konfigurierbar durch Filialleitung). Während seinem Einkauf weist ihm die Brille den Weg zum nächsten Produkt, analog zu einem Navigationsgerät im Auto.

5.3 Prototyp

Basierend auf den gesammelten Ideen und nach weiteren Diskussionen mit Migros wurde ein Use Case definiert, welcher als Prototyp für HoloLens während dieser Arbeit realisiert werden soll. Dabei wird man sich mit Problemen sowohl technischer Art als auch mit Fragen zur Benutzerfreundlichkeit weiter auseinandersetzen.

Die Beschreibungen aus Benutzersicht (nachfolgend User Story) bei der Verwendung des Prototyps wurden von den verantwortlichen Personen bei Migros ausgearbeitet. Daraus lassen sich die zu lösenden technischen Probleme sowie von externen Schnittstellen benötigte Daten ableiten.

User Story 1: Filialen Rundgang

Der Leiter einer Filiale oder Rayonchef begeben sich auf einen Rundgang durch ihre Filiale bzw. ihr Rayon. Es gilt sich einen schnellen Überblick darüber zu verschaffen, wie sich die Sortimente in den letzten Wochen sowie zur vergleichbaren Vorjahresperiode entwickeln. Der Blick vom Eingang aus über die Abteilungen und Regale hinweg zeigt dank entsprechender Farbgebung und den eingeblendeten Werten wo sich der Umsatz und die Frequenz (SLW) positiv oder negativ entwickelt haben. Beim genaueren Blick auf das Kaffeeregal im sich positiv entwickelnden Rayon Frühstück/Warmgetränke zeigt sich aber ein etwas nuancierteres Bild: nicht alle Artikel verkaufen sich gleich gut. Da gibt es Kaffeesorten, die hervorragend laufen und andere, die deutlich rückläufig sind.

User Story 2: Überprüfung der Bestände

Der Rayonchef ist dabei, die Bestände zu prüfen und die Mengen für die heutige Bestellung festzulegen. Dazu reicht ein Blick ins Regal: die grün eingefärbten Produkte sind noch in genügender Zahl vorhanden. Deshalb wendet er sich den gelb und rot eingefärbten zu. Dank der eingeblendeten Werte sieht er, wie hoch die Bestände noch sind und passt auf Basis dieser Information die vorgeschlagenen Bestellmengen noch an. Ganz besonders hilfreich ist die neue HoloLens-Brille bei den Frischeartikeln, die dank dynamischer Prognosewerte besonders effektiv planbar sind. Bei diesen muss eigentlich nur noch bestellt werden, die vorgeschlagenen Bestellmengen entsprechen immer sehr genau den Bedürfnissen.

Für den Benutzer wird es drei verschiedene Ansichten geben, die er bei der Verwendung des Prototyps in einer Filiale präsentiert bekommt. Die Ansichten unterscheiden sich in der Art, wie Informationen mittels HoloLens in die reale Welt eingebettet werden, und welchen Detailgrad diese Informationen aufweisen. Ausserdem kommen für die Realisierung der Ansichten jeweils unterschiedliche technische Fähigkeiten von HoloLens zum Einsatz.

1. **Übersicht Filiale:** Der Benutzer steht auf einem leicht erhöhten Punkt oder an einer anderen Stelle, von der aus er eine gute Übersicht über die Filiale hat und zusammengefasste Informationen zu Produkten oberhalb der Regale ablesen kann (tiefster Detailgrad)
2. **Übersicht Regal:** Der Benutzer steht direkt vor einem Regal und erhält durch die darauf/daneben platzierten Informationen eine etwas genauere Übersicht (mittlerer Detailgrad)
3. **Übersicht Produkt:** Der Benutzer richtet seinen Blick auf ein ganz bestimmtes Produkt im Regal, um die dazugehörigen Daten etwas genauer studieren zu können (höchster Detailgrad)

5.3.1 Technische Probleme

1. **Optische Erkennung von Objekten:** In der HoloLens sind diverse Kameras verbaut. Diese eignen sich nicht nur für die Vermessung und die Orientierung, sondern auch für die Erkennung bestimmter Gegenstände. In diesem Fall sind das Produkte in einem Regal, zu denen die Brille nach der Erkennung bestimmte Informationen darstellen soll. Lösungsansätze und gewonnene Erkenntnisse dazu in Kapitel 6.2.
2. **Ortung im Raum:** Die zu visualisierenden Produktinformationen beziehen sich grösstenteils auf eine einzelne Migros Filiale. Die HoloLens muss also imstande sein, eine Filiale eindeutig zu identifizieren, um dann die dazugehörigen Daten abrufen zu können. Für die Visualisierung braucht es ausserdem Wissen über die Position und Ausrichtung von Regalen. Dasselbe gilt für die Organisation der Produkte innerhalb einzelner Regale. Lösungsansätze und gewonnene Erkenntnisse dazu in Kapitel 6.3.
3. **Gestaltung einer Benutzerumgebung:** Zur benutzerfreundlichen Gestaltung grafischer Oberflächen für Bildschirme hat man bereits relativ viel Erfahrung gesammelt. Es gibt

klar definierte Punkte, die man dabei beachten sollte. Nun befinden sich die Bedienelemente plötzlich in der echten Welt, dem Benutzer als Hologramme auf der Brille dargestellt. Es stellt sich deshalb die Frage, wie man die Bedienung für den Benutzer in dieser Mischung aus künstlicher und echter Welt gestaltet. Lösungsansätze und gewonnene Erkenntnisse dazu in Kapitel 6.4.

4. **Anbindung an externe Datenquellen:** Der Prototyp benötigt eine Anbindung an externe Datenquellen. Er bezieht darüber Informationen zur Filiale sowie zu einzelnen Produkten, die darin zum Verkauf angeboten werden. Somit ist sichergestellt, dass immer der aktuellste Stand der Informationen genutzt wird (z.B. Mengenvorschläge für Lagerbestand). Die WLAN Schnittstelle der HoloLens kommt für diese Anbindung zum Einsatz. Solange in den Filialen eine Internetverbindung zur Synchronisation vorhanden ist, sollte das technische Risiko relativ klein sein.

5.3.2 Benötigte Daten

Der Prototyp wird für seine Visualisierungen eine Schnittstelle kontaktieren, die ihm Daten zu einer oder mehreren Filialen liefern kann. Da während dieser Arbeit in erster Linie eine Machbarkeitsstudie entsteht, wurden gewisse Annahmen zur Verfügbarkeit von Daten getroffen. Während der Entwicklung werden diese simuliert bzw. statische Werte dafür eingesetzt.

- Eindeutiges Identifikationsmerkmal pro Filiale, damit der Prototyp weiss, welche Daten er abfragen muss
- Auflistung der Regale in einer Filiale sowie eine Beschreibung der darin enthaltenen Produkte (inkl. deren Anordnung)
- Definition bestimmter Produktgruppen, die in der Übersicht eines Regals zusammengefasst werden sollen (bspw. Kaffee)
- Umsatzzahlen für einzelne Produkte bzw. Produktgruppen sowie die Umsatzentwicklung über einen längeren Zeitraum
- Soll- und Meldebestände für einzelne Produkte bzw. Produktgruppen
- Zusätzliche Informationen zu einzelnen Produkten (z.B. Verderb)
- Metainformationen zu den Daten (bspw. ab wann gilt ein Wert als besonders negativ oder positiv)

6 Experimente

6.1 Testaufbau

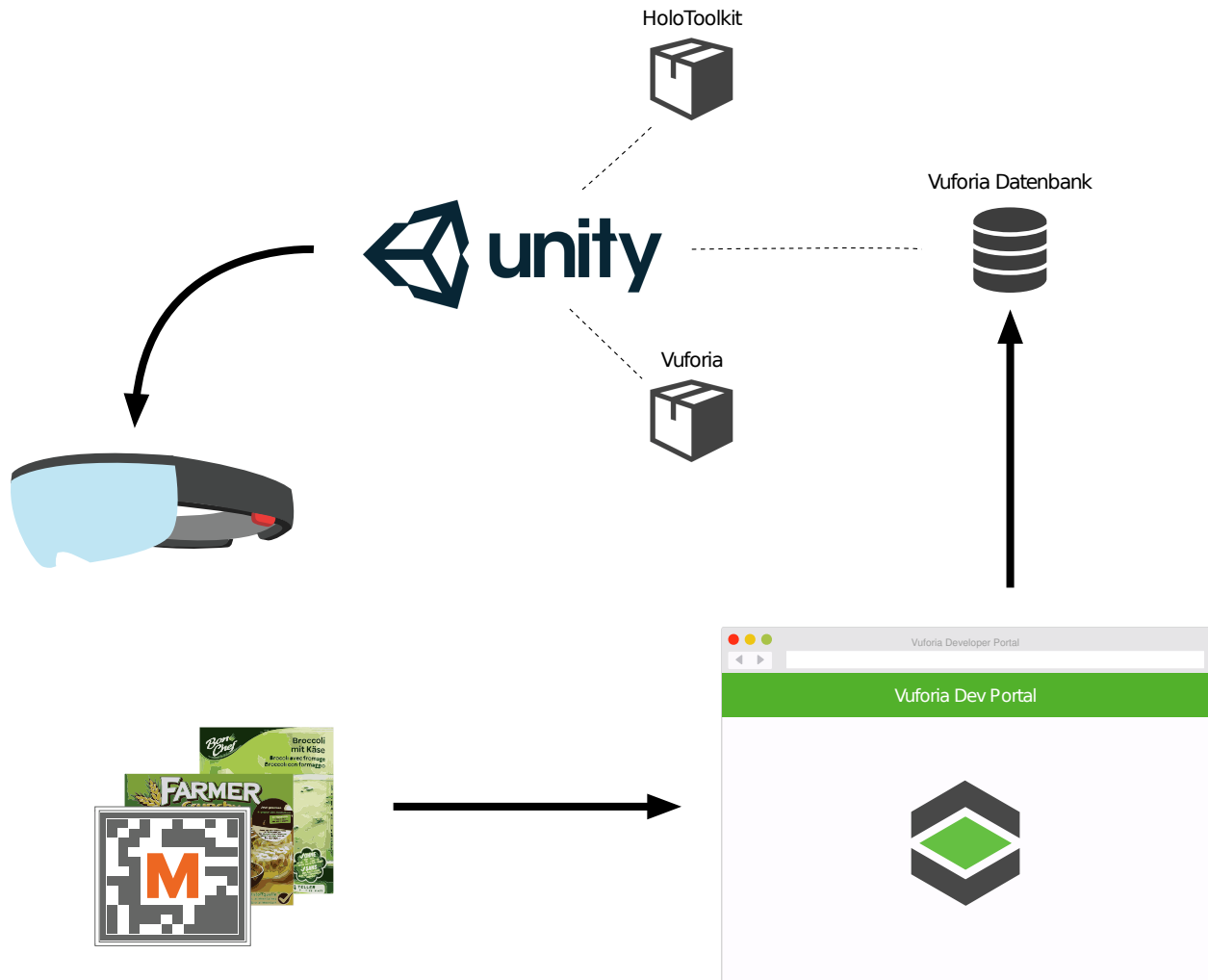


Abbildung 2: Schema für die Komponenten der HoloLens Applikation

Für die durchgeführten Experimente wurde immer etwa der gleiche Aufbau verwendet. Microsoft empfiehlt für die Entwicklung von HoloLens-Anwendungen die Verwendung von Unity. In der Dokumentation zu HoloLens finden sich alle nötigen Schritte zur Einrichtung der Entwicklungsumgebung. [Mic17h]

Als Ergänzung dazu gibt es HoloToolkit für Unity auf GitHub. Das Toolkit enthält eine Vielzahl an vorgefertigten Assets, die man mit wenigen Mausklicks in die eigene HoloLens Applikation integrieren kann. Dadurch hat man gleich Beispiele, wie die verschiedenen Funktionen von HoloLens angesteuert werden. [Mic17f]

Mit Vuforia wird die Erkennung von Objekten über die eingebaute Kamera der HoloLens ermöglicht. Gegen Ende des Jahres 2016 wurde von den Entwicklern offiziell die Unterstützung

für HoloLens angekündigt. In Zusammenarbeit mit Unity wurde die Integration so stark vereinfacht, dass man als Entwickler selbst mit wenig Vorkenntnissen innerhalb relativ kurzer Zeit beeindruckende AR Applikationen inklusive Bilderkennung für HoloLens erstellen kann. [\[Vuf17a\]](#)

Als Ausgangspunkt wurden die von den jeweiligen Herstellern zur Verfügung gestellten Beispiele verwendet.

6.2 Experiment 1: Bilderkennung

Motivation

HoloLens bietet eine gute Basis für die Realisierung der eigenen AR Applikation. Vieles wird bereits automatisch gemacht, bietet aber gleichzeitig durch Schnittstellen die nötige Erweiterbarkeit. Im Bezug auf die Entwicklung des finalen Prototyps dieser Studienarbeit sieht man sich allerdings mit einem sehr entscheidenden Problem konfrontiert: - HoloLens kann zwar darüber Auskunft geben, welche Dimensionen ein Objekt hat und wo es sich ungefähr befindet (Siehe Kapitel 6.3). Offen bleibt allerdings die Frage, um was es sich bei diesem Objekt handelt. Darum muss man sich in einer Applikation selber kümmern. Mögliche Lösungsansätze dafür wurden während dieses Experiments untersucht.

Ansatz

Bilderkennung ist aus technischer Sicht eine relativ komplizierte Angelegenheit und erfordert verhältnismässig viel Rechenaufwand. Im Prinzip hat ein Programm nur die einzelnen Farbanteile (Pixel) eines Bildes vor sich, und muss basierend auf deren Anordnung und Farbe gewisse Formen oder Gruppierungen herauslesen. Die erkannten Merkmale ergeben dann als Ganzes einen Kontext, dem man ein Bild zuordnen kann (z.B. auf dem Bild befindet sich eine Katze). In den letzten Jahren wurden in diesem Bereich durch Einsatz künstlicher Intelligenz sehr beeindruckende Fortschritte erzielt. Eigene Applikationen lassen sich mit Funktionsbibliotheken verschiedenster Hersteller ebenfalls mit dieser Fähigkeit ausstatten. Eine Eigenentwicklung wäre also für den Prototyp völlig unverhältnismässig und würde zudem den Rahmen der Studienarbeit sprengen. Aus diesem Grund liegt der Fokus bei diesem Experiment eher auf der **Evaluation** sowie der **Integration** einer solchen Bibliothek in eine Applikation für HoloLens.

- **Barcode:** Ein häufig anzutreffender Ansatz für die Erkennung von Objekten ist der Einsatz von Barcodes. Die dafür verwendeten Muster bestehend aus schwarzen und weissen Flächen weisen einen hohen Kontrast auf und erleichtern somit die Analyse durch einen Computer. Jedes Produkt in einer Migros Filiale verfügt bereits über einen solchen Barcode, womit es sich an der Kasse eindeutig identifizieren lässt. Allerdings wird bei dieser Methode vorausgesetzt, dass der Barcode zum lesenden Gerät (beispielsweise zur Kamera eines Mobiltelefons) ausgerichtet wird. Oder genauer, der Benutzer muss ein Produkt in die Hand nehmen, um dessen Erkennung zu ermöglichen, denn Barcodes werden in den seltensten Fällen auf der Vorderseite eines Produkts angebracht. Im Bezug auf den Prototyp würde das bedeuten, dass der Filialleiter jedes Produkt vor sein Gesicht halten muss, damit es von der Brille erkannt wird. Eigentlich wäre es viel praktischer, wenn man das Produkt direkt im Regal anschauen kann, ohne es in die Hand nehmen zu müssen. Aufgrund dessen wurde dieser Ansatz während dem Experiment nicht genauer verfolgt, wobei er für andere Anwendungsfälle durchaus besser geeignet sein kann.
- **Vuforia:** Viele der gezeigten Anwendungsbeispiele gehen in eine ähnliche Richtung wie der Prototyp, welcher am Ende dieser Studienarbeit gezeigt werden soll. Deshalb wurden die Versuche dieses Experiments darauf ausgelegt, die Funktionsweise und den Umfang von Vuforia sowie das Zusammenspiel mit HoloLens genauer zu untersuchen. Zu-

dem galt es abzuklären, ob sich eine zuverlässige optische Erkennung von Produkten verschiedenster Farben und Formen überhaupt damit realisieren lässt.

Bemerkung

Neben Vuforia gibt es natürlich auch noch diverse andere Anbieter von Frameworks, die einen ähnlichen Umfang an Funktionalität bieten. Ein Beispiel dafür ist Catchoom¹, deren Framework zurzeit in der Smartphone App der Migros zur Erkennung von Produkten mittels eingebauter Kamera eingesetzt wird.

Der Fokus lag bei diesem Experiment vor allem auf der Erkennung der Frontansicht eines Produkts, da solche im Regal typischerweise aus ästhetischen Gründen nach vorne ausgerichtet werden. Oft wird das gleiche Produkt sogar mehrfach nebeneinander platziert, was die Chancen einer Erkennung direkt im Regal erhöht, selbst wenn ein Exemplar etwas verschoben oder verdreht im Regal steht. Ein ebenfalls wichtiger Aspekt ist die Identifizierung der Migros Filiale. Trotz Bilderkennung ist es relativ schwer, eine Filiale nur Anhand eines Fotos eindeutig einzuordnen. Schliesslich finden in der Anordnung von Regalen und Produkten je nach Saison laufend Änderungen statt. Deshalb benötigt man ein Identifikationsmerkmal, das möglichst immer am gleichen Ort ist und unverändert bleibt. Vuforia bietet verschiedene Techniken zur Erkennung von Objekten via *Targets* an. Drei davon wurden in diesem Experiment ausprobiert.

Versuch - VuMark Targets

Beschreibung: Für die Erkennung wird ein spezieller Marker verwendet, analog zu einem 2D Barcode, wie man ihn öfters auch auf Werbeplakaten oder Flyern antrifft. Der Unterschied besteht darin, dass der Marker bis auf ein paar technische Einschränkungen relativ frei gestaltet werden kann. Vuforia bietet dafür ein Hilfswerkzeug, welches einem bei der Gestaltung eines eigenen Markers unterstützt und sicherstellt, dass dieser später auch erkannt wird. [Vuf17d]

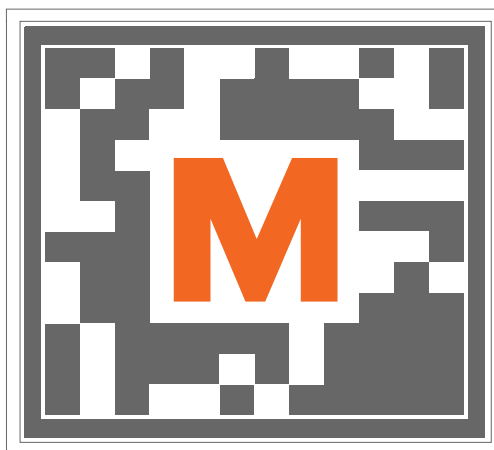


Abbildung 3: Eigens gestalteter Marker für das Experiment

¹Hersteller eines Frameworks mit ähnlichem Funktionsumfang wie Vuforia. [Cat17]

Abbildung 3 zeigt den Marker, welcher während des Versuchs erstellt wurde. In diesen Marker lassen sich Informationen einbetten, konkret sind das in diesem Fall insgesamt 10 ASCII Zeichen. Dafür werden die einzelnen dunklen Rechtecke ein- bzw. ausgeschaltet, womit sich Informationen binär darstellen lassen. Im Prinzip gestaltet man also eine Vorlage für den Marker, und Vuforia generiert daraus viele, leicht abgewandelte Instanzen, die einen vorgegebenen Text repräsentieren. Der gezeigte Marker enthält beispielsweise den Text "Test". Die Dimensionen eines Markers ergeben sich grundsätzlich aus der Menge der darin zu speichernden Informationen sowie der Distanz, ab der ein Marker erkannt werden muss.

Auswertung: Während des Versuchs hat sich gezeigt, dass Vuforia den Marker mit seinem *Inhalt von 10 Zeichen* bei einer *Grösse von 10 x 10 cm* ab einer *Distanz von ca. 1 Meter* gut erkennt. Zudem funktioniert die Erkennung auch aus verschiedenen Blickwinkeln (z.B. schräg von der Seite). Die Messresultate waren allerdings stark abhängig von den Lichtverhältnissen, bei wenig Licht oder bei starken Lichtreflektionen sinkt die Zuverlässigkeit dementsprechend. Man kann allerdings davon ausgehen, dass eine Filiale der Migros typischerweise gut beleuchtet ist, um die angebotenen Produkte hervorzuheben. Problematisch wird die Erkennung erst dann, wenn ein Teil des Markers verdeckt ist (z.B. durch Befestigung mit einem Magneten an der Wand oder durch einen Finger).

Bemerkung

Beim Ausdrucken der Marker muss zwingend darauf geachtet werden, dass das Seitenverhältnis des dafür in der Datenbank hinterlegten Bildes beibehalten wird. Der Marker wird zwar trotzdem noch von Vuforia erkannt, allerdings liefert die Bibliothek bei der Erkennung eine falsche Ausrichtung des Markers im Raum. Dies ist sehr wahrscheinlich darauf zurückzuführen, dass anhand der gemessenen Seitenlängen nach der Erkennung die Ausrichtung des Markers berechnet wird.

Eine ausführliche Anleitung zur Erstellung sowie Beschreibungen zur Funktionsweise von VuMarks findet man auf der Webseite von Vuforia. [[Vuf17e](#)]

Versuch - Image Targets

Beschreibung: Die Erkennung findet über ein vorgegebenes Bild statt, also beispielsweise die Vorderseite eines Produkts aus der Migros Filiale. Sobald Vuforia das Bild wiedererkennt kann man davon ausgehen, dass der Benutzer das dazugehörige Produkt gerade vor sich hat. Soll ein Produkt aus verschiedenen Perspektiven erkannt werden, müssen dementsprechend viele Bilder davon in die Datenbank von Vuforia eingepflegt werden. Während diesem Versuch wurden Produkte unterschiedlicher Art verwendet, um die Zuverlässigkeit der Erkennung besser einschätzen zu können.



Abbildung 4: Für die Erkennung aufbereitete Fotos der getesteten Produkte

Produkt	Zweck
Getreideriegel	Prüfung der eigentlichen Frage, ob sich ein Produkt nur anhand eines Fotos seiner Vorderseite mit Vuforia erkennen lässt. Die boxförmige Verpackung bietet dafür eine gute Ausgangslage.
Tütensuppe Broccoli / Spargel	Evaluation möglicher Fehltreffer, was die Erkennung von Produkten betrifft, die sehr ähnlich aussehen. Zudem haben die Verpackungen eine glänzende Oberfläche, deren Aufdruck bei direkter Lichteinstrahlung schon für das menschliche Auge schwer erkennbar wird.
Olivenöl	Die verwendete Flasche besteht aus dunklem Glas und ist sehr schmal. Dementsprechend klein ist auch die darauf angebrachte Beschriftung, die so ziemlich das einzig brauchbare Erkennungsmerkmal darstellt. Dazu kommt, dass die Beschriftung beim Fotografieren von vorne Verzerrungen aufgrund der Rundung der Flasche aufweist.

