

Game Design:  
Anforderungen von Mixed Reality mit  
Unbekannter Spielumgebung

Hausarbeit im Studiengang  
Game Design  
-  
Vorlesung  
Forschung und Technik

vorgelegt von:

Anja Haumann

GD1014



## **Abstrakt**

Diese Arbeit befasst sich mit individuellen Anforderungen von Mixed Reality an das Game Design. Im Gegensatz zu den etablierten Plattformen haben Mixed Reality Anwendungen das Problem, dass sich das Gameplay auf die unbekannte Umgebung des Users einstellen muss und keinerlei räumliche Begebenheiten während der Entwicklung bekannt sind.

Um dieses Problem zu lösen werden Mindestanforderungen an die Umgebung gestellt, Spielorte vorgegeben oder das Gameplay dynamisch an die Umgebung angepasst. Dabei soll untersucht werden, wie man mit möglichst wenig Einschränkungen der Spielumgebung auskommen kann.

Im Vorfeld werden die verschiedenen Methoden zur Eingrenzung der Umgebung in Mixed Reality Games an Hand von Beispielen erarbeitet und verglichen. Anschließend wird darauf eingegangen, wie und ob man komplett ohne Beschränkung der Spielumgebung auskommen kann. Im Hinblick darauf werden Techniken aus dem Bereich der prozeduralen Generierung untersucht.

Ziel der Arbeit ist eine Gegenüberstellung verschiedener Lösungswege für das Umgebungsproblem in Mixed Reality Anwendungen zu erarbeiten. Die Ergebnisse können dabei auch potenziell auf Serious-Anwendungen oder VR-Game Design übertragen werden.



# Inhaltsverzeichnis

Abstrakt.....	3
1. Problemstellung, Zielsetzung und Forschungsfrage .....	1
2. Anpassen der Umgebung an das Spiel .....	2
2.1 Räumliche Anforderungen .....	3
2.1.1 Freiraum .....	3
2.1.2 Oberflächen .....	4
2.1.3 Fazit.....	5
2.2 Marker Tracking .....	6
2.2.1 Dynamisch .....	6
2.2.2 Statisch .....	7
2.2.3 Fazit.....	8
2.3 Vorgabe der Spielorte .....	9
3. Anpassen des Spiels an die Umgebung.....	10
3.1 Limitationen von Prozeduraler Generierung.....	10
3.2 Optimieren des Raums für die prozedurale Generierung.....	11
4. Fazit.....	13
4.1. Fragenkatalog zur Lösung des Umgebungsproblems .....	14
4.2. Ausblick .....	15
Literaturverzeichnis .....	16
Spieleverzeichnis .....	18
Abbildungsverzeichnis.....	18
Tabellenverzeichnis .....	19



## 1. Problemstellung, Zielsetzung und Forschungsfrage

Mit dem Erscheinen des Augmented-Reality Spiels *Pokémon Go* (Niantic, 2016) im Juli 2016 ist eine große Begeisterung für Spiele mit Elementen aus der realen Welt aufgekommen. Nach einem Jahr ist die Begeisterung der breiten Bevölkerung für *Pokémon Go* abgeklungen, aber das Interesse an neuen Augmented-Reality und Virtual-Reality Anwendungen ist geblieben (Grabmann, 2016; Jansen, 2017). Daraus folgend interessieren sich immer mehr Entwickler für die neuen Möglichkeiten und Probleme bei einer Verbindung von Realität und virtuellen Inhalten. Dieser Bereich wird übergeordnet als Mixed-Reality (In Zukunft abgekürzt als MR) bezeichnet und umfasst laut Paul Milgram und Fumio Kishino alle Abstufungen im Realität-Virtualität Kontinuum zwischen unserer Realität und virtuellen Realitäten, vergleiche Abbildung 1 (Milgram, et al., 1994).

Dabei unterscheidet man zwischen Augmented-Reality (In Zukunft abgekürzt: AR) und Augmented-Virtuality (In Zukunft abgekürzt: AV). AR Anwendungen zeichnen sich dadurch aus, dass sie die Realität mit virtuellen Objekten erweitern (Spacey, 2016). Bei AV Spielen hingegen werden Objekte aus der realen Welt genutzt um die Virtuelle zu erweitern (Spacey, 2016). Bei allen Bereichen der MR wird also die reale Umgebung des Spielers mit in das Spielerlebnis mit eingebunden. Dies führt dazu, dass die variable Umgebung des Spielers beim Game Design des MR Spiels mit beachtet werden muss. Es ist während der Entwicklung nicht bekannt, wie die Räume des Spielers aussehen oder ob er in einem Gebäude spielen wird. Somit

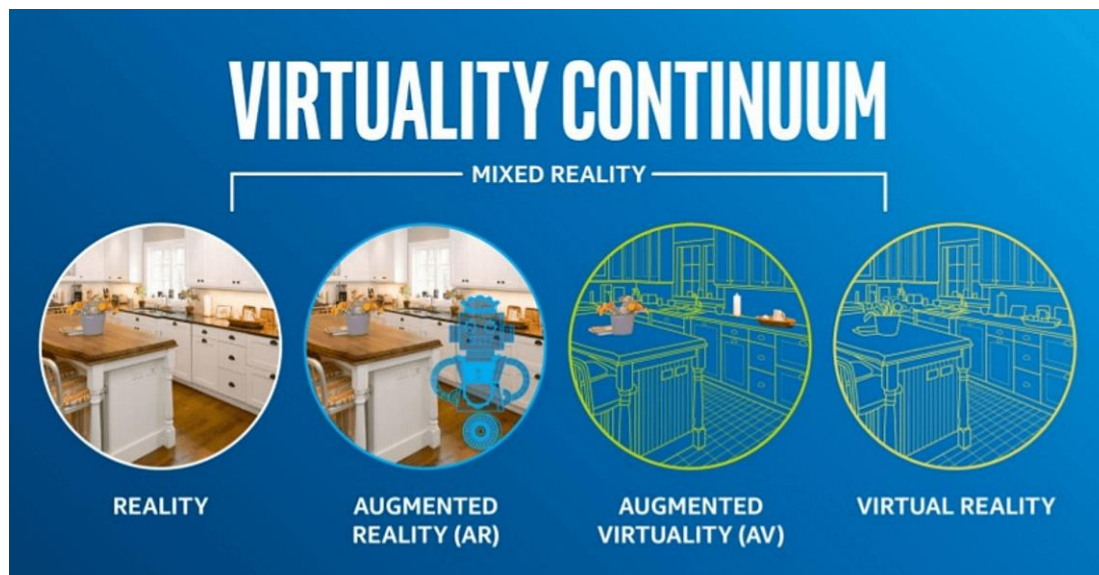


Abbildung 1: Realität-Virtualität Kontinuum nach Milgram. Quelle: Valoriani, Matteo. 2016.

können keinerlei Begebenheiten der Spielumgebung als gegeben betrachtet werden (Barbagallo, 2015). Das MR Spiel muss in einem kleinen Apartment genauso gut funktionieren wie in einer Villa oder in der freien Natur. Wobei sich das Spiel an die unterschiedlichsten Voraussetzungen wie Grundrisse, Deckenhöhen und Möbel anpassen muss (Lang, 2017). Diese Problematik wird im weiteren Verlauf als Umgebungsproblem bezeichnet.

Im Verlauf dieser Arbeit werden die verschiedenen Möglichkeiten mit dem Umgebungsproblem umzugehen erarbeitet und verglichen. Außerdem wird prozedurale Generierung als Mittel, um das MR Spiel an möglichst viele verschiedene Umgebungen anzupassen, untersucht. Womit sich die Frage stellt, welche Möglichkeiten es gibt mit dem Umgebungsproblem umzugehen und welche Vor- und Nachteile diese Möglichkeiten haben. Es wird keine allgemeingültige Lösung für das Umgebungsproblem bei MR Spielen vorgestellt, sondern ein Überblick über die verschiedenen Möglichkeiten gegeben, mit diesem Problem im Game Design umzugehen. Zusätzlich wird ein Fragen-Katalog erarbeitet, um das Finden der am besten geeigneten Lösungsstrategie für das betreffende Spiel zu ermöglichen.

## **2. Anpassen der Umgebung an das Spiel**

Die Umgebung kann an das Spiel angepasst werden oder das Spiel an die Umgebung. Dabei liegt der hauptsächliche Aufwand bei der Lösung des Umgebungsproblems entweder beim Spieler, welcher seine Umgebung an die Anforderungen anpassen muss, oder beim Game Designer, der das Spiel an die unbekannte Umgebung anpassen muss. Im Hinblick darauf ist es für das Game Design mit wenig bis gar keinem Aufwand verbunden Anforderungen an die Umgebung zu stellen. Da dies dafür sorgt, dass die für das Spiel benötigten Umgebungsmerkmale während der Entwicklung als gegeben betrachtet werden können. Diese Anforderungen können in verschieden starkem Maße durch Mindestanforderungen an die Umgebung, die Vorgabe der Spielorte oder das Tracken von Objekten erreicht werden. Dabei bergen Anforderungen allerdings das Risiko, dass der Spieler nicht in der Lage ist die nötigen Umgebungsverhältnisse herzustellen um das Spiel spielen zu können. Dies kann potenziell die Zielgruppe des Spiels weiter auf Personen mit den passenden Räumlichkeiten eingrenzen. Dies muss insbesondere bei der Wahl der Anforderung in Kombination mit der Zielgruppe beachtet werden. Außerdem könnte es dem Spielerlebnis abträglich sein, wenn der Spieler vor jeder neuen Spielsession erst einmal eine gewisse Zeit seine Umgebung umbauen muss, was als Nachteil gewertet werden könnte. Hier muss man abwägen, inwieweit man das Umsetzen der Anforderungen dem Spieler zumuten kann ohne eine zu große Einstiegshürde zu schaffen.



## 2.1 Räumliche Anforderungen

Insbesondere bei AR-Spielen benötigt man einen Platz in der Umgebung des Spielers an dem man die Spielwelt oder deren Elemente platzieren kann. Um sicher zu stellen, dass ein solcher Platz in der Umgebung existiert, können räumliche Anforderungen gestellt werden. Dazu zählen Anforderungen an Freiräume und Oberflächen. Diese gehören zu dem im Game Design am schwierigsten abzuschätzenden Anforderungen was die Umsetzbarkeit auf der Seite des Spielers angeht. Zentral dabei ist, wie groß die Freiräume und Oberflächen sind zu denen die gewählte Zielgruppe Zugang hat. Hierbei gilt, weniger ist mehr. Bei einer durchschnittlichen Wohnfläche von ca. 44,6 m<sup>2</sup> pro Person (Statistisches Bundesamt Deutschland, 2017) ist 1 m<sup>2</sup> Freiraum oder Oberfläche noch zu finden, 10 m<sup>2</sup> aber wahrscheinlich nicht mehr<sup>1</sup>. Wenn man jedoch größere Anforderungen an die Umgebung stellen muss, sollte man darauf achten, dass das Spiel im Freien spielbar ist. Eine für eine Wohnung zu große Quadratmeteranzahl ist wenig im Vergleich mit einem großen Parkplatz oder einem Fußballfeld. Im Folgenden werden als Beispiele für räumliche Anforderungen die Spiel *ARise* (Climax Studios, 2017) und *The Machines* (Directive Games, 2017) untersucht.

### 2.1.1 Freiraum

Als Freiraum wird ein 3D-Raum in der Umgebung des Spielers bezeichnet, der bis zur Decke des Raumes reicht, komplett frei von Möbelstücken ist und vollständige Bewegungsfreiheit für den Spieler garantiert. Eine Freiraum-Anforderung hat für das Game Design den Vorteil, dass man davon ausgehen kann, dass es einen komplett leeren 3D-Raum der geforderten Größe gibt, den man frei gestalten kann. Von diesem nutzbaren Freiraum muss man allerdings noch einen Laufbereich für den Spieler abziehen, welcher ebenfalls frei von Hindernissen sein muss. Dieser dient der Sicherheit des Spielers, da der unachtsame Spieler ansonsten riskiert mit Möbeln in seinem Laufweg zu kollidieren. Dies kann für ihn zu Verletzungen führen oder dafür sorgen, dass der Spieler zu sehr auf seine Umgebung achten muss und sich nicht mehr in ausreichendem Maße mit dem Spiel beschäftigen kann. Dies ist in beiden Fällen abträglich für das Spielerlebnis und sollte vermieden werden indem man bei einer Freiraum-Anforderung den Laufbereich für den Spieler mit einberechnet. Wenn man zum Beispiel 1 m<sup>2</sup> Freiraum für das Spiel benötigt sollte man noch einen weiteren ½ m<sup>2</sup> für den Laufbereich dazu rechnen. Sollte

---

<sup>1</sup> „Als Repräsentativraum ist das Wohnzimmer naturgemäß der flächenmäßig größte Raum [...]“ (Wohnung.com, o.J.) dabei wird „[...] etwa ein Drittel der gesamten Wohnfläche für das Wohnzimmer verplant“ (Wohnung.com, o.J.). Ein Wohnzimmer entspricht somit bei 44,6m<sup>2</sup> Wohnraum 14,9 m<sup>2</sup>. Davon sind die Wohnzimmermöbel noch abzuziehen. 14,9m<sup>2</sup> – 9 m<sup>2</sup> (2 m<sup>2</sup> Schrank, 2m<sup>2</sup> Fernseher, 1m<sup>2</sup> Wohnzimmertisch, 4m<sup>2</sup> Sitzmöbel, etc.) = 5,9 m<sup>2</sup> Freiraum

der Spieler dazu aufgefordert werden ausladende Bewegungen zu machen, sollte der Laufbereich noch einmal vergrößert werden. So wird der geforderte Freiraum für den Spieler größer und potenziell schwerer zu finden, schützt ihn aber vor Verletzungen und gewährleistet seine uneingeschränkte Aufmerksamkeit für das Spiel. Ein Spiel das Freiraum-Anforderungen nutzt um das Umgebungsproblem zu lösen ist das AR-Handyspiel *ARise* für IOS. In *ARise* wird der Spieler zu Beginn aufgefordert die Spielwelt in einem Freiraum zu platzieren, in der von diesem Zeitpunkt an das komplette Spielgeschehen stattfindet. Auf einer fliegenden Insel muss man nun einen kleinen Ritter über Abgründe bis zur Spitze der Insel führen. Dabei werden die auftretenden Rätsel durch das Ändern der Perspektive des Spielers gelöst. Dies hat zur Folge, dass der Spieler sich konstant um die Insel herumbewegt und teilweise seine Sichthöhe ändern muss. In *ARise* wird der Laufbereich des Spielers nicht mit in die Freiraum-Anforderung mit einberechnet. „Es empfiehlt sich, dafür einen Ort mit relativ viel Platz auszusuchen [...] damit man sich ohne Hindernisse frei bewegen kann“ (Mel, 2017). Dies führt dazu, dass der Spieler selber dafür sorgen muss, dass er den nötigen Laufbereich um den Freiraum schafft, was er aber unter Umständen nicht tun wird. In *ARises* Game Design wurde der Laufweg des Spielers nicht mit beachtet, aber die Größe der Spieler. Es ist möglich die Höhe der fliegenden Insel auf die eigene Größe anzupassen (Mel, 2017). Dies ist in einem Spiel bei dem man durch die Perspektive auf die Insel spielt ein entscheidendes Feature. Jeder Spieler kann es auf seine Augenhöhe einstellen und Personen die körperlichen Probleme haben können die Höhe der Freiraum-Anforderung nutzen um die Insel nach oben zu schieben anstatt sich zu bücken um die richtige Perspektive zu erreichen.

Eine Freiraum-Anforderung bedeutet somit nicht, dass die Verantwortung für die Umgebung komplett an den Spieler abgegeben werden kann. Das Game Design ist weiterhin für die Sicherheit der Spielumgebung, in Form von freien Laufwegen, verantwortlich und der Möglichkeit das Spiel im 3D-Raum zum Beispiel an die Augenhöhe des Spielers anzupassen.

### **2.1.2 Oberflächen**

Oberflächen sind das zweidimensionale Äquivalent einer Freiraum-Anforderung. Eine Oberflächen-Anforderung unterscheidet sich dadurch von einer Freiraum-Anforderung, dass hauptsächlich eine ebene Fläche gefordert ist. Spiele mit Oberflächen-Anforderung können wie das MR Äquivalent eines Brettspieles behandelt werden. Sie können genauso auf Tischen, Böden und anderen ebenen Oberflächen gespielt werden. Je nach Spiel ist auch hier ein Laufbereich für den Spieler um die Oberfläche notwendig. Ein Beispiel dafür ist das AR-Handyspiel *The Machines*. Bei diesem Spiel wird die Spielwelt in Form eines Terrains auf die

geforderte Oberfläche projiziert. Auf dieser Oberfläche kämpft man mit mehreren Einheiten um die Eroberung der gegnerischen Basis. Um alle Teile des Spielfeldes einzusehen ist es notwendig um das Spielfeld herumzugehen und sich zum Spielfeld herunter zu beugen. Dies wird zu einem Hindernis für Spieler mit körperlichen Problemen, falls auf dem Boden gespielt wird. Dies ist nötig sobald die geforderte Oberflächen-Anforderung die Größe eines durchschnittlichen Esstisches übersteigt.

### 2.1.3 Fazit

Zusammenfassend kann man sagen, dass räumliche Anforderungen an Freiräume oder Oberflächen das Umgebungsproblem komplett lösen, da das Spiel sich seinen eigenen Raum in der Umgebung des Spielers schafft. Die bestehende Umgebung wird nicht mit in das Spiel eingebunden. Bei diesen Ansätzen ist es allerdings fraglich, ob die Spiele durch die MR-Komponente einen Mehrwert erhalten. Ein Beispiel bei dem das nicht der Fall ist, ist das oben genannte Spiel *The Machines*. Es wird im Game Design nur die Kamerasteuerung durch die Bewegung des Spielers ersetzt und keine weiteren Vorteile von MR genutzt. Bei dem Spiel *ARise* hingegen ist die Perspektive des Spielers auf die Spielwelt das zentrale Gameplayelement und nutzt somit die Bewegung des Spielers um das Gameplay immersiver zu machen. Des Weiteren sind auch Kombinationen der räumlichen Anforderungen denkbar. Ein Beispiel dafür ist das Spiel *Robo Raid* (Microsoft Corporation, 2016) für HoloLens. Bei diesem Spiel wird eine Wandoberfläche für das Erscheinen von Robotern benötigt, aber auch gleichzeitig ein Freiraum vor dieser Wand den der Spieler nutzt um den Angriffen der Roboter auszuweichen und sie abzuschießen. Alle erarbeiteten Vor- und Nachteile der verschiedenen räumlichen Anforderungen sind in der Tabelle 1 noch einmal zusammengefasst.

#### Mindestanforderungen

<i>Vorteile</i>	<ul style="list-style-type: none"> <li>- Vollständige Lösung des Umgebungsproblems</li> <li>- Einfach umzusetzen im Game Design</li> </ul>	
<i>Nachteile</i>	<ul style="list-style-type: none"> <li>- Nur im beschränktem Maße verfügbar</li> <li>- Potenzielle Einstiegshürde für den Spieler</li> </ul>	
	<b>Freiraum</b>	<b>Oberflächen</b>
<i>Vorteile</i>	- Komplette freier 3D-Raum	- Komplette freie 2D-Fläche
<i>Nachteile</i>	- Benötigt freien Laufbereich um die geforderte Umgebung	- Potenziell problematisch für Spieler mit körperlichen Problemen

**Tabelle 1:** Zusammenfassung und Gegenüberstellung aller genannten Vor- und Nachteile von Mindestanforderungen in Form von Freiräumen und Oberflächen. Quelle: Selbst erstellte Grafik

## 2.2 Marker Tracking

Marker basierte AR ist ein Teilbereich der AR bei dem sogenannten Marker benutzt werden um die Position und Orientierung der Virtuellen Objekte in der Physischen Welt zu erkennen (Levski, o.J.). Dabei werden die Spielinhalte auf die Marker-Karten projiziert. Ein großer Vorteil von Marker-Karten ist, dass sie technisch einfacher vom Spiel erfasst werden können als die Unbekannte Umgebung des Spielers (Levski, o.J.). Daraus folgend benötigt man die Marker-Karten um das Spiel spielen zu können. Diese stehen auch in Verbindung mit einem der größten Nachteile von Marker basierten AR-Spielen. Es werden gute Lichtverhältnisse benötigt, damit die Marker-Karten problemlos von der Kamera der Spielplattform erkannt werden können. Dabei können Lichtverhältnisse vom Spieler unter Umständen nicht in größerem Maß verändert werden. Außerdem wird bei manchen Spielen ebenfalls eine ebene Oberfläche benötigt, auf der die Karten abgelegt und bewegt werden können. Dies ist aber nicht bei allen Marker basierten AR-Spielen der Fall. Aus diesem Grund unterscheiden wir zwischen statischem und dynamischem Marker basierten MR-Spielen. Als Beispiel für ein statisches Marker Tracking dient das Handyspiel *Genesis Augmented Reality* (Genesis Augmented Reality, 2016) und als Beispiel für dynamisches Marker Tracking das PS Vita Spiel *PulzAR* (XDEV Studio Europe, 2012).

### 2.2.1 Dynamisch

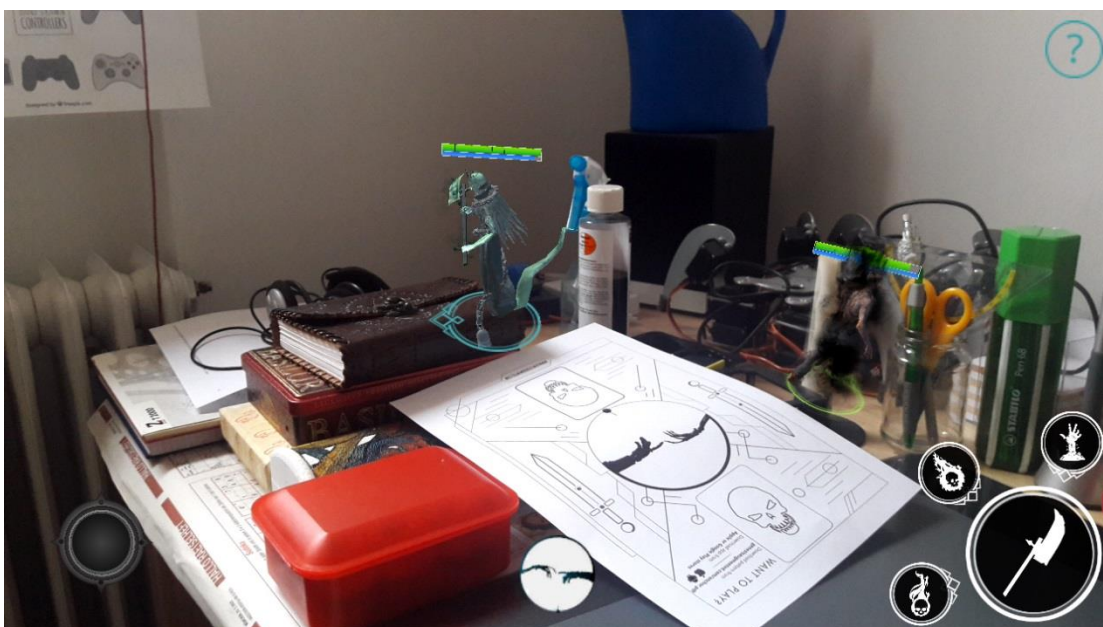
Als dynamisch werden Marker basierte AR-Spiele bezeichnet deren Marker während des Spielens bewegt werden müssen. Eines dieser Spiele ist das Rätselspiel *PulzAR* für PS Vita. Ziel des Spiels ist es einen Laser mithilfe der auf den Marker-Karten erscheinenden Spiegel so umzuleiten, dass er den Auslöser der Raketenstartrampe erreicht. Der Spieler startet das Spiel indem er die Marker-Karte der Raketenbasis vor sich auf einer geraden Oberfläche platziert. Anschließend wird ein Level mit Hindernissen und Bonusmünzen um den Marker herum erscheinen. Mit jeder zusätzlichen Marker-Karte die der Spieler vor sich platziert erscheint ein Spiegel. Die Spiegel können über die Bewegung der Marker-Karten im Level bewegt werden, vergleiche Abbildung 2. Dynamische Marker basierte AR-Spiele gehen wie Mindestanforderungen an Oberflächen mit dem Umgebungsproblem um, benötigen aber zusätzlich die Anwesenheit von Marker-Karten um die Spielinhalte zu platzieren. Dies ist eine weitere Einstiegshürde, gibt dem Spieler aber mehr Kontrolle darüber, wo die Spielinhalte auf seiner Oberfläche platziert werden und bieten eine zusätzliche Ebene um mit den virtuellen Inhalten zu interagieren, wie es im Beispiel *PulzAR* der Fall ist.



**Abbildung 2:** Platzierung einer Marker-Karte in dem PS Vita Spiel *PulzAR* um den Laser durch einen Spiegel in Richtung Raketenbasis abzulenken. Quelle: Vincent, Ryan. 2012

### 2.2.2 Statisch

Statische Marker basierte AR-Spiele zeichnen sich dadurch aus, dass die Marker während des Spielens nicht bewegt werden müssen. Als Beispiel dient hier das Handyspiel *Genesis Augmented Reality* (In Zukunft abgekürzt als *Genesis*). Bei diesem Spiel wird zu Beginn eine Marker-Karte in der Umgebung des Spielers platziert, welche als Ankerpunkt für die Spielcharaktere dient. Auf dem Marker erscheint die Spielfigur, welche über das Handy durch



**Abbildung 3:** Beispiel für die Bewegung des Spielercharakters auf einer unebenen Oberfläche, in dem Spiel *Genesis*, außerhalb der weißen Marker-Karte. Quelle: Selbst erstellte Grafik

die Umgebung des Spielers bewegt werden kann. Dabei ist keine gerade Oberfläche notwendig. Die Figuren benutzen die Ausrichtung der Marker-Karte als Richtwert für den Winkel des Bodens. Dabei bewegen sie sich scheinbar gerade über unebene Oberflächen, vergleiche Abbildung 3. Das Spiel ist in jeder Umgebung spielbar, in der die Marker-Karte ausgelegt werden kann. Somit gehen statische, Marker basierte Spiele anders mit dem Umgebungsproblem um als Dynamische. Es wird keine Mindestanforderung an die Oberfläche gestellt. Es ist von Vorteil auf einer ebenen Oberfläche zu spielen, aber es ist nicht notwendig, wie man in Abbildung 3 sehen kann.

### 2.2.3 Fazit

Je nachdem welche Art von Marker Tracking man in einem AR-Spiel benutzt, verstärken sich die Anforderungen an die Umgebung, vergleiche dynamisches Marker Tracking, oder verringern sich, vergleiche statisches Marker Tracking. In jedem Fall benötigt der Spieler aber gute Lichtverhältnisse und die Marker-Karten. Diese bilden den Hauptunterschied zu den räumlichen Anforderungen, da bestimmte Objekte in der Umgebung, die Marker-Karten, vorausgesetzt werden. Alle erarbeiteten Vor- und Nachteile der verschiedenen Marker basierten Methoden zur Lösung des Umgebungsproblems sind in der Tabelle 2 noch einmal zusammengefasst.

#### Marker Tracking

<i>Vorteile</i>	<ul style="list-style-type: none"> <li>- Vollständige Lösung des Umgebungsproblems</li> <li>- Technisch einfacher umzusetzen<sup>2</sup></li> <li>- Der Spieler hat mehr Kontrolle über die Platzierung der Spielinhalte</li> </ul>	
<i>Nachteile</i>	<ul style="list-style-type: none"> <li>- Benötigt gute Lichtverhältnisse</li> <li>- Benötigt Marker-Karten</li> </ul>	
	<b>Statisch</b>	<b>Dynamisch</b>
<i>Vorteile</i>	<ul style="list-style-type: none"> <li>- Benötigt keine ebene Oberfläche</li> </ul>	<ul style="list-style-type: none"> <li>- Zusätzliche Inputebene durch die Bewegung der Marker-Karten</li> </ul>
<i>Nachteile</i>		<ul style="list-style-type: none"> <li>- Benötigt ebene, freie Oberfläche</li> </ul>

**Tabelle 2:** Zusammenfassung und Gegenüberstellung aller genannten Vor- und Nachteile von Marker basierten AR-Spielen in Form von statischem und dynamischem Marker Tracking. Quelle: Selbst erstellte Grafik

<sup>2</sup> Im Vergleich mit der Erkennung unbekannter Umgebungen.

## 2.3 Vorgabe der Spielorte

Die Vorgabe der Spielorte ist ein Konzept das besonders erfolgreich in dem Handyspiel *Pokémon Go* umgesetzt wurde. In diesem AR-Spiel bewegt man sich, indem man in der realen Welt läuft wobei man über GPS verfolgt und darüber im virtuellen Äquivalent der realen Welt bewegt wird. Dabei ist das Ziel des Spiels zufällig in der Umgebung auftauchende Pokémon zu fangen und mit Pokéstops und Arenen zu interagieren. Die Pokéstops und Arenen sind dabei an festen Orten in der Realität verankert und geben somit die Spielorte vor, an denen mit diesen besonderen Gebäuden interagiert werden kann. Dabei bildet diese Spielorte in Kombination mit Google Maps eine Umgebungseinschränkung die einer Anpassung des Spiels an die Umgebung schon sehr nah kommt. *Pokémon Go* ist überall auf der Welt spielbar, vorausgesetzt der Ort kann von einem GPS-Signal erreicht werden. Daneben werden die vom Spiel vorgegebenen Orte, Pokéstops und Arenen, in der Nähe benötigt. Dies bildet auch einen entscheidenden Nachteil für Spiele mit vorgegebenen Spielorten. Es kommt drauf an, wie umfangreich die Datenbank der Spielorte ist um das Spiel auf der ganzen Welt mit gleichbleibender Qualität spielen zu können (Barbagallo, 2016).

Die Verfolgung der Spielerposition über GPS bildet zurzeit noch einen größeren Nachteil. Bei Handys beläuft sich die durchschnittliche GPS-Genauigkeit auf ca. 1 - 3 Meter, was durch eine schlechte Verbindung, kein direkter Sichtkontakt zu den Satelliten, noch weiter sinken kann (Katzenberg, 2017). Dies führt auch zu sehr schlechten Ortungsgenauigkeiten in Gebäuden. Die insgesamte GPS-Ungenauigkeit sollte allerdings in einigen Jahren kein Problem mehr darstellen. Ab 2018 werden in Handys neue GPS-Chips verbaut, die eine Ortung auf bis zu 30 cm ermöglichen sollen (futurezone, 2017). Wenn sich diese Handys weit genug verbreitet haben kann der Nachteil der GPS-Ungenauigkeit als nicht mehr existent betrachtet werden. Außerdem wird die Problematik der GPS-Ungenauigkeit in Gebäuden in naher Zukunft gelöst, da Google an einer so genannten VPS-Ortung (Visual Positioning Service) arbeitet, welche eine Ortung in Gebäuden mit einer Abweichung von wenigen Zentimetern ermöglichen soll (Google Developers, 2017).

Ein weiterer Punkt ist, dass unsere Welt sich im konstanten Wandel befindet. Womit alle Änderungen der realen Welt laut Jin Ha Lee auch in der virtuellen Welt stattfinden müssen, um Diskrepanzen zwischen den beiden Welten zu vermeiden und die Immersion nicht zu stören (Lee, et al., 2017). Dazu zählen das Beseitigen von vorgegebenen Spielorten die in der Realität nicht mehr existieren (Lee, et al., 2017), sowie die Anpassung von Straßennetzen und anderen

Wandlungen mit Auswirkung auf die Spielwelt. Alle Vor- und Nachteile von vorgegebenen Spielorten sind in der Tabelle 3 noch einmal zusammengefasst.

### Vorgabe der Spielorte

<i>Vorteile</i>	<ul style="list-style-type: none"> <li>- Vollständige Lösung des Umgebungsproblems</li> <li>- Bei einer großen Spielort-Datenbank ist die Position des Spielers auf der Welt irrelevant</li> </ul>
<i>Nachteile</i>	<ul style="list-style-type: none"> <li>- Zurzeit noch eine GPS-Ungenauigkeiten von ca. 1-3 Metern (Ab 2018 bei Handys mit einem neuem GPS-Chip nur noch 30 cm)</li> <li>- Ortsbedingte Spielqualität ist abhängig von der Größe der Spielort-Datenbank</li> <li>- Komplette Synchronisation mit der realen Welt notwendig</li> </ul>

**Tabelle 3: Zusammenfassung aller genannten Vor- und Nachteile von vorgegebenen Spielorten. Quelle: Selbst erstellte Grafik**

## 3. Anpassen des Spiels an die Umgebung

Im Gegensatz zur Anpassung der Umgebung an das Spiel steht das Anpassen des Spiels an die Umgebung. Ein Mittel um dies zu erreichen ist die prozedurale Generierung. Da die Umgebungen der Spieler im Vorfeld nicht bekannt sind, ist es nicht möglich für jede Spielumgebung eigens Level zu designen. Um dieses Problem zu lösen kann auf prozedurale Generierung zurückgegriffen werden. Diese macht es möglich die Level auf die Umgebung des Spielers zuzuschneiden. Prozedurale Generierung stellt dabei allerdings nicht die perfekte Lösung für das Umgebungsproblem dar. Sie besitzt einige Limitationen auf die im Game Design Rücksicht genommen werden müssen. Ein Kernpunkt von prozeduraler Generierung ist dabei, dass man nicht zu 100% weiß, was der Algorithmus generieren wird (Bycer, 2016).

### 3.1 Limitationen von Prozeduraler Generierung

Laut Bycer ist eine der größten Limitationen, dass die Interaktion des Spielers mit der Welt einfach gehalten werden muss.

*“In order for procedural generation to work, the player's own interaction with the game must be kept basic. The reason is that the algorithm has to be designed to work around every mechanic and system the player has access to. The more mechanics or abilities in the player's toolkit, the harder it will be to create something that makes use of all of it.” (Bycer, 2016)*



Dies sorgt dafür, dass nicht jedes beliebig komplexe MR-Spiel auf prozedurale Level Generierung zurückgreifen kann. Umso komplexer die Interaktion des Spielers mit dem Level ist umso eher muss man Abstand von der prozeduralen Level Generierung nehmen und auf das Anpassen der Umgebung an das Spiel zurückgreifen. Außerdem ist das generierte Level stark von der physischen Umgebung des Nutzers abhängig. Dies sorgt dafür, dass der Spieler darauf achten muss, eine Spielumgebung zu wählen, die ein gutes Spielerlebnis verspricht (Riedl, 2016). Dies bedeutet das auch bei prozeduraler Generierung weiterhin der Nutzer gefordert ist seine Umgebung an das Spiel anzupassen um ein möglichst gutes Spielerlebnis zu haben. Der Nutzer wird dabei nicht in dem Maße gefordert, wie bei der Anpassung der Umgebung an das Spiel. Er muss nur darauf achten, dass seine Umgebung bestimmte Eigenschaften aufweist. Der Plattformer *Young Conker* (Microsoft Corporation, 2016) für HoloLens profitiert dabei zum Beispiel von einer Vielzahl von Oberflächen in Form von Möbelstücken. Dies sorgt durch die Höhenunterschiede der verschiedenen Oberflächen für abwechslungsreichere und interessantere Platzierung der einzelnen Level Bestandteile. Dabei werden neben den Möbeln und dem Boden auch die Wände genutzt. Dies sorgt dafür, dass auch in relativ leeren Räumen ansprechende Level generiert werden können. Dabei spielt es laut Riedl auch eine entscheidende Rolle, wie gut der prozedurale Generierungsalgorithmus implementiert wurde.

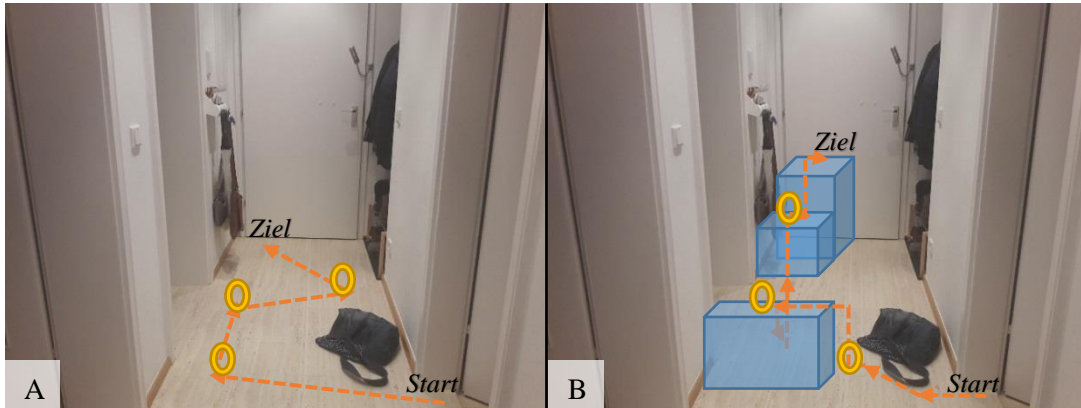
*„A good design makes for an engrossing time. A bad design, well... odds are good that your living room was not designed to be a good Mario game level or first person shooter level.”* (Riedl, 2016)

Dabei kann nicht verhindert werden, dass die Räume der Spieler unter Umständen nicht gut für die Level Generierung geeignet sind. Eine der Gegenmaßnahmen wäre das Game Design so zu verändern, dass die Spielmechaniken auf einer ebenen Fläche genauso interessant und anspruchsvoll sind, wie in einem Raum mit vielen Hindernissen.

### **3.2 Optimieren des Raums für die prozedurale Generierung**

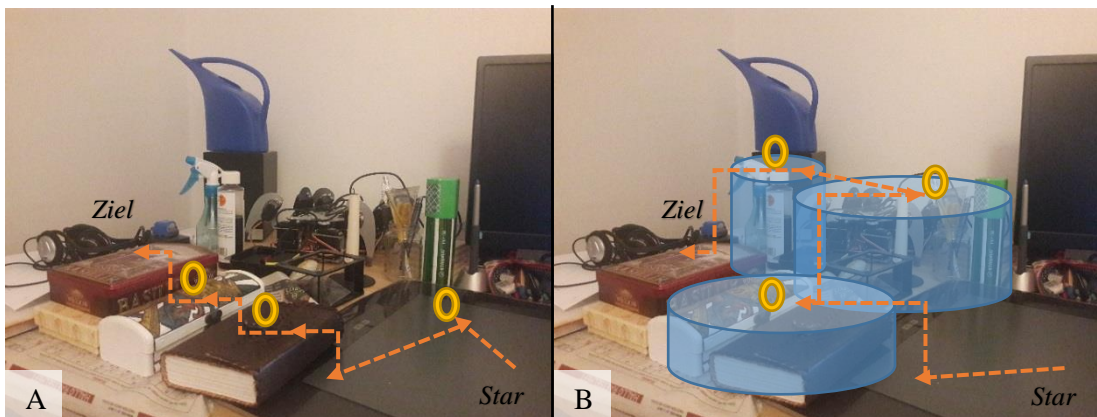
Eine andere Möglichkeit ist es den physischen Raum durch zusätzliche virtuelle Objekte hin zu einer für die Generierung besser geeigneten Umgebung zu verändern. Dazu können zusätzliche Hindernisse in leeren Räumen platziert werden oder ebene Flächen in Räumen mit vielen Hindernissen um die räumliche Komplexität zu verringern. Um diesen Umstand zu verdeutlichen dient im Folgenden ein imaginäres MR-Spiel, bei dem der Spieler einen Charakter vom Startpunkt des Levels zum Ziel bewegen muss und dabei möglichst viele goldene Ringe einsammeln soll. Ein Beispiel für das Einfügen von virtuellen Hindernissen in

leeren Räumen sieht man in [Abbildung 4](#). Im Teilbereich A sieht man ein potenzielles Level bei dem sich der Weg des Charakters aus Mangel an Hindernissen lediglich in Zick-Zack-Linien über den Boden bewegt. Dies ist kein für den Spieler interessantes Level, da keinerlei



**Abbildung 4:** Beispiel für das Einfügen von virtuellen Objekten in einem leeren Raum um ihn für die Level Generierung interessanter zu gestalten. Dabei zeigt Teil A ein mögliches Level in einem leeren Raum und Teil B den selben Raum mit einer Erweiterung durch virtuelle Objekte um die möglichen generierten Level zu verbessern. Quelle: Selbst erstellte Grafik

Hindernisse oder Höhenunterschiede im Weg existieren. In Teil B von [Abbildung 4](#) wurden zur Verbesserung der räumlichen Voraussetzungen blaue Hindernisse eingefügt. Diese sorgen dafür, dass es mehrere verschieden hohe Bereiche gibt. Dies macht die generierten Level interessanter und abwechslungsreicher, da jetzt während der prozeduralen Generierung verschiedene Oberflächen miteingebunden werden können. Dieselben Prinzipien können in abgewandelter Form genutzt werden um hindernisreichen Spielumgebungen ihre räumliche Komplexität zu nehmen. Ein Beispiel dafür ist [Abbildung 5](#). Im Teil A ist der hintere Bereich des Raums auf Grund seiner Vielzahl an Hindernissen nicht geeignet um in das Level mit eingebunden zu werden. Dies beschränkt das Level auf den vorderen Bereich des Raums und



**Abbildung 5:** Beispiel für das Einfügen von virtuellen Objekten um die räumliche Komplexität der Spielumgebung zu verringern. Dabei zeigt Teil A ein mögliches Level in einer komplexen Umgebung und Teil B dieselbe Umgebung, aber vereinfacht durch die Vereinheitlichung von Oberflächen durch virtuelle Objekte. Quelle: Selbst erstellte Grafik

lässt die generierten Level sehr kurz und gestaucht ausfallen, wie man in Teil A von Abbildung 5 sehen kann. Um auch den hinteren Teil der Umgebung einzubinden wurden in Teil B mehrere komplexe Bereiche in größere Flächen zusammengefasst. Dies sorgt dafür, dass sie wieder mit in die Level Generierung miteingebunden werden können. Dadurch kann nun ein Level generiert werden, dass die komplette Tiefe des Raumes nutzt, was es länger und interessanter zu spielen macht. Wird die Umgebung des Spielers mit zu vielen Objekten erweitert, verliert das Spiel jedoch seine MR Eigenschaften und bewegt sich immer weiter in den Bereich der virtuellen Realität. An diesem Punkt muss man Abwägen in wie weit man eine Veränderung der Umgebung des Spielers zulassen kann, ohne dass die Charakteristiken des betreffenden Raumes verloren gehen. Alle Vor- und Nachteile von prozeduraler Generierung zur Anpassung des Spiels an die Umgebung sind in Tabelle 4 noch einmal zusammengefasst.

#### **Prozedurale Generierung**

<i>Vorteile</i>	<ul style="list-style-type: none"> <li>- Einbinden der physischen Umgebung des Spielers in das Spiel</li> <li>- Kann mit jeder Art von Raum umgehen</li> </ul>
<i>Nachteile</i>	<ul style="list-style-type: none"> <li>- Qualität des Spielerlebnisses ist stark von der physischen Umgebung des Spielers abhängig</li> <li>- Nur für Spiele mit simpler Spieler-Welt Interaktion anwendbar</li> <li>- Spieler ist gefordert Orte mit bestimmten Eigenschaften aufzusuchen um das Spielerlebnis zu verbessern</li> </ul>

**Tabelle 4:** Zusammenfassung aller genannten Vor- und Nachteile von prozeduraler Generierung zur Anpassung des Spiels an die physische Umgebung des Spielers. Quelle: Selbst erstellte Grafik

## **4. Fazit**

Die Anpassung der physischen Umgebung ist genauso wie die Anpassung der virtuellen Level nicht für jedes Spiel geeignet. Dies ist stark vom Genre und der Art des Gameplays abhängig. Welche die passende Lösungsstrategie für das Umgebungsproblem ist, muss für jedes Spiel neu entschieden werden. Um dies zu erleichtern wurden alle wichtigen Bereiche des Themas, in Form von räumlichen Mindestanforderungen, Marker Tracking, Vorgabe der Spielorte und prozeduraler Generierung mