Bachelor-Projekt “Mixed Reality”

Internationaler Studiengang Medieninformatik,
Hochschule Bremen

Fachsemester: 3. Semester
Zeitraum: 03.10.2019 - 31.01.2020

Betreuer: Prof. Dr. Volker Paelke,
Jendrik Bulk, Andreas Lochwitz

Teilnehmer*innen:
Moumita Ahmad, Annie Berend, Moritz Hodler, Robin Jacobse, Reda Khalife, Bjarne Lehmkuhl, Philipp Moritzer, Joscha Sattler, Tobi Schmitt, Alisa Schumann, Leonard Tuturea

1. Abstract **3**

2. Grundlagen **5**
Projection-based-AR 5
Computer Vision 6
Gestenbasierte Interaktion 6
Tangible User Interfaces 8
Prototyping Techniken 9
Sonification 11

3. Technologien **13**
Unity 13
Vorstellung und Nutzen für das Projekt 13
Best Practices 13
Intel RealSense SDK 18
3D-Druck 20
Ultimaker Cura 20
Pure Data 21
Open Sound Control Protocol 21

4. Hardware **23**
Tischaufbau 23
Beamer-Halterung 24
Intel Realsense Depth Camera 24
Halterung 25
Hololens 26
Mixed Reality Toolkit 27
Griffhalterung der Marker 27

Soundsystem **28**

5. Projekte **29**
Pong MR 29
Space Invaders 32
Holo-Cubes 35

6. Ausblick **39**

7. Anhang / Fotos **39**

8. Zusammenfassung **41**

Quellenverzeichnis **43**
Gestenbasierte Interaktion 43

Tangible User Interfaces 43
Intel Realsense 43
Realsense-Halterung 43
3D-Druck 43
Unity 43
RealSense SDK 44
Projection-based-AR: (Stand: 18.02.2020 14:42) 44
Prototyping: (Stand: 16.02.2020 20:59) 44
Mixed Reality Toolkit 45

Verantwortlichkeiten 45

1. Abstract

Durch den technischen Fortschritt werden Kamera- und Display-Technologien stetig besser und für Verbraucher sowie Industrie zugänglicher. Dadurch ergeben sich neue Möglichkeiten die physische Umwelt mit einer virtuellen Welt zu erweitern. Im Bachelorprojekt Mixed Reality werden einige dieser Möglichkeiten erforscht und implementiert.
Hierfür werden verschiedene Technologien, z. B. Tiefenbildkameras und Augmented Reality Brillen, genutzt und sich mit Aufgaben der Augmented Reality, wie z.B. Image Tracking und Tangibles, beschäftigt.

Für die Implementierung der entwickelten Systeme wurde ein modular erweiterbarer Tisch konstruiert. Auf dessen Oberfläche können mithilfe eines an dem Tisch angebrachten Kurzdistanzprojektors Anwendungen angezeigt werden. Somit können viele Menschen gleichzeitig um den Tisch stehen und mit den Anwendungen interagieren.

2. Grundlagen

Projection-based-AR

Augmented Reality ist schnell populär und beliebt geworden. Kurz ein Bild mit “Hunde-Filter” gemacht oder eine Runde Pokemon-Go gespielt und schon wurde gebrauch von Augmented Reality gemacht. Das Prinzip dabei ist immer recht ähnlich. Eine Kamera erfasst einen Ausschnitt aus der realen Welt und ergänzt es mit einem oder mehreren virtuellen Objekten.

Genutzt wird AR über Smartphones, Tablets, AR-Brillen oder Desktop-AR mittels einer Webcam. Diese Geräte haben jedoch einige Nachteile vorzuweisen. Bei Smartphones & Tablets besteht das Problem, dass das Gerät stets festgehalten und vor das zu erweiternde Objekt aus der realen Welt gehalten werden muss. Dies schränkt den Nutzer in der Bewegungsfreiheit der Hände enorm ein. AR-Brillen hingegen können nur einen Nutzer zur Zeit haben. Sollten mehrere Nutzer Zugriff auf AR mittels einer Brille bekommen, müssen auch mehr AR-Brillen gekauft werden, was schnell sehr teuer werden kann. Zudem kann das Tragen einer solchen Brille als unangenehm empfunden werden. Desktop-AR ist dagegen in der Mobilität und Flexibilität sehr eingeschränkt. Der Bereich der AR ist auf die Größe und den Standort des Bildschirms mit seinen Hardwarekomponenten beschränkt.

Als Alternative ist Projection-based-AR immer mehr im Trend. Dabei projiziert ein Projektor virtuelle Informationen auf ein physisches Objekt und lässt es so “zum Leben erwecken” oder im Erscheinungsbild verändern. Damit können einige Nachteile der vorherigen AR-Systeme gut umgangen werden. Jeder Zuschauer kann die Projektionen ohne Hilfsmittel betrachten und ist in seiner Bewegung uneingeschränkt. Zudem sind die Projektionen zwar auch statisch, dafür aber punktgenau, sofern die Konfigurationen korrekt vorgenommen wurden.

Doch Projection-based-AR bietet auch Nachteile. Darunter zählt, dass es nur eine optimale Perspektive gibt, sofern lediglich ein Projektor genutzt wird. Zudem werfen Objekte zwischen dem Projektor und der Projektionsfläche schatten, sodass die Projektion verdeckt werden kann.

Projection-based-AR hat großes Potential und findet in nahezu allen Gesellschaftlichen Bereichen ein Anwendungsfeld. Darunter fällt Unterhaltung, Design, Medizin, Bildung, Produktion/Reparatur und Marketing. Am meisten verwendet wird es bisher in der Unterhaltung und im Marketing.

Computer Vision

Moderne Kamerasysteme können Bilder in einer Auflösung und Detail erfassen, die die des menschlichen Auges übertreffen. Computer können zudem Unterschiede zwischen Farben und Formen extrem genau erkennen. Den Inhalt von Bildern allerdings zu erkennen und einzuordnen ist ein Problem, das Entwickler schon seit Jahrzehnten nicht lösen konnten.

Computer Vision ist ein Feld der Informatik das sich darauf konzentriert, Teile des komplexen menschlichen Sehens zu replizieren und es Computern zu ermöglichen, Objekte auf Bildern und Videos auf gleiche Weise zu identifizieren und verarbeiten wie Menschen. Bis vor wenigen Jahren waren die Möglichkeiten von Computer Vision noch stark limitiert. Aufgrund des erheblichen Fortschrittes in verschiedenen Disziplinen der Informatik, wie künstlicher Intelligenz, Deep Learning und neuronalen Netzwerken, haben sich auch die Kapazitäten der Computer Vision stark vergrößert und den Menschen in manchen Bereichen der Objekterkennung, wie zum Beispiel Gesichtserkennung, bereits übertroffen.

Deep Learning spielt eine wichtige Rolle in der Computer Vision. Deep Learning basiert auf neuronalen Netzwerken. Neuronale Netzwerke können, nach Eingabe von großen Mengen beschrifteter (beschriebener) Daten, wiederkehrende Muster in diesen erkennen und diese Muster in mathematische Formeln transformieren, welche wiederum dabei helfen, zukünftige Informationen zu klassifizieren. In der Computer Vision bedeutet dies, dass neuronale Netzwerke mit sehr großen Mengen an beschrifteten Bildmaterial trainiert werden und die daraus resultierenden “Erkenntnisse” nutzt um unbekanntes Bildmaterial zu verarbeiten und darin bereits bekannte Objekte zu erkennen.

Die Anwendungsfelder der Computer Vision erweitern sich in rasantem Tempo. Neben schon lange zuverlässigen Verfahren, wie zum Beispiel Fingerabdrucksensoren, finden auch moderne Techniken wie beispielsweise die Gesichtserkennung immer mehr Einsatz. Die beiden größten Entwicklungsgebiete der Computer Vision sind zurzeit die Erkennung von Hindernissen bei selbstfahrenden Autos (Genauigkeit und Geschwindigkeit wichtig) sowie das Erkennen von Personen und Orten, oft mit dubiosem Hintergrund wie finanzielle Interessen bei Social Media Giganten oder Kontrolle im Falle autoritärer Regierungen.

Gestenbasierte Interaktion

Eine Geste ist eine Bewegung oder Statische Körperhaltung die eine Information enthält. Steckt die Information in der Bewegung so spricht man auch von kontinuierlichen; steckt sie in einer Haltung von diskreten Gesten.

Die Gestensteuerung ist eine komplexere Form der Mensch-Computer Interaktion im Gegensatz zu traditionellen Eingabemethoden. Dies liegt vor allem in der höheren Anzahl an Freiheitsgraden sowohl im dreidimensionalen Raum als auch in Abhängigkeit von der Zeit

begründet. Dementsprechend werden auch höhere Anforderungen an die Eingabegeräte so wie Algorithmen für die Gestenerkennung gestellt. Dabei erweitert sie jedoch das Vokabular mit dem kommuniziert werden kann und zielt darauf ab die Interaktion intuitiver zu gestalten und die Interaktionsmöglichkeiten zu erweitern.

Bei den Eingabegeräten in der gestenbasierten Interaktion wird unterschieden zwischen Geräten die am Körper geführt werden und externer Sensorik. Die am Körper geführten Geräte besitzen dabei zumeist Beschleunigungs-, Positions- sowie Drucksensoren für die Erkennung der Gesten. Aber auch die Krümmung der Finger kann z.B. durch Glasfaserkabel in speziellen Handschuhen bestimmt werden indem das austretende Licht an den Gelenken bestimmt wird. Bei der externen Sensorik hingegen kommen hauptsächlich visuelle Sensoren wie Kameras zum Einsatz. Bei diesen wird die Bildanalyse herangezogen um Information aus den Sensordaten zu gewinnen. Diese Informationen können mit einer Datenbank abgeglichen werden um eine bestimmte Geste zu erkennen oder ein Skelettmodell berechnet werden, welches die Geste nachstellt.

Gesten sind ein großer Bestandteil unserer Kommunikation mit anderen Menschen im Alltag. Hierbei lassen sich Gesten nach verschiedensten Kriterien kategorisieren. Gesten die eine einzige bestimmte Entsprechung haben, Gesten die zu etwas auffordern oder hinweisen so wie Gesten, die Eigenschaften eines Objektes beschreiben sind nur einige Beispiele hierbei.

Neben den unterschiedlichen Arten von Gesten gibt es zusätzliche Aspekte die bei der Implementierung einer gestenbasierten Interaktion zu beachten sind. Denn die Gesten sollten für den Anwender natürlich und intuitiv sein. Dies bedeutet, dass die Gesten vom Benutzer einfach auszuführen sein sollten und möglichst wenig Aufwand benötigen um ihn auch längerem Benutzen des Systems aussetzen zu können. Hilfreich ist es auch Gesten aus dem Alltag zu implementieren die nicht erst von dem Nutzer erlernt werden müssen.

Durch den Einsatz von verbaler Sprache lässt sich die gestenbasierte Interaktion zusätzlich erweitern und der intuitiven alltäglichen Kommunikation annähern. Sprache kann hierbei eine optimale Ergänzung sein, die gleichzeitig die Genauigkeit der Gesten verbessert, da manche Gesten auch kontextabhängig unterschiedliche Bedeutungen haben können.

Tangible User Interfaces

User Interfaces beschränken sich ursprünglich auf den klassischen Computer: Desktop, Mouse und Tastatur bieten Interaktionsmöglichkeiten mit dem jeweiligen System und geben über den Bildschirm Feedback. Dabei werden die gewohnte Geschicklichkeit und Fähigkeiten des Menschen, Dinge zu verschieben, zu verformen und anderweitig zu manipulieren, völlig außer Acht gelassen (Ishii 2008). An Werkzeuge und ähnliches gewohnt, erfordert die erfolgreiche Nutzung von digitalen User Interfaces einen Prozess, in dem der Umgang mit der Technologie erst erlernt werden muss. Dieser erfolgt je nach der Erfahrung des Nutzenden mehr oder weniger schnell.
Tangible (*deutsch*: greifbar) User Interfaces setzen bei diesem nicht genutzten Potential an und versuchen neue Interaktionsmöglichkeiten im Digitalen zu schaffen. Bei Tangible User Interfaces wird eine Anwendung über Gegenstände gesteuert, die die nutzende Person anfassen, verschieben, hinzufügen oder wegnehmen kann. Somit gestaltet sich ein TUI weitaus intuitiver als die Standard-GUIs. Die interaktiven Gegenstände sind mit dem System verbunden, sodass sie als Eingabegeräte funktionieren. Ein Feedback des Systems kann dann visuell via Projektion o.ä. erfolgen, sodass sich die beiden Kerneigenschaften von physischen Objekten in TUI ergeben: Kontrolle und Repräsentation.

Die ersten Tangible User Interfaces wurden Mitte der 90er Jahre konzipiert und seitdem stetig weiterentwickelt (Ishii 2008). Dabei wurde festgestellt, dass die Technologie vor allem in den Bereichen Stadtplanung, industrielle Abläufe, Musik oder auch in der Unterhaltungsindustrie und der Bildung viele Möglichkeiten erschließt.

Vorteile der Nutzung sind dabei vor allem, dass digitalen Informationen eine physische Form gegeben wird, die für den Menschen viel verständlicher ist und einen intuitiven Umgang ohne Lernprozess ermöglichen. Besonders im Bildungsbereich spielt die Interaktivität eine zentrale Rolle, da durch sie Wissen besser verfestigt wird. Ein weiterer Vorteil ist, dass mehrere Nutzer*innen zur gleichen Zeit mit einem System interagieren können (Creed, Sivell, and Sear, 2013; East, 2015) und somit neue Arbeitsweisen denkbar werden. Dabei eignen sich die Anwendungen allerdings meistens für kleinere Zielgruppen und erfordern für eine erfolgreiche Konzeption eine enge Absprache mit den potentiell nutzenden Personen. Weiterhin muss beachtet werden, dass die Anwendung bei zu vielen Elementen und Interaktionsmöglichkeiten Überforderung hervorrufen kann. So müssen bei der Entwicklung von Tangible User Interfaces andere Aspekte berücksichtigt werden als bei Graphical User Interfaces.

Die Interaktivität bei TUIs können durch zwei verschiedene Arten von Input realisiert werden: *camera-sensitive* und *capacitive-sensitive* (Spadaccini, McDonald 2017). Letztere reagieren auf elektrische Impulse und basieren somit auf speziellen Sensoren, die diese verarbeiten können. Bei *camera-sensitive* TUIs tracken Kameras Gegenstände oder Gesten und verarbeiten diese als Input (zum Beispiel mit der SDK Vuforia). Für das Tracken von Gegenständen werden häufig sogenannte fiducial markers eingesetzt, die vor allem bereits in den Anfängen von Bilderkennung getrackt werden konnten und über ihre einzigartige Gestaltung Auskunft über Position und Ausrichtung der Gegenstände geben können (Bencina, Kaltenbrunner 2005). Durch die Begrenzung der Formen auf die Farben schwarz und weiß gestaltet sich das Tracken durch die Kamera deutlich schneller, als bei farbigen und dreidimensionalen Formen oder Figuren.

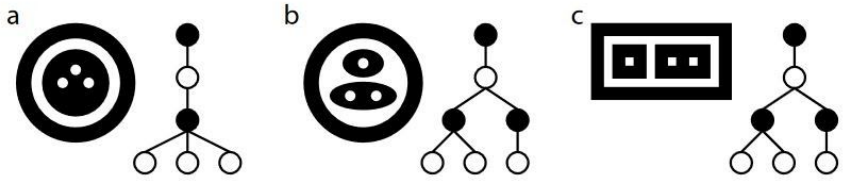
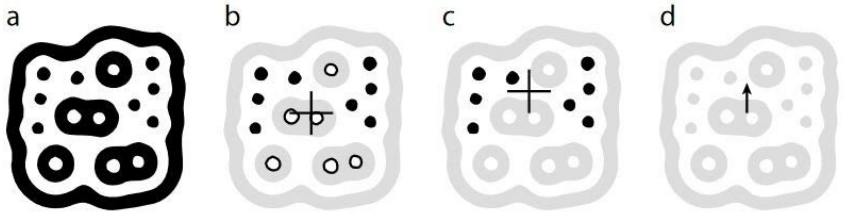


Figure 2: Some simple topologies and their corresponding region adjacency graphs.



Prototyping Techniken

Auch im Bereich AR ist das Prototyping ein wichtiger Bestandteil bei der Erstellung eines Produkts.

Der Begriff wird verstanden, dass eine Visualisierung der Idee erschaffen wird, die meist eine Rohversion des angestrebten Resultats darstellt. Dieses Verfahren ist sehr wichtig, da beispielsweise bei der Arbeit mit einem Auftraggeber das Verständnis von den Anforderungen und dem fertigen System von beiden Seiten ein unterschiedliches sein kann. Hält man mit dem Kunden Rücksprache und präsentiert bereits während des Arbeitsprozesses kleinere Ergebnisse, so ist es unwahrscheinlicher im Nachhinein eine böse Überraschung zu erleiden. In dem Fall wäre schon viel Zeit und Geld in ein Produkt investiert worden, den der Kunde nicht gebrauchen kann. Das könnte die Kooperation mit dem Kunden, sowie die Motivation des Teams negativ beeinflussen.

Prototypen können in unterschiedlichen Entwicklungsstadien eines Prozesses erstellt werden, daher werden sie in Kategorien eingeteilt. Eine davon sind die Demonstratoren, das

sind Prototypen, die noch weit von der eigentlichen Realisierung entfernt sind, sie werden meist am Anfang eines Arbeitsprozesses erstellt.

Diese dienen dazu dem Kunden eine grobe Vorstellung zu vermitteln wie das Produkt aussehen kann. Prototypen in dieser Phase werden meist schnell angefertigt und nach der Präsentation nicht mehr weiter verwendet, daher kommen die Begriffe Rapid-Prototyping und Throwaway-Prototyping.

Eine weitere Kategorie sind die sogenannten Labormuster, bei denen darauf eingegangen wird ob das Produkt aus technischer Sicht prinzipiell Umsetzbar ist. Diese Modelle sind nicht für den Kunden bestimmt, funktionierende Teile werden jedoch im fertigen Produkt aufgegriffen. Zum Beispiel kann hierbei die Tragfähigkeit einer Architektur getestet werden.

Bei der letzten Kategorie handelt es sich um die Pilotensysteme, die ziemlich am Ende eines Entwicklungsprozesses entstehen. Es sind Produkte, die sehr nah am finalen System sind und alle benötigten Funktionen beinhalten. Pilotensysteme können obendrein in der Einsatzumgebung getestet werden. Der Auftraggeber sowie Endnutzer kann hierbei einbezogen werden.

Zusätzlich gibt es bei dem Vorgang von Prototyping verschiedene Modelle. Die Begriffe vertikale sowie horizontale Prototypen liegt die Unterscheidung in der Softwarearchitektur. Bei horizontale Prototypen wird eine Schicht, meist die Anwendungsschicht, unter die Lupe genommen. Dabei soll das äußere Erscheinungsbild des Produkts getestet werden. Funktionalitäten oder Bedienbarkeit werden dabei außer Acht gelassen. Anders ist es bei vertikalen Prototypen, bei diesen wird eine bestimmte Funktion getestet, bei der es unerlässlich ist, dass alle Schichten der Software miteinander interagieren.

Ein weiteres Modell ist das Explorative Prototyping, welches dazu verwendet wird die Anforderungen an das fertige Produkt zu sammeln und so weit es geht zu präzisieren. Dieses wird in dem Fall benutzt, wenn der Kunde noch nicht weiß oder nur teilweise weiß, welche Funktionen das System benötigt.

Des Weiteren gibt es einige Prototyping-Methoden, eine davon ist das Storyboard. Dabei werden einfache Skizzen, die einem Comic ähneln, angefertigt um die Interaktion zu visualisieren. Diese Technik wird in der frühen Phase der Entwicklung hergestellt, damit die Entwickler sich ein Bild davon machen können in was für einer Situation und mit welcher Erwartung ein Nutzer das System in Gebrauch nimmt. Das kann sehr hilfreich sein, jedoch ist dafür eine hohe Vertrautheit mit dem Endverbraucher vorausgesetzt. Ein Vorteil von dieser Technik ist, dass es schnell anzufertigen ist und keinen hohen Aufwand benötigt, somit auch keine hohen Kosten. Ein Nachteil ist jedoch, dass bei dieser Technik keine Details erforscht werden können. Ein Verfahren welcher mit dem Storyboard kombiniert werden kann ist das Roleplay. Dabei versetzt man sich in die Lage des Kunden und versucht seine Bedürfnisse nachzuvollziehen. Man spielt eine Situation in der der Nutzer sich befinden kann nach, wobei es hilfreich sein kann sich in eine Umgebung zu begeben oder andere äußere Umstände anzupassen, in denen sich der Nutzer befindet. Diese Technik kann hilfreich sein um das emotionale Erfahrung des Nutzers zu erfassen.

Eine weiteres Verfahren ist der Papierprototyp, bei dem man Skizzen eines Interfaces auf Papier aufgemalt und dieses einer Testperson vorliegt. Wird von dieser ein gemalter Button per Berührung „bedient“ wechselt man die Skizze durch eine andere, bei der man sieht wie das System darauf reagieren würde. Ein ähnliches Verfahren ist das Wizzard of Oz Prototyping, dabei wird ebenfalls eine Interaktion gefälscht. Der Nutzer soll wird zum testen ein fertig wirkendes System bedienen. Zunächst denkt er, dass die Reaktionen auf sein Handeln, Systemhandlungen sind. In Wirklichkeit werden diese von einem Menschen ausgeführt und somit gefälscht. Ein Berühmtes Beispiel dazu ist das IBM Speech-to Text Experiment, bei dem Tester dachten, dass ihre Sprache von einem Computer zu Text generiert werden, in Wirklichkeit saß eine Person im Nebenraum, die das Gesagte abgetippt hat. Dieses Verfahren hat bei der Entwicklung später dem Unternehmen Kosten einsparen können.

Alle aufgezählten Methoden können zum Entwickeln von AR-Anwendungen genutzt werden, jedoch gibt es Methoden, die die Einbindung der Umgebung beinhalten. Zum Beispiel das Augmented Sketching, dabei werden Fotografien angefertigt, die die Umgebung eines Anwendungsbereichs zeigen. Auf diesen Fotografien werden mit digitaler Zeichensoftware Skizzen von dem AR-Interface angefertigt. Eine weitere Technik ist das Skizzieren auf transparenten Unterlagen, wie zum Beispiel auf Folien oder Plexiglas. Beim Testen wird eine Person diese Skizze je nach Blickrichtung der Testperson halten, so kann eine Skizze mit der Umgebung in Bewegung getestet werden.

Sonification

Sonification bezeichnet das Vorgehen aus Daten und Informationen Sounds zu gestalten. Diese Sounds sollen wiederum die Daten wiederspiegeln, so dass der Hörer Rückschlüsse auf die Informationen der Daten schließen kann. Einfache Anwendungen aus dem Alltag sind beispielsweise das Piepen eines EKG's im Krankenhaus, welches die Herzfrequenz eines Patienten hörbar macht. Oder das Knacken eines Geiger-Müller-Zählrohrs, das die Strahlungsintensität an einem Ort widerspiegelt.

Sonification kann in der Wissenschaft als alternative Erkenntnismethode über Informationen dienen. Es kann als Gegenstück der graphischen Darstellung angesehen werden. Dort wo man visuell mit Diagrammen und Graphen Rückschlüsse auf Daten schließt, kann man auch Klänge verwenden, um den Informationsgehalt der Daten hörbar zu machen. Dies kann zum Beispiel auch hilfreich für Menschen sein, die unter Beeinträchtigung oder dem Fehlen der Sehkraft leiden. Sonification kann diesen Menschen als Zugang zu den Informationen dienen.

Des Weiteren bietet Sonification auch die Möglichkeit Visuelles zu verstärken oder zu ergänzen. Man hat somit die Möglichkeit Objekten weitere Eigenschaften durch Sounds zuzuschreiben oder Soundfeedbacks auf Ereignisse zu geben. Durch die zusätzliche Nutzung des Hörsinns kann ein*e Nutzer*in mehr Informationen gleichzeitig wahrnehmen

und hat beispielsweise bestimmte Informationen als Sound im Ohr und kann sich dadurch besser auf weitere Geschehnisse konzentrieren.

Wie genau die Sonification in der Praxis umgesetzt wird, ist an keine Grenzen gebunden. Durch die Nähe von Sound und Musik passiert es jedoch gerade in der medialen Öffentlichkeit recht oft, dass durch starke Ästhetisierung der Darstellung die Sonification "musikalisiert" wird. Hierbei wird die Ästhetik der Darstellung in den Vordergrund und die Objektivität in den Hintergrund gerückt. Gerade im Bereich von Musik kann die Wahl der Umsetzung, wie zum Beispiel der ausgewählte Klang, der Synthesizer, das Tempo oder die Tonhöhe, den Hörer stark subjektiv beeinflussen. In der Wissenschaft wird daher auf die Unterscheidung von Sound und Musik geachtet. An sich gilt in der Wissenschaft, dass Sonification algorithmisch reproduzierbar sein sollte und die Erzeugung transparent gemacht wird.

Grundsätzlich wurde festgestellt, dass sich das menschliche Gehör sehr gut eignet um schnell und präzise Informationen aus Sound zu analysieren. Das Gehör kann viele Parameter gleichzeitig aufnehmen und diese differenzieren. Zudem kann der Mensch sehr gut seine visuelle Wahrnehmung mit Geräuschen kombinieren und somit das Bild seiner Umwelt durch Sound ergänzen.

In unserem Projekt bietet es sich an, Sound zu dem auf den Tisch projizierten Bild hinzuzufügen, um damit die Wahrnehmung des Nutzers zu erweitern. Die in dem Projekt genutzten Technologien, wie beispielsweise die RealSense, Vuforia und der Einsatz von Tangibles, produzieren viele Daten die für Sonification genutzt werden können.

3. Technologien

Unity

Vorstellung und Nutzen für das Projekt

Unity ist eine Laufzeit- und Entwicklungsumgebung, die primär als Spiel-Engine verwendet wird. Den Spielersteller versorgt sie mit den notwendigen Funktionen, um schnell und effizient Spiele zu entwickeln.

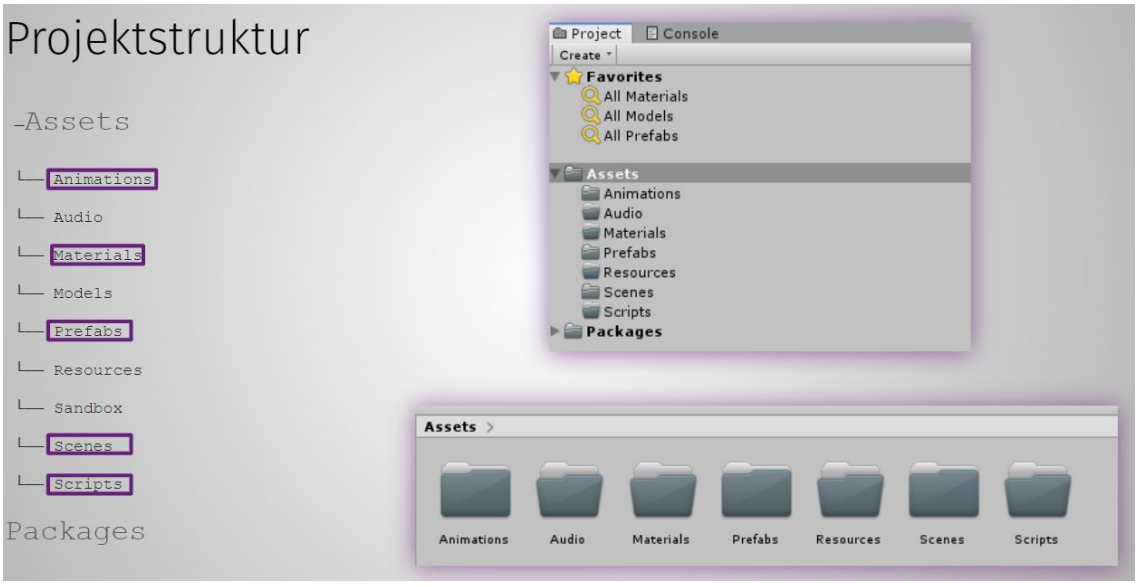
Die Game-Engine fungiert als Gerüst für die Entwicklung von Spielen oder 3D-Applikationen und bietet Werkzeuge für den Import von Grafiken und der Zusammenstellung von Szenen, die Beleuchtung, Audio, Effekte, Physik, Animationen, Interaktivität und Gameplay-Logik zusammenfügen und für die jeweilige Zielplattform bereitstellen.

Unity in unserem Projekt ermöglicht es, die Virtualisierung die wir mit dem Tisch umsetzen wollen, schnell und effizient über vorgefertigte Funktionen zu implementieren. Unity bietet ebenfalls den Support für Mixed-Reality-Toolkits oder ein SDK für die genutzte RealSense. So können wir diese Geräte für unsere Applikation ansteuern und die Applikation so verändern, dass sie unterstützend mit diesen Geräten funktionsfähig ist.

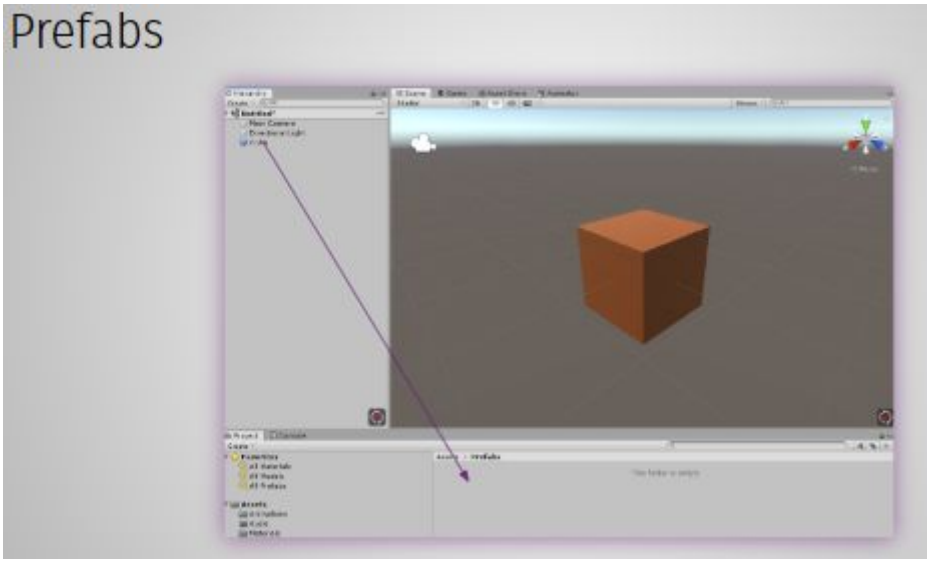
Best Practices

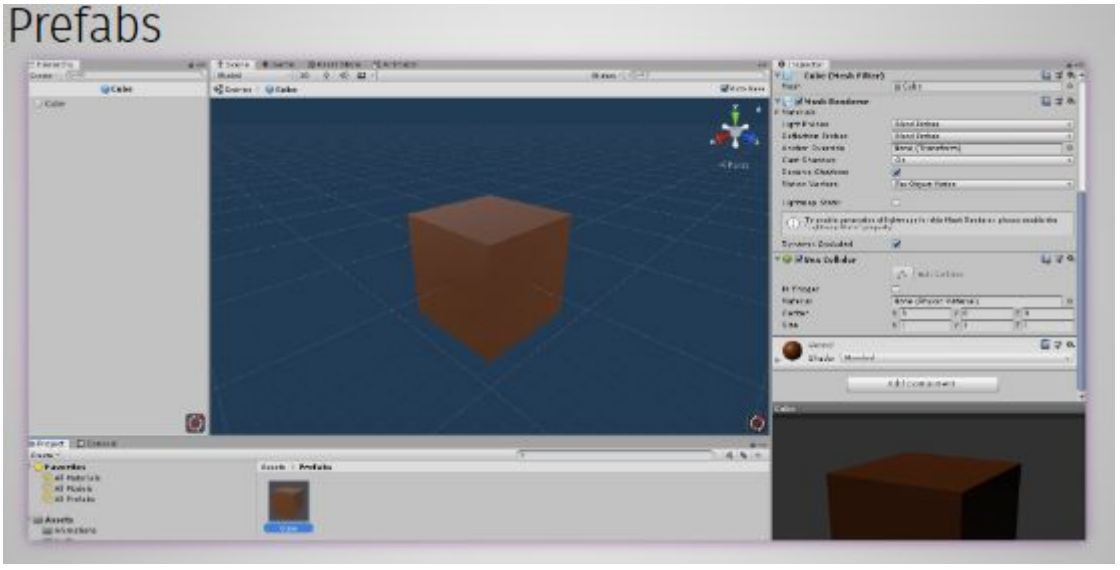
Projektstruktur

Die Ordnerstruktur für ein neu angelegtes Unity Projekt sollte eine organisierte Trennung von Entitäten des selben Typens vorsehen. Der Vorteil hierbei ist, dass weniger vergessen wird, die Produktivität steigt und ungewollte Nebeneffekte vermieden werden. Im Folgenden Sieht man eine Standardmäßige Ordnerstruktur in Unity. Es werden Ordner für Animations, Audio, Materials, Prefabs, Scenes und Scripts angelegt.



Wichtige Grundregeln bei der Erstellung dieser Ordnerstruktur sind es keine weiteren Ordner zu erstellen, keine Assets oder Files im “Root”-Directory zu platzieren, eine konsistente Namensgebung zu verwenden und kontextspezifische Assets nicht in generelle Ordner abzulegen. Experimentelle Ansätze gehören in den dafür vorgesehenen Sandbox-Ordner. Desweiteren sollte eine Versionsverwaltung, wie z.B. git für das Projekt verwendet werden. Außerdem empfiehlt es sich für jegliche Erzeugung von Game-Objects Prefabs zu verwenden:

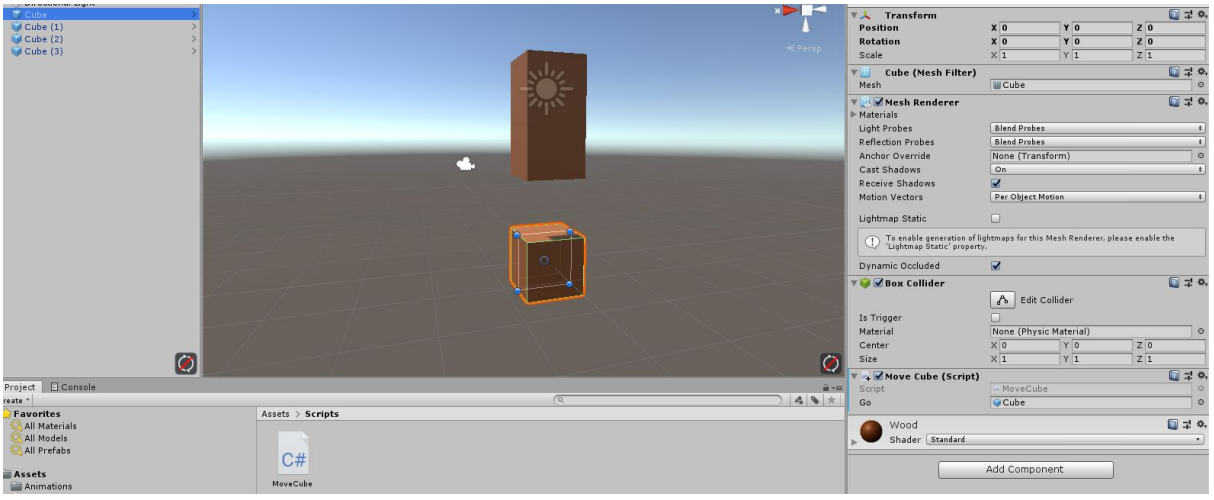




Scripts

Variablen, die in einem Mono-Behaviour Script als Public deklariert wurden, können direkt in Unity während der Laufzeit in das Script eingespeist werden. Die Lifecycle-Methode Update() wird jede Sekunde des Öfteren aufgerufen. In Start() werden initiale Sachen erledigt, wie z.B. Objekte initialisiert oder Ressourcen geladen.

```
MoveCube.cs
Assembly-CSharp                                   MoveCube
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
    0 Verweise
5   public class MoveCube : MonoBehaviour
6   {
7       public GameObject go;
        0 Verweise
8       void Start()
9       {
10          //do initial stuff
11      }
12
13
        0 Verweise
14      void Update()
15      {
16          Vector3 pos = go.transform.position;
17          go.transform.position = new Vector3(pos.x+0.01f, pos.y, pos.z);
18      }
19  }
20
```



Scenes

Alle leeren GameObjects in einer Scene werden an die Koordinaten x:0, y:0, z:0 plaziert. Objekte, die zur Laufzeit initialisiert werden, werden in ein _Dynamic-GameObject verschoben, welches immer am ende der Liste steht und somit die statischen GameObjects oben an der Liste leichter referenzierbar zu machen. Für GameObjects, die nur als Script-Container dienen wird “@” als Prefix verwendet.

Weitere Best Practices

Weitere Best Practices, die verwendet werden sollen sind: Das Schreiben von Editor Scripts, z.B. zufälliges Platzieren von Bäumen in der Scene, Shortcuts für den Programmfluss, um nicht alles durchspielen zu müssen um eine im Programmfluss später auftretende Mechanik zu ändern, Test Driven Development durch Unit-Tests und den in Unity integrierten TestDriver und Verwendung von sinnvollen Strukturen und Design Patterns, wie z.B. das Factory-Pattern zum Erstellen von Objekten oder die Beachtung des Single-Responsibility-Prinzip.

Naming Conventions

GameObject-Namen werden in PascalCase vergeben. Für die Naming-Convention in C#, bzw. den Mono-Behaviour Scripts, siehe folgende Abbildung:

Verwendung des Debuggers

Die Verwendung des Debuggers bringt den Vorteil schnell herauszufinden, welche Bedingungen an welcher Stelle erfüllt sind, bzw. An welcher Stelle welche Variablen welche Werte haben. Das Debugging mit Print-Statements sollte vermieden werden. In Visual Studio Code hat man die Möglichkeit im C#-Script das Script “An Unity anzufügen”. Im Code können dann verschiedene Haltepunkte gesetzt werden und der Status der Applikation evaluiert werden.

Intel RealSense SDK

Für Die Beschreibung der Intel RealSense siehe 4. Hardware - Intel Realsense Depth Camera

Das IntelRealSense SDK ist ein Toolkit zur Verwendung der Funktionen der RealSense in Unity. Über ein .unitypackage-File ist es möglich die von dem SDK in Verbindung mit der Kamera bereitgestellten Tools in Unity zu verwenden. Dazu gehören die Erkennung von Farben-, Tiefen- und Infrarot-Texturen.
Es wird mit Prefabs gearbeitet, die Kameras bereitstellen, um die jeweiligen Bilder in der Unity-Scene bereitzustellen. Das SDK bietet dann verschiedene Konfigurationsmöglichkeiten, um das Bild und die Erkennung im Unity-Inspector anzupassen:

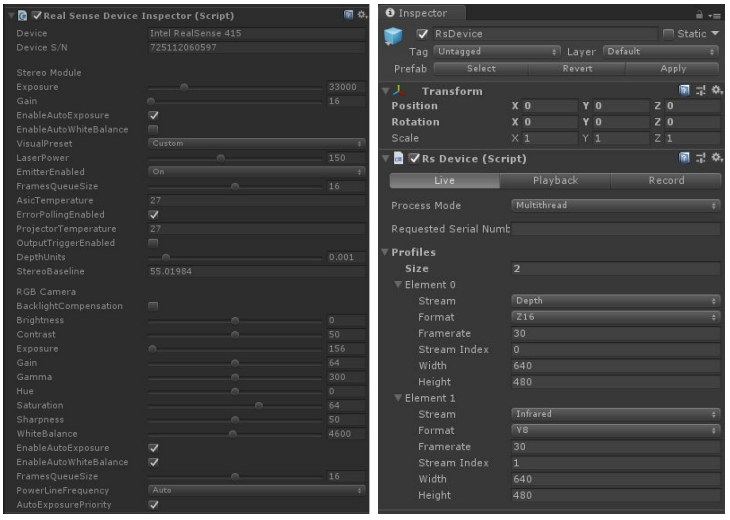


Abbildung: Anpassung der RealSense Kamera über Unity-Inspector

Das Bild in der Scene kann über den eigenen Update-Loop des SDK-Scripts abgefragt werden, die Texturen (Tiefe, Farbe, Infrarot) entnommen werden, die verschiedenen Werte dann abgefragt und dementsprechend programmatisch reagiert werden. Im Folgenden als Beispiel aus dem Projekt Pong-MR wo die Tiefe eines Objekts über die RealSense erkannt und eingefärbt wurde:

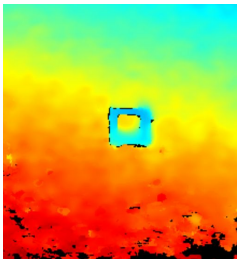


Abbildung: Eingefärbtes Depth Image der Intel Realsense

Als Beispiel die Veränderung einer Textur im Code:

```
outTexture = (Texture2D)outImageDisplay.GetComponent<RawImage>().texture;

            for (int i = boxXPos; i < boxXPos + boxSize; i += 10) {
                for (int j = boxYPos; j < boxYPos + boxSize; j += 10) {
                    int redPixels = 0;
                    int bluePixels = 0;
                    int greenPixels = 0;
                    int yellowPixels = 0;
                    for (int k = 0; k < 10; k++) {
                        for (int l = 0; l < 10; l++) {
                            Color c = outTexture.GetPixel(i + k, j + l);
                            //red
                            if (c.r > colorValues.redMod * c.g
                                && c.r > colorValues.redMod * c.b
                                && c.r > colorValues.redBrightnessLevel) {
                                outTexture.SetPixel(i + k, j + l, Color.red);
                                redPixels++;
```

Über die Methode GetPixel() kann in einem Bild, hier als zweidimensionales Array aus Pixeln, der Farbwert des jeweiligen jeweiligen Pixel abgefragt werden. Übersteigt dieser bestimmte vorgegebene Werte (redMod, ...), wird darauf reagiert. In diesem Fall werden alle Pixel die den Wert übersteigen rot eingefärbt. So können Objekte die eine bestimmte Farbe haben in dem Bild identifiziert werden.

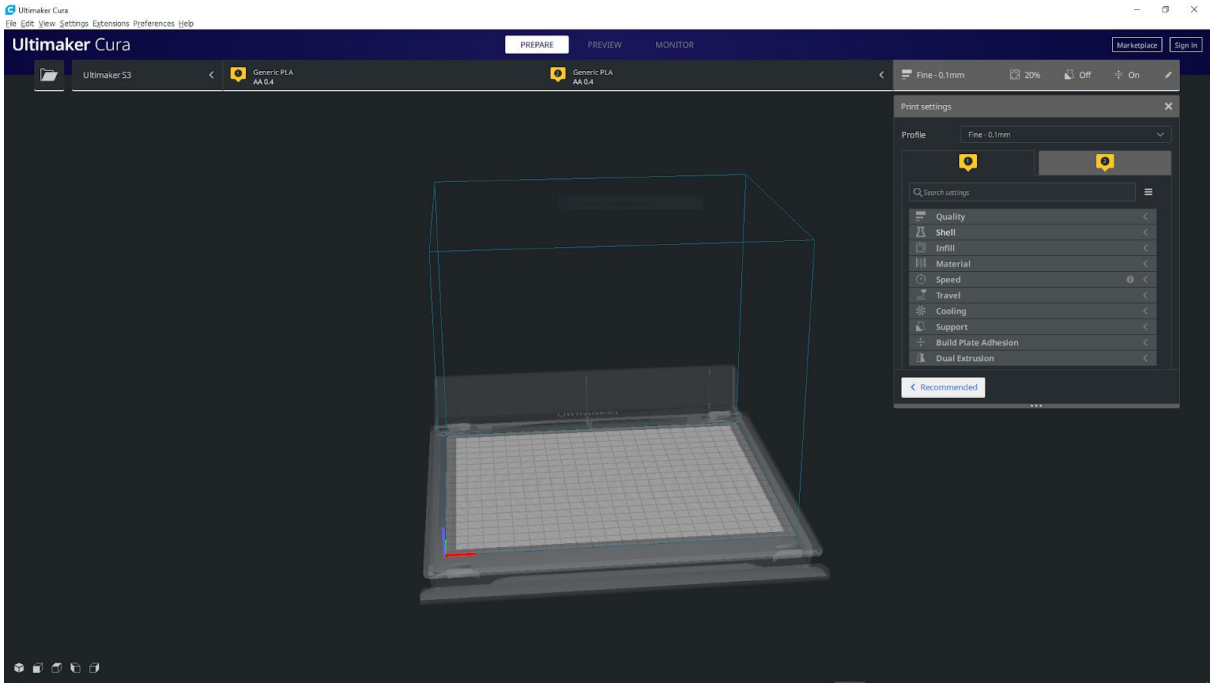
Für den Anwendungsfall in diesem Projekt siehe 5. Projekte - Pong MR

3D-Druck

Der 3D-Druck ist ein additives Fertigungsverfahren, das Gegenstände durch das schichtweise Auftragen von Fertigungsmaterial erzeugt. Hierzu werden in einem CAD/3D-Modellierungs Programm digital erstellte Modelle - mithilfe einer Slicer-Software - in Drucker-Anweisungen umgewandelt. Mithilfe dieser Anweisungen wird zumeist eine Düse eines 3D-Druckers gesteuert, die das Fertigungsmaterial schmilzt und aufträgt. Bei der Wahl des zu verwendeten Materials können neben der Belastung auch weitere Kriterien wie Oberflächenbeschaffung, Kosten, Zeit, Verarbeitung nach dem Druck, so wie Elastizität eine Rolle spielen.

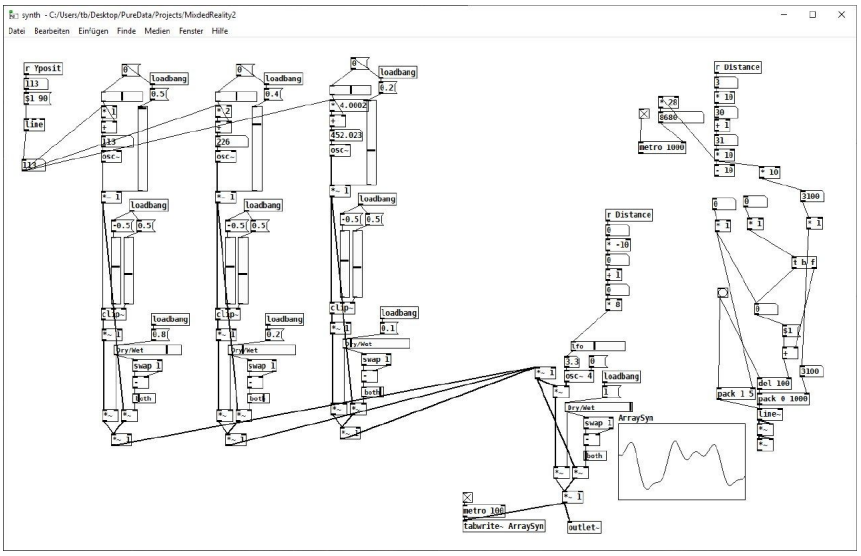
Ultimaker Cura

Cura ist eine Slicing-Software, die neben der Detailtreue und Anteil der Füllung auch Einstellungen zu Temperatur und Geschwindigkeit des Druckprozesses bietet. Letztere sind zumeist abhängig von dem verwendeten Druckmaterial, das nur bei bestimmten Werten ein optimales Druckergebnis liefert.



Pure Data

Pure Data ist eine datenstromorientierte Programmiersprache und Entwicklungsumgebung. Sie wird viel im Bereich von Sound Design und für die Programmierung von digitalen Synthesizern genutzt. Somit lässt sich Pure Data sehr gut als Werkzeug für Sonification benutzten. Synthetische Klänge können von Grund auf programmiert werden und können dadurch beliebig an die Ansprüche der Sonification angepasst werde. Der Datenstrom der für die Wiedergabe des Sounds in Pure Data benutzt wird kann in Echtzeit verändert werden. Damit eignet sich Pure Data auch sehr gut für interaktive Multimedia-Anwendungen.



Alle Daten aus Unity können mithilfe des Open Sound Control Protocol in Echtzeit an Pure Data gesendet werden und dort den von Pure Data erzeugten Sound beeinflussen.

Open Sound Control Protocol

Open Sound Control (OSC) ist ein Netzwerkprotokoll, das sich gut für die Datenübertragung in lokalen Netzwerken in Echtzeit eignet. Gerade in Multimedia- und Soundanwendungen wird es viel benutzt, um Daten zur Beeinflussung und Kontrollieren der Anwendungen zu versenden. OSC wird sehr viel in Open Source Projekten verwendet was den Vorteil hat, dass es viele offene Bibliotheken für die Verwendung gibt. Beispielsweise auch für Unity und Pure Data.
OSC kann mit UDP oder auch mit TCP verwendet werden. Da man in der Regel eine möglichst schnelle Übertragung für die Echtzeitverarbeitung haben möchte wird meistens UDP verwendet.
OSC kann in lokalen Netzwerken sowie auch für das Internet verwendet werden. In lokalen Netzwerken lassen sich Verbindungen mit Clients aller Art aufbauen. Unicast, Multicast und

Broadcast ist ohne Probleme möglich. Beispielsweise kann man mehrere Smartphones mit einem Computer verbinden und schnell Daten hin und her schicken.

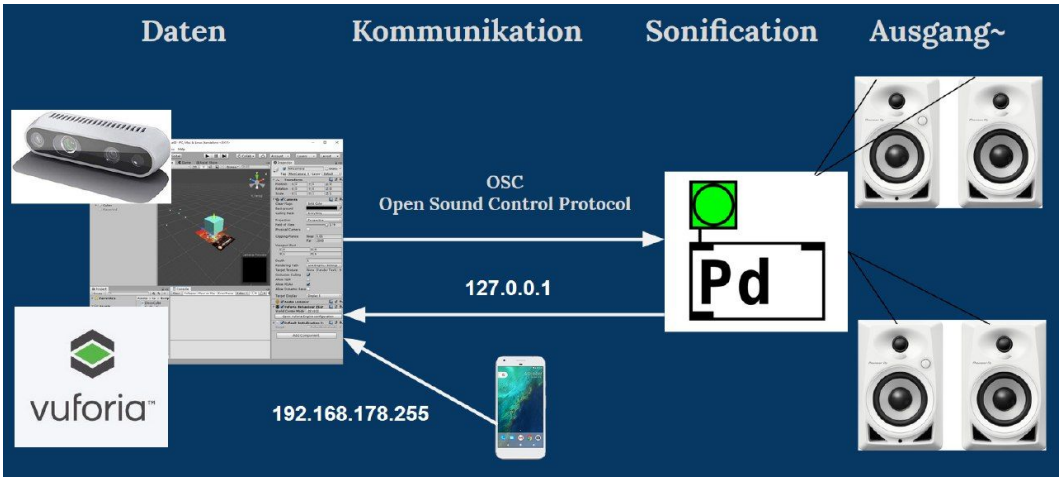


Abbildung: Beispiel von Datenübertragungen bei der OSC verwendet werden kann

Unity und Pure Data können sich beispielsweise mittels Local Host und einem festgelegten Port einen Socket aufbauen, über welchen Daten gesendet werden. Die Verwendung an sich ist sehr einfach. Die Daten, welche gesendet werden sollen, werden mit einem Slash und einer Message eingeleitet. Auf der Empfängerseite werden diese mit dieser Message dann herausgefiltert. Intern arbeitet das OSC viel mit BLOB's (Binary Large Object's).

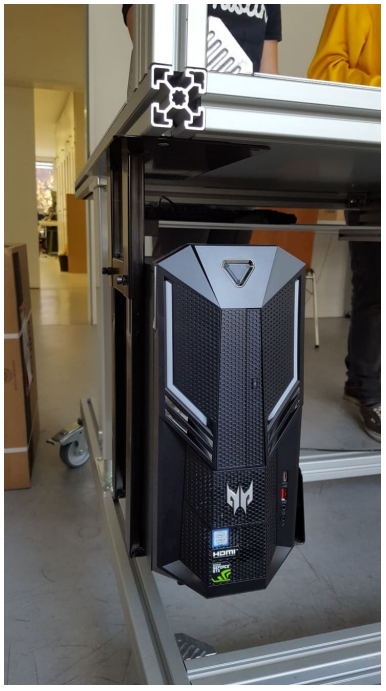
```
void Update()
{
    //sendet x position an Pure Data
    osc.GetComponent<OSCSendReceive>().PlaySoundOSC("/Xwert " + transform.position.x);

    osc.GetComponent<OSCSendReceive>().PlaySoundOSC("/LocalY " + transform.localPosition.y);

}
```

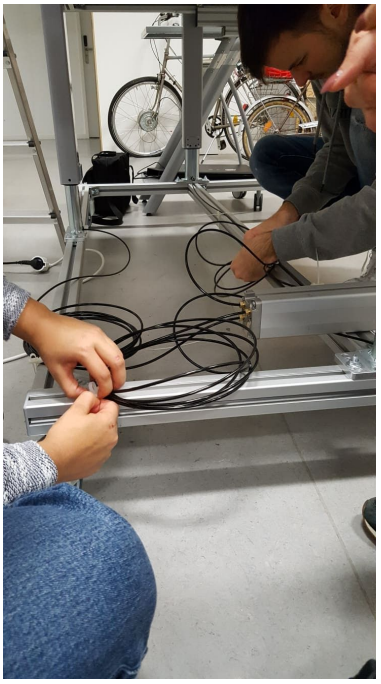
Abbildung: Beispiel wie in einem C# Script Daten von Unity mittels OSC versendet werden

4. Hardware

Tischaufbau

Der Aufbau des Tisches beanspruchte mehrere Wochen.
Nachdem das Grundgerüst zusammengeschraubt wurde, wurde eine Aluminiumplatte als Tischplatte zugesägt und befestigt. Um weiterhin die glatte Optik der Platte zu bewahren, haben wir uns dafür entschieden diese nicht anzuschrauben, sondern nur festzukleben.
Anschließend musste die Hydraulik für das Auf- und Absenken des Tischen, die Beamer-Halterung und die PC Befestigungen angebracht werden.
Dabei hat besonders die Beamer-Halterung Probleme bereitet.





Beamer-Halterung

Beamer-Test

Beamer-Details: Optoma EH460

Beim Testen des Projektorbildes auf die Tischplatte hat sich herausgestellt:

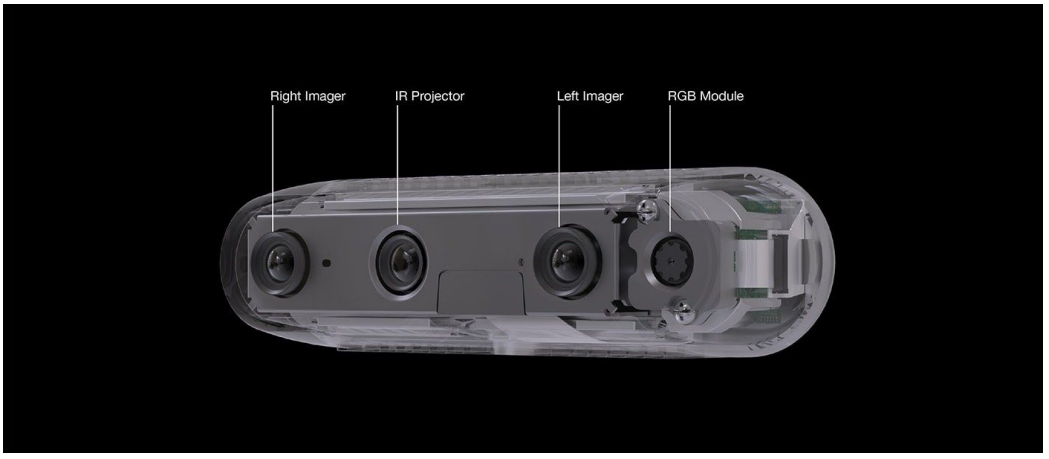
- Das projizierte Bild ist kleiner als die Tischplatte
- Das projizierte Bild ist etwas schräg
- Das Bild wird in einer Ecke abgeschnitten
- Der Winkel in dem das Bild von dem Projektor abgestrahlt wird ist sehr groß.

Mögliche Korrekturen in den Beamer-Einstellungen

Der Beamer bietet die Funktion der Trapezkorrektur. Hiermit lassen sich die Ecken des projizierten Bildes verschieben. Die Korrekturmöglichkeit reicht jedoch nicht aus, um das Bild komplett quadratisch zu projizieren. Es scheint zudem, als ob dadurch das Bild unschärfer wird.
Es wurde noch in Erwägung gezogen, die Beamerhalterung anders zu platzieren. Hierbei wurde jedoch keine Möglichkeit entdeckt, das Bild besser zu projizieren.
Fazit: Insgesamt ist die Halterung des Beamers nicht optimal. Das Bild kann nicht quadratisch, unverzerrt und den Tisch ausfüllend projiziert werden.

Intel Realsense Depth Camera

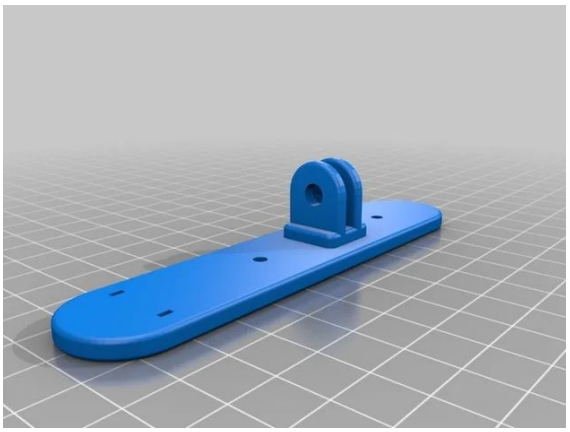
Die *Intel Realsense Depth Camera* ist eine von Intel entwickelte Tiefenbildkamera mit der Abstände von Objekten im Raum von der Kamera gemessen werden können. Überdies erzeugt die Kamera ein echtzeitbild mit den im Raum gemessenen Proportionen. Sie verfügt über zwei Tiefensensoren mit deren Daten in einem Stereotiefenalgorithmus ein Tiefenbild erzeugt werden kann. Da die Tiefen Erkennung auf einem gleichfarbigen Grund schwer ist, ist die Intel Realsense auch mit einer Infrarot Sensor ausgestattet, welcher ebenfalls in der Lage ist entfernungen zu Messen und ein räumliches Bild zu erstellen. Für Bilder in Farbe ist der integrierte RGB-Sensor zuständig.

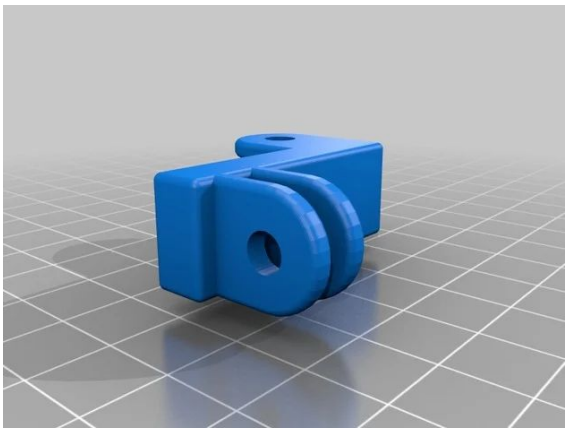


Im Rahmen des Projekts soll die Kamera als integraler Bestandteil für Interaktionen mit dem Tisch fungieren. Solche Interaktionen können mithilfe von Tracking-Algorithmen, oder externen Tracking-Libraries zur Auswertung des Farb- oder Tiefenbildes, ermöglicht werden. Da das FOV (Field of View) der Kamera lediglich ca 87 Grad beträgt, kann nicht die gesamte Fläche des Tisches erfasst werden. Um den erfassten Bereich auf dem Tisch zu erweitern, wird derzeit über ein Multikamerasetup nachgedacht. Diese könnte hinzukommend noch die durch Hände oder andere Gegenstände entstehende Verdeckung verkleinern, da nun 2 Kameras aus verschiedenen Winkeln die vom Beamer bestrahlte Fläche beobachten.

Halterung

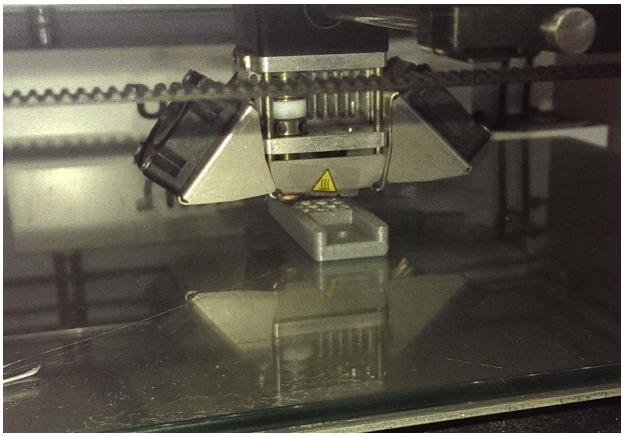
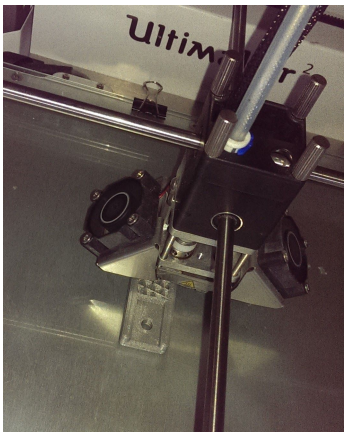
Für die Befestigung der Realsense-Kamera am Projekttisch wurden 3D-schematics von der Seite *thingiverse.com* vom user “vatbrain” genutzt um einen Intel RealSense D435 camera GoPro adapter zu drucken.



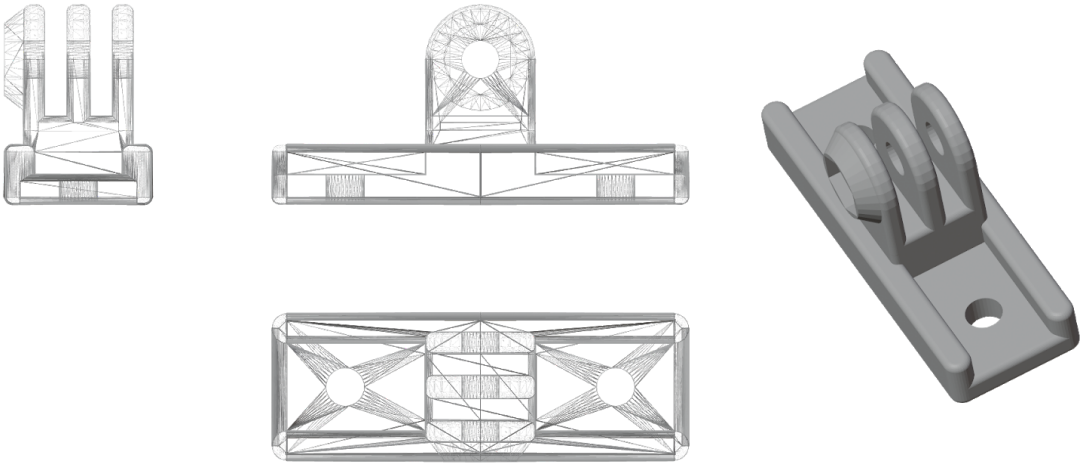


Durch Überkopfstellen der Kamera entspricht das erhaltene Bild auch gleich dem Koordinatensystem der Spielwelt. Hierdurch muss der Kamera-Feed nicht zuvor digital gespiegelt werden.

Weiterhin ist aufgefallen, dass das Verbindungsstück von Halterung zur Schienenaufhängung nicht einwandfrei angebracht werden konnte. Ein überstehendes Teil des Verbindungsstückes stieß hierbei an die Gewindeteile in der Schiene. Das Teil wurde in Blender angepasst und nach Einführung in den Umgang mit dem 3D-Drucker und zugehöriger Software Ultimaker Cura gedruckt. Das Teilstück stimmte mit der restlichen Halterung nahtlos überein und konnte samt

Kamera am Projekttisch befestigt werden.

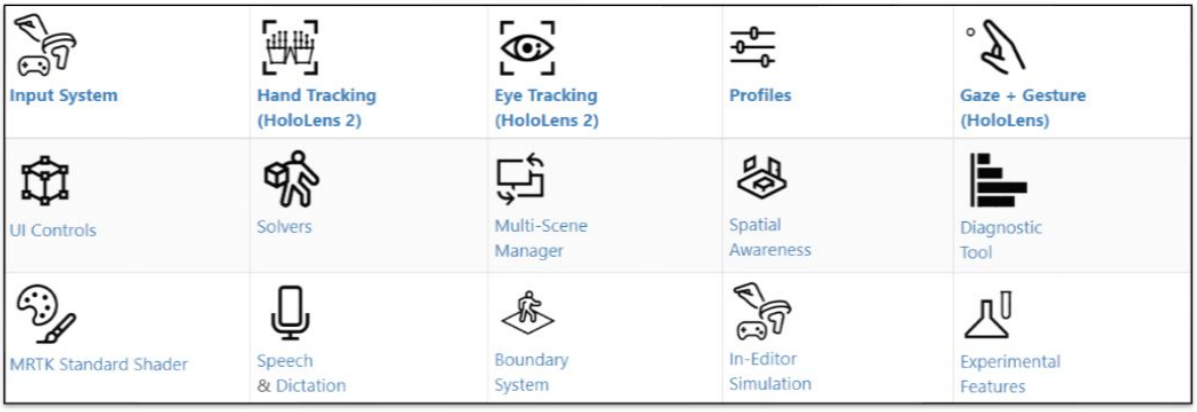


Hololens

Die Hololens ist ein Augmented Reality Technologie, mit der es möglich ist Hologramme in der realen Umgebung zu platzieren und diese anschließend mit Gesten- und Spracherkennung zu steuern und manipulieren.
Anwendungen für die Hololens können in Unity programmiert werden, wodurch die Hololens mit unseren anderen Projekten kompatibel ist und auch der Arbeitsprozess sich ähnlich gestaltet hat.

Mixed Reality Toolkit

Um nicht jedes Feature eigenhändig programmieren zu müssen, haben wir das Mixed Reality Toolkit genutzt. Dadurch wurden einige Aufgaben erleichtert und konnten schneller umgesetzt werden.



Das Toolkit verfügt über mehrere Features, das folgende Bild zeigt ein Überblick über die Features, die durch die zweite Version des Toolkits angeboten werden:

Im unserem Projekt haben wir vorrangig mit folgenden Features gearbeitet:

- Input System
- Gaze + Gesture
- UI Controls
- Spatial Awareness

Die UI Controls haben uns beim schnellen Prototyping geholfen, hierbei haben wir besonders die Buttons genutzt.

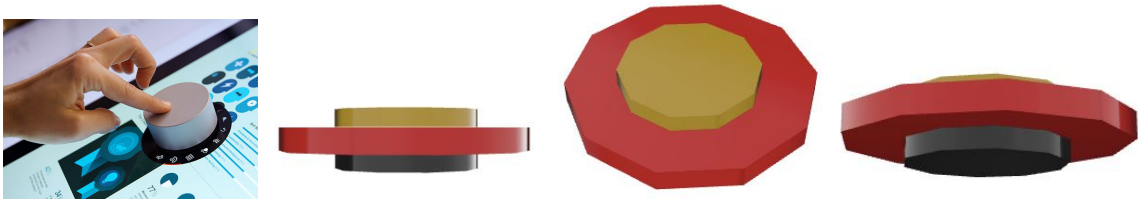
Spatial Awareness war ein Feature mit welchem wir mehrere Probleme hatten.
Innerhalb des Mixed Reality Toolkit setzt sich Spatial Awareness in Placement, Occlusion, Physics, Navigation und Visualisation zusammen.
Jedes dieser Features hilft dabei die Hologramme realer erscheinen zu lassen.

Griffhalterung der Marker

Zur Verbesserung der Marker wurden vier Konzepte zur Modifikation entwickelt. Hierbei wurden primär Lösungsansätze aufgezeigt, bei denen die Marker Griffe erhalten. Ein Problem, welches dabei jedoch in Kauf genommen wird, ist die verschlechterte Handhabung der Marker aufgrund der ergonomischen Eigenschaften in Verbindung mit den zusätzlichen Griffen. Außerdem könnten potenziell weitere Interaktionsmöglichkeiten in Verbindung mit den Griffen implementiert werden.

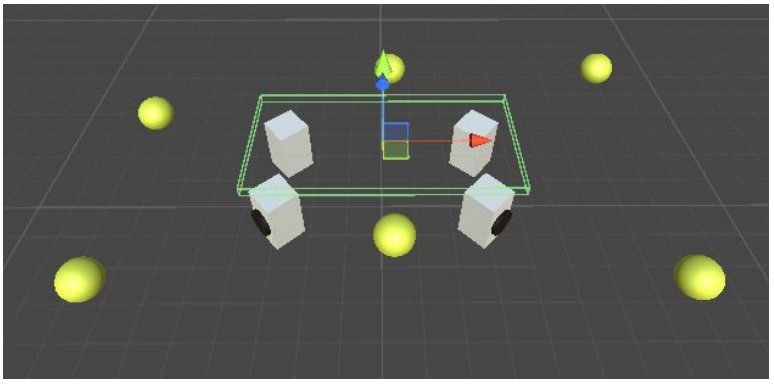
Um das Problem der Ergonomie zu beseitigen, wurde eine Lösung von Microsoft als Beispiel herangezogen. Nämlich das Microsoft Dial. Hierbei wird die Steuerung mehrerer Marker mit einem einzelnen Griff ermöglicht. Durch das Drehen und Schieben dieses Griffs können verschiedene Parameter adjustiert werden, sodass die Steuerung verschiedener Marker damit kombiniert wird. Auf Basis des Microsoft Dials wurde das ursprünglich konzipierte Modell des Griffs erweitert, sodass dies zusätzliche Interaktionsmöglichkeiten bietet (s. Abbildung oben).

Es gibt 3 Platten, mit verschiedenen Farben und Größen, wobei jede Platte eine Funktionalität repräsentiert. Perspektivisch sollte das Konzept weiter durchdacht und verbessert werden, um mit verschiedenen Ansätzen weitere Interaktionsmöglichkeiten zu entwickeln.



Soundsystem

Der Aufbau des Raumklangs an dem Tisch ist sehr individuell. Bei herkömmlichen Soundsystemen wie 2.0 oder 5.0 soll der*die Hörer*in optimalerweise in der Mitte der Lautsprecher platziert werden. Bei der Umsetzung für den "Tischklang" stehen die Personen jedoch um den Tisch herum. Es ergibt Sinn ein 4.0 oder 4.1 System aufzubauen, bei dem man Lautsprecher an jeder Ecke des Tisches platziert.



Um einen Raumklang auf dem Tisch durch einen Panning-Effekt zu erzeugen, müssen die vier Lautsprecher einzeln anzusteuern sein. Nach Beratung von Herrn Schleicher wurde hierfür ein Audiointerface mit 4 Ausgängen gewählt. Im Soundstudio der Hochschule Bremen wurde ein Test mit Pure Data gemacht, vier verschiedene Ausgangskanäle einzeln anzusteuern. Für die Kombination mit dem Audiointerface eignen sich aktive Monitorlautsprecher.

Ein Interface mit den entsprechenden Eigenschaften wäre z.B. das
Behringer U-Phoria UMC204HD

Aktive Monitorlautsprecher mit den entsprechenden Eigenschaften wären z.B die
Behringer C50A Studio Monitore

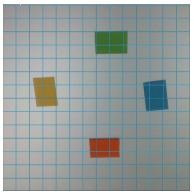
5. Projekte

Pong MR

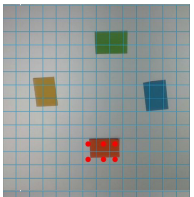
Pong MR ist eine Abwandlung des Retro-Videospiels Pong für 4 Personen einer Mixed-Reality Umgebung. Die Aufgabe mit der sich bei der Entwicklung von Pong MR beschäftigt wurde war die Auswertung eines von einer Intel RealSense bereitgestellten Videostreams. Es wurde dabei ein eigener Object-Tracking Algorithmus geschrieben, der verschiedenfarbige Objekte in einem Bild erkennt. Mithilfe dieses Algorithmus ist es möglich, in Echtzeit physische Tangibles auf dem Tisch zur Steuerung des auf den Tisch projizierten Spiels zu benutzen.

Das Kernproblem das noch vor Beginn der Entwicklung thematisiert wurde war das Erkennen von Farben in einem Videostream. Dafür wurde das Bild des Streams in jedem Frame Pixel für Pixel durchiteriert und geprüft, ob die RGB-Farbwerte des Pixels in eine bestimmte Reichweite von Parametern fällt. So eine Reichweite könnte für die Farbe Rot beispielsweise sein, dass der Rotwert mindestens 1,5 mal größer sein muss als der Grün- und Blauwert, und dass die Helligkeit mindestens 75% betragen muss. Diese Parameter müssen präzise gewählt werden, um auf der einen Seite zu vermeiden, dass ähnlich farbige Objekte auch in das Schema fallen und auf der anderen Seite, dass die gesuchte Farbe auch unter verschiedenen Beleuchtungssituationen und Winkeln zuverlässig erkannt wird. Pixel die das angegeben Farbschema erfüllen werden im Programm markiert und zu einer Liste hinzugefügt.
Im nächsten Schritt wurde ein Algorithmus entwickelt, der aus dieser großen Zahl an markierten Pixeln die Position des gesuchten Objektes zuverlässig ermittelt. Die Funktionsweise dieses Algorithmus wird im Folgenden schrittweise erläutert:

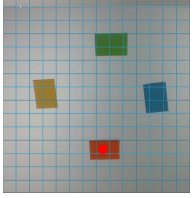
1. Das Bild das auf dem das Rote Objekt erfasst werden soll



2. Das Bild wird wird in 10x10 Pixel Kacheln aufgeteilt.

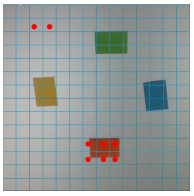


3. In Jeder Kachel werden alle Pixel auf das Farbschema geprüft. Sind mindestens 80 Pixel markiert worden, so werden die Koordinaten der Kachel in einem Array festgehalten.

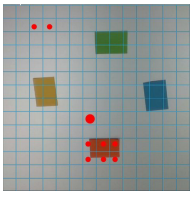


4. Der Mittelpunkt der markierten Kacheln wird berechnet.

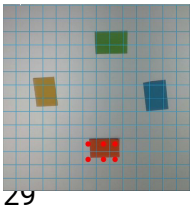
Betrachtet man nun einen hypothetischen Fall in dem noch andere, in das Farbschema passende, rote Objekte im Bild sind wird ersichtlich, warum die Aufteilung des Bildes in Kacheln nützlich ist.



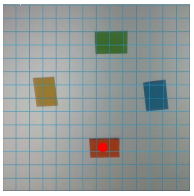
1. Alle Kacheln des Bildes wurden geprüft. In der oberen linken Ecke des Bildes wurde noch weiter Kacheln als rot ermittelt.



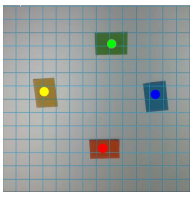
2. Der berechnete Mittelpunkt liegt zwischen den beiden Objekten. Nun wird noch einmal über den Array mit den markierten Kacheln iteriert. Dabei wird jede Kachel, die zu weit vom berechneten Mittelpunkt entfernt liegt aus dem Array entfernt.



3. Es bleiben nun nur die Kacheln übrig, die nah am gesuchten Objekt liegen.



4. Von diesen wird erneut der Mittelpunkt berechnet.



5. Dieses Verfahren wird nun auch für die anderen Farben wiederholt. Am Ende wurden 4 Koordinaten ermittelt, die jeweils den Mittelpunkt der Objekte angeben.

Mithilfe dieses Verfahrens lassen sich die vier verschiedenfarbigen Tangiglbes zuverlässig und weitgehend fehlerresistent tracken. Die Position der Objekte wird nun in jedem Frame berechnet und die ermittelten Koordinaten fungieren als Steuerung für die 4 Paddles der Spieler.

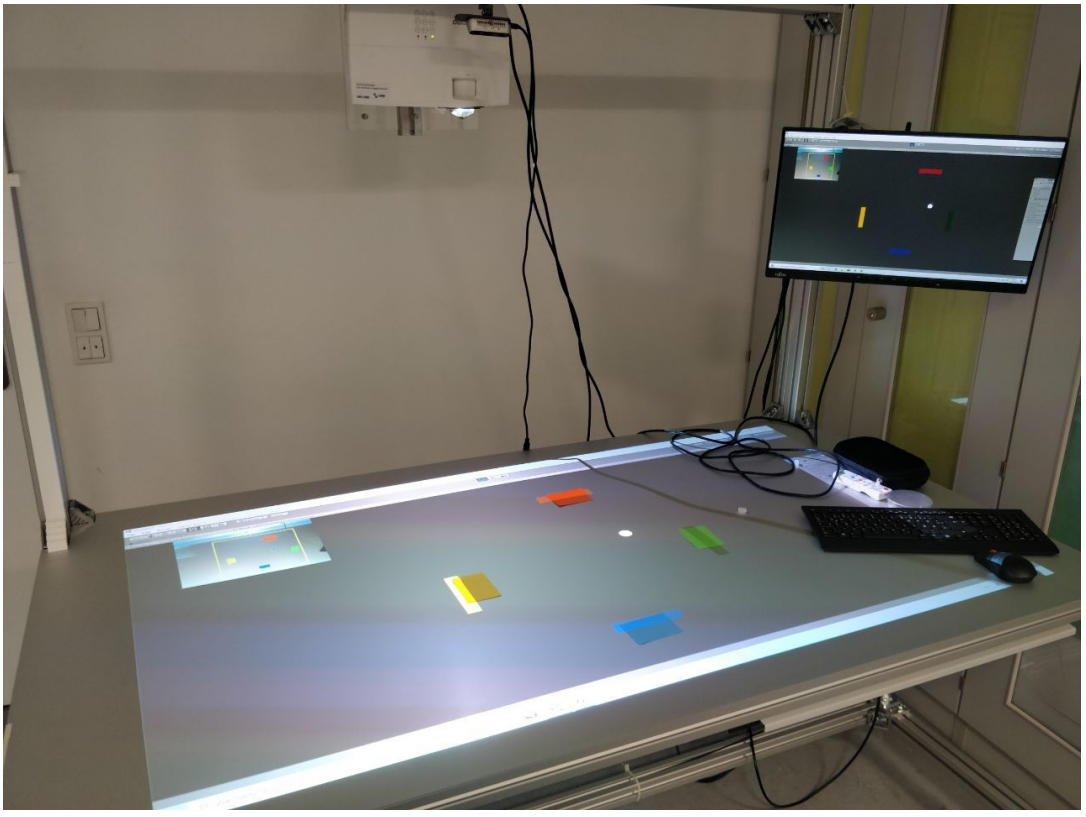


Abbildung: Finales Setup von Pong MR

Der nächste Schritt in der Entwicklung wäre anstatt ein Farbbild das Tiefenbild der Realsense auszuwerten und somit besseres tracking und Benutzung von effektiveren Tangibles zu ermöglichen. Eine Möglichkeit dies zu erreichen wäre das eingefärbte Depth-Image (siehe Abbildung unten) der Realsense zu nutzen und den oben erklärten Algorithmus mit Farbwerten für bestimmte Abstände einsetzen. Der Algorithmus müsste in diesem Falle gegebenenfalls noch um Funktionalität erweitert werden, multiple gleichfarbige Objekte zu erkennen.

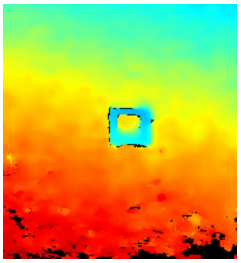


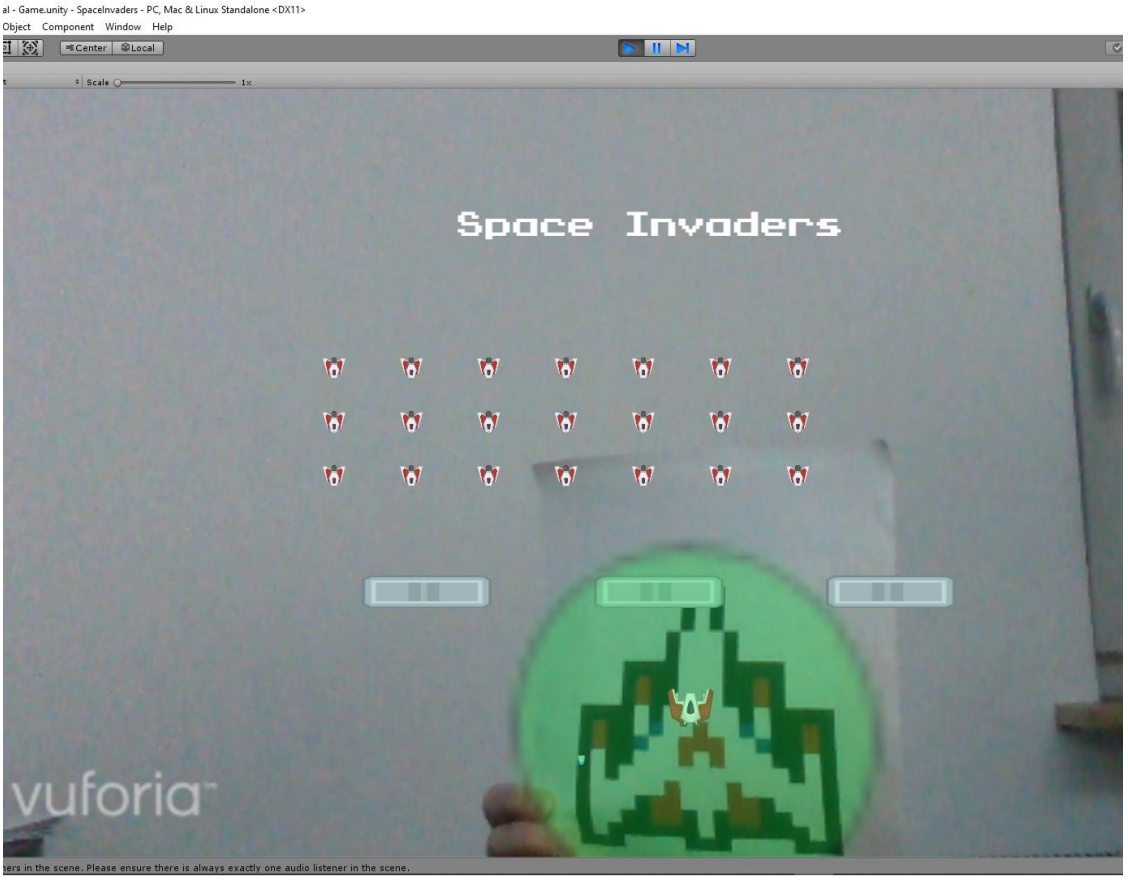
Abbildung: Eingefärbtes Depth Image der Intel Realsense

Space Invaders

Als weitere Testumgebung für die Mixed Reality Werkzeuge des Projekttisches wurde der Spielklassiker Space Invaders in Unity umgesetzt. Zunächst erfolgte die Steuerung des Spielers nur über die Tastatur.

Anschließend wurde mithilfe des Trackings von Vuforia die Bewegung des Spielers realisiert. Für die Tracking-Marker wurde ein individuelles Bild herausgesucht und mit Ausdrucken verschiedener Größen das Optimum für die Arbeit am Mixed-Reality Tisch gesucht. Hierbei stellte sich ein ca. 10cm großer Marker als gut für die Arbeit mit der Realsense in Verbindung mit Vuforia heraus. Kleinere Motive wurden aus der Entfernung der Kamera zur Tischplatte nicht hinreichend erkannt.

Das Tracking mit Vuforia lieferte zunächst 3D-Koordinaten was zur Folge hatte, dass sich das Schiff im Spiel, abhängig vom Abstand des Trackers zur Realsense, unterschiedlich schnell bewegte. Um dies zu verbessern wurde die durch Vuforia erhaltene Tracking-Position mathematisch auf eine Ebene projiziert, sodass der Abstand des Markers keinen Unterschied mehr bei der Bewegung des Schiffes darstellte.

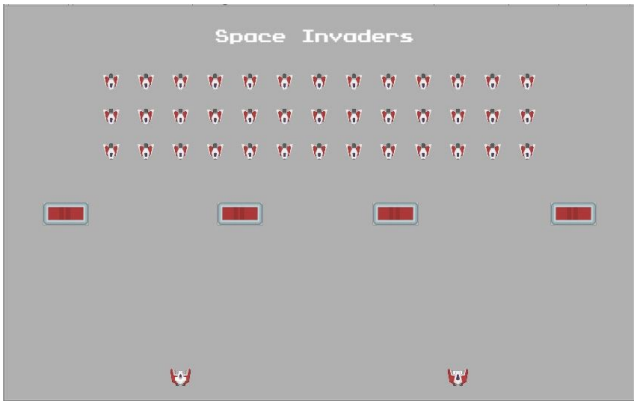


Das Schießen des Spielers wurde außerdem über die Rotation des Tangibles auf dem Projekttisch realisiert, sodass das gesamte Spiel ohne konventionelle Eingabe auf der Tastatur spielbar ist.

Das Unity Projekt wurde zusätzlich um die Platzierung von Hindernis-Wänden erweitert. Diese blockieren eine bestimmte Anzahl an Projektilen bevor sie zerstört und aus der Spielszene entfernt werden.

Der Kamera-Feed der Realsense wurde von Vuforia standardmäßig im Hintergrund des Spiels angezeigt. Bei der Projektion mit dem Beamer wurde dieser somit auch über die gesamte Tischplatte projiziert und erneut von der Kamera aufgenommen, wodurch ein Endlosbild entstand. Um diesem Effekt entgegenzuwirken wurde der Kamera-Feed skaliert und in eine Bildecke verschoben.

Als zusätzliche Features wurde das Platzieren der Hindernis-Wände mithilfe von Tracking-Objekten sowie zwei Multiplayer-Modi umgesetzt. Der eine Multiplayer ist ein Spieler gegen Spieler - Modi, während der Andere eine Kooperation zwischen zwei Spielern, gegen den Computer darstellt. Um zwischen diesen Modi wechseln zu können wurde ein Menü mit Navigation hinzugefügt.


Space Invaders

1v1

Holo-Cubes

Der Prototyp “Holo-Cubes” stellt eine Anwendung dar, in der Möglichkeiten und Grenzen der Hololens für das Projekt getestet wurden. Um für das Projekt nützliche Funktionen zu erlernen, wurde in der Erarbeitung des Prototyps das Tracking von der Umgebung mit der Interaktion von virtuellen Gegenständen kombiniert. Diese Verknüpfung ist besonders unter dem Aspekt der Mixed Reality interessant für unser Projekt, da an diesem Punkt virtuelle Objekte auf unsere Umgebung reagieren.

Im Rahmen des Prototypen wurde zu Beginn ein Würfel erschaffen, mit dem der*die Nutzer*in interagieren kann, das heißt verschieben, vergrößern und drehen (dazu verwendet wurden die Scripts *ManipulationHandler*, *NearInteractionGrabbable* und *Bounding Box* des Mixed Reality Toolkits).

Der Würfel sollte im nächsten Schritt durch den*die Benutzer*in erst erstellt werden, sodass beliebig viele Würfel im Raum erschaffen werden können. Umgesetzt wurde dies mit einem Button, der sich zuerst frei im Raum befand. Pro Klick wurde ein Würfel geschaffen, der nun bewegt werden kann. Den Übergang zur Mixed Reality geschah im nächsten Schritt. Der frei im Raum schwebende Button hatte zunächst keinen wirklichen Bezugspunkt und verdeckte unnötig das Sichtfeld, wenn zum Beispiel mit den Würfeln interagiert wurde.

Der Button sollte somit im nächsten Schritt mit der direkt realen Umwelt interagieren. Mit Hilfe von Bildern sollte die Anwendung durch Verwendung von Image Targets und der SDK Vuforia Buttons anzeigen, wenn sich das jeweilige Bild im Sichtfeld befindet. Dadurch hat der*die Nutzer*in die Möglichkeit, selbst zu bestimmen, wann und wo ein Button auftaucht.

Um dies sinnvoll umzusetzen haben wir verschiedene Marker ausprobiert. Dabei war schnell klar, dass wir diese Marker auf einem Würfel platzieren wollten, um die später erschaffenen, virtuellen Würfel nachzuahmen und mehrere Seiten für verschiedene Interaktionen zu haben.

Anfangs haben wir überlegt einen Rubiks Cube als Grundlage zu verwenden und Bilder mit Mustern als Marker daran zu befestigen.

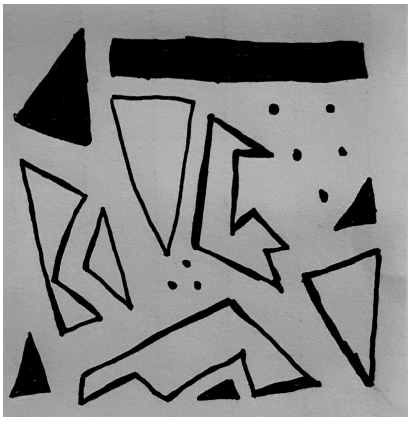


Abb.: Beispiel der selbst erdachten Marker

Doch aufgrund der schlechten Reproduzierbarkeit und um Lasercutter-Techniken auszuprobieren, entschieden wir uns letztendlich dagegen.
Für die konkrete Umsetzung nutzten wir die oben (Tangible User Interfaces) beschriebenen fiducial markers, die mit Hilfe eines Lasercutters auf eine Holzplatte eingraviert wurden.

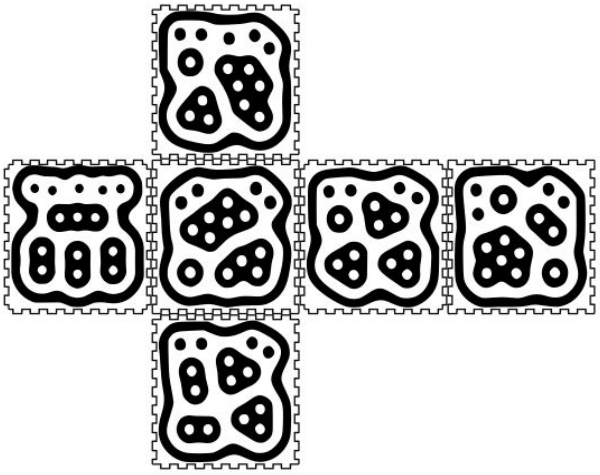


Abb.: Vorlage für den Lasercutter

Abb.: Tangible User Interface - Würfel

Als Grundlage für die Entwicklung von spielerischen Anwendungen sollen durch verschiedene Marker verschiedenfarbige Würfel erstellt werden. Denkbar wäre zum Beispiel das Nachbauen von vorgegebenen Figuren oder das Platzieren von Würfeln einer bestimmten Farbe an einer markierten Stelle im Raum. Insgesamt sechs Marker wurden auf einem Würfel angeordnet und für den*die Nutzer*in entsprechend händisch eingefärbt. So sollte dem Nutzer vermittelt werden, welche Farbe der virtuelle Würfel haben wird.

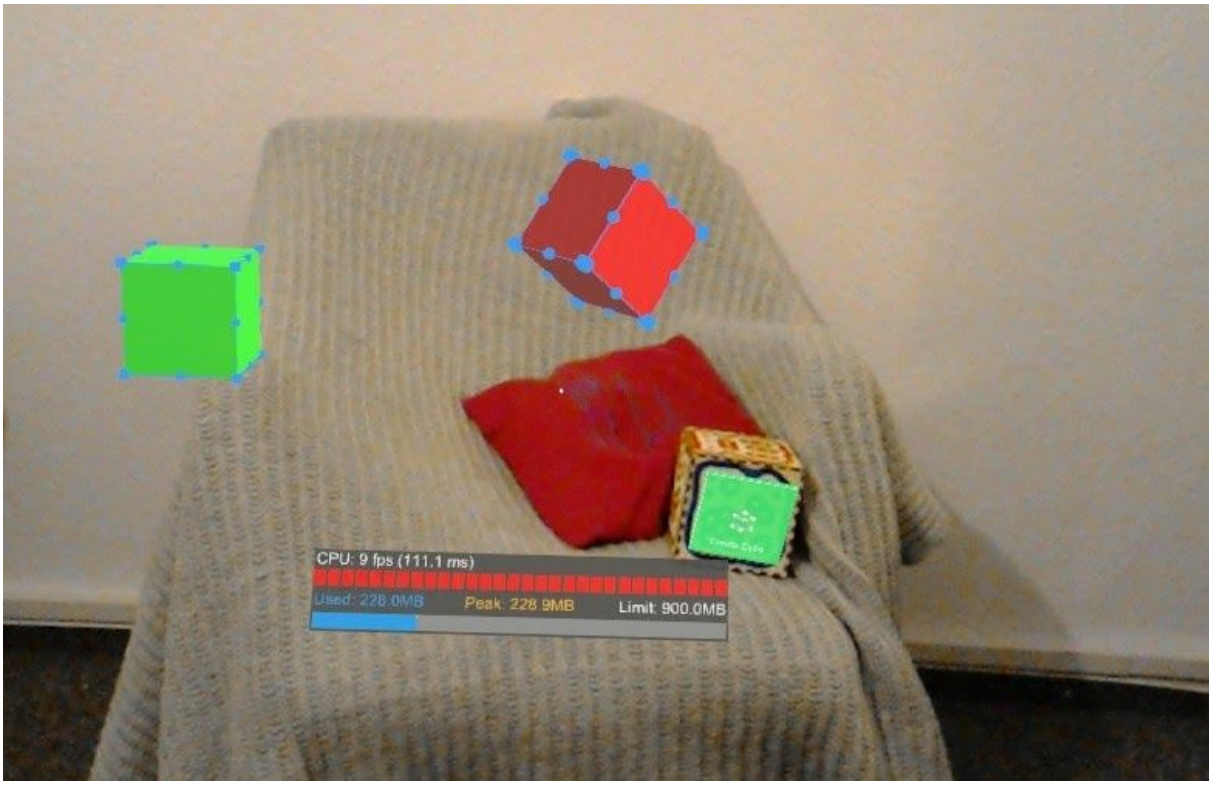


Abb.: Verschiedenfarbige Würfel in der Umgebung

Zum Abschluss wurde ein Feedback für richtig erledigt Aufgaben eingebaut. Im aktuellen Fall erscheint eine Mitteilung, wenn ein Würfel der richtigen Farbe (blau) erstellt wurde. Dies lässt sich jedoch beliebig auf Farbkombinationen oder komplexere Anforderungen erweitern. Somit sind neben spielerischen Anwendungen auch Anwendungen im Bereich Bildung oder Simulation denkbar.

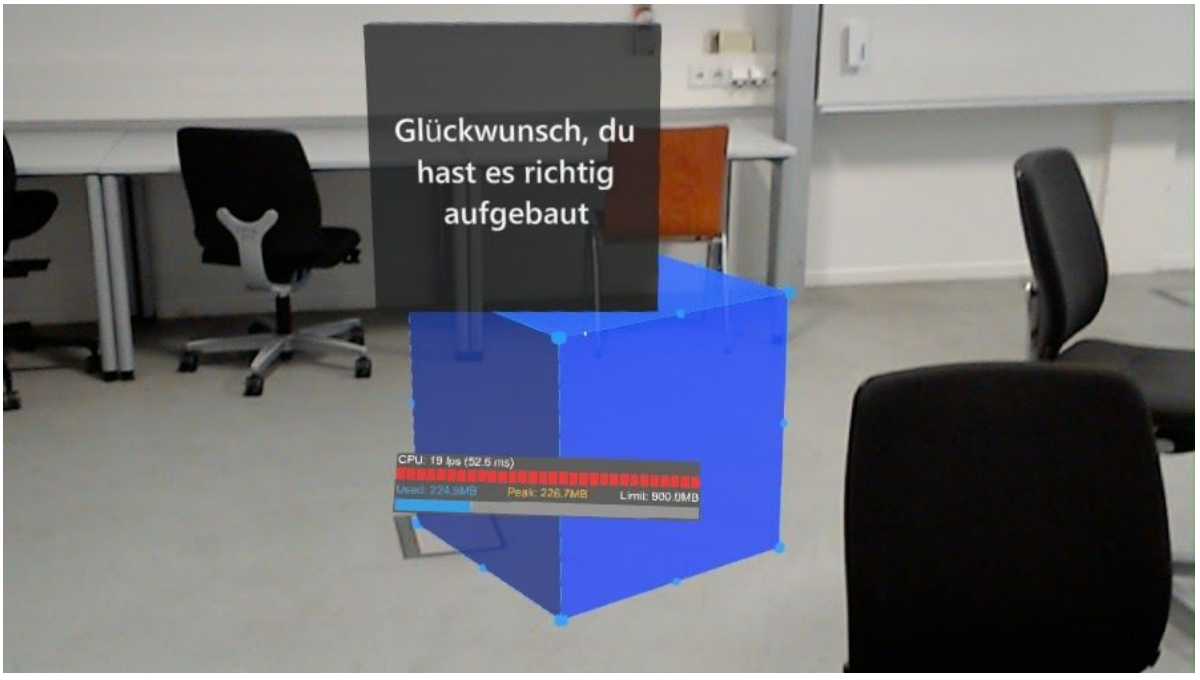


Abb.: Feedback für erfolgreiches Lösen einer Aufgabe

Außerdem haben wir uns noch viel mit dem Bereich Spatial Awareness (Tracking des realen Raumes) beschäftigt, dabei haben wir folgende Erkenntnisse gewonnen**:**

Um Spatial Awareness des Mixed Reality Toolkits in ein Projekt einzubauen ist es nötig, dass ein Spatial Manager in der aktuellen Szene vorhanden ist. Dieser muss das Profil aus dem Mixed Reality Toolkit Playground implementieren. Dort ist es auch möglich spezifische Einstellungen über das Profil zu treffen, so ist es zum Beispiel möglich den Scan konstant, teilweise oder gar nicht anzeigen zu lassen.

Allerdings müssten noch weitere Justierungen getroffen werden, die sonst Probleme erzeugen. Doch durch die unvollständige Dokumentation und verschiedenen Versionen war es uns leider nicht mögliche dieses Feature komplett umzusetzen.

Eine andere Möglichkeit wäre es das Spatial Mapping von Unity zu nutzen, doch auch dies löste Probleme aus, welche wahrscheinlich auf unsere intensive Nutzung der Mixed Reality Toolkits zurückzuführen ist.

Sollten wir uns weiterhin mit der Hololense beschäftigen**,** könnten folgende weiteren Features interessant werden: Erdanziehung, Spracheingabe.

Wir haben uns grob mit diesen Features beschäftigt, doch leider ist unsere Bearbeitung dessen noch nicht ausgereift genug, um sie in unserer Anwendung einzubauen.

Das Projekt kann hier heruntergeladen werden: https://drive.google.com/open?id=1qOT4-zNivPLloEVuMEAjtAObpIX7crhk. Für die Benutzung muss es mit Unity geöffnet werden und auf eine Hololens gespielt werden. Die Anwendung funktioniert mit den oben abgebildeten Mustern.

6. Ausblick

In diesem Semester war das Ziel sich mit der Hard- und Software vertraut zu machen und mit deren Hilfe kleinere Projekte zu erstellen. Diese Kenntnisse sollen im nächsten Semester hilfreich sein um mit der gesamten Gruppe ein großes Projekt auf die Beine zu stellen. Um keine Zeit zu verlieren, haben wir uns bereits am Ende des ersten Semesters Gedanken gemacht wie, die zweite Hälfte des Projekts verlaufen soll und welche Aufgaben noch zu erfüllen sind.

Zuerst einmal sollte der MR-Tisch fertiggestellt und optimiert werden. Der Fokus liegt dabei auf der mobilität des Tisches, im Moment ist der Tisch vollständig aufgebaut und funktioniert einwandfrei, jedoch ist er zu breit um ihn aus dem Raum, in dem er sich zur Zeit befindet, raus zu schieben. Im neuen Semester sollte eine Konstruktion entwickelt werden, die es zulässt den Tisch zwischen verschiedenen Räumen zu befördern ohne diesen auseinanderbauen zu müssen. Außerdem sollten Kleinigkeiten sowie die Stabilität der Halterung des Monitors und die Abdichtung der scharfen herausstehenden Kante verbessert

werden. Zusätzlich sollte der Tisch mit einer geeigneten Konstruktion von Lautsprechern versehen werden.

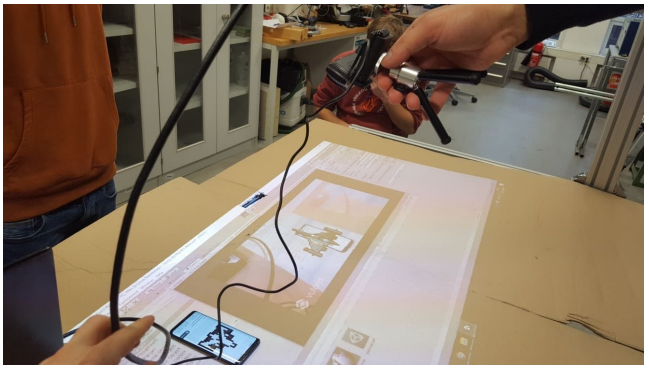
Des weiteren sollen sich die Funktionalitäten des Tisches ausweiten. Neben Farberkennung und Bild Tracking sollten weitere Methoden sowie Tangible-Tracking und Gestenerkennung erforscht und in das Projekt eingebracht werden. Es sollte ebenfalls erforscht werden inwiefern es sinnvoll ist Sonifikation in das Projekt mit einzubringen und in welcher Form dies geschehen kann.

Der nächste Schritt wäre die Verwirklichung eines neuen Projektes, dabei wurden einige Vorschläge gemacht. Gesprochen wurde neben der modularen Produktionsstraße, über eine Verkehrssimulation, ein Agentenspiel bei dem Menschen über Tangibles gesteuert werden und eine Anwendung bei der Kindern spielerisch das programmieren näher gebracht wird. Letztendlich fiel die Entscheidung auf die Produktionsstraße. Neben der Idee für das Projekt wurde ebenfalls über die Beschaffung eines möglichen Kooperationspartner gesprochen, eine Idee war, mit dem Science Center – Universum Bremen eine Kooperation zu bilden.

7. Anhang / Fotos

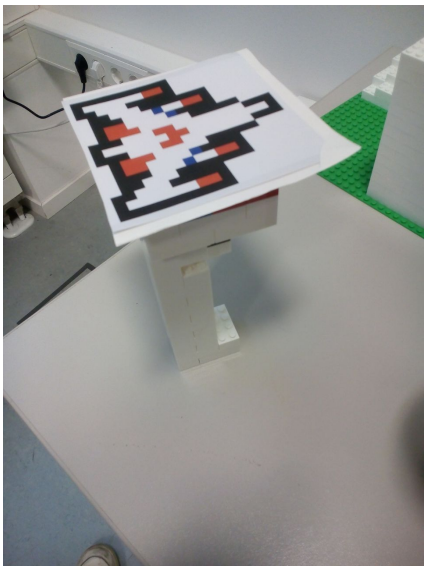
Testen der Bildanpassung mit einem Spiegel

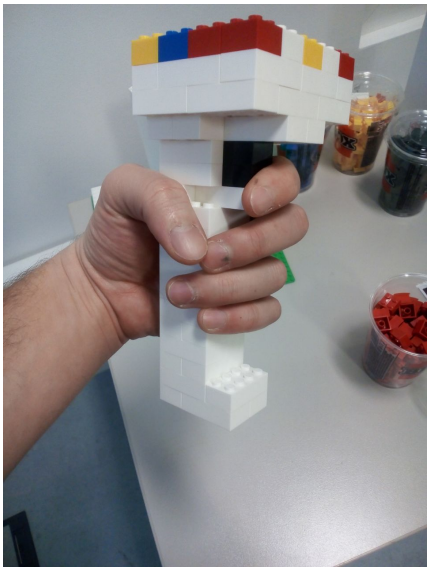
Vuforia Image-Tracking.

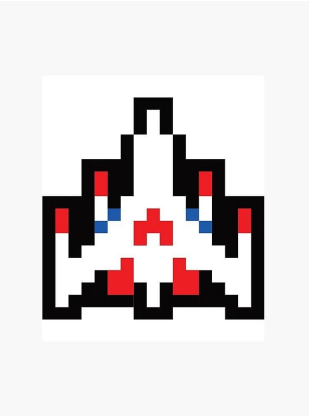


Erster Space Invader Prototype, für das Testen von Color-Tracking.

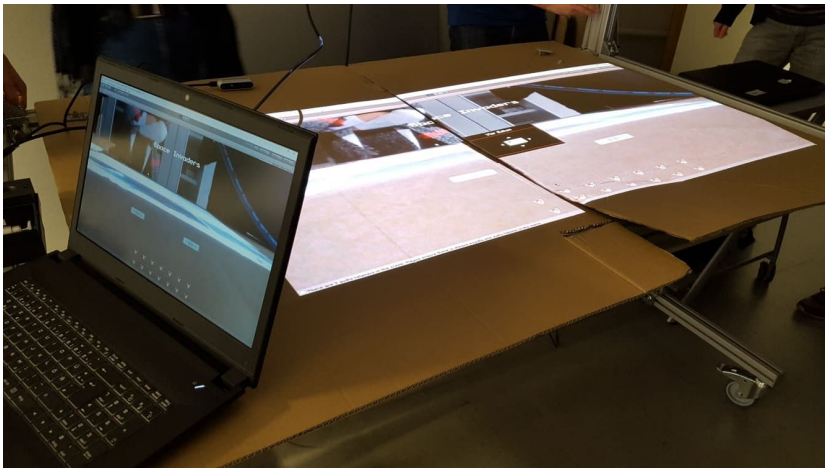
Lego-Tangible-Prototype für Space Invader.







Beamer Konfiguration



8. Zusammenfassung

Dieses Dokument stellt einen Überblick über das erste Semester des Bachelor-Projekts “Mixed Reality” des Studiengangs Medieninformatik, Hochschule Bremen, dar.

Im Laufe des ersten Semesters wurden drei Prototypen entwickelt, um mit der Hard- und Software vertraut zu werden und Möglichkeiten sowie Grenzen der verwendeten Technologien zu testen. Dabei entstanden Prototypen, die im Bereich des Game Developments zu verorten sind: Pong MR, Space Invaders und Holo-Cubes.

Pong MR nutzt für das bekannte Videospiel Pong einen eigens für das Projekt entwickelten Tisch mit Beamer sowie die Intel Realsense Depth Camera und ermöglicht ein Spielen mit farbigen Karten statt mit Controllern auf der Tischoberfläche.

Space Invaders arbeitet mit der selben Hardware wie Pong MR und trackt mit der SDK Vuforia Bewegungen eines Controllers, der an der entsprechenden Stelle im Spielfeld gehalten wird, sodass der Spieleklassiker ebenfalls auf dem Tisch gespielt werden kann.

Holo-Cubes ist eine Anwendung für die Hololens, in der Marker im realen Raum erkannt werden, die das Erstellen von unterschiedlich farbigen virtuellen Cubes ermöglichen. Ein Feedback auf Aktionen der nutzenden Person, beispielsweise das Erstellen von Cubes in einer korrekten Reihenfolge, ermöglicht die Umsetzung von unterschiedlichen Spielen.

Beschreibungen der grundlegenden Themen des Bereichs Mixed Reality (u.a. Projection-based-AR, Computer Vision, Tangible User Interface) sowie der konkret verwendeten Technologien (u.a. 3D Druck) und Hardware geben einen guten Einblick in das Feld, in dem sich die Prototypen und das folgende Projekt verorten lassen.

Quellenverzeichnis

Gestenbasierte Interaktion

http://www.billbuxton.com/input14.Gesture.pdf
https://www.cs.rutgers.edu/~vladimir/pub/pavlovic97pami.pdf
https://mediatum.ub.tum.de/doc/1138300/743052.pdf
Letzter Aufruf: 23.01.2020 21:40

Tangible User Interfaces

http://delivery.acm.org/10.1145/1350000/1349026/communications200806-dl.pdf?ip=195.37.177.70&id=1349026&acc=ACTIVE%20SERVICE&key=2BA2C432AB83DA15%2E9428DC60D3406798%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1574414414_bbbf3cb995cf25f713e4e6c36fa821ed#page=34
https://www.sciencedirect.com/science/article/pii/B9780124166912000064
https://ideum.com/spec-sheets/tangible-user-interfaces-TUI-UX-UI-v6.pd
https://eprints.soton.ac.uk/270958/1/vvg.pd
https://www.researchgate.net/publication/262398484_Combinatorix_a_Tangible_User_Interface_that_Supports_Colaborative_Learning_of_Probabilitie
Letzter Aufruf: 19.02.2020 16:05

Intel Realsense

https://www.intelrealsense.com/depth-camera-d435/
https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/RealSense_Multiple_Camera_WhitePaper.pdf
Stand: 19.02.2020 2:56

Realsense-Halterung

https://www.thingiverse.com/thing:3420218

3D-Druck

https://ultimaker.com/de/software/ultimaker-cura

Unity

https://blog.theknightsofunity.com/7-ways-keep-unity-project-organized/
https://www.gamasutra.com/blogs/HermanTulleken/20160812/279100/50_Tips_and_Best_Practices_for_Unity_2016_Edition.php
Stand: 17.02.2020 10:55

RealSense SDK

https://github.com/IntelRealSense/librealsense
https://github.com/IntelRealSense/librealsense/tree/development/wrappers/unity
Stand: 17.02.2020 10:55

Sonification (Stand: 18.02.2020)
http://www.zfdg.de/2018_002
http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmi_ws0506/essays/uebung2-pollner.html
https://www.youtube.com/watch?v=-hY9QSdaReY

Projection-based-AR: (Stand: 18.02.2020 14:42)

https://www.youtube.com/watch?v=Df7fZAYVAIE&t=112s
https://www.youtube.com/watch?v=_30dN1rWzh0
https://www.youtube.com/watch?v=x4sVTUAdoLg

https://omnia360.de/blog/die-entwicklung-von-augmented-reality/

https://www.springerprofessional.de/fertigungstechnik/vr-technologien/komfortabel-produzieren-mit-augmented-reality/7069808

https://www.elektroniknet.de/markt-technik/industrie-40-iot/neue-augmented-reality-methode-loest-datenbrille-ab-124932.html

https://www.ferchau.com/de/de/blog/details/24-05-2016-gute-aussichten-ar-projektionen-statt-datenbrille

https://apps.dtic.mil/dtic/tr/fulltext/u2/a606245.pdf

https://www.researchgate.net/publication/220476239_Merging_Fossil_Specimens_with_Computer-Generated_Information

https://www.mitpressjournals.org/doi/abs/10.1162/105474605775196562

https://www.researchgate.net/publication/30868226_What%27s_Real_About_Virtual_Reality

https://www.researchgate.net/figure/Basic-workflow-of-an-AR-application-using-fiducial-marker-tracking_fig1_216813818

https://ofmechanicsandcode.wordpress.com/2014/12/26/project-4-augmented-reality-instruction-instructar/

Prototyping: (Stand: 16.02.2020 20:59)

https://www.gruenderszene.de/lexikon/begriffe/prototyping
https://www.gruenderszene.de/lexikon/begriffe/throwaway-prototyping

https://www.gruenderszene.de/lexikon/begriffe/rapid-prototyping

https://www.gruenderszene.de/lexikon/begriffe/throwaway-prototyping

https://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Prototyping/index.html

https://www.lead-innovation.com/blog/prototyping-methoden-und-best-practices

https://www.dev-insider.de/was-ist-software-prototyping-a-794959/

https://webdesign.tutsplus.com/articles/storyboarding-vs-prototyping-when-to-use-each--cms-28707

Mixed Reality Toolkit

https://microsoft.github.io/MixedRealityToolkit-Unity/README.html#ui-andinteraction-building-blocks

https://medium.com/@dongyoonpark/open-source-building-blocks-for-windows-mixed-reality-experiences-hololens-mixedrealitytoolkit-28a0a16ebb61

https://docs.microsoft.com/de-de/windows/mixed-reality/spatial-mapping

https://microsoft.github.io/MixedRealityToolkitUnity/Documentation/SpatialAwareness/SpatialAwarenessGettingStarted.html

https://microsoft.github.io/MixedRealityToolkitUnity/Documentation/README_ObjectCollection.html

Verantwortlichkeiten

Moumita Ahmad	Tischaufbau Hololens (mit Mixed Reality Toolkit) Holo-Cubes
Annie Berend	Tangible User Interfaces Holo-Cubes Zusammenfassung
Moritz Hodler	Hardware - Intel RealSense (Teilweise Überarbeitet, teilweise selbst geschrieben)
Robin Jacobse	Gestenbasierte Interaktion

	Realsense-Halterung Space Invaders
Reda Khalife	Projection-based-AR Griffhalterung der Marker
Bjarne Lehmkuhl	
Philipp Moritzer	Software - Unity Software - Intel RealSense SDK (Pong - Text nicht verfasst, aber mit erarbeitet)
Joscha Sattler	Projection-based-AR, Vuforia, Unterstützung beim Tischbau, Space-Invaders Multiplayer, Fotos
Tobi Schmitt	Sonification Pure Data Open Sound Control Protocol Soundsystem
Alisa Schumann	Prototyping Ausblick
Leonard Tuturea	Abstract Intel Realsense Computer Vision Pong MR