Tabelle 1: Übersicht der getesteten Produkte für die Bilderkennung

Bemerkung
Bei der Erkennung über Produktfotos steckt der Teufel im Detail (unabhängig vom verwendeten Framework). Wenn Produktlogos auf den Verpackungen vorhanden sind, kann dies dazu führen, dass die Erkennung zu schnell anschlägt und dadurch Produkte falsch identifiziert werden. Insbesondere sehr ähnlich aussehende Produkte wie beispielsweise die M-Budget Linie stellen in manchen Fällen eine Problem dar. Aus diesem Grund ist für die Einpflegung eines kompletten Sortiments zur Erkennung immer noch viel Handarbeit bei der Erstellung und Aufbereitung von Produktfotos nötig.

Auswertung: In einem ersten Anlauf wurden die Produktfotos mit einem Smartphone erstellt. Die Erkennung hat damit bereits funktioniert, allerdings war sie nur bei den Getreideriegeln wirklich zuverlässig. Im zweiten Test wurden die Produktfotos mit einer Spiegelreflexkamera

und optimierter Beleuchtung erstellt. Nach zusätzlicher Aufbereitung in einem Bildbearbeitungsprogramm funktionierte die Erkennung für alle getesteten Produkte, wobei Abstand und Blickwinkel beim Betrachten des Olivenöls deutlich kritischer sind als bei anderen Produkten.



Abbildung 5: Der Weg vom Produktfoto zum Image Target für Vuforia

Bezüglich der Aufbereitung der Fotos hat sich gezeigt, dass man vor allem durch Erhöhung des Kontrastes schlagartig bessere Resultate erzielt. Das lässt sich dadurch erklären, dass für die Erkennung auffällige Merkmale (sogenannte *Features*) aus einem Bild extrahiert werden. Solche Merkmale ergeben sich durch auffällige Farbunterschiede bzw. Kanten im Bild, also beispielsweise die Umrisse einer schwarzen Schrift aufweissem Hintergrund (siehe Abbildung 5 rechte Seite). Je mehr Features ein Bild aufweist, desto höher die Chancen, dass es korrekt wiedererkannt wird. Deshalb spielt die Qualität der Fotos, aus denen diese Features extrahiert werden eine entscheidende Rolle. Durch Erhöhung des Kontrastes hat sich die Anzahl Features tendenziell deutlich erhöht, was sich sehr positiv auf die Erkennung ausgewirkt hat. Objekte wurden schneller und zuverlässiger auch aus grösserer Distanz oder anderen Blickwinkeln erkannt.

Weitere Tricks zur Optimierung sowie Hinweise für die Erkennung von Image Targets können der Vuforia Dokumentation entnommen werden. [Vuf17b]



Abbildung 6: Das Produkt wird anhand seiner Vorderseite erkannt

Abbildung 6 zeigt ein beeindruckendes Beispiel für die Zuverlässigkeit der Erkennung. Die Aufnahme stammt von einer kleinen USB Kamera mit massiv schlechterer Bildqualität als die in

der HoloLens verbaute Kamera aufweist. Trotz schrägem Blickwinkel, leichten Lichtreflektionen auf der Verpackung und schlechter Bildqualität werden die Getreideriegel immer noch anhand eines Fotos ihrer Vorderseite erkannt. Die Erkennung wird in diesem Fall durch Platzieren eines Würfels auf dem erkannten Objekt symbolisiert.

Versuch - Object Targets

Beschreibung: Über eine Android App von Vuforia können Objekte mit der Kamera des Smartphones oder Tablets aus allen Perspektiven erfasst und im Anschluss für deren Erkennung verwendet werden. [Vuf17d]

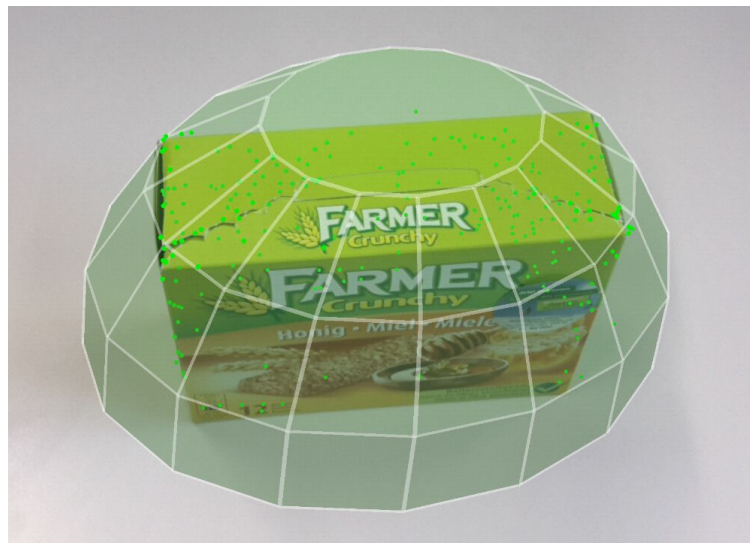


Abbildung 7: Erfassung eines Produkts aus verschiedenen Perspektiven

Abbildung 7 zeigt die dreidimensionale Erfassung eines Produkts. Über das Objekt wird eine Art Haube gelegt, bestehend aus Flächen, deren Farbe den Status der Erkennung aus der entsprechenden Perspektive symbolisiert. Es müssen nicht zwangsweise alle Perspektiven erfasst werden, sondern nur die, aus denen Vuforia das Objekt später erkennen soll. Die eingezeichneten Punkte repräsentieren Merkmale, an denen das Objekt wiedererkannt wird. Beim Versuch hat sich gezeigt, dass die Erfassung aus allen Perspektiven viel Zeit in Anspruch nehmen kann. Für ein einzelnes Produkt lag der Aufwand bei rund zwei Stunden.

Bemerkung

Man muss dazu noch erwähnen, dass diese Methode eigentlich fast schon eine professionelle Ausrüstung voraussetzt. Die Ausleuchtung des Objekts spielt hier eine wichtige Rolle. Zudem darf das Objekt während der Erfassung keinesfalls bewegt werden. Somit muss man sich mit der Kamera ständig um das Objekt herum bewegen, um es aus allen Perspektiven erfassen zu können. In einer dafür geschaffenen Umgebung würde der Vorgang vermutlich weniger lange dauern und das Ergebnis wäre überzeugender.

Auswertung: Trotz anfänglicher Schwierigkeiten bei der Erfassung war das Ergebnis im Endeffekt zufriedenstellend, das Produkt wurde aus den meisten Perspektiven zuverlässig durch

Vuforia erkannt. Abbildung 8 zeigt Auszüge der Erkennung im Testmodus der App, mit der das Produkt zuvor erfasst wurde.



Abbildung 8: Das Produkt wird aus verschiedenen Perspektiven erkannt

Vorteilhaft ist zwar, dass die Erkennung aus verschiedenen Perspektiven funktioniert. Damit müsste man sich also nicht darauf verlassen, dass Produkte in einem Regal immer nach vorne ausgerichtet sind. Eine solche Erfassung aller Produkte bei Migros scheint allerdings aufgrund des Zeitaufwands eher unrealistisch, selbst wenn man den Ablauf durch bessere Ausrüstung und mehr Erfahrung optimiert. Ausserdem hat sich beim vorhergehenden Versuch mit Image Targets gezeigt, dass die Erkennung der Vorderseite auch dann noch funktioniert, wenn das Produkt leicht verdreht zur Kamera positioniert wird.

Resultat

Im Endeffekt haben sich im Laufe dieses Experiments drei wichtige Erkenntnisse herauskristallisiert:

1. **VuMark Targets** eignen sich, um eine Migros Filiale oder ein spezifisches Regal innerhalb einer Filiale zu identifizieren. Es können darin Informationen eingebettet werden, die von einer HoloLens-Applikation interpretiert und entsprechend verarbeitet werden können. Die Marker nehmen nicht viel Platz in Anspruch und können nach Bedarf so unauffällig gestaltet werden, dass sie nicht sofort den Blick des Kunden im Laden auf sich ziehen.
2. **Image Targets** eignen sich, um mit verhältnismässig kleinem Aufwand eine Erkennung von Produkten für HoloLens zu ermöglichen. Mit einfachen Fotos als Grundlage erkennt Vuforia ein Produkt zwar nicht aus allen Perspektiven, doch nur schon die Erfassung der Vorderseite gibt einem eine zuverlässige Erkennung aus diversen Blickwinkeln. Für den Prototyp ist diese Methode hinreichend, um das Prinzip aufzuzeigen.
3. Die Erkennung von Targets mit Vuforia kann über eine **lokale Datenbank** realisiert werden, wodurch nicht permanent Anfragen an einen Server gemacht werden müssen, bevor ein Objekt möglicherweise erkannt wird.

Zudem wurde festgestellt, dass das Zusammenspiel zwischen Vuforia, Unity und HoloLens gut funktioniert. In der Dokumentation findet man viele Anleitungen, und die vorgefertigten Beispiele geben einem eine gute Grundlage für eigene Experimente. Sichtbare Reaktionen auf

die Erkennung von Objekten der realen Welt intensivieren das Erlebnis mit HoloLens für den Benutzer ungemein.

6.3 Experiment 2: Räumliches Verständnis



Abbildung 9: Studienarbeitszimmer aus der Sicht von HoloLens

Motivation

Für die drei Ansichten des in Kapitel 5.3 beschriebenen Use Case müssen Hologramme in der Filiale korrekt platziert und gespeichert werden können. In diesem Experiment wird untersucht, wie Hologramme mit der realen Welt verbunden werden können, so dass sie sich bei erneutem Starten der Applikation wieder am gleichen Ort befinden. Dabei soll die Positionierung im virtuellen Raum möglichst exakt mit der realen Welt übereinstimmen. Zudem wird untersucht, wie die Stabilität der Hologramme gewährleistet werden kann. Dazu werden verschiedene Versuche durchgeführt und die Resultate entsprechend ausgewertet.

Ansatz

Die HoloLens kann sich bereits dank der eingebauten Sensoren und Kameras im Raum orientieren und Dimensionen von Räumen und Hindernissen vermessen. Man kann sich das in etwa vorstellen wie ein virtuelles Netz, das über die Umgebung des Trägers gelegt wird (*Spatial Mapping Mesh*). Dieses Netz kann vom Entwickler einer Applikation abgefragt und analysiert werden. So lassen sich beispielsweise Flächen finden, auf denen Hologramme platziert werden können. Bewegt man sich nun in der realen Welt um eine solche Fläche herum, lässt sich ein darauf platziertes Hologramm von allen Seiten betrachten. Dieser beeindruckende Effekt erfordert eine regelmässige Lokalisierung der Brille bzw. des Benutzers im Raum. Die HoloLens nimmt einem auch diese Aufgabe ab durch Erkennung von optischen Merkmalen (Mustern) sowie gemessenen Distanzen zu Hindernissen wie beispielsweise Wänden oder Möbeln. Für dieses Experiment wird davon ausgegangen, dass der Raum (später dann die Filiale) bereits ausreichend erfasst wurde. Für die Erfassung genügt es, sich während einer gewissen Zeit (je nach Raumgrösse und Komplexität) in einem Raum aufzuhalten und sich mit aufgesetzter HoloLens umzuschauen. Mehr Informationen dazu in der Microsoft Dokumentation. [[Mic17k](#)]

Bemerkung

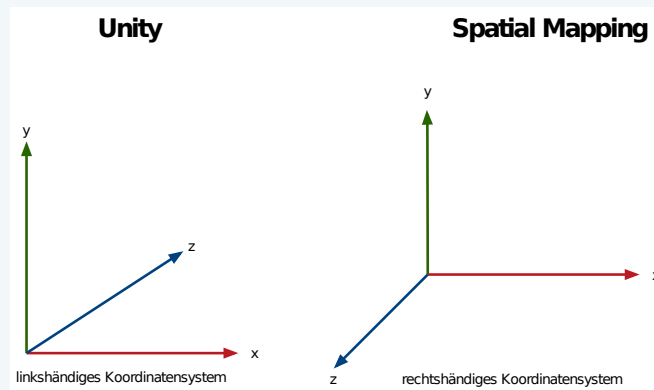


Abbildung 10: Ausrichtung Koordinatensysteme

Zu beachten ist ausserdem, dass sich die Ausrichtungen der Koordinatensysteme von Unity und dem des Spatial Mapping der HoloLens unterscheiden. Unity verwendet ein *linkshändiges* und die HoloLens ein *rechtshändiges* Koordinatensystem.

Versuch - Koordinatensystem pro Filiale

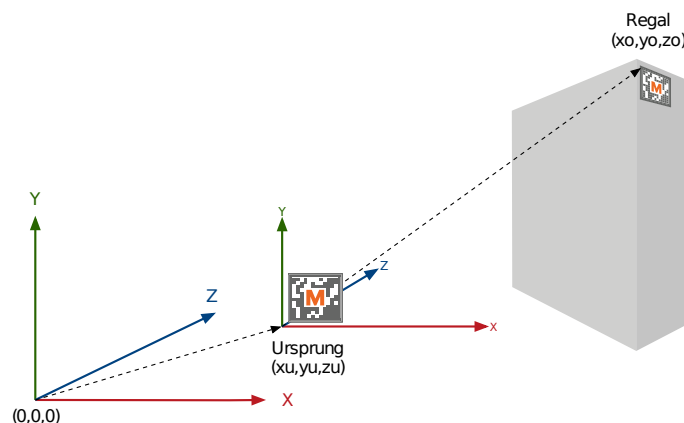


Abbildung 11: Regale werden relativ zu einem gemeinsamen Ursprung ausgerichtet

Beschreibung: In einem ersten Versuch wird ein Marker (VuMark), welcher über Bilderkennung lokalisiert und identifiziert wird, als globaler Ursprung definiert. Praktischerweise kann dieser Marker gleichzeitig zur Identifikation einer Filiale verwendet werden. Es ist essenziell, dass der Marker sich immer am gleichen Ort befindet, so dass Positionen im Verständnis von HoloLens mit Positionen in der realen Welt verbunden werden können. Anschliessend werden weitere Marker an den Regalen zur Identifikation gescannt. Dabei wird die Position jedes Regals relativ zum globalen Ursprung gespeichert und in einer separaten Datei abgelegt (siehe Abbildung 11). Um die gespeicherten Positionen während des Versuchs mit der jeweils erwarteten Position in der realen Welt abgleichen zu können, wird an der entsprechenden Stelle

in Unity ein einfacher Block dargestellt. Beim nächsten Start der Applikation muss dann der globale Marker für die Filiale erneut gescannt werden, um den Ladevorgang für die gespeicherten Regale auszulösen. Mithilfe der absoluten Positions- bzw. Rotationsdaten des Markers und den relativen Positionen der gespeicherten Regale wird das Koordinatensystem der Filiale zusammengesetzt und anschliessend verschoben und rotiert, so dass die Regale in der realen Welt wieder am richtigen Ort sind. Wenn alles funktioniert hat, müssen die Blöcke in der realen Welt wieder exakt auf den Markern dargestellt werden, die zuvor für die Erfassung der Regale gescannt wurden.

Auswertung: Leider hat dieser Ansatz nicht funktioniert. Das Koordinatensystem von Unity verändert sich bei jedem Start der Applikation. Die Achsen und der Ursprung sind bei jedem Start - im Vergleich zur vorherigen Ausführung - jeweils verschoben und rotiert. Das Koordinatensystem wird nach der Position und der Orientierung des Benutzers ausgerichtet, wobei die Brille selbst den Ursprung dieses Systems darstellt. Die Hologramme müssen aber ihre Position relativ zur realen Welt beibehalten können. Die Positions- und Rotationsdaten in Unity beziehen sich allerdings immer auf das aktuell gültige Koordinatensystem und sind bei einem erneuten Aufruf der Applikation somit unbrauchbar.

Versuch - Ursprung im Raum verankern

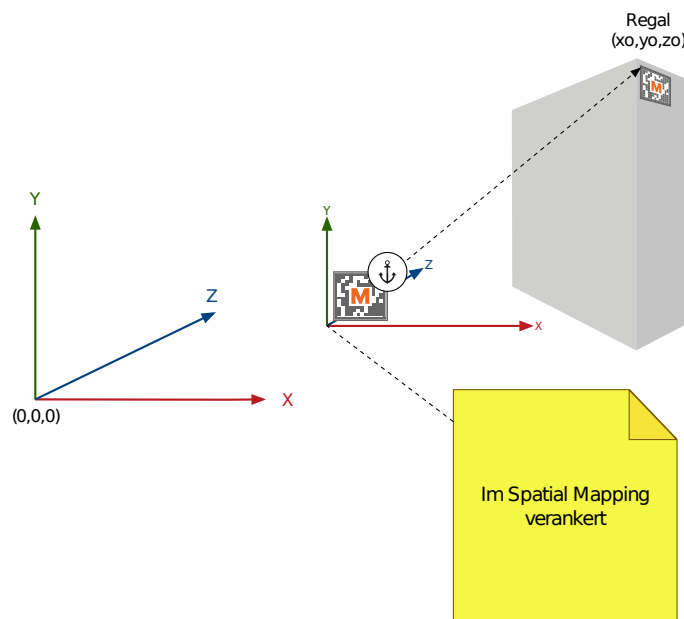


Abbildung 12: Der gemeinsame Ursprung wird in der echten Welt verankert

Beschreibung: Wie sich im letzten Versuch gezeigt hat, muss neben dem Unity Koordinatensystem auch das räumliche Verständnis von HoloLens miteinbezogen werden. Um sicherzustellen, dass Hologramme ihre Position und Rotation im Vergleich zur wirklichen Welt beibehalten, muss ihr Ursprung irgendwie verankert werden. Nur so bleiben relative Koordinaten zu diesem Ursprung über mehrere Starts hinweg gültig, selbst wenn sich der Benutzer dazwischen bewegt hat. Microsoft empfiehlt dafür die Verwendung von sogenannten *Spatial Anchors*. Ein Spatial Anchor definiert einen statischen Punkt im generierten Spatial Mapping

Mesh der HoloLens. Man übergibt diesen Punkt quasi an HoloLens mit der Aufgabe dafür zu sorgen, dass dieser Punkt in der echten Welt immer am selben Ort bleibt. Der Versuchsaufbau ist im Kern gleich wie im ersten Versuch. Es gibt einen Marker als Ursprung plus jeweils einen zusätzlichen Marker pro Regal. Jedoch wird in diesem Fall der Ursprung als Spatial Anchor definiert, also in der Realität verankert. Somit verändert er seine Position nur dann, wenn sich das räumliche Verständnis der HoloLens verändert (siehe Abbildung 12). Die Regale werden weiterhin relativ zum Ursprung ausgerichtet.

Auswertung: Leider hat auch dieser Ansatz nicht den gewünschten Erfolg gebracht. Grundsätzlich würde der Ansatz funktionieren, jedoch kamen beim Versuch starke Ungenauigkeiten zum Vorschein. Teilweise konnte eine Positionsabweichung von bis zu 1 Meter beobachtet werden. Dies kommt daher, dass HoloLens exakte Abmessungen und Distanzen der Umgebung nur in einem bestimmten Radius um einen Spatial Anchor herum wirklich garantieren kann. Je grösser die Distanz eines Hologramms zu seinem Spatial Anchor, desto ungenauer wird die Position im Vergleich zur Realität. So kann es z.B. sein, dass zwei Objekte in der realen Welt 10 Meter voneinander entfernt sind, im Modell von HoloLens die Distanz allerdings nur 9.8 Meter beträgt. Wenn nun Hologramme über solche Distanzen hinweg platziert werden, verfälscht sich die Position proportional zur Distanz (laut Microsoft wird dieser Fehler sichtbar ab ca. 3 Metern Abstand zum Spatial Anchor). Da Regale aber jeweils über eine ganze Filiale hinweg verteilt sind, ist ein einzelner Spatial Anchor als Ursprung somit keine akzeptable Lösung. Mehr zur Handhabung der verschiedenen Koordinatensysteme findet man in der Dokumentation von Microsoft. [[Mic17b](#)]

Versuch - Regale im Raum verankern

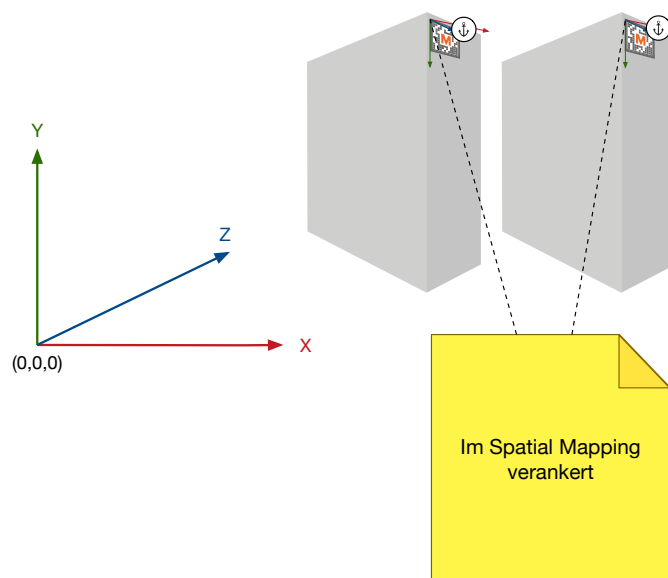


Abbildung 13: Jedes Regal wird direkt in der echten Welt verankert

Beschreibung: Basierend auf den Resultaten der vorhergehenden beiden Versuche wird nun pro Regal ein eigener Spatial Anchor definiert, der gemeinsame Ursprung fällt weg. Jedes Hologramm, welches später relativ zu einem Regal dargestellt werden soll, wird somit in dessen

Koordinatensystem platziert. Dies hat zur Folge, dass Position und Ausrichtung von Elementen, die sich in der Nähe des Regals befinden, vom Spatial Mapping Algorithmus der HoloLens permanent optimiert werden.

Auswertung: Dieser Versuch hat die besten Ergebnisse geliefert. In der Nähe der Regale waren nur kleine Verschiebungen von maximal 3 Zentimetern beobachtbar. Bei grösseren Entfernungen zum Regal entstanden Abweichungen von ca. 10 - 15 Zentimetern. Je besser die HoloLens den Raum vermessen hat, desto kleiner wurden die Abweichungen.

Bemerkung

Wenn man die HoloLens trägt, bewegt sich der Kopf relativ stark. Für den Benutzer macht es den Anschein, als wären die Hologramme instabil bzw. unruhig. Hier muss der Entwickler für eine optische Stabilisierung sorgen. Mithilfe der "Gaze Stabilizer" und "Stabilization Plane Modifier" Komponenten aus dem HoloToolkit lassen sich die Hologramme stabilisieren, was das Erlebnis für den Benutzer deutlich angenehmer macht.

Resultat

Da HoloLens das räumliche Abbild der Welt dynamisch anpasst, müssen für fixe Positionierungen im Raum zwingend Spatial Anchors verwendet werden. Die besten Ergebnisse erhält man, wenn pro Hologramm ein eigener Spatial Anchor definiert wird. Unter Umständen müssen im finalen Prototyp mehrere Verankerungen pro Regal definiert werden, wenn die relativ dazu positionierten Hologramme weiter als 3 Meter davon entfernt sind. Alternativ könnte man ein grosses Regal in Teilabschnitte unterteilen, die jeweils einzeln verankert werden. Die Genauigkeit ist aber für den aktuellen Anwendungsfall ausreichend, so dass vorerst keine weiteren Optimierungen vorgenommen werden müssen.

Bemerkung

Das HoloToolkit beinhaltet eine Reihe von Hilfsfunktionen um High-Level Mesh Analysen durchführen zu können. Damit ist es möglich Wände, Böden, Decken und Flächen mit vorgegebenen Dimensionen (z.B. Sitzflächen) im Spatial Mapping Mesh der HoloLens zu erkennen. Zurzeit können allerdings nur zweidimensionale Flächen erkannt werden [Mic17k]. Ein möglicher Anwendungsfall für diese Hilfsfunktionen wäre z.B. das Erkennen von Regalen innerhalb einer Filiale anhand ihrer Form, statt über spezielle Marker. Dies sprengt aber den Umfang dieser Studienarbeit.

6.4 Experiment 3: Benutzeroberfläche

Motivation

Benutzeroberflächen werden typischerweise für Bildschirme (2 Dimensionen) entworfen, dabei kann man sich an vielen Erfahrungsberichten und Studien zur Benutzerfreundlichkeit orientieren. Die Umsetzung von Benutzeroberflächen im Raum (3 Dimensionen) stellt Entwickler vor eine neue Herausforderung. Plötzlich hat man viel mehr Möglichkeiten der Darstellung und Interaktion. Hinzu kommt, dass die Wahrnehmung mit einer holografischen Brille sich anders verhält als mit einer “klassischen” Virtual Reality Brille. Insbesondere Platzierung von Benutzerelementen und Informationen sowie Farbwahl müssen genauer untersucht werden. Für HoloLens hat Microsoft bereits Richtlinien herausgegeben, welche ihre Erfahrungen bezüglich Benutzerfreundlichkeit und technischen Limitationen widerspiegeln. Ziel dieses Experimentes ist, mit dem Entwerfen einer Benutzeroberfläche im dreidimensionalen Raum (mit Unity) vertraut zu werden und Probleme, welche währenddessen auftreten könnten, frühzeitig aus der Welt zu schaffen.

Ansatz

Dieses Experiment wird in drei Versuche unterteilt. Die Versuche beschäftigen sich mit den drei verschiedenen Ansichten des finalen Prototyps (vgl. Beschreibung Use Case in Kapitel 5.3). Bisher gibt es relativ wenig Informationen zur optimalen Gestaltung von Benutzeroberflächen für Augmented Reality Applikationen. Aus diesem Grund stützt sich das Experiment auf die spezifischen Richtlinien von Microsoft für HoloLens.

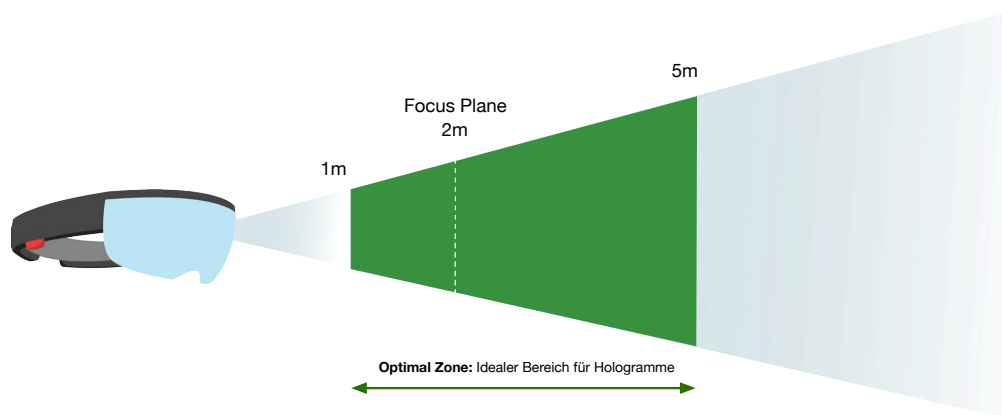


Abbildung 14: Distanzen für Hologramme (eigene Darstellung gemäss [Mic17i])

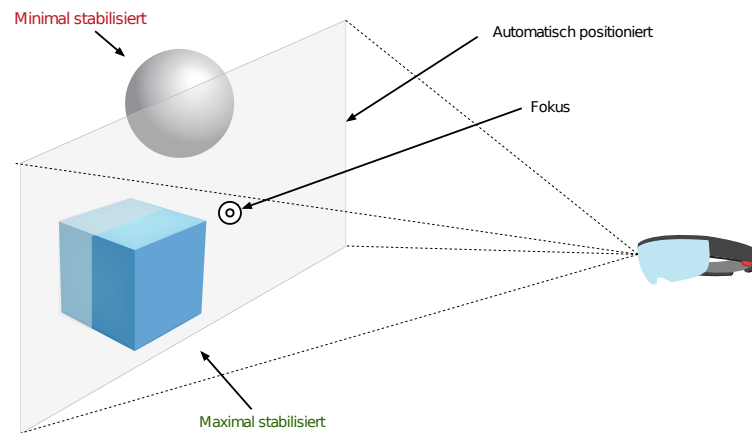


Abbildung 15: HoloLens Stabilization Plane für Hologramme

- **Abstände:** Microsoft spezifiziert einen optimalen Bereich zur Platzierung von Hologrammen (siehe Abbildung 14). Die sogenannte “*Optimal Zone*” ist für Hologramme, mit denen der Benutzer über verschiedene Wege interagieren kann. Ausserhalb davon können Hologramme zwar platziert werden, allerdings sollten sich Interaktionen damit auf ein Minimum beschränken. Wichtig ist, dass keine Hologramme näher als 1 Meter vor dem Benutzer platziert werden.
- **Einbezug der Umgebung:** Um die Möglichkeiten von HoloLens voll auszuschöpfen, soll die Umgebung bei der Platzierung der Hologramme berücksichtigt werden (z.B. Hologramm auf Tisch platzieren, statt es in der Luft schweben zu lassen).
- **Platzierung:** Da das sogenannte *Field of View* der HoloLens nicht das komplette Sichtfeld des Benutzers abdeckt, sollten neue Objekte immer im Zentrum und mit ausreichend Abstand platziert werden. Wichtig ist ausserdem, dass keine Hologramme im Sichtfeld des Benutzers fixiert werden, da sie es dadurch zusätzlich einschränken.
- **Farbenwahl:** Auf der HoloLens wird schwarz als transparent dargestellt und erlaubt somit den Blick auf die “reale Welt” durch die Brille. Im Gegensatz dazu wird Weiss als besonders hell wahrgenommen. Microsoft empfiehlt als maximale Werte für Weiss $RGB = (235, 235, 235)$. Zudem sollte bedacht werden, dass jedes Hologramm leicht transparent ist.
- **Frame Rate:** Sowohl Stabilität als auch Farbechtheit der Hologramme hängen stark von der Aktualisierungsrate des Bildes auf den verbauten Bildschirmen ab. Microsoft empfiehlt ein Minimum von 60 Frames pro Sekunde für Applikationen. Dadurch wirken auch Bewegungen der Hologramme flüssiger.
- **Stabilization Plane:** Die HoloLens versucht alle Hologramme zu stabilisieren. Aus Gründen der Performance können aber unter Umständen nicht alle Hologramme gleichzeitig durch den Algorithmus korrigiert werden. Aus diesem Grund versucht die HoloLens eine Ebene durch die wichtigsten (fokussierten) Hologramme zu legen. Hologramme, die von dieser Ebene geschnitten werden, sind hinsichtlich ihrer Stabilität (siehe Abbildung 15) optimiert.

Bemerkung

Entwickler können diesen Fokus über die API von HoloLens mit der Funktion “set-

FocusPoint” manuell steuern.

- **Color Separation:** Aufgrund des additiven Verhaltens bei der Projektion der Hologramme (Addition von Farbkanälen), können sich einzelne Farbkanäle vom Hologramm lösen. Aus Sicht des Benutzers entsteht dadurch ein Regenbogeneffekt. Dies geschieht vor allem bei schnellen Kopfbewegungen respektive bei sich schnell bewegendem Hologrammen, die sich nicht im Bereich der Stabilization Plane befinden. Microsoft empfiehlt deshalb, den Fokus entsprechend zu setzen, Kamerabewegungen zu stabilisieren und Farben zu verwenden, die hauptsächlich Rot, Grün oder Blau enthalten (Weiss ist ungeeignet, weil es hohe Anteile von allen drei Farben aufweist).

Mehr Details zu diesen Themen findet man in der Microsoft Dokumentation. [[Mic17c](#)] [[Mic17d](#)] [[Mic17a](#)]

Versuch - Regal Overlay

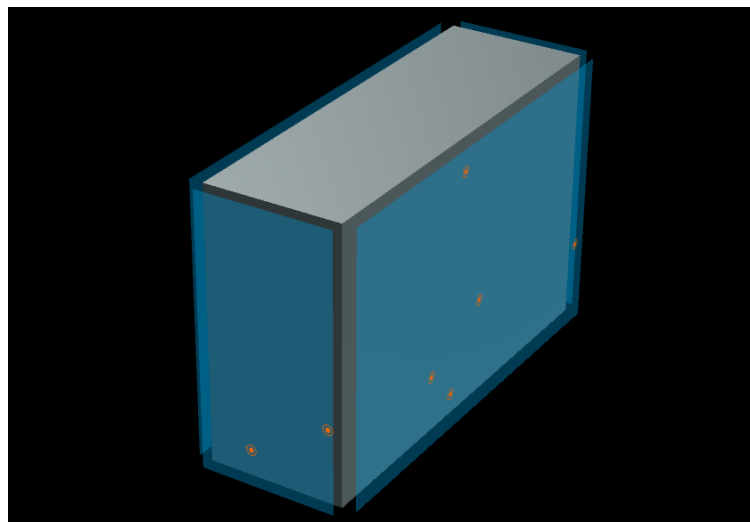


Abbildung 16: Auf allen Seiten des Regals können Markierungen dargestellt werden

Beschreibung: Der Versuch setzt sich mit der Übersicht für ein Regal auseinander. Die Aufmerksamkeit des Benutzers soll durch Markierungen auf dem Regal auf bestimmte Produkte bzw. Produktgruppen gelenkt werden. Hier gibt es ein grundsätzliches Problem, denn die HoloLens hat keinerlei Informationen darüber, wie Produkte innerhalb eines Regals einsortiert sind. Laut Migros gibt es diesbezüglich zwar grobe Vorgaben, allerdings kann es in den Filialen durchaus zu Abweichungen kommen.

Bemerkung

In Experiment 1 (vgl. Kapitel 6.2) wurden bezüglich der Erkennung von Produkten durch HoloLens bereits gute Resultate erzielt. Somit müsste es theoretisch auch möglich sein, erkannte Produkte einem Regal zuzuordnen, wenn die Positionen der Regale einmal erfasst worden sind. Die Zugehörigkeit zu einem bestimmten Regal kann über den Abstand eines erkannten Produkts geprüft werden. Je länger man durch eine Filiale geht und sich

Produkte anschaut, desto besser weiss HoloLens Bescheid, wo ein Produkt in einer Filiale zu finden ist.

Unabhängig davon benötigt der Prototyp eine Datenstruktur, mit der die Anordnung von Produkten in einem Regal beschrieben werden kann (siehe Abbildung 17). Zur Demonstration des Prototyps werden diese Informationen statisch im Voraus definiert.

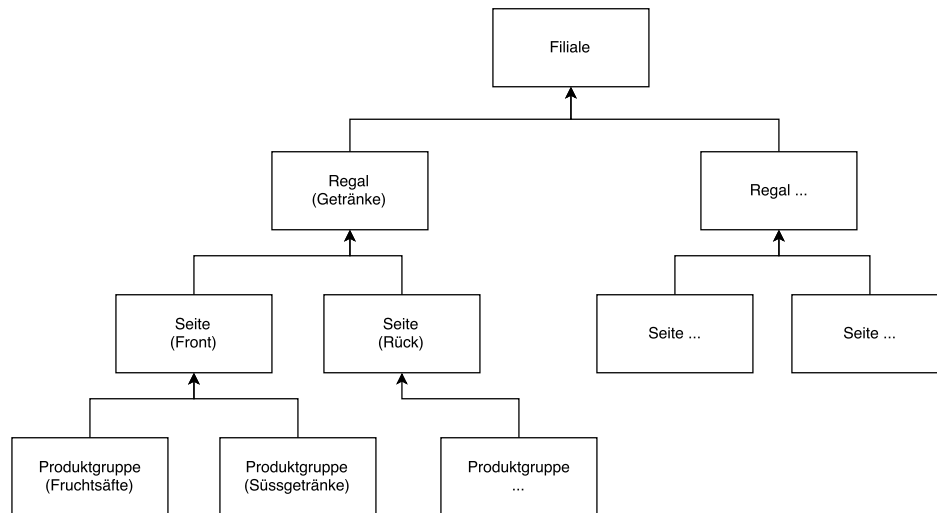


Abbildung 17: Baumartige Struktur der Daten für die Regalübersicht

Ein Regal hat verschiedene Seiten, jede davon definiert durch ihre Ausrichtung (z.B. Vorderseite oder Rückseite) und ihre Grösse. Stellt man sich eine Seite nun als einfache Fläche vor (siehe dazu Abbildung 18), dann kann die Position von jedem Produkt im Regal durch zweidimensionale Koordinaten auf dieser Fläche angegeben werden. Der Ursprung liegt in der Mitte der Fläche und hat die Koordinaten $X = 0$ und $Y = 0$.

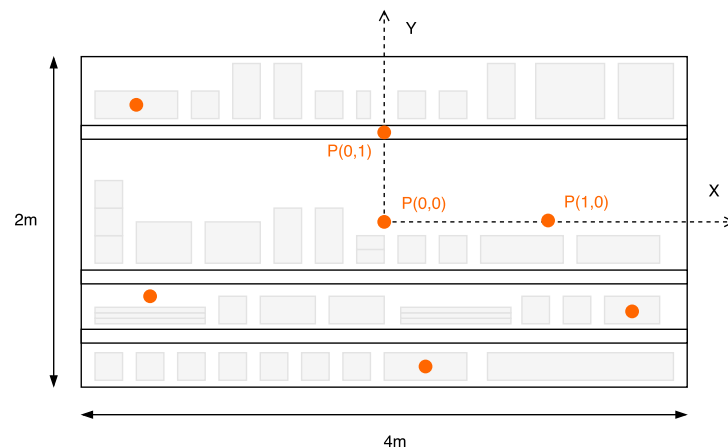


Abbildung 18: Seite des Regals als Koordinatensystem

Auswertung: Basierend auf einer Kombination der beschriebenen Datenstruktur und der in Experiment 2 (vgl. Kapitel 6.3) gezeigten Positionserfassung für Regale kann der Prototyp nun

eine Art virtuelles Netz über ein Regal in der Nähe des Benutzers spannen, um visuelle Hinweise zu einzelnen Produkten oder ganzen Produktgruppen auf den Seiten des Regals darzustellen. So lässt sich beispielsweise durch farbige Symbole aufzeigen, welche Produkte in näherer Zeit nachbestellt werden müssen. Die vorgestellte Datenstruktur erlaubt viel Flexibilität, womit sich die meisten in der Realität existierenden Regale abbilden lassen sollten. Natürlich ist die Zusammenstellung aller benötigten Informationen mit einem gewissen Aufwand verbunden. Für die Praxis müsste man die Erfassung also in jedem Fall vereinfachen. Es wäre durchaus denkbar, dass man z.B. ein grafisches Programm zur Verfügung stellt, in dem ein Regal Schritt für Schritt für die Verwendung mit HoloLens erfasst werden kann.

Bemerkung

Theoretisch wäre auch eine dynamische Erfassung (bzw. Vermessung) mittels der in der HoloLens verbauten Kameras und Sensoren denkbar. Sehr wahrscheinlich müsste dafür allerdings der Algorithmus für das Spatial Mapping optimiert werden, die Auflösung des resultierenden Modells ist in der Standardimplementation von Microsoft doch relativ grob. Aufgrund des Zeitaufwands konnten dazu keine weiteren Versuche durchgeführt werden.

Versuch - Regal Marker

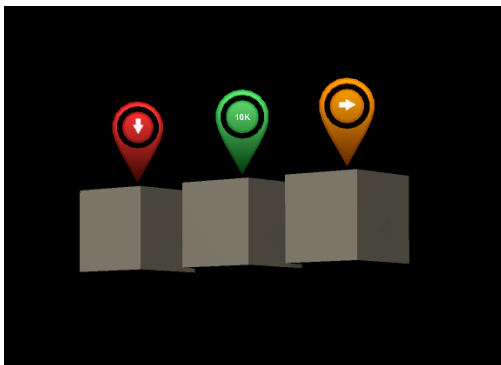


Abbildung 19: Marker zur Ortung interessanter Regale innerhalb einer Filiale

Beschreibung: Dieser Versuch befasst sich mit der Darstellung eines Markers, welcher beim Blick in eine Filiale über jedem Regal schweben soll. Wie bei einem Wegweiser soll der Benutzer darauf aufmerksam gemacht werden, welches Regal er sich noch genauer anschauen muss. Der während des Experiments erstellte Marker besitzt zwei visuelle Zustände. Entweder wird der Umsatz darauf als gerundete Zahl dargestellt oder die Umsatzentwicklung wird durch ein Symbol repräsentiert. Die Farbe des Markers dient als zusätzlicher Indikator, und ändert bei Betrachtung der Umsatzentwicklung jeweils für: “Steigender Umsatz”, “Stagnierender Umsatz” und “Sinkender Umsatz” (siehe Beispiel in Abbildung 19). Zudem richtet sich der Marker immer in Richtung des Benutzers aus, damit die darauf dargestellten Informationen aus jeder Perspektive erkennbar bleiben. Falls die Ansicht in Zukunft über mehrere Brillen verteilt werden sollte, muss dieses Verhalten natürlich überdacht werden. Als Vorlage dient ein Asset aus dem Unity Asset Store. [[For17](#)]

Auswertung: Die Marker sehen auch auf der HoloLens gut aus. Die Farben sind gut erkennbar und es tritt in den meisten Fällen keine Color Separation auf. Für den Versuch wurden die Komponenten zur Stabilisierung (“Gaze Stabilizer” und “Stabilization Plane Modifier”) aus dem HoloToolkit verwendet. Diese sorgen für ein relativ stabiles Verhalten und unterbinden das sonst beobachtete, ruckartige Herumspringen der Hologramme.

Versuch - Bedienelemente

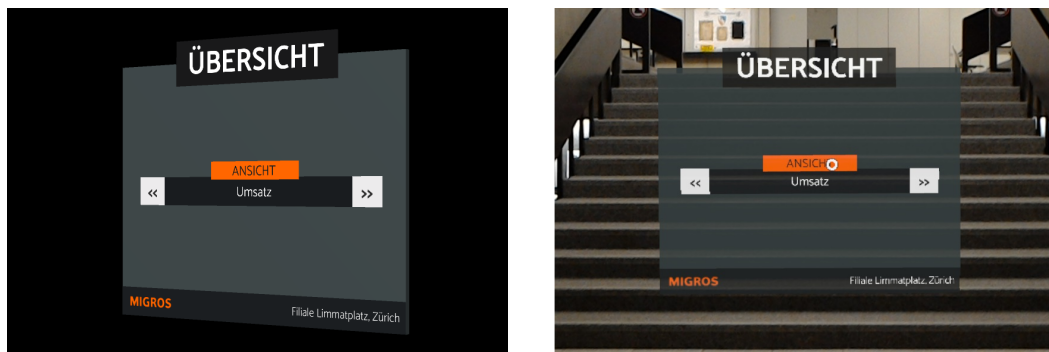


Abbildung 20: Die Bedienung erfolgt über ein fensterähnliches Hologramm

Beschreibung: Dieser Versuch beschäftigt sich mit der Frage, wie auf der HoloLens Bedienelemente für den Benutzer dargestellt werden können. Dies beinhaltet insbesondere die Darstellung von Text im Zusammenspiel mit Unity. Wie gross muss der Text dargestellt werden, damit er noch gut lesbar ist? Wie wird dem Benutzer eine möglichst einfache Bedienoberfläche präsentiert, die er über die HoloLens problemlos bedienen kann? Für das Experiment wurden eine Kombination aus Unity Standardkomponenten und im Unity Asset Store erworbenen Bedienelementen verwendet. [Stu17]

Auswertung: Elemente lassen sich auf der HoloLens gut bedienen, solange sie eine gewisse Mindestgrösse nicht unterschreiten. Diese lässt sich am einfachsten durch Ausprobieren herausfinden. Dank vorgefertigter Komponenten aus dem HoloToolkit klappt das Zusammenspiel zwischen Bedienelementen und HoloLens Gesten problemlos. Zu beachten ist allerdings, dass die standardmässig zur Verfügung gestellten Komponenten für Benutzeroberflächen in Unity eher für einen normalen Bildschirm ausgelegt sind. Hier müssen diverse Einstellungen angepasst werden, um eine brauchbare Darstellung mit der HoloLens zu erreichen. Dies betrifft in erster Linie die Skalierung, weshalb die Einstellungen “Scaling” und “Dynamic Pixels Per Unit” von Canvas Objekten zwingend angepasst werden müssen. Erst dann haben darauf platzierte Bedienelemente in der realen Welt die erwartete Grösse. Während des Experiments hat sich eine Skalierung von $\frac{1}{256}$ bewährt.

Resultat

Grundsätzlich empfiehlt sich, die Richtlinien von Microsoft gleich am Anfang durchzulesen, um bei der Gestaltung der Benutzeroberfläche weniger böse Überraschungen zu erleben. Da-

durch lernt man gleichzeitig die Funktionsweise der Brille besser kennen und kann sich darauf einstellen.

Das Experiment war zudem eine gute Gelegenheit, um sich weiter in Unity einzuarbeiten. Dank der standardmässig zur Verfügung gestellten Komponenten für Benutzeroberflächen musste nicht alles von Grund auf neu erfunden werden. Die erarbeiteten technischen Konzepte können praktisch unverändert in den finalen Prototyp übernommen werden. Insofern wurde hiermit also bereits Vorarbeit für später geleistet.

7 Prototyp

7.1 Konzept

Basierend auf den Erkenntnissen der durchgeführten Experimente (siehe Kapitel 6) wurde für den Prototyp ein Gesamtkonzept ausgearbeitet. Dieses umfasst sowohl grafische Elemente, als auch Ideen zur Benutzerführung. Während der Ausarbeitung des umzusetzenden Anwendungsfalls wurden bereits drei gewünschte Ansichten identifiziert (siehe Kapitel 5.3). Einem Benutzer soll es möglich sein, produktbezogene Informationen auf verschiedenen Ebenen betrachten zu können.

Mögliche Ansätze zur Umsetzung dieser Ebenen wurden gleich zu Beginn der Implementierungsphase des Prototyps mittels Handskizzen festgehalten. Ideen lassen sich damit relativ schnell und unkompliziert mit anderen Personen besprechen und verfeinern. Die nachfolgenden Abbildungen zeigen links jeweils die ursprüngliche Skizze und rechts die finale Umsetzung auf der HoloLens.

Bemerkung

Die gezeigten Abbildungen des Prototyps weisen eine deutlich schlechtere Qualität auf, als es bei direkter Betrachtung mit der HoloLens der Fall ist. Dies gilt insbesondere für Texte und Symbole, welche auf der Brille über kürzere Distanzen problemlos erkennbar sind. Ausserdem sind schwarze Objekte auf den Bildern sichtbar, was in der Realität nicht der Fall ist.

Für ein besseres Verständnis versetzt man sich am besten in die Rolle eines Filialleiters. Bisher kommen für die täglichen Verwaltungsaufgaben in einer Filiale typischerweise kleine Handgeräte mit Barcode-Scanner oder Ausdrücke auf Papier zum Einsatz. Dank dieses Prototyps kann er diese Aufgaben nun auch mit HoloLens bewerkstelligen. Er setzt sich die Brille auf und sieht als erstes ein im Raum schwebendes Bedienelement, welches ihn begrüsst und dazu auffordert, die Filiale zu identifizieren. Abbildung 21 zeigt Konzept und konkrete Umsetzung dieses Bedienelements.

Die Identifikation ist nötig, damit HoloLens weiss, zu welcher Filiale Informationen angezeigt werden sollen. Zudem erinnert sich die Brille an Filialen, und wo die Regale darin platziert sind. Bereits bekannte Filialen können direkt ausgewählt werden, neue Filialen müssen mittels VuMark zuerst erfasst werden (siehe Experiment 6.2 für Details).

Nach erfolgreicher Identifikation über den VuMark der Filiale bewegt sich das Bedienelement auf die linke Seite des Benutzers. Dies geschieht in der Annahme, dass die Marker typischerweise an einer Wand befestigt werden. Damit soll verhindert werden, dass das Element plötzlich in bzw. hinter einer Wand schwebt und damit schwer zugänglich wird. Ein Signalton auf dem linken Ohr des Benutzers soll ihn dazu bringen, seinen Blick nach links zu richten, wo er das Bedienelement wiederfindet.



Abbildung 21: Ein Bedienelement dient zur Steuerung und für Anweisungen

Bemerkung

Die optimale Platzierung von Benutzerelementen im Raum ist nicht so einfach. Sie müssen einerseits leicht auffindbar sein, dürfen aber gleichzeitig nicht die Sicht auf andere Hologramme versperren. Zudem bricht eine ungünstige Platzierung in Wänden oder Objekten das Gefühl der Immersion und schmälert damit das Erlebnis. Der Prototyp löst dieses Problem, indem er dem Benutzer erlaubt, das Bedienelement jederzeit mittels Sprachbefehl zurück in sein Sichtfeld zu holen.

Ab diesem Zeitpunkt dient das Bedienelement als zentrale Anlaufstelle, welche dem Filialleiter die Steuerung des Prototyps ermöglicht. So wählt er darüber beispielsweise aus, welche Informationen er zum Sortiment der Filiale sehen will. Möchte er den Lagerbestand prüfen, dann wechselt er darüber in die entsprechende Ansicht (siehe Abbildung 21). Der Prototyp passt nun alle Hologramme auf sowie oberhalb der Regale dieser Auswahl an. Die Daten werden je nach Ansicht durch Symbole, Farben und Text repräsentiert. Ein Beispiel dafür zeigt Abbildung 22.

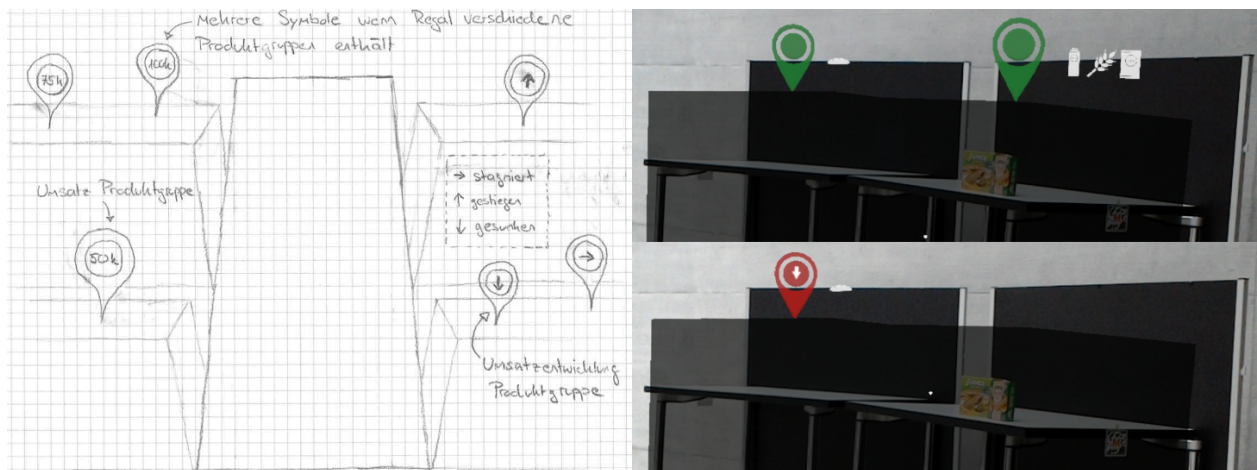


Abbildung 22: Marker über den Regalen geben einen guten Überblick der Filiale

Die grossen Marker über den Regalen haben zwei Funktionen. Einerseits zeigen sie dem Filialleiter schon von weitem, zu welchen Regalen er sich begeben soll, um genaueres herauszufinden. Gleichzeitig fassen sie die zur Ansicht ausgewählten Daten der Produkte in einem Regal zusammen. Im Gegensatz zur Skizze in Abbildung 22 gibt es im Prototyp nur noch genau einen Marker oberhalb eines Regals.

Beispiel: Beim Lagerbestand wird der Marker über dem Regal rot und zeigt einen Pfeil nach unten, wenn mindestens eines der darin einsortierten Produkte nur noch in niedriger Stückzahl vorhanden ist. Kleine Symbole rechts vom Marker verweisen auf die betroffenen Produktgruppen. Sind im Gegensatz dazu alle Produkte eines Regals in genügend hoher Stückzahl an Lager, dann wird über diesem Regal kein Marker dargestellt.

Bemerkung

Der Prototyp ist so flexibel aufgebaut, dass verfügbare Daten zu einer Filiale praktisch beliebig durch Hologramme über einem Regal und auf seinen Seiten visualisiert werden können. Die Kombination aus Daten und Visualisierungen entspricht einer Ansicht. Jede Ansicht definiert für sich, mit welchen Farben/Symbolen/Text die dazugehörige Statistik dargestellt wird. Kundenspezifische Wünsche lassen sich somit leicht umsetzen, sofern die zu visualisierenden Daten beim Detailhändler vorhanden sind. Weitere Ansichten können im Code des Prototyps via Controller (siehe Kapitel 7.2) implementiert werden.

Begibt sich der Filialleiter in die Nähe eines Regals, dann erscheinen auf den Seiten abhängig von der gewählten Ansicht kleine Symbole. Diese Symbole repräsentieren Produktgruppen, und sind ungefähr dort platziert, wo sich die Produkte dieser Gruppe im Regal befinden. Ein Symbol ist typischerweise nur dann sichtbar, wenn bei Produkten dieser Gruppe Handlungsbedarf besteht. Ein Beispiel dafür zeigt Abbildung 23.

Beispiel: Beim Lagerbestand werden die Symbole der betroffenen Produktgruppen in roter Farbe auf den Seiten der Regale dargestellt. Eine Gruppe gilt als betroffen, wenn mindestens ein Produkt davon seinen minimalen Lagerbestand unterschreitet. Der Filialleiter sieht dank der Symbole mit einem Blick, welche Produkte im Regal eine genauere Betrachtung erfordern.

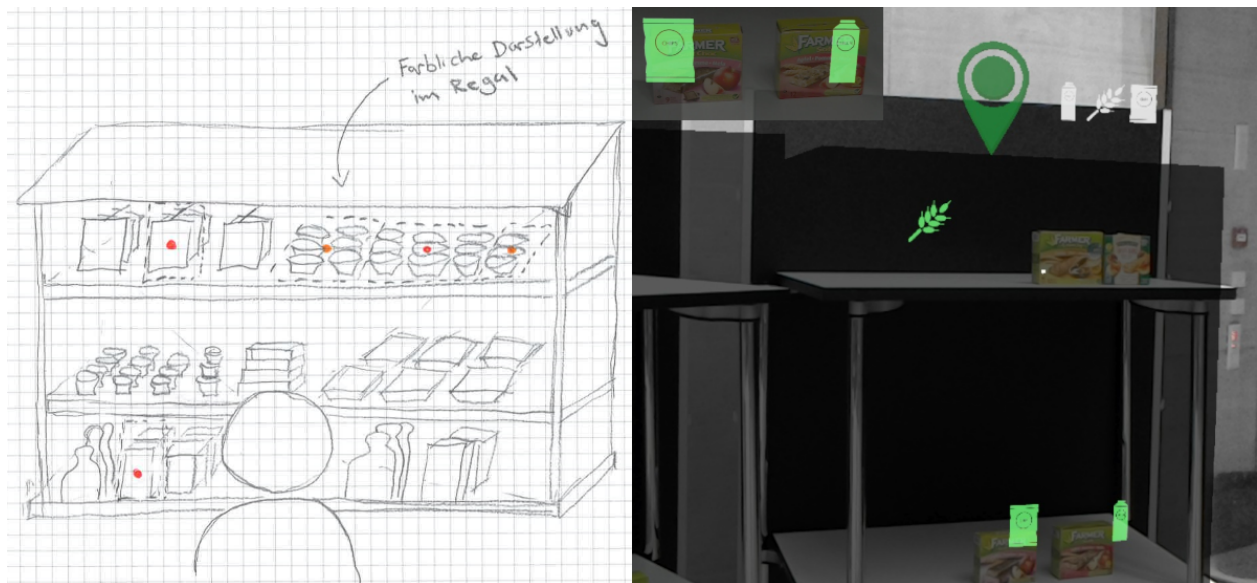


Abbildung 23: Interessante Stellen im Regal werden durch Symbole hervorgehoben

Früher hätte der Filialleiter die Produkte einzeln aus dem Regal genommen, und über ihren Barcode Informationen zum Produkt auf dem Handgerät abgefragt. Der Prototyp allerdings erkennt Produkte anhand der Vorderseite ihrer Verpackung, ohne dass man sie dafür in die Hand nehmen muss. Diese Funktionalität ist allerdings aufgrund technischer Limitationen nicht permanent aktiv, sondern muss mittels Sprachbefehl ein- und ausgeschaltet werden.

Sprachbefehl	Funktion
“Select”	Alternative zum Klick mittels Handgeste
“Show controls”	Bedienelement zurück ins eigene Sichtfeld holen (Zuruf)
“Show me products”	Aktiviert die Erkennung von Produkten
“Hide products”	Deaktiviert die Erkennung von Produkten

Tabelle 2: Auflistung der vom Prototyp unterstützten Sprachbefehle

Bemerkung

Für die Erkennung muss der Prototyp auf die Frontkamera der HoloLens zugreifen. Das Bildsignal wird an Vuforia zur Erkennung von Produkten und VuMarks weitergeleitet. Der erhöhte Bedarf an Rechenleistung führt dazu, dass weniger Leistung zur Darstellung der Hologramme bleibt. Deshalb wird diese Funktionalität bewusst nur sehr selektiv im Prototyp eingesetzt.



Abbildung 24: Produkte werden beim Anblick automatisch erkannt

Ist die Erkennung einmal aktiv, reicht ein Blick ins Regal. Vom Prototyp erkannte Produkte werden mit einem Rahmen versehen. Richtet sich der Blick auf ein spezifisches Produkt, dann erscheint zusätzlich ein kleines Symbol in der Mitte, wie es in Abbildung 24 rechts zu sehen ist. Klickt der Filialleiter auf das anvisierte Produkt, dann werden die dazu bekannten Daten als Hologramme rund herum angeordnet.

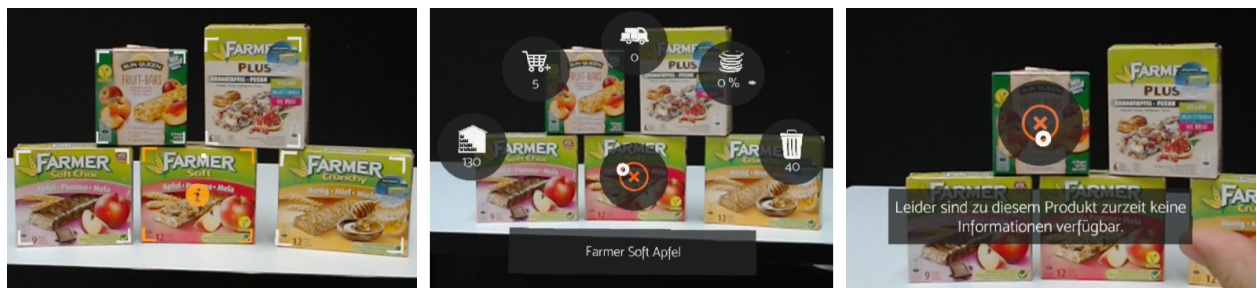


Abbildung 25: Zu Produkten kann der Benutzer genauere Informationen abrufen

	Lagerbestand (Anzahl)
	Bestellvorschlag (Anzahl)
	Offene Bestellungen (Anzahl)
	Umsatzentwicklung (Prozent)
	Verderb (Anzahl)

Tabelle 3: Beschreibung der Symbole zur Darstellung von Produktinformationen

Eine weitere Funktion des Prototyps ist die Erfassung von Regalen in einer Filiale. Dieser Vorgang findet im Normalfall nur dann statt, wenn die HoloLens zum ersten Mal in einer neuen Filiale zum Einsatz kommt, oder wenn innerhalb der Filiale eine Umstrukturierung vorgenommen wurde. Zuvor muss der geschulte Filialleiter oder ein technischer Mitarbeiter an jedem Regal ein VuMark befestigen. Ein VuMark muss seinem Regal innerhalb der Filiale eindeutig zugeordnet werden können. Die Erfassung kann ebenfalls über das Bedienelement aus Abbildung 21 gestartet werden. Das Prinzip ist analog zur Identifizierung einer neuen Filiale. Die VuMarks müssen vom Benutzer lediglich ein paar Sekunden länger betrachtet werden. Sobald der Prototyp ein Regal anhand seines VuMarks erkannt hat, werden Position und Orientierung bestimmt und dauerhaft auf der HoloLens gespeichert. Die VuMarks können von den Regalen entfernt werden, sobald alle Regale erfasst worden sind.

Bemerkung

Der Einsatz von VuMarks lässt sich damit rechtfertigen, dass eine rein optische Erkennung einer Filiale inklusive der darin angeordneten Regale zumindest im Zeitrahmen dieser Studienarbeit nicht realisierbar waren. Über VuMarks erlangt der Prototyp trotzdem die Fähigkeit, Dinge der echten Welt einem zurzeit noch statisch hinterlegten Datensatz zuzuordnen.

Kurz zusammengefasst: Der Filialleiter hat mit dem Prototyp eine praktische Arbeitshilfe zur Hand. Dank positionsgetreuer Hologramme wird eine Form von Navigation innerhalb einer Filiale möglich. Mit einem Blick lassen sich relevante Stellen in einer Filiale identifizieren, um dann entsprechend reagieren zu können. Lässt man die zurzeit noch sehr aufwändige Aufbereitung der benötigten Daten einmal aussen vor, dann können solche Applikationen durchaus die Effizienz der Angestellten einer Filiale steigern. Zudem werden Routineaufgaben dadurch möglicherweise etwas spannender.

Trotz vorangegangenen Experimenten sind gewisse Probleme, seien sie nun konzeptioniellen oder auch technischen Ursprungs, erst während der Entwicklung des Prototyps aufgetaucht. Kapitel 7.3 geht darauf etwas genauer ein und zeigt Lösungsansätze dafür auf.

7.2 Architektur

Da das Projekt mit der 3D Engine von Unity umgesetzt wird, orientiert sich auch die Architektur des Prototyps stark an dieser Umgebung. Abbildung 26 zeigt eine vereinfachte Darstellung der internen Datenstruktur von Unity. Ein Grundverständnis dafür erleichtert den Einstieg in den Aufbau des Prototyps, weshalb an dieser Stelle kurz darauf eingegangen wird.

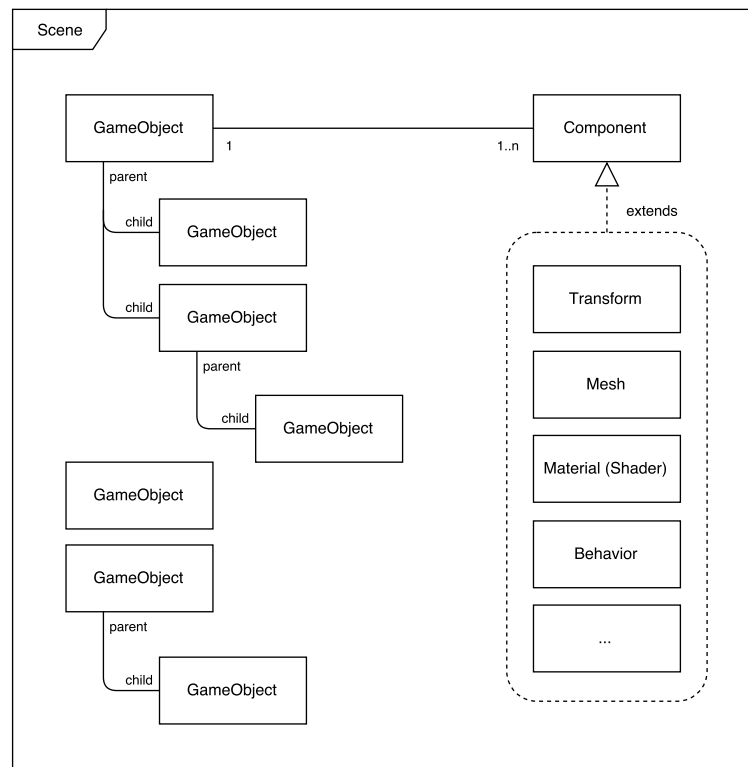


Abbildung 26: Aufbau einer Szene in Unity

Jedes darzustellende Objekt wird durch ein *GameObject* repräsentiert und ist gleichzeitig Teil einer übergeordneten Szene. In einem Spiel wäre die Analogie zu einer Szene beispielsweise ein einzelnes Level. Objekte können innerhalb einer Szene beliebig verschachtelt werden. Diese Hierarchie nutzt Unity für die Positionierung der Objekte im 3D Raum. Ein in der Hierarchie untergeordnetes Objekt (*child*) wird immer relativ zum übergeordneten Objekt (*parent*) positioniert, rotiert und skaliert. Somit besitzt jedes Objekt ein eigenes, lokales Koordinatensystem, in dem weitere Objekte platziert werden können. Für Objekte der obersten Stufe in der Hierarchie gilt das globale Koordinatensystem (auch *World* genannt) der Szene, welcher sie zugeordnet sind.

Bemerkung

Die *Transform* Komponente eines *GameObject* bietet eine Vielzahl an nützlichen Funktionen zur Transformation von Objekten innerhalb der verschiedenen Koordinatensysteme. Manuelle Berechnungen mit Vektoren und Matrizen fallen dadurch in vielen Situationen weg. [Tec17d]

Objekte sind wie ein Behältnis für eine oder mehrere Komponenten (*Component*). Komponenten definieren Eigenschaften und Verhalten eines Objekts. Dazu gehört zwingend die Position, Rotation und Skalierung im 3D Raum (*Transform*). Weitere Komponenten wie zum Beispiel Form (*Mesh*) oder Material (*Material*) kommen je nach Art des Objekts dazu.

Bemerkung

Genau genommen sind nicht die Objekte selbst, sondern ihre *Transform* Komponenten hierarchisch organisiert. Im Unity Editor² entspricht die Hierarchie allerdings der in Abbildung 26 gezeigten Struktur. Diese Diskrepanz macht sich dann bemerkbar, wenn man beispielsweise mittels *Behavior* die Position eines Objekts im Code verändern möchte. Allerdings erlaubt die Schnittstelle eine nahtlose Navigation zwischen *GameObject* und dem dazugehörigen *Transform*.

Mittels C# Skripts (*Behavior*) kann die Verhaltensweise eines Objekts definiert werden. Mit einem *Behavior* lässt sich beispielsweise umsetzen, dass sich ein Objekt rot einfärbt, sobald es vom Benutzer in der virtuellen Welt angeschaut wird. Der Prototyp nutzt einen Hybrid aus *Behavior* und eigenständigen C# Klassen zur Umsetzung der Benutzerinteraktion und zur Darstellung der Hologramme. Der Aufbau lässt sich grundsätzlich in drei Hauptbestandteile aufbrechen. Es ist der Versuch, trotz starker Integration in Unity eine gewisse Entkopplung zwischen 3D Visualisierung und Logik zu erreichen.

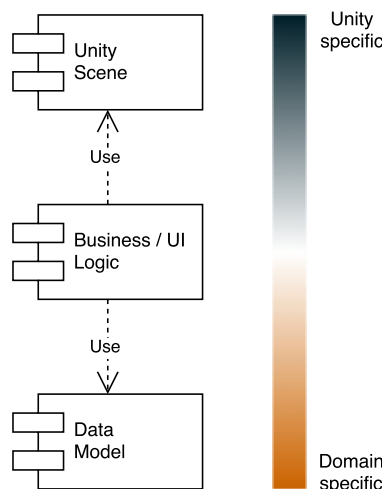


Abbildung 27: Hauptbestandteile, aus denen sich der Prototyp zusammensetzt

Das Prinzip folgt der in der Anwendungsentwicklung typischerweise angewandten Trennung zwischen Benutzeroberfläche, Logik und Daten. Bei einer sauberen Trennung lässt sich der Code für die Logik völlig unabhängig von HoloLens, Unity oder Vuforia automatisch testen, was den Entwicklungsprozess beschleunigt und Fehler schneller zu Tage fördert.

Bemerkung

Während der Entwicklung hat sich gezeigt, dass manuelle Tests mit HoloLens sehr zeit-

²Grafische Anwendung zur Entwicklung von 3D Anwendungen mit Unity

raubend sein können. Es fallen immer wieder ein Reihe von teilweise lang andauernden Schritten an, bis die Applikation auf der Brille ausgeführt und getestet werden kann. Mit dem Umfang steigt auch die Anzahl der Variationen, die jedes Mal von Hand ausprobiert werden müssen. Klassische Logikfehler oder Seiteneffekte liessen sich viel einfacher durch automatische Tests finden und beheben. Deshalb sollte der Kern der Applikation mehr oder weniger entkoppelt von externen Komponenten bleiben.

Szene

Sowohl die Szene als auch der Aufbau der sich darin befindenden Objekte ist natürlich sehr Unity-spezifisch. Deshalb übernimmt dieser Teil des Prototyps eher allgemeine Aufgaben, die relativ unabhängig vom konkreten Anwendungsfall für eine HoloLens-Applikation mit Unity nötig sind.

- **Anbindung HoloLens:** Unity abstrahiert im Prinzip schon das von Microsoft zur Verfügung gestellte SDK für HoloLens. Die minimal nötigen Unity Einstellungen für eine HoloLens-Anwendung beschreibt Microsoft in der Dokumentation zu HoloLens [Mic17e]. Im Zusammenspiel mit dem frei erhältlichen HoloToolkit lassen sich in wenigen Schritten spezifische Fähigkeiten von HoloLens in Unity einbinden. Ein Beispiel dafür ist die Gestenerkennung, welche über vorgefertigte Komponenten des Toolkits in Eingabe-Ereignisse umgewandelt werden. Macht der Benutzer mit seiner Hand eine Klick-Geste, führt das zu einem Klick auf ein Objekt in der Unity Szene. Spracherkennung, Spatial Mapping und Spatial Sound von HoloLens lassen sich ähnlich einfach einbinden.
- **Anbindung Vuforia:** Vuforia stellt ein fertiges Packet für die Integration in Unity zur Verfügung [Vuf17c]. Die zu erkennenden Targets müssen statisch in der Szene hinterlegt und konfiguriert werden. Deshalb befinden sich in der Szene des Prototyps ein VuMark zur Identifizierung von Filialen und Regalen sowie Image Targets zur Erkennung von Produkten.
- **3D Objekte:** Alle im Prototyp dargestellten Objekte wurden mit dem Unity-Editor in der Szene entworfen und anschliessend als Vorlage (auch als *Prefab* bezeichnet) hinterlegt. Zur Laufzeit greift der Prototyp situationsbedingt auf diese Vorlagen zurück, um konkrete Instanzen daraus der Szene hinzuzufügen. Ein Grossteil der verwendeten Objekte existiert also noch nicht, wenn die Szene im Editor betrachtet wird. Sie werden erst dann erstellt, wenn sie für den Benutzer sichtbar werden sollen.
- **Verbindung zwischen 3D-Objekten und Code:** Prefabs und statische Objekte in der Szene müssen irgendwie mit dem Code verbunden werden. Das geschieht im Normalfall über *Behaviors*, die man *GameObjects* als *Component* zuweist. Diese Verknüpfung kann man über die "Inspector"-Ansicht im Unity Editor nachvollziehen. Die Szene fungiert hier wie eine Art Verbindung und fügt die einzelnen Teile zu einem grossen Ganzen zusammen.

Der Prototyp besteht nur aus einer einzelnen Szene, deren Objekte je nach Zustand vom C# Code versteckt oder dargestellt werden. Die Szene ist ein guter Einstiegspunkt, wenn man sich den Prototyp etwas genauer ansehen möchte. Das dazugehörige Unity Projekt lässt sich mit dem Unity Editor öffnen und ist Teil der Abgabe dieser Studienarbeit (siehe Anhang A).

Logik

Programmcod verleiht einer Szene erst die Fähigkeit, auf Ereignisse des Benutzers zu reagieren. Im Falle der HoloLens werden typischerweise Geräusche abgespielt oder Hologramme ein- und ausgeblendet. Für die Darstellung der Hologramme werden Daten benötigt, die geladen, interpretiert und in die Hologramme eingebettet werden. Der C#-Code des Prototyps realisiert ausserdem Abläufe und Zustände der in Kapitel 5.3 beschriebenen Geschäftslogik.

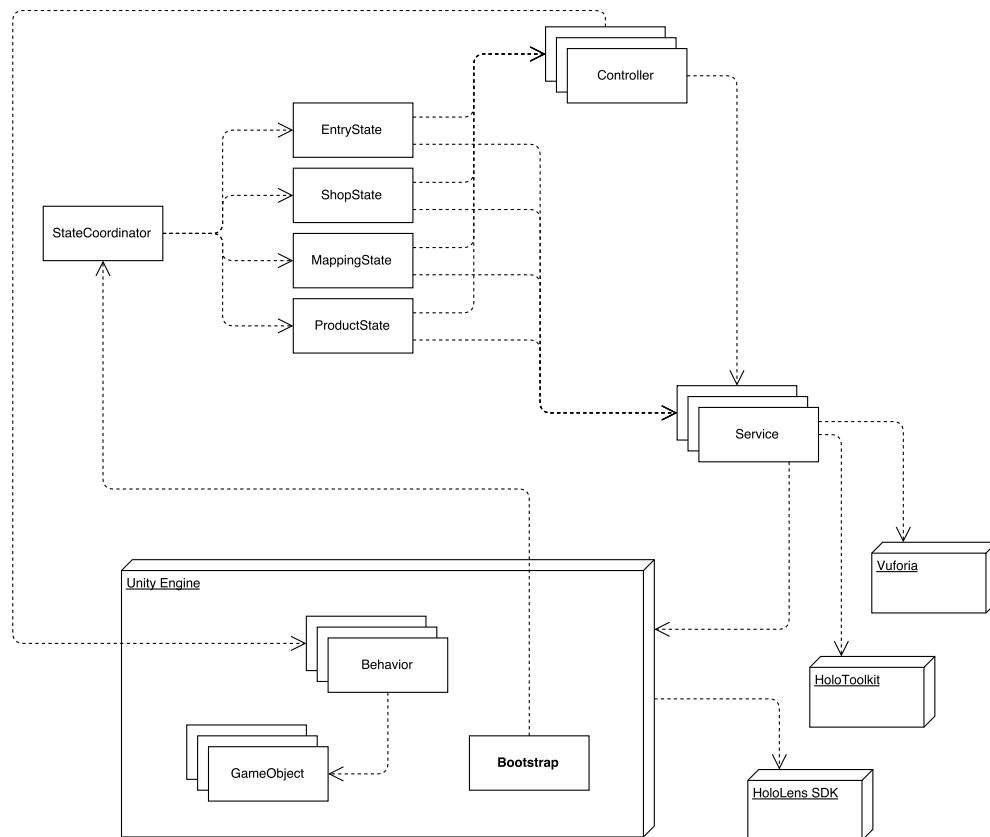


Abbildung 28: Übersicht der Code Architektur

Nachfolgend soll kurz auf die wichtigsten Aspekte eingegangen werden:

- **Entkopplung:** Der Code besteht mehrheitlich aus eigenständigen C# Klassen, also keine direkt von Unity abhängigen Skripts. Dadurch liesse sich theoretisch ein Grossteil der Logik automatisch testen. HoloLens- bzw. Vuforia-Ereignisse können simuliert werden, um die Reaktion darauf entsprechend zu prüfen. Aus Zeitgründen konnten allerdings keine automatischen Tests eingerichtet werden. Die Grundlage dafür wäre allerdings geschaffen.
- **Start:** Die Ausführung der Applikation beginnt mit einem speziellen *Behavior*, welches den Namen “Bootstrap” trägt (siehe Abbildung 28). Es kann einem beliebigen Objekt der Unity Szene zugeordnet werden. Es gilt lediglich zu beachten, dass das Skript durch Unity beim Start aufgerufen wird. Danach wird die eigentliche Applikation hochgefahren.

Bemerkung

Objekten zugeordnete Skripts (*Behaviors*) werden von Unity in ganz bestimmten Situationen aufgerufen. Dafür muss ein Skript Funktionen zur Verfügung stellen, deren Name von Unity vorgegeben wird. Über solche Funktionen definiert ein Skript typischerweise, wie es initialisiert wird (Funktion “Start”) oder was es bei jedem Neuzeichnen der Szene (bzw. Frame) ausführen möchte (Funktion “Update”). Eine gute Übersicht aller Funktionen und wann sie aufgerufen werden, findet man in der Unity-Dokumentation. [[Tec17a](#)]

- **Zustände:** Der Prototyp befindet sich immer in einem von vier möglichen Zuständen. Ein Zustand (*State*) legt sich beim Eintritt jeweils die Szene über Controller (*Controller*) und weitere Helfer (*Services*) so zurecht, wie er sie gerne haben möchte. Zustandswechsel erfolgen über Ereignisse, die durch den Benutzer ausgelöst werden.

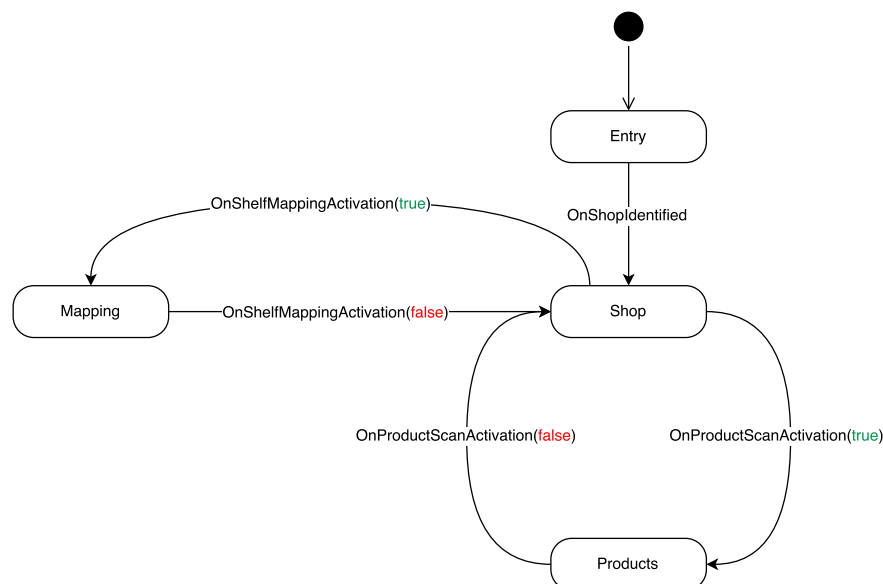


Abbildung 29: Zustände und Übergänge des Prototyps

OnShopIdentified	Der Benutzer identifiziert die Filiale
OnShelfMappingActivation(true)	Der Benutzer startet den Erfassungsmodus für Regale
OnShelfMappingActivation(false)	Der Benutzer navigiert zurück zur Übersicht
OnProductScanActivation(true)	Der Benutzer aktiviert die Erkennung von Produkten
OnProductScanActivation(false)	Der Benutzer deaktiviert die Erkennung von Produkten

Tabelle 4: Zuordnung von Zustandsübergängen zu Benutzeraktionen

- **Testen:** Grundsätzlich kann der Prototyp auch direkt im Unity Editor ausgeführt werden (im sogenannten “Play” Modus). Kopfbewegungen und Gesten des Benutzers können über Tastatur und Maus imitiert werden. Für die Bilderkennung kann eine normale USB-Kamera als Eingangssignal in den Vuforia Einstellungen konfiguriert werden. Zur Simulation spezieller Ereignisse wie beispielsweise Spracheingaben wurde ein Skript mit dem Namen “EventSimulation” erstellt, welches ebenfalls auf Tastatureingaben reagiert. Somit lassen sich alle Benutzerinteraktionen direkt im Unity Editor auslösen, ohne die Applikation für jeden Test auf die HoloLens laden zu müssen.

Daten

Während der Ausarbeitung des Anwendungsfalls für den Prototyp hat sich bereits gezeigt, dass viele Informationen zu einer Filiale und zu den darin angebotenen Produkten bekannt sein müssen. Siehe dazu auch die Auflistung in Kapitel 5.3.2. Lässt man die Existenz bzw. die Beschaffung solcher Daten kurz ausser Betracht, dann bleibt immer noch die Frage, wie man diese Informationen intern in der Applikation abbilden soll. Grundsätzlich gibt es drei Kategorien:

- Organisatorische Informationen zur Gestaltung einer Filiale
→ Sortiment, Produktanordnung in Regalen auf Verkaufsfläche
- Informationen zu einem Produkt, welches innerhalb einer Filiale angeboten wird
→ Lagerbestand, Bestellungen, Verkaufszahlen, Verderb
- Statistiken und Vorhersagen für Verkauf und Lagerverwaltung
→ Potentielle Ladenhüter, Bestellvorschläge für Filiale von Zentrale

Für jede Kategorie wurde ein Datenmodell entworfen, das den jeweiligen Anforderungen gerecht werden soll. Teile davon orientieren sich an echten Beispieldaten, die von Migros zur Verfügung gestellt wurden. Andere sind rein technischer Natur und werden benötigt, damit sich die HoloLens innerhalb einer Filiale orientieren kann. Diese Kombination erlaubt dem Prototyp eine positionsgetreue Darstellung der Hologramme in der Nähe von Regalen.

Abbildung 30 zeigt das Datenmodell zur Abbildung von Informationen rund um die Gestaltung und das Sortiment einer Filiale. Erwähnenswert ist hier die Art, wie sich Regale mit diesem Modell beschreiben lassen. Im Prinzip werden nur die Regalseiten definiert, mit ihren Dimensionen und ihrer Ausrichtung relativ zum Mittelpunkt des Regals. Das schafft eine hohe Flexibilität und erlaubt sowohl die Erfassung von quaderförmigen Regalen, als auch von komplexeren Formen. Produktgruppen können mittels zweidimensionalen Koordinaten auf den Regalseiten platziert werden. Mit diesem Modell hat der Prototyp **genug Informationen, um Hologramme oberhalb des Regals bzw. nahe bei den Produkten im Regal zu platzieren. Der Benutzer kann auf Regale und auf spezifische Produktgruppen in diesen Regalen aufmerksam gemacht werden.**

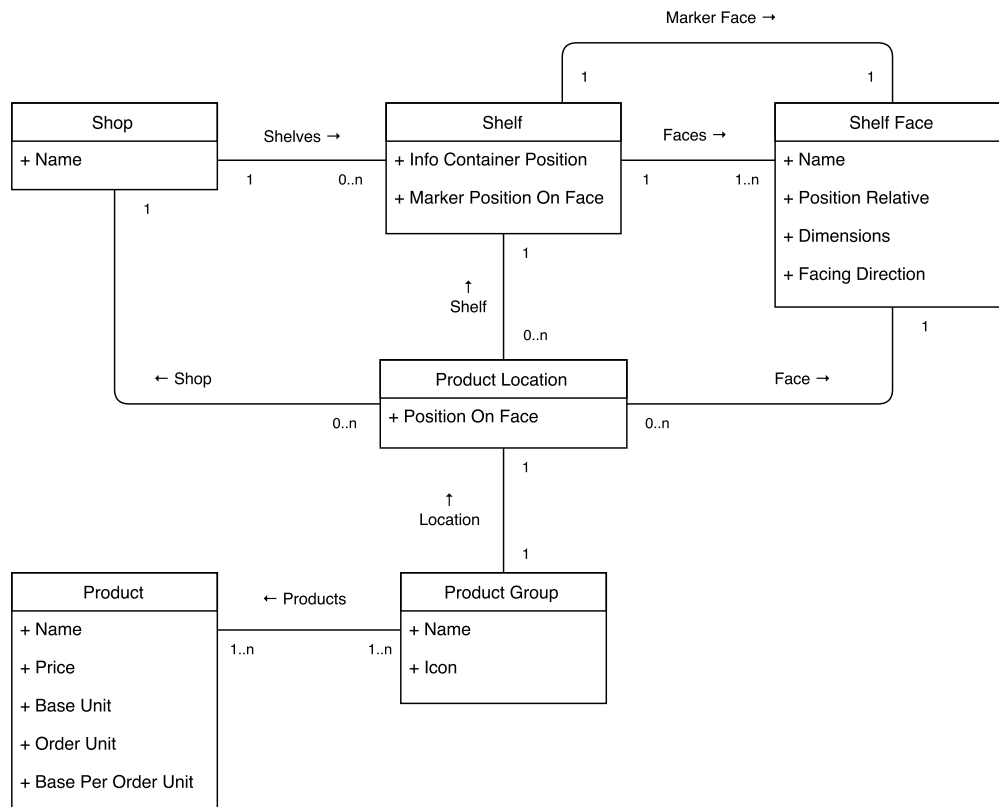


Abbildung 30: Datenmodell zur Abbildung einer Filiale mit Regalen und Produkten

Abbildung 31 zeigt das Datenmodell für Informationen rund um den Verkauf und die Lagerung von Produkten. Solche Informationen sind essenziell für den Detailhandel, weshalb in diesem Fall auf bereits vorhandene Daten als Vorlage für das Modell zurückgegriffen werden konnte. Für offene Bestellungen und für Bestellvorschläge der Zentrale konnte die gleiche Datenstruktur wiederverwendet werden, was das Modell zusätzlich vereinfacht hat. **Der Prototyp wertet diese Informationen aus und generiert daraus eine Zusammenfassung, die dann im Endeffekt für die Darstellung der Hologramme genutzt wird.**

Bemerkung

In der Realität würde man ein Datenmodell dieser Art eher als Teil eines zentralen Web-service erwarten, welcher die Daten speziell für die HoloLens Applikation aufbereitet mit Prognosen und Trends über einen längeren Zeitraum hinweg. Auch ein Vergleich zwischen Filialen wäre denkbar. Die Auswertung von Daten auf der Brille selbst ist in Anbetracht der limitierten Ressourcen nicht wirklich sinnvoll, für den Prototyp allerdings aufgrund der stark limitierten Menge an Testdaten ausreichend.

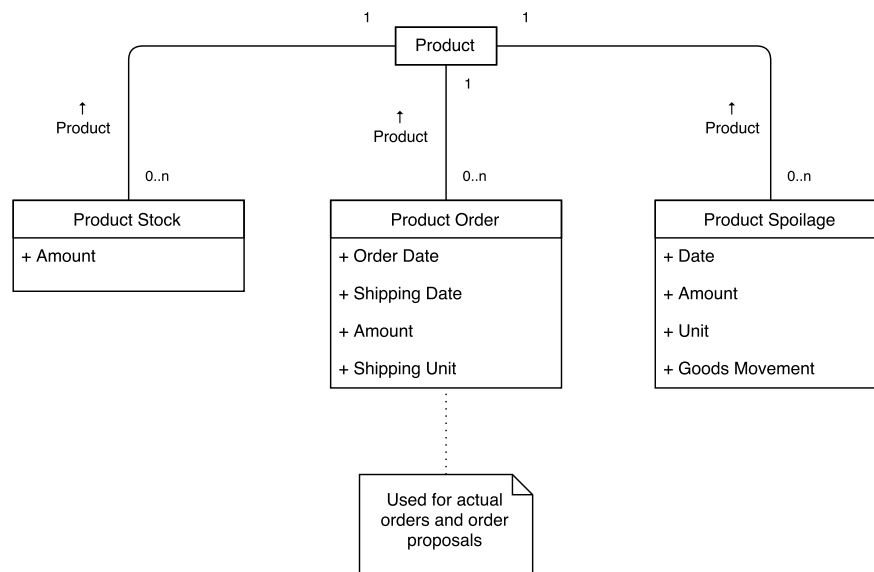


Abbildung 31: Datenmodell für gesammelte Informationen zu einem Produkt

Abbildung 32 zeigt das Datenmodell für die Zusammenfassung verschiedener Informationen rund um ein Produkt in einer Filiale. **Der Prototyp nutzt diese Informationen, um zu entscheiden, wohin der Benutzer abhängig von der ausgewählten Ansicht geleitet werden soll. Hologramme weisen ihm dann den Weg zu den interessanten Stellen der Filiale.** Interessiert er sich beispielsweise für Produkte mit schlechtem Absatz, dann kann ihn der Prototyp innerhalb der Filiale direkt zu Regalen mit betroffenen Produkten leiten.

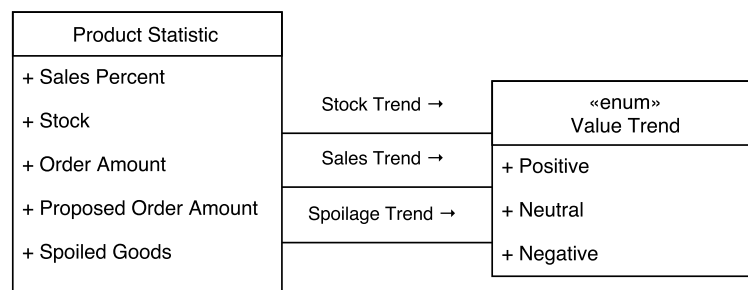


Abbildung 32: Datenmodell für produktbezogene Statistik und Vorschläge

Bemerkung

Die Daten aus Abbildung 30 und 31 sind für den Prototyp in XML Dateien auf der HoloLens hinterlegt und können vor Präsentationen modifiziert werden, ohne die Applikation neu installieren zu müssen. Die Daten in Abbildung 32 werden dann zur Laufzeit daraus berechnet. Eine Anleitung zur Modifikation dieser Daten gibt es in der Installationsanleitung (siehe Anhang B).

7.3 Probleme & Lösungsansätze

Grössenverständnis der HoloLens

Bei der Ausrichtung von Regalen zu ihrem Marker wurde festgestellt, dass die Abmessungen verglichen zur realen Welt proportional zur Grösse zu klein wirken, wenn man sie auf der HoloLens betrachtet. So fehlten bei einem Regal mit 4.30 Metern Länge am Ende ganze 10 cm.

Teilweise gelöst

Der Effekt lässt sich kompensieren, indem die Masse eines Regals entsprechend etwas grösser angegeben werden, als sie es tatsächlich sind. Aufgrund dieser Ungenauigkeit sollten erfasste Regale nicht zu gross sein. Bei sehr grossen Exemplaren bietet es sich an, diese in Abschnitte zu unterteilen, die jeweils mit einem separaten VuMark erfasst werden.

Mess-Ungenauigkeiten Vuforia



Abbildung 33: Erfassung eines Regals mit dem Prototyp

Vuforia liefert bei erfolgreicher Erkennung eines VuMarks dessen Position sowie Rotation im Raum. Diese Werte ändern sich allerdings permanent, solange der Marker im Sichtfeld ist (Mess-Ungenauigkeit). Kopfbewegungen des Benutzer verfälschen diese Werte zusätzlich. Dies stellt ein Problem dar, wenn man ein Regal basierend auf seinem Marker ausrichten will. Verwendet man gleich den ersten gemessenen Wert, kann dies dazu führen, dass ein Regal nach der Erfassung leichte Verdrehungen bzw. Verschiebungen aufweist. Je nach Positionierung des Markers auf dem Regal können schon leichte Abweichungen in der Rotation zu heftigen Fehlern bei der Ausrichtung führen (Hebelwirkung).

Teilweise gelöst

Der Marker sollte an einem Regal so befestigt werden, dass er sich möglichst nahe beim

Zentrum des Regals befindet. Die Distanz zu allen Eckpunkten des Regals sollte also so kurz wie irgendwie möglich sein. Dadurch sind Abweichungen weniger gut sichtbar. Um Mess-ungenauigkeiten sowie Kopfbewegungen des Benutzers auszugleichen, werden zur genaueren Bestimmung der Position und Rotation eines Markers mehrere Messungen durchgeführt. Davon wird der Median berechnet, um die eigentlichen Werte zu bestimmen, die vom Prototyp zur Ausrichtung eines Regals verwendet werden sollen. Der Benutzer wird über eine Animation informiert, wann die Messung abgeschlossen ist (vgl. Abbildung 33). Während der Messung muss er seinen Blick auf den Marker richten und den Kopf möglichst ruhig halten.

Verdeckung durch Umgebung

Hologramme auf der HoloLens sind grundsätzlich von überall her sichtbar. Damit die Hologramme von der Umgebung verdeckt werden, muss das in Kapitel 6.3 bereits beschriebene Spatial Mapping Mesh mit einem speziellen Material versehen und dargestellt werden. Dieses Material muss einen Shader verwenden, der dahinterliegende Objekte verbirgt, wobei das Mesh selbst für den Benutzer unsichtbar bleibt. Er sieht also im Prinzip nur, wie Hologramme durch Hindernisse in seiner Umgebung verdeckt werden. Im HoloToolkit gibt es bereits ein solches Material. Bei der Verwendung im Prototyp führt es aber zu neuen Problemen. Regale der echten Welt kollidieren mit ihrem Overlay in der virtuellen Welt, wodurch Markierungen auf den Seiten eines Regals nur noch teilweise oder gar nicht mehr sichtbar sind. Folgende Quellen haben einen direkten Einfluss auf die Ungenauigkeiten:

- **Vuforia:** Versucht über Bilderkennung die Position eines Objekts im dreidimensionalen Raum zu ermitteln. Je nach Winkel, Lichtverhältnissen und Qualität des VuMark können hier Ungenauigkeiten bei der Ausrichtung eines Regals auftreten.
- **Benutzer:** Die Dimensionen der Regale werden in der realen Welt von Hand ausgemessen. Hier besteht ebenfalls ein Potenzial, dass Ungenauigkeiten auftreten.
- **HoloLens:** Die HoloLens ist kein Messgerät. Sie besitzt zwar ein räumliches Verständnis, muss aber Abstände und Dimensionen dauernd anpassen. Eine auf Zentimeter-genaue Vermessung der Umgebung ist damit nicht möglich.

Die Repräsentation des Regals in der virtuellen Welt müsste perfekt mit dem Regal in der echten Welt übereinstimmen, damit dieser Ansatz vernünftig funktioniert. Das Spatial Mapping Mesh müsste bei der Ausrichtung und Platzierung eines Regals miteinbezogen werden. Nur wenn die Dimensionen bzw. das räumliche Verständnis von HoloLens verwendet wird, um die Konturen eines Regals zu erkennen, kann sichergestellt werden, dass keine Überschneidungen mit dem Overlay auftreten. Dies erfordert aber aufwändige Analysen der Umgebung. Vermutlich müssten für diesen Ansatz auch detailliertere 3D-Modelle der Regale zur Verfügung stehen. Was das Problem weiter verkompliziert ist die Tatsache, dass unter Umständen noch nicht die ganze Umgebung von der HoloLens erfasst worden ist. Somit bräuchte es eine zusätzliche Phase, in der eine Filiale in einem ersten Schritt komplett erfasst wird. Im Rahmen dieser Studienarbeit ist das Problem aus Zeitgründen nicht auf diesem Weg lösbar.

Teilweise gelöst

Um den Effekt der Verdeckung wenigstens teilweise zu erreichen, wird jedes Regal mit einer schwarzen Oberfläche versehen. Auf der HoloLens führt dies dazu, dass Hologramme hinter einem erfassten Regal nicht mehr sichtbar sind. Da in einer Filiale typischerweise Regale die einzigen grossen Hindernisse sind, funktioniert der Trick für diesen Anwendungsfall.

Performance Probleme

Insbesondere auf mobilen Geräten wie der HoloLens-Brille muss besonders auf die Auslastung der Hardware geachtet werden. Wie in Kapitel 6.4 bereits beschrieben wurde, sollte die HoloLens stets mit 60 FPS laufen, um dem Benutzer ein optimales Erlebnis bieten zu können. Im Laufe der Entwicklung des Prototyps sind regelmässige Einbrüche runter bis auf 30 FPS aufgefallen. Je nachdem, welche Hologramme betrachtet wurden, fiel die Bildrate teilweise sogar unter 15 FPS. Eine Analyse hat gezeigt, dass die CPU durch die GPU über einen grossen Zeitraum hinweg blockiert wurde. Aufgrund dieser Erkenntnisse wurden einige Optimierungen an der Szene und am Verhalten des Prototyps vorgenommen.

Bemerkung

Achtung: Wenn die Applikation im “Debug” Modus gestartet wird, hat dies einen enormen Einbruch der Bildrate zur Folge. Der Zugriff auf die Frontkamera der HoloLens für die Bilderkennung führt ausserdem dazu, dass die Bildrate automatisch auf 30 FPSs begrenzt wird. Tests bezüglich Performance sollten aus diesem Grund immer im “Release” Modus und - wenn möglich - mit ausgeschalteter Kamera durchgeführt werden.

Bemerkung

Mithilfe des Unity Profiler, des Unity Frame Debugger und des HoloLens Device Portal kann die Performance von Applikationen für die HoloLens genauer analysiert werden. Weitere Informationen dazu in Anhang C.

Teilweise gelöst

Microsoft hat Richtlinien herausgegeben, welche dabei helfen sollen, die Performance von HoloLens-Applikationen zu verbessern [Mic17j]. Folgende Optimierungen wurden auf der Basis dieser Richtlinien und den Empfehlungen von Unity am Prototyp vorgenommen:

- **Statische Objekte als “Static” markieren:** Objekte, welche sich nicht bewegen und das gleiche Material mit anderen Objekten teilen, sollten in Unity im “Inspector” Fenster als “Static” markiert werden. Dies führt dazu, dass weniger Aufrufe an die GPU nötig sind.
- **Low-Poly Modell:** In der gesamten Szene wurden nur 3D Modelle verwendet, wel-

che über einen niedrigen Detailgrad verfügen.

- **Collider:** Collider sollten nur verwendet werden, wenn sie tatsächlich gebraucht werden. Insbesondere auf die Verwendung der Mesh-Collider sollte nach Möglichkeit verzichtet werden, da diese meist eine komplizierte Form besitzen und die Berechnung einer Kollision darum massiv aufwändiger ist.
- **Kamera deaktivieren:** Sobald die Kamera verwendet wird, sorgt HoloLens dafür, dass die Bildrate auf maximal 30 FPS beschränkt wird. Deshalb sollte nur selektiv auf die Kamera zugegriffen werden.
- **Performante Shader:** Für das Rendering auf einer Plattform mit wenig Rechenleistung (wie der HoloLens), sollten möglichst performante Shader verwendet werden. Im HoloToolkit sind einige solche Shader enthalten.

Natürlich lösen diese Anpassungen nicht alle Probleme. Wenn ernsthafte Performance-Probleme bestehen, empfiehlt es sich auf jeden Fall das Problem individuell zu analysieren. Grundsätzlich hilft es aber, die Richtlinien von Microsoft und Unity zu befolgen, um Engpässen bereits frühzeitig aus dem Weg zu gehen.

Der Prototyp läuft nun meist konstant mit 60 FPS und weitere Performance Optimierungen werden aus diesem Grund nicht vorgenommen. Sobald man sich allerdings sehr nahe an ein Hologramm begibt, bricht die Bildrate wieder ein. Dies könnte allerdings mit der in Kapitel 6.4 beschriebenen Stabilization Plane zu tun haben.

Speicherverbrauch

Memory

Installed Memory: 2.0 GB



Total: 1.9 GB
In use: 1.0 GB
Available: 886.4 MB
Committed: 1.0 GB
Paged: 78.3 MB
Non-paged: 180.4 MB

Abbildung 34: Durch Betriebssystem belegter Arbeitsspeicher

Das aktuelle Modell der HoloLens verfügt über 2 Gigabyte Arbeitsspeicher. Davon sind, ohne dass eine eigene Applikation läuft, bereits rund 1 Gigabyte durch das Betriebssystem belegt (siehe Abbildung 34). Eine Analyse hat ergeben, dass der Prototyp verhältnismässig viel Arbeitsspeicher verbraucht. Fast **400 Megabyte** werden zusätzlich reserviert.

Teilweise gelöst

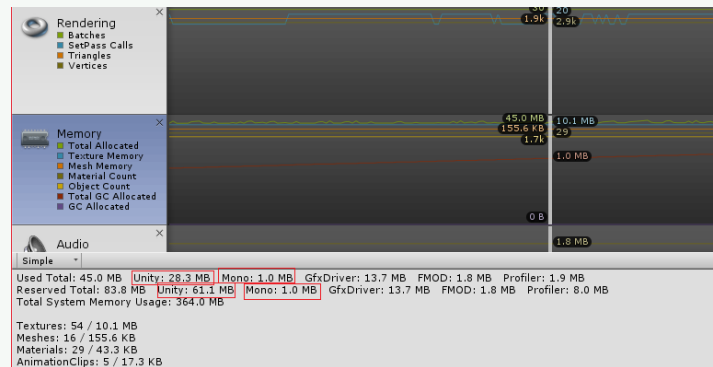


Abbildung 35: Auszug des Unity Profiler bezüglich Speicherverbrauch

Wie in Abbildung 35 zu sehen ist, wird der grösste Teil des Arbeitsspeichers von der Unity-Engine und Grafiktreibern belegt. Assets wie Texturen, Materialien, Audio usw. machen nur einen Bruchteil des verbrauchten Speichers aus. Eine Verbesserung ist aus diesem Grund vermutlich schwer zu erzielen und hat nur einen kleinen Einfluss auf den gesamt-haft verbrauchten Speicher.

Informationen zur Filiale

Der Prototyp braucht diverse Daten zur Darstellung der Hologramme in einer Filiale (siehe Kapitel 7.2 für Details). Die Daten sind aktuell statisch in XML Dateien auf der HoloLens hinterlegt. Dieses Dateien können zwar für Demonstrationszwecke ausgetauscht werden (siehe Anhang B), dies ist allerdings etwas umständlich und für eine produktive Anwendung nicht geeignet.

Teilweise gelöst

Um den Prototyp produktiv einsetzen zu können, müsste eine Schnittstelle auf einem Server angeboten werden, über welche die HoloLens aufbereitete Daten beziehen kann. Aus Zeitgründen wurde dieses Problem nicht weiter untersucht. Zudem stellt die Anbindung an einen Server keine grosse Herausforderung dar und sollte im Falle einer Umsetzung im produktiven Umfeld kein wirkliches Problem darstellen.

Position von Produkten

Im Prototyp sind die Positionen der Produktgruppen innerhalb der Regale statisch erfasst. In der Realität müssten diese Informationen zuerst pro Filiale komplett erfasst werden. Zudem wäre es bei jeder Umstrukturierung der Filiale nötig, die Daten entsprechend zu aktualisieren.

Ungelöst

Aus Zeitgründen wurde dieses Problem nicht weiterverfolgt. Um die Applikation produktiv einsetzen zu können, müsste zumindest der Prozess für die Erfassung dieser Daten vereinfacht werden. Vorstellbar wäre eine Desktop-Applikation, über welche ein Filialleiter die Produkte den virtuellen Regalen zuteilen kann. Als Alternative könnte eine Korrekturmöglichkeit direkt auf der HoloLens angeboten werden, was eine einfache Aktualisierung der Produktpositionen ermöglichen würde.

Ein weiterer Ansatz könnte darin bestehen, die Produkte über die HoloLens direkt im Regal zu erkennen und ihre Position jeweils dynamisch anzupassen. Im Moment ist dazu aber die Erkennungsdistanz von Vuforia noch etwas zu kurz. Insbesondere wenn die Produkte von der Seite oder von oben betrachtet werden, funktioniert die Erkennung noch nicht ausreichend gut.

Sporadische Abstürze

Während der Erfassung von Regalen und während der genaueren Betrachtung von Produkten kommt es sporadisch zu Abstürzen der Applikation. Eine Analyse der Crash Dumps hat ergeben, dass in Vuforia eine Speicherschutzverletzung (Access Violation) verursacht wird. Es scheint so, als ob schnelle Kopfbewegungen mit aktivierter Bilderkennung tendenziell schneller in diese Situation führen. Zuverlässig reproduzieren lässt sich der Fehler nicht.

Ungelöst

Leider geben die Crash Dumps nicht viele Hinweise darauf, wo das Problem liegen könnte. Der Prototyp selbst verwendet keinen "unsafe" Code, und innerhalb der .NET Common Language Runtime sind solche Verletzungen bei Speicherzugriffen normalerweise nicht möglich. Es ist nicht auszuschliessen, dass der Prototyp eine der vielen Komponenten falsch verwendet und sie dadurch in einen ungültigen Zustand bringt. Im Forum von Vuforia wurde für dieses Problem ein Beitrag erfasst ³.

³<https://developer.vuforia.com/forum/hololens/access-violation-when-enabling-vuforia-unity>

8 Schlussfolgerungen

8.1 Technologie

Setzt man jemandem die HoloLens zum ersten Mal auf, dann sind die meisten sehr fasziniert davon, was mit Technik heutzutage alles möglich ist. Doch wie lange hält diese Faszination an? Im Verlauf der Studienarbeit konnte viel Erfahrung gesammelt werden, was die Funktionalität und Praktikabilität der Brille im Alltag betrifft. Dabei sind sowohl positive wie auch negative Aspekte zum Vorschein gekommen.

Ein Grossteil der negativen Aspekte lässt sich vermutlich darauf zurückführen, dass es sich bei der HoloLens eher um einen Prototyp, als um ein marktreifes Produkt für die grosse Masse handelt. Das macht sich auch am hohen Preis von 3000 USD⁴ bemerkbar. Die Brille soll eher zeigen, was jetzt und möglicherweise in der Zukunft alles möglich sein wird. Aus Sicht eines Entwicklers ist es die ideale Gelegenheit, um sich in die Welt von Mixed Reality Applikationen einzuarbeiten.

Bei längerem Tragen wird das Gewicht der Brille und die Wärmeentwicklung teilweise etwas unangenehm. Brillenträger können die HoloLens zwar nutzen, allerdings wird der Tragekomfort dadurch zusätzlich beeinträchtigt. Der mitgelieferte Überkopfbügel entschärft dieses Problem ein wenig. Ein sehr viel grösseres Manko ist das stark eingeschränkte Sichtfeld. Im Verhältnis zum kompletten Sichtfeld eines Menschen ist der Bereich für die Darstellungen der Hologramme definitiv zu klein. Bei grösseren Objekten reisst es den Benutzer völlig aus dem Erlebnis, weil das Gefühl aufkommt, nur noch durch zwei kleine Fenster zu schauen. Wenn eine Applikation ihre Hologramme ungünstig platziert, dann bedeutet das für den Benutzer viel Beinarbeit und Kopfbewegungen, bis die Hologramme wieder als ganzes wahrnehmbar sind (sofern er sie wieder findet). Im Gegensatz zu einer VR-Brille läuft man dabei wenigstens nicht Gefahr, über irgendwelche Kabel zu stolpern.

Technisch gesehen funktioniert HoloLens überraschend gut. Es ist beeindruckend, wie ein kompletter Computer in einer Brille Platz findet. Bei der Entwicklung von eigenen Applikationen sollte man sich vorzugsweise an die Richtlinien von Microsoft halten. Die Rechenleistung ist durchaus begrenzt und es gilt, diese so optimal wie möglich auszunutzen. Teilweise problematisch sind äussere Lichtverhältnisse, die das räumliche Verständnis der Brille etwas durcheinander bringen können. An der Decke befestigte Lampen sind diesbezüglich ein potentieller Störfaktor, wenn man sich genau darunter stellt. Ausserdem heben sich Hologramme immer ein wenig von der Realität ab, weil sie sich nicht der Beleuchtung ihrer Umgebung anpassen.

Sehr faszinierend ist das Erlebnis, wenn man sich mit der HoloLens zum ersten Mal ein virtuelles Objekt von allen Seiten genau anschauen und sich physikalisch darum herumbewegen kann. Die verbauten Lautsprecher tragen dank Raumklang dazu bei, dass man in diese kombinierte Welt aus Realität und Virtualität abtauchen kann. Die HoloLens erkennt die Umgebung mit Wänden und Hindernissen in den meisten Fällen sehr gut. Die Einbettung der realen Umgebung in die eigene Applikation eröffnet neue Möglichkeiten der Visualisierung, die viel mehr Freiheiten erlauben, als das auf einem klassischen Bildschirm der Fall ist.

⁴Stand Mitte 2017 für die Entwicklerversion

Abgesehen von zu erwartenden technologischen Verbesserungen müssen Geräte wie die HoloLens erst noch in der Gesellschaft ankommen. Im Moment ist es eher unangenehm, wenn man mit dieser überdimensionalen Brille in die Öffentlichkeit tritt, und mit den Händen wild in der Luft herumfuchelt. Trotzdem ist HoloLens ein Schritt in die richtige Richtung, um Entwickler, Designer und Hardware-Hersteller dazu zu animieren, sich neue Wege der Darstellung und Interaktion zwischen Mensch und Computern zu überlegen. Schliesslich findet zwischenmenschliche Interaktion über Augen, Hände, Ohren und Stimme statt und nicht über Maus oder Tastatur.

8.2 Entwicklungsumgebung

Unity bietet dem Entwickler einen verhältnismässig einfachen Einstieg in die Welt der 3D-Programmierung. Auch wenn man noch nie mit einer 3D Engine gearbeitet hat, findet man sich schnell zurecht. Unity hat eine integrierte Unterstützung für VR und insbesondere für HoloLens. Auch wenn dieser Teil noch stetig weiterentwickelt wird und sich dort vermutlich noch einiges ändern wird, ist die Integration bereits sehr gut. Innerhalb von wenigen Minuten hat man ein kleines lauffähiges Projekt für HoloLens erstellt.

Die Dokumentation ist sehr detailliert und man findet zu praktisch jedem Aspekt der Engine einen Beitrag oder eine Anleitung. Unity ist für kleinere Projekte sowohl im privaten wie auch im kommerziellen Umfeld frei verwendbar. Das führt unter anderem dazu, dass es durch eine grosse Community getragen wird. Auch die Dokumentation von Microsoft zu HoloLens und insbesondere zur Integration in Unity ist bereits sehr ausgereift.

Für die Arbeit im Team ist Unity nur bedingt geeignet. Solange die Teammitglieder an einzelnen Assets oder Scripts arbeiten entstehen kaum Probleme. Aufpassen muss man allerdings bei der Bearbeitung der Szenen. Änderungen von mehreren Personen lassen sich diesbezüglich nur sehr schwer zusammenführen. Dieses Problem lässt sich umgehen, indem man die Arbeit auf mehrere Szenen verteilt. So kann trotzdem parallel an einem Projekt gearbeitet werden, ohne dass man permanent Konflikte auflösen muss.

Der Abstraktionsgrad ist relativ hoch, so dass die Engine einem vieles abnimmt und den Einstieg auch für Personen mit wenig Programmiererfahrung einfacher gestaltet. Dies führt aber auch dazu, dass einem manchmal etwas die Kontrolle und die Übersicht über die genauen Mechanismen der Engine fehlen. Insbesondere wenn man für schwächere Hardware entwickelt, muss man sich zu einem grossen Teil darauf verlassen, dass die Engine entsprechende Optimierungen vornimmt.

8.3 Use Case

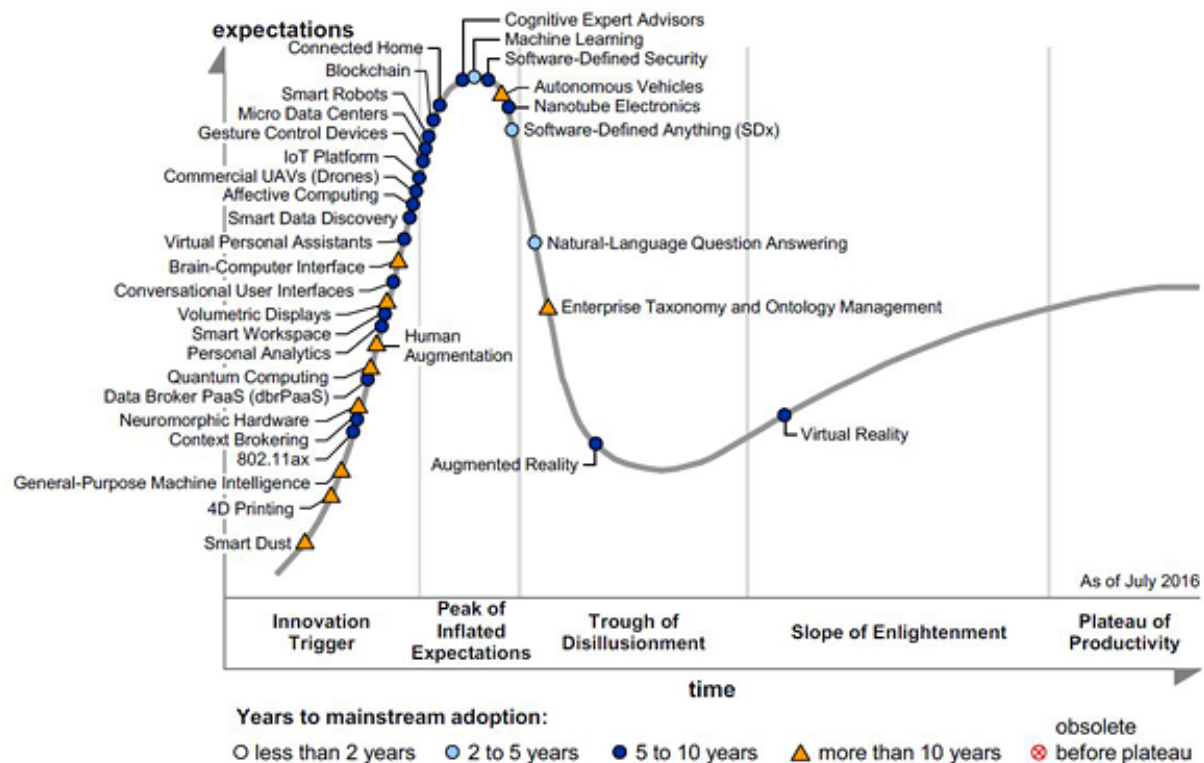
Mithilfe des Prototyps könnte ein Filialleiter in Zukunft direkt im Laden die Verwaltung seiner Filiale vornehmen. Statt sich einen Überblick auf dem Papier zu verschaffen, wird er sich gemütlich durch die Filiale bewegen und sich wo nötig weitere Informationen anzeigen lassen. Seine Wahrnehmung wird direkt mit den entsprechenden Informationen angereichert. Dadurch kann er effizienter und auf intuitive Art und Weise seine Arbeit erledigen. Die alten Spreadsheet-Tabellen gehörten der Vergangenheit an.

Ein Problem besteht allerdings darin, die dafür nötigen Informationen zusammenzutragen und aktuell zu halten. Jedes Regal und seine dazugehörigen Produkte müssen erfasst und permanent auf dem neusten Stand gehalten werden. Dies verursacht einen riesen Verwaltungsaufwand. Zusammen mit den Kosten für die Entwicklung einer marktreifen Applikation wären die Gesamtkosten vermutlich grösser als der effektive Nutzen, der dadurch erzielt wird.

Eine Lösung für dieses Problem könnte darin bestehen die aufwändige Datenerfassung durch komplexere Software zu automatisieren. Effektiv hinterlegt als Vorlage wären in diesem Fall dann nur noch die zu erkennenden Produkte. Die Brille würde selbständig feststellen, in welcher Filiale sie sich gerade befindet und wo die Produkte einsortiert sind. Bis dies möglich sein wird muss sich die Technologie vorallem im Bereich der Bilderkennung noch stärker weiterentwickeln.

Die grosse Stärke von HoloLens liegt zurzeit im visuellen Erlebnis. Es ist beeindruckend zu sehen, wie gut sich Hologramme bereits in die Realität einbetten lassen. Man kann sich ausmalen, zu was diese Technologie in Zukunft fähig sein wird. Wer HoloLens bereits heute einsetzen will, sollte den primären Fokus auf dieses Erlebnis legen. In Bereichen wie Marketing, Design oder Ausbildung bringt die Brille bereits jetzt schon einen grossen Mehrwert. Sie erlaubt neue Perspektiven auf Dinge, die man sich vielleicht sonst gar nicht oder nur schwer vorstellen kann.

8.4 Ausblick



Source: Gartner (July 2016)

Abbildung 36: Hype-Zyklus nach Gartner Inc. [Gar16]

In letzter Zeit hat sich einiges im Bereich von Mixed Reality getan. Trotzdem ist die Technologie noch nicht ganz soweit, um bereits produktiv im grossen Stil eingesetzt werden zu können. Dies könnte sich aber im Verlauf der nächsten paar Jahre schnell ändern. Der Hype-Zyklus der Firma Gartner Inc. zeigt welche Phasen der öffentlichen Aufmerksamkeit eine Technologie nach ihrer Ankündigung durchlaufen wird. Gemäss der letzten Einschätzung aus dem Jahr 2016 befindet sich Augmented Reality im sogenannten „Tal der Enttäuschung“ (siehe Abbildung 36). Laut Gartner wird es noch ca. 5 - 10 Jahre dauern, bis die Technologie in der Allgemeinheit angekommen ist. Wenn man sich allerdings die rasante Entwicklung anderer Technologien wie zum Beispiel dem Smartphone ansieht, könnte diese Entwicklung durchaus auch schneller stattfinden.

Grosskonzerne wie Google und Microsoft investieren zurzeit kräftig in Mixed Reality. Mit dem „Creators Update“ wurden beispielsweise erst kürzlich neue Funktionalitäten für die Verwendung von Mixed Reality Geräten in Windows 10 eingebaut. Hardware Hersteller wie HP, Dell, Acer, ASUS und Lenovo wollen in naher Zukunft neue und vorallem günstigere Geräte auf den Markt bringen. Damit wäre die Technologie für ein breiteres Publikum zugänglich. [Sha17]

Dies sind vielversprechende Aussichten und man darf gespannt sein, was die Zukunft bringen wird. Selbst wenn es sich nur um einen temporären Hype handelt, manch einer Zukunftsvision ist man damit bereits ein Stück nähergekommen.

Anhang

A Lieferumfang

Jegliche Inhalte, die während dieser Studienarbeit produziert wurden, sind auf einem privaten Git Server abgelegt. In der Abgabe der Studienarbeit sind keine Binärdateien enthalten, der Source Code der Applikation muss selber kompiliert werden. Diesen kann man über folgende Adresse via Git klonen:

<https://git.narcodex.ch/hsr/student-research-project.git>

Den beteiligten Personen wurden zu Beginn des Projekts die benötigten Zugangsdaten via E-Mail ausgehändigt. Die Ordnerstruktur des Git Repository ist in Tabelle 5 beschrieben.

Ordner	Inhalt
journal	X _Y TeX Quelldateien, Illustrationen und Quellenverweise des Projektjournals.
report	X _Y TeX Quelldateien, Illustrationen und Quellenverweise des technischen Berichts.
src ▶ 01-Experiment	Unity Projekt für das erste Experiment, beschrieben in Kapitel 6.2
src ▶ 02-Experiment	Unity Projekt für das zweite Experiment, beschrieben in Kapitel 6.3
src ▶ 03-Experiment	Unity Projekt für das dritte Experiment, beschrieben in Kapitel 6.4
src ▶ 04-Migros-Showcase	Unity Projekt für den finalen Prototyp, beschrieben in Kapitel 7
src ▶ 05-Migros-Showcase-Data	Visual Studio Projekt für C# Klassenbibliothek, die das Datenmodell des Prototyps speichert

Tabelle 5: Ordnerstruktur des Git Repository mit den Resultaten der Studienarbeit

Der Prototyp verwendet teilweise käuflich erworbene Assets aus dem Unity Asset Store. Im Falle einer Weiterentwicklung des Projekts müssten die kostenpflichtigen Assets dementsprechend erneut gekauft werden, da sie nicht übertragbar sind. Tabelle 6 zeigt eine Auflistung der betroffenen Assets.

Name	Preis
UMenu: World Space	7 USD
Quest Markers 3D	7.99 USD
UI Sound Effects Collection Pack 2: Buttons	-

Tabelle 6: Auflistung von Assets aus dem Unity Asset Store

B Installationsanleitung

Die nachfolgenden Schritte beschreiben, wie man den Prototyp auf einer HoloLens installiert. Die Anleitung orientiert sich dabei an der Microsoft Dokumentation zur Einrichtung der Entwicklungsumgebung für die HoloLens [Mic17g]. Während der Entwicklung kam folgende Umgebung zum Einsatz:

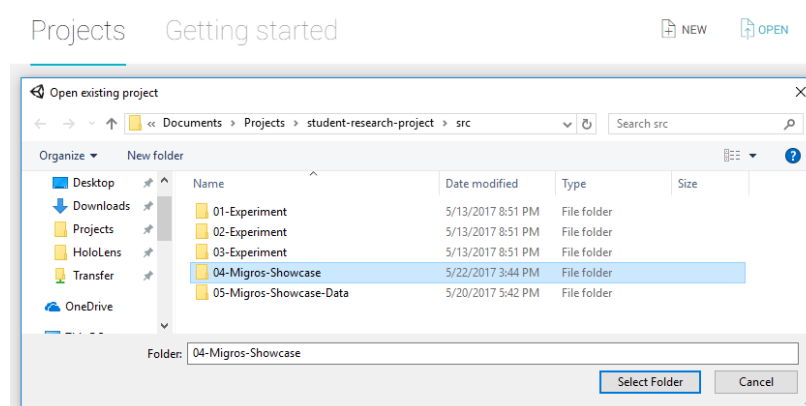
- Windows 10
- Visual Studio 2015 Update 3 (Community Edition)
- Windows App Development Tools 1.4 und SDK 10.0.10586 (via Visual Studio Installation)
- Unity 5.5.2
- Windows Metro Integration für Unity

Bemerkung

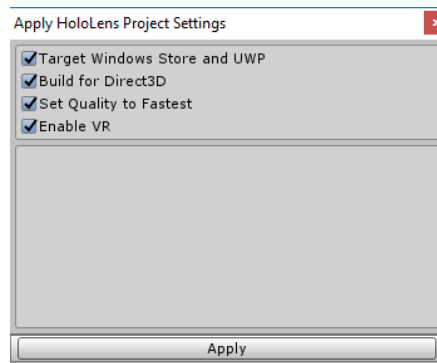
Theoretisch stellt Microsoft auch einen Emulator für die HoloLens zur Verfügung. Dieser kam in diesem Projekt allerdings so gut wie nie zum Einsatz. Glücklicherweise bestand für den gesamten Zeitraum der Arbeit die Möglichkeit, auf ein richtiges Exemplar zurückzugreifen. Viel lässt sich allerdings auch direkt im Unity Editor ausprobieren.

Die beschriebene Umgebung muss vorhanden sein, um den Prototyp installieren zu können. Zudem setzen alle nachfolgenden Schritte voraus, dass eine Kopie der Resultate dieser Arbeit (siehe Anhang A) auf dem eigenen Computer vorhanden ist.

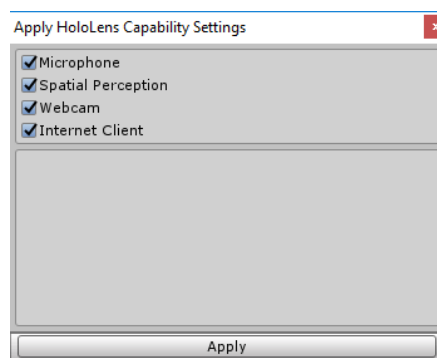
1. Prototyp Projekt im Startfenster des Unity Editors öffnen mit **Open**.



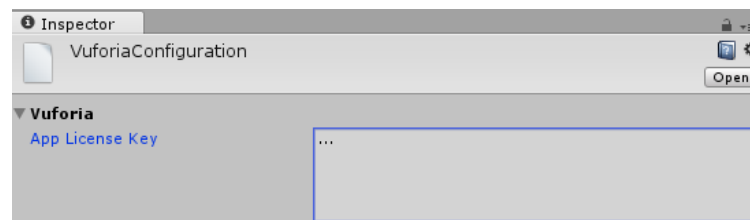
2. In der Menüleiste des Unity Editors über **HoloToolkit > Configure > Apply HoloLens Project Settings** alle aufgelisteten Einstellungen aktivieren.



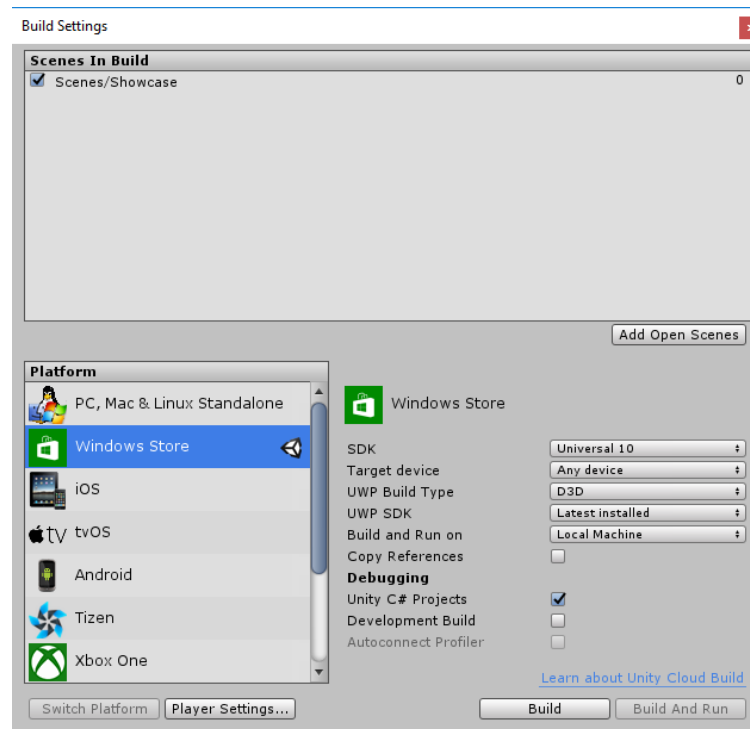
3. In der Menüleiste des Unity Editors über **HoloToolkit** » **Configure** » **Apply HoloLens Capability Settings** alle aufgelisteten Berechtigungen aktivieren.



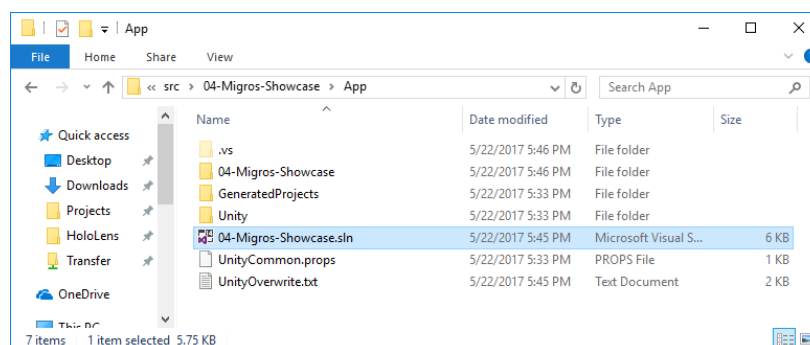
4. Für Vuforia wird ein Lizenzschlüssel benötigt, welcher in der Unity Editor Menüleiste über **Vuforia** » **Configuration** eingetragen werden kann. Der für den Prototyp verwendete Schlüssel darf nach der Studienarbeit nicht weiterverwendet werden.



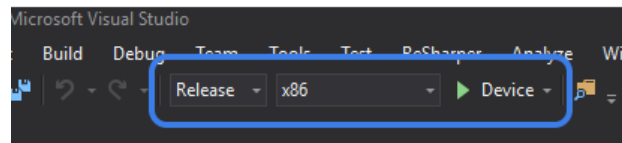
5. In der Menüleiste des Unity Editors über **File** » **Build Settings...** die nachfolgend gezeigten Einstellungen übernehmen. Bei einer frischen Installation weist Unity darauf hin, dass zuerst die Windows Store Integration installiert werden muss. Der Hinweis enthält weitere Anweisungen zur Installation dieser Erweiterung, weshalb hier nicht näher darauf eingegangen wird.



6. Im Dialog aus letztem Schritt über **Build** die Erstellung der Visual Studio Solution für die HoloLens Applikation starten.
7. Als Zielordner einen neuen, leeren Ordner auswählen. Der Name kann frei gewählt werden, beispielsweise "App". Beim nächsten Mal wird Unity nur noch die effektiv veränderten Dateien in diesem Ordner ersetzen, wenn man den existierenden Ordner als Ziel auswählt.
8. Die erstellte Solution (siehe *.sln Datei in zuvor definiertem Zielordner) kann jetzt in Visual Studio geöffnet werden. Ab diesem Zeitpunkt braucht man den Unity Editor im Prinzip nicht mehr, ausser es müssen Änderungen an der Szene vorgenommen werden. Danach müssen jeweils die Schritte 5 - 7 wiederholt werden.



9. In der Visual-Studio-Konfiguration die nachfolgend gezeigten Einstellungen übernehmen. Für Debugging-Aktivitäten von "Release" auf "Debug" umschalten, die Applikation läuft dann allerdings mit deutlich reduzierter Frame-Rate.



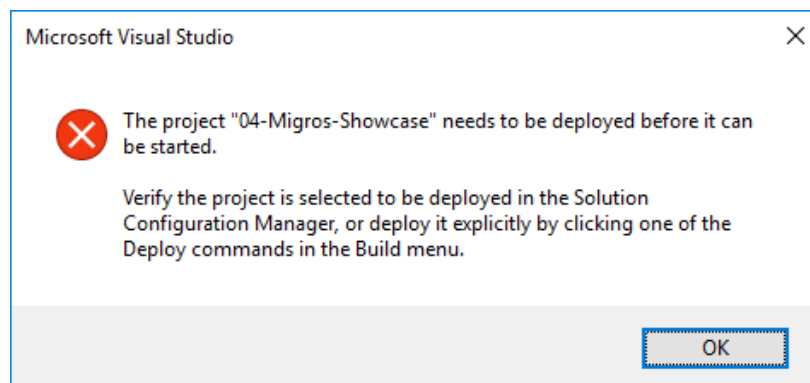
10. Die HoloLens via USB Kabel mit Computer verbinden und sicherstellen, dass sie eingeschaltet und aktiv ist. Je nach Systemkonfiguration fällt die Brille relativ schnell in einen Stromsparmodus, aus dem sie vor der Installation durch kurzes Antippen des Einschaltknopfs geweckt werden muss.
11. In der Visual Studio Menüleiste über **Debug** » **Start Without Debugging** die Installation der Applikation auf der HoloLens starten. Für Debugging Aktivitäten kann man über **Debug** » **Start Debugging** problemlos direkt in Visual Studio auf Fehlersuche gehen.
12. Der Prototyp sollte nun auf der HoloLens installiert sein und direkt ausgeführt werden.

Mögliche Fehlerquellen

Bemerkung

Die Einstellungen in Schritt 2 und 3 können in Unity auch von Hand vorgenommen werden, bei Problemen mit der Installation sollten in einem ersten Schritt die Einstellungen in Unity mit den Vorgaben von Microsoft auf der Webseite abgeglichen werden. Allerdings hat sich die gezeigte Methode bewährt und stellt zudem sicher, dass man keine Einstellungen vergisst.

Während der Entwicklung funktionierte die Installation manchmal erst beim zweiten Anlauf. Dieses Problem trat meistens dann auf, wenn die Applikation bereits auf der HoloLens installiert war. Visual Studio versucht in diesem Fall die alte Version zu entfernen und die neue aufzuspielen, und dabei kann der folgende Fehler auftreten:



Erneutes Starten der Installation löste das Problem in den meisten Fällen.

Testdaten

Wie in Kapitel 7.3 bereits angesprochen, sind alle visualisierten Informationen zu einer Filiale statisch als XML Dateien hinterlegt. Zugriff auf diese Dateien erhält man über das in Anhang

C gezeigte Device Portal der HoloLens. Die Struktur des XML orientiert sich an der in Kapitel 7.2 beschriebenen Datenstruktur. Bei der Installation wird ein Satz an Testdaten mitgeliefert. Diese können dann beispielsweise für Demonstrationen entsprechend verändert werden. Die Dateien befinden sich auf der HoloLens im Verzeichnis:

📁 User Files ▶ LocalAppData ▶ 04-Migros-Showcase ▶ LocalState ▶ Migros

Bemerkung

Der Prototyp muss nach der Installation einmal gestartet werden, damit die XML Dateien in das hier beschriebene Verzeichnis kopiert werden. Dieser zusätzliche Schritt ist deshalb notwendig, weil die Daten bei der Installation in ein Verzeichnis kopiert werden, auf das man über den File Explorer im Device Portal keinen Zugriff hat.

Zur Modifikation der Testdaten sind folgende Schritte nötig:

1. XML Dateien über Device Portal herunterladen
2. Änderungen in den XML Dateien vornehmen (Syntax prüfen!)
3. Alte XML Dateien über Device Portal löschen
4. Bearbeitete XML Dateien via Device Portal hochladen
5. Prototyp neu starten (Dateien werden nur beim Start ausgelesen)

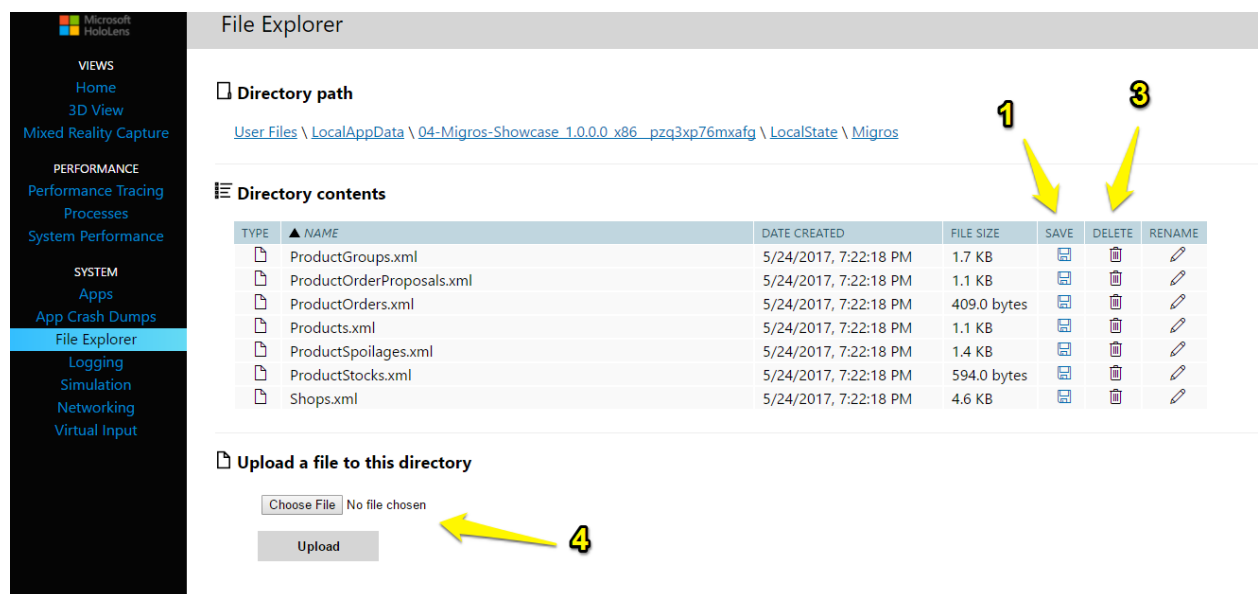


Abbildung 37: Die Testdaten des Prototyps können über das Device Portal modifiziert werden

Datei	Inhalt
Shops.xml	Filialen mit den darin platzierten Regalen inklusive Abmessungen
ProductGroups.xml	Listet Produktgruppen auf und weist sie einem Regal zu
Products.xml	Listet alle Produkte im Sortiment auf mit Basisinformationen wie Bezeichnung oder Preis
ProductStocks.xml	Listet alle Lagerbestände der einzelnen Produkte auf
ProductOrderProposals.xml	Listet alle Bestellvorschläge der einzelnen Produkte auf
ProductSpoilages.xml	Listet Statistiken zum Verderb von Produkten auf

Tabelle 7: Auflistung der XML Dateien mit Testdaten

C Entwicklungsumgebung

Profiler

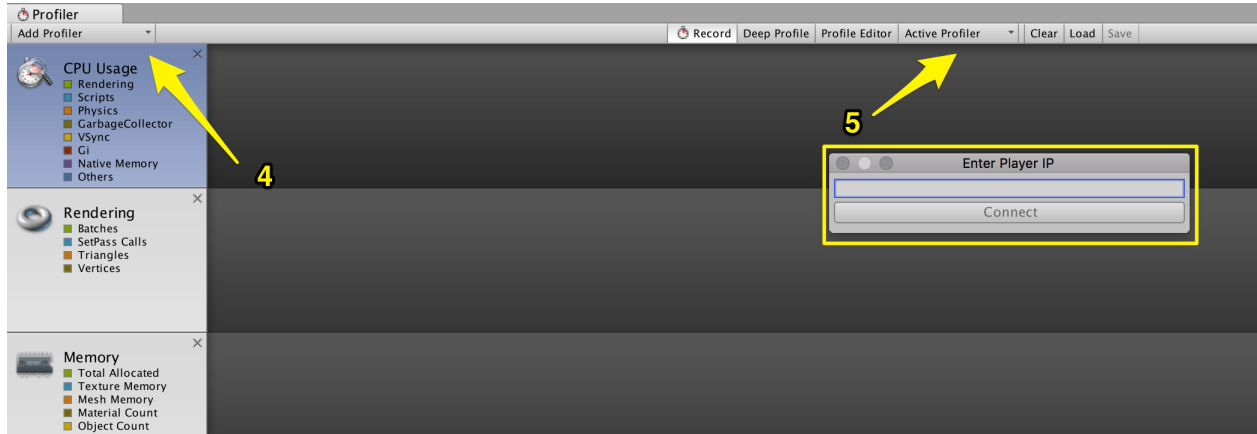


Abbildung 38: Unity Profiler

Unity bietet ein Werkzeug an, mit welchem die Performance einer Applikation auf der HoloLens analysiert werden kann. Dazu benötigt die Applikation aber die Berechtigung um auf Netzwerkressourcen zuzugreifen. Folgende Schritte müssen befolgt werden, um den Profiler zu verwenden:

1. In Menüleiste von Unity über `File` » `Build Settings...` » `Player Settings...` » `Windows Store` » `Publishing Settings` » `Capabilities` folgende Einträge aktivieren: *PrivateNetworkClientServer*, *InternetClient*, *InternetClientServer*
2. Applikation auf der HoloLens installieren wie in Anhang B beschrieben
3. In Menüleiste von Unity `Window` » `Profiler` auswählen
4. Optional: `Add Profiler` » `GPU` für Details zur Auslastung der Grafikeinheit
5. Profiler über `Active Profiler` via IP Adresse der HoloLens verbinden

Genauere Informationen zur Anwendung und Interpretation sind auf der Unity Webseite beschrieben. [[Tec17c](#)]

Frame Debugger

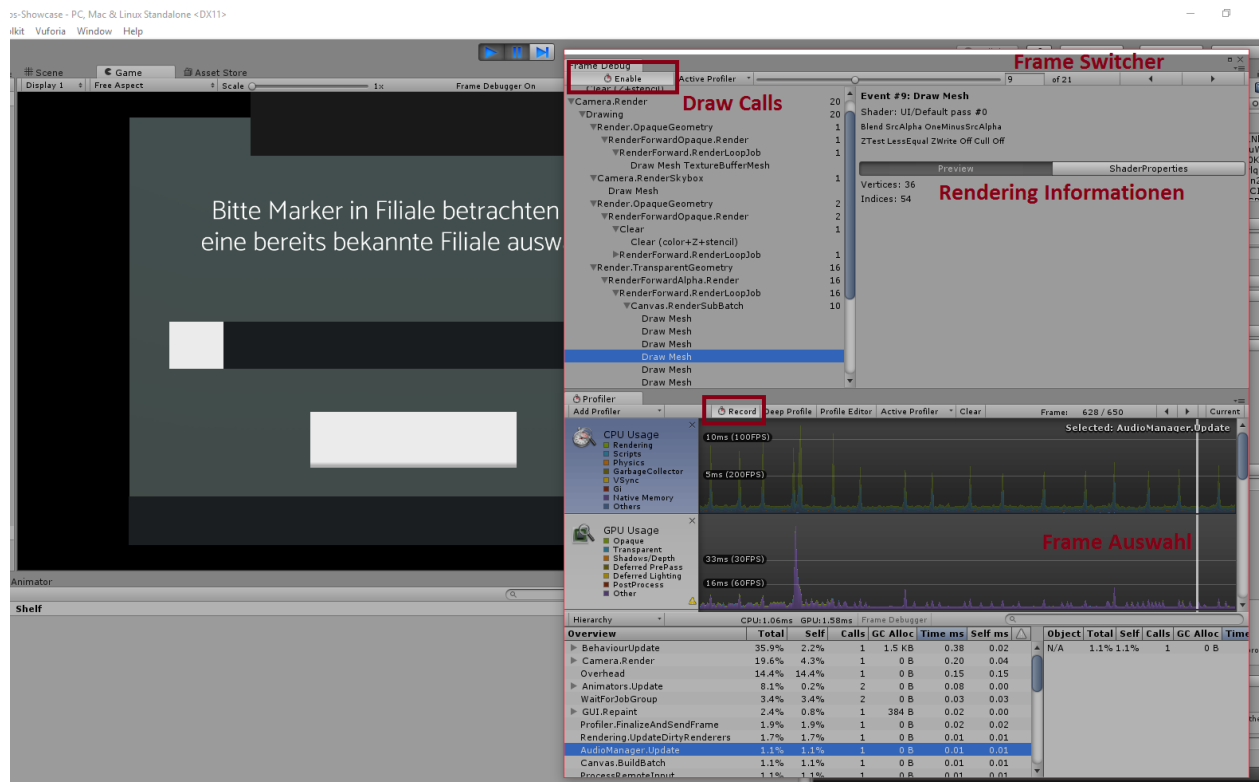


Abbildung 39: Unity Frame Debugger

Mittels Frame-Debugger kann man in Unity den Rendering Vorgang für jedes Frame genau analysieren. Der Frame-Debugger kann nur in Kombination mit dem Unity Profiler verwendet werden. Er wird wie folgt gestartet:

1. Unity Profiler starten
2. In Menüleiste von Unity **Window** > **Frame Debugger** auswählen
3. Debugger mit **Enable** aktivieren

Bemerkung

In der verwendeten Version von Unity scheint ein Problem im Zusammenhang mit HoloLens zu existieren. Jedes mal wenn der Frame Debugger mit einem Profiler verknüpft wurde, der gerade mit der HoloLens verbunden war, ist der Profiler abgestürzt. Eine Verwendung im "Play Modus" des Unity Editors ist aber möglich. Eine grobe Analyse des Rendering-Vorgangs ist theoretisch möglich.

Weitere Informationen zum Frame Debugger können der Unity Webseite entnommen werden. [\[Tec17b\]](#)

HoloLens Device Portal



Abbildung 40: HoloLens Device Portal

Das Device Portal ist eine Web-Anwendung, über welche man als Entwickler die HoloLens verwalten und steuern kann. Das Portal bietet die Möglichkeit, das Bild der HoloLens abzugreifen, Prozesse und Applikationen zu verwalten, Dateien anzupassen, die System Performance zu überwachen und vieles mehr. Der Zugriff auf das Portal findet entweder über ein USB Kabel⁵ oder, sofern die HoloLens mit einem drahtlosen Netzwerk verbunden ist, via Netzwerk⁶ statt. Eine ausführlichere Anleitung dazu findet man auf der Microsoft Webseite. [Mic171]

⁵Im Browser über IP Adresse 127.0.0.1

⁶Im Browser über IP Adresse der HoloLens im drahtlosen Netzwerk

Git mit Unity

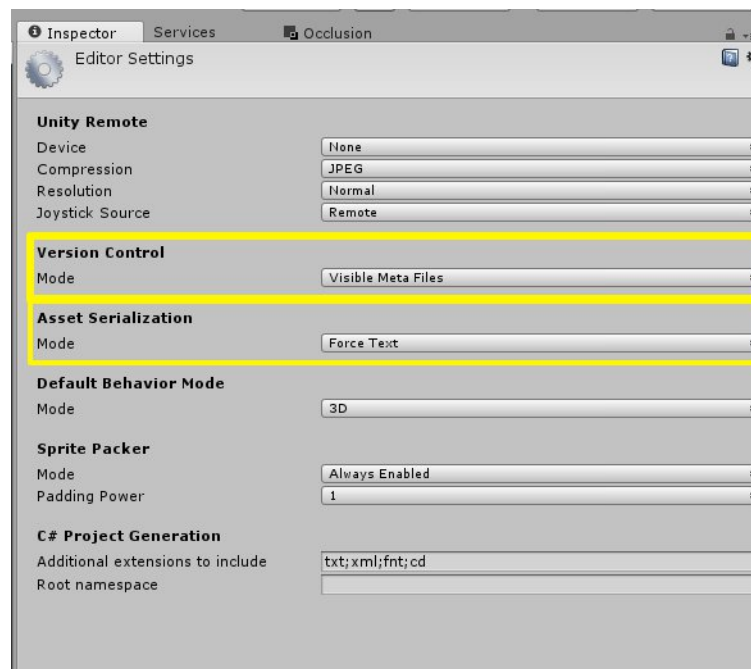


Abbildung 41: Einstellungen für Git in Unity

Damit Unity besser zusammen mit Git verwendet werden kann, müssen einige Einstellungen am Unity Projekt vorgenommen werden. Zuerst müssen die Meta-Informationen zu einzelnen Assets in Textform abgespeichert werden. Dazu öffnet man das entsprechende Unity Projekt und anschliessend **Edit** > **Project Settings** > **Editor** und wählt bei **Version Control** > **Mode** “Visible Meta Files” aus. So wird für jedes Verzeichnis und jedes Asset eine separate Datei mit Meta-Informationen angelegt. In diesen Dateien sind die wichtigsten, von Unity benötigten Informationen abgelegt. Zudem sollte man dafür sorgen, dass die Assets in Textform serialisiert werden. Dazu wählt man im gleichen Menü unter **Asset Serialization** > **Mode** die Einstellung “Force Text” aus (siehe Abbildung 41). Da in einem Projekt die Assets einen Grossteil des benötigten Speicherplatzes ausmachen, sollte man zusätzlich die Verwendung von [Git LFS \(Large File Storage\)](#) in Betracht ziehen.

Bemerkung

Für umfangreichere Projekte empfiehlt Unity die Verwendung von [Plastic SCM](#) oder [Perforce](#).

D Glossar

AR ist eine Abkürzung und steht für “Augmented Reality”. 5, 13, 14

ASCII Standardisiertes Format zur numerischen Abbildung von Buchstaben, Zahlen und einer limitierten Anzahl Sonderzeichen auf 1 Byte Grösse (entspricht 8 Bit bzw. 8 Stellen mit dem Wert 0 oder 1). 16

Asset Bestandteil einer grafischen Applikation, im Bezug auf Unity sind das z.B. Scripts, 3D Modelle, Texturen und Shader. 12, 31, 32, 52, 55, 58, 59, 69

Augmented Reality Als Augmented Reality z. Dt. erweiterte Realität bezeichnet man eine Erweiterung in der Wahrnehmung der echten Welt durch computergestützte Technologien. 5, 27, 57

Canvas Ein Canvas z. Dt. Leinwand bezeichnet ein Element von Unity auf welchem 2D Komponenten dargestellt werden können (z.B. zur Gestaltung von Benutzeroberflächen). 32

Collider Ein Collider ist eine Komponente aus Unity und dient dazu, Kollisionen mit anderen 3D Objekten von Unity festzustellen. Ein Collider ist unsichtbar und muss nicht die gleichen Konturen wie das 3D Objekt selbst haben. 51

Color Separation Optischer Effekt auf der HoloLens, bei dem sich einzelne Farbkanäle von ihrem Hologramm lösen. 29, 32

CPU ist eine Abkürzung für “Central Processing Unit” z. Dt. zentrale Recheneinheit und bezeichnet die zentrale Komponente in einem Computer, welche die Befehle eines Programmes ausführt. 50

Crash Dump Immer wenn unter Windows ein unzulässiger Vorgang ausgeführt wird, beendet das Betriebssystem die Applikation und erstellt ein Dokument, welches die wichtigsten Informationen zum Fehler beinhaltet. 53

Debugger Werkzeug zur Fehlersuche in einem Programm. 50, 67

Engine Als Engine bezeichnet man im Kontext der Entwicklung von grafischen Anwendungen (z.B. Spiele) ein Grundgerüst für Software, das einen Grossteil der Aufgaben zur Ansteuerung der Grafikkarte übernimmt. 40, 55

Feature Im Bezug auf die Bilderkennung sind Features z. Dt. Merkmale die Eigenschaft eines Bildes, anhand dessen es durch einen direkten Vergleich wiedererkannt werden kann (z.B. Kanten). 18

Field of View Teilbereich des menschlichen Sichtfelds, in dem auf der HoloLens Hologramme dargestellt werden können.. 28

FPS ist eine Abkürzung und steht für “Frames per Second” z. Dt. Bilder pro Sekunde. Der Begriff wird vor allem in der Videotechnik verwendet und ist ein Mass für die Bildfrequenz. 50, 51

Frame Als Frame bezeichnet man ein einzelnes Bild in einer schnell ablaufenden Sequenz von Bildern (Film, Animation). 28, 44, 50, 62, 67

Git Software zur Versionsverwaltung von Dateien. 58, 69

GPU ist eine Abkürzung für “Graphics Processing Unit” z. Dt. grafische Recheneinheit und bezeichnet die Komponente in einem Computer, welche für die Berechnung von grafischen Operationen zuständig ist. 50

Hologramm Als Hologramm bezeichnet man die dreidimensionale Abbildung eines Gegenstandes. 5, 8, 11, 22, 24–29, 32, 35–39, 41, 43, 45–47, 49–52, 54, 56

HoloToolkit Unity Erweiterung mit vorgefertigten Komponenten und Beispielen für die Entwicklung von HoloLens Applikationen. 12, 26, 32, 42, 49, 51

Immersion bezeichnet das sprichwörtliche “Eintauchen” in eine virtuelle Welt, so dass man für einen kurzen Moment vergisst, dass es sich dabei nicht um die Realität handelt. 5, 35

Koje Nach oben und nach der Vorderseite hin offener, durch provisorische Wände abgeteilter Bereich in einem grösseren Raum, der für einen bestimmten Zweck eingerichtet ist [Dud17]. 7, 8

Low-Poly Modell Ein 3D Modell besteht in der Regel aus einzelnen Polygonen, welche für die Darstellung in einzelne Dreiecke aufgeteilt werden. Je grösser die Zahl der Dreiecke desto mehr Rechenleistung wird für das 3D Modell benötigt. Ein Low-Poly Modell besitzt besonders wenig dieser Polygone und braucht somit weniger Rechenleistung um dargestellt zu werden. 51

Mesh In der Computergrafik bezeichnet ein Mesh ein Netz aus einzelnen zusammenhängenden Polygonen. Die Gesamtheit eines Meshs bilden, dann einen Körper. 22, 25, 26, 41, 49, 51

Mixed Reality Als Mixed Reality z. Dt. vermischte Realität bezeichnet man im Allgemeinen eine Kombination aus virtueller und echter Realität. 2, 3, 5, 54, 57

Prefab ist eine Abkürzung und steht für “Prefabricated” z. Dt. vorfabriziert und bezieht sich im Kontext von Unity auf Assets, die man als Vorlage abspeichert, damit man sie zu einem späteren Zeitpunkt einer Szene hinzufügen kann. 42

Profiler Werkzeug zur detaillierten Performance Analyse (Speicherverbrauch, CPU Auslastung) eines Programms. 50, 52, 66, 67

Rayon Einzelne Abteilungen innerhalb einer Filiale werden auch als Rayon bezeichnet. 9

SDK ist eine Abkürzung und steht für “Software Development Kit”. 42, 60

Shader Shader ist ein Begriff aus der 3D-Computergrafik und bezeichnet eine Software Komponente, welche auf eine bestimmte Anzahl von Pixeln jeweils einen Effekt anwendet (z.B. ein Lichteffect). 49, 51

Spatial Anchor Funktion der HoloLens zur dauerhaften Verankerung von Hologrammen im Spatial Mapping Mesh (also in der räumlichen Abbildung der Brille). 24–26

Spatial Mapping Funktion der HoloLens für die Erfassung der Umgebung (Wände, Gegenstände usw.) und zur Orientierung in Räumen. 22–24, 26, 31, 42, 49

Spatial Sound Funktion der HoloLens, die das Abspielen von Geräuschen aus einer beliebigen Richtung über die verbauten Lautsprecher erlaubt (Raumklang). 42

Stabilization Plane Fläche im dreidimensionalen Raum, auf der HoloLens alle durch diese Fläche geschnittenen Hologramme stabilisiert (ähnlich wie der Fokus bei einer Kamera). 28, 29, 51

Target Im Bezug auf die Bilderkennung ist ein Target z. Dt. Ziel eine Repräsentation eines Gegenstands, welche von einem Programm genutzt wird, um den Gegenstand in einem Bildsignal wiederzuerkennen (quasi eine Vorlage). Ein Target beschreibt eine Liste von Features. 15, 18, 20, 42

Unity Software zur Entwicklung von 3D Applikationen. 12, 13, 20, 23, 24, 27, 31–33, 40–45, 50–52, 55, 58–63, 66, 67, 69

Use Case Ein Use Case z. Dt. Anwendungsfall dient typischerweise zur Ausformulierung von Anforderungen an ein (Software) System. 7, 9, 22, 27

User Story Eine User Story z. Dt. Anwendererzählung beschreibt die Verwendung eines (Software) Systems aus Benutzersicht. 9

Verderb Unter dem Verderb versteht man eine Kennzahl die angibt, wie oft ein Produkt aufgrund Überschreiten des Haltbarkeitsdatums vom Händler entsorgt werden musste. 11, 39, 45, 65

Virtual Reality Als Virtual Reality z. Dt. virtuelle Realität bezeichnet man die Darstellung und Wahrnehmung einer durch den Computer generierten Realität. 5, 27

VR ist eine Abkürzung und steht für “Virtual Reality”. 5, 54, 55

Vuforia Funktionsbibliothek (Software) zur Bilderkennung, ausgelegt für die Entwicklung von AR Applikationen. 2, 12, 14–20, 37, 41–43, 45, 48, 49, 53, 61

E Verzeichnisse

E.1 Abbildungsverzeichnis

1	Pressebild HoloLens	5
2	Schema für die Komponenten der HoloLens Applikation	12
3	Eigens gestalteter Marker für das Experiment	15
4	Für die Erkennung aufbereitete Fotos der getesteten Produkte	17
5	Der Weg vom Produktfoto zum Image Target für Vuforia	18
6	Das Produkt wird anhand seiner Vorderseite erkannt	18
7	Erfassung eines Produkts aus verschiedenen Perspektiven	19
8	Das Produkt wird aus verschiedenen Perspektiven erkannt	20
9	Studienarbeitszimmer aus der Sicht von HoloLens	22
10	Ausrichtung Koordinatensysteme	23
11	Regale werden relativ zu einem gemeinsamen Ursprung ausgerichtet	23
12	Der gemeinsame Ursprung wird in der echten Welt verankert	24
13	Jedes Regal wird direkt in der echten Welt verankert	25
14	Distanzen für Hologramme (eigene Darstellung gemäss [Mic17i])	27
15	HoloLens Stabilization Plane für Hologramme	28
16	Auf allen Seiten des Regals können Markierungen dargestellt werden	29
17	Baumartige Struktur der Daten für die Regalübersicht	30
18	Seite des Regals als Koordinatensystem	30
19	Marker zur Ortung interessanter Regale innerhalb einer Filiale	31
20	Die Bedienung erfolgt über ein fensterähnliches Hologramm	32
21	Ein Bedienelement dient zur Steuerung und für Anweisungen	35
22	Marker über den Regalen geben einen guten Überblick der Filiale	36
23	Interessante Stellen im Regal werden durch Symbole hervorgehoben	37
24	Produkte werden beim Anblick automatisch erkannt	38
25	Zu Produkten kann der Benutzer genauere Informationen abrufen	38
26	Aufbau einer Szene in Unity	40

27	Hauptbestandteile, aus denen sich der Prototyp zusammensetzt	41
28	Übersicht der Code Architektur	43
29	Zustände und Übergänge des Prototyps	44
30	Datenmodell zur Abbildung einer Filiale mit Regalen und Produkten	46
31	Datenmodell für gesammelte Informationen zu einem Produkt	47
32	Datenmodell für produktbezogene Statistik und Vorschläge	47
33	Erfassung eines Regals mit dem Prototyp	48
34	Durch Betriebssystem belegter Arbeitsspeicher	51
35	Auszug des Unity Profiler bezüglich Speicherverbrauch	52
36	Hype-Zyklus nach Gartner Inc. [Gar16]	57
37	Die Testdaten des Prototyps können über das Device Portal modifiziert werden	64
38	Unity Profiler	66
39	Unity Frame Debugger	67
40	HoloLens Device Portal	68
41	Einstellungen für Git in Unity	69

E.2 Tabellenverzeichnis

1	Übersicht der getesteten Produkte für die Bilderkennung	17
2	Auflistung der vom Prototyp unterstützten Sprachbefehle	37
3	Beschreibung der Symbole zur Darstellung von Produktinformationen	39
4	Zuordnung von Zustandsübergängen zu Benutzeraktionen	44
5	Ordnerstruktur des Git Repository mit den Resultaten der Studienarbeit	58
6	Auflistung von Assets aus dem Unity Asset Store	59
7	Auflistung der XML Dateien mit Testdaten	65

E.3 Quellenverzeichnis

- [Cat17] Catchoom. *Easy, flexible and reliable Image Recognition and Augmented Reality tools*. 2017. URL: <https://catchoom.com/> (besucht am 31.05.2017).
- [Dud17] Duden. *Koje, die*. 2017. URL: <http://www.duden.de/rechtschreibung/Koje> (besucht am 31.05.2017).

- [For17] Frost Forged. *Quest Markers 3D*. 2017. URL: <https://www.assetstore.unity3d.com/en/#!/content/83389> (besucht am 31.05.2017).
- [Gar16] Gartner. *Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage*. 2016. URL: <https://www.gartner.com/newsroom/id/3412017> (besucht am 31.05.2017).
- [Mic17a] Microsoft. *Color Design*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/color_design (besucht am 31.05.2017).
- [Mic17b] Microsoft. *Coordinate systems*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/coordinate_systems (besucht am 31.05.2017).
- [Mic17c] Microsoft. *Focus point in Unity*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/focus_point_in_unity (besucht am 31.05.2017).
- [Mic17d] Microsoft. *Hologram stability*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/Hologram_stability.html (besucht am 31.05.2017).
- [Mic17e] Microsoft. *Holograms 100*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/holograms_100 (besucht am 31.05.2017).
- [Mic17f] Microsoft. *HoloToolkit-Unity*. 2017. URL: <https://github.com/Microsoft/HoloToolkit-Unity> (besucht am 31.05.2017).
- [Mic17g] Microsoft. *Install the tools*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools (besucht am 31.05.2017).
- [Mic17h] Microsoft. *Mixed Reality Academy*. 2017. URL: <https://developer.microsoft.com/en-us/windows/mixed-reality/academy> (besucht am 31.05.2017).
- [Mic17i] Microsoft. *Optimal distance for placing holograms from the user*. 2017. URL: <https://az835927.vo.msecnd.net/sites/mixed-reality/Resources/images/DistanceGuideRendering.png> (besucht am 31.05.2017).
- [Mic17j] Microsoft. *Performance recommendations for HoloLens apps*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/performance_recommendations_for_hololens_apps (besucht am 31.05.2017).
- [Mic17k] Microsoft. *Spatial mapping*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_mapping (besucht am 31.05.2017).
- [Mic17l] Microsoft. *Using the Windows Device Portal*. 2017. URL: https://developer.microsoft.com/en-us/windows/mixed-reality/using_the_windows_device_portal (besucht am 31.05.2017).
- [Sha17] Dave W. Shanahan. *Windows 10 Creators Update: What's new with Mixed Reality*. 2017. URL: <https://www.onmsft.com/news/windows-10-creators-update-whats-new-with-mixed-reality> (besucht am 31.05.2017).
- [Stu17] Lovatto Studio. *UMenu: World Space*. 2017. URL: <https://www.assetstore.unity3d.com/en/#!/content/59588> (besucht am 31.05.2017).
- [Tec17a] Unity Technologies. *Execution Order of Event Functions*. 2017. URL: <https://docs.unity3d.com/Manual/ExecutionOrder.html> (besucht am 31.05.2017).

- [Tec17b] Unity Technologies. *Frame Debugger*. 2017. URL: <https://docs.unity3d.com/Manual/FrameDebugger.html> (besucht am 31.05.2017).
- [Tec17c] Unity Technologies. *Profiler overview*. 2017. URL: <https://docs.unity3d.com/Manual/Profiler.html> (besucht am 31.05.2017).
- [Tec17d] Unity Technologies. *Transform*. 2017. URL: <https://docs.unity3d.com/ScriptReference/Transform.html> (besucht am 31.05.2017).
- [Vuf17a] Vuforia. *Developing Vuforia Apps for HoloLens*. 2017. URL: <https://library.vuforia.com/articles/Training/Developing-Vuforia-Apps-for-HoloLens> (besucht am 31.05.2017).
- [Vuf17b] Vuforia. *Recommendations for Optimizing Target Detection and Tracking Stability*. 2017. URL: https://library.vuforia.com/articles/Best_Practices/Recommendations-for-Improving-Target-Detection-and-Tracking-Stability (besucht am 31.05.2017).
- [Vuf17c] Vuforia. *SDK*. 2017. URL: <https://developer.vuforia.com/downloads/sdk> (besucht am 31.05.2017).
- [Vuf17d] Vuforia. *Tools*. 2017. URL: <https://developer.vuforia.com/downloads/tool> (besucht am 31.05.2017).
- [Vuf17e] Vuforia. *VuMark Design Guide*. 2017. URL: <https://library.vuforia.com/articles/Training/VuMark-Design-Guide> (besucht am 31.05.2017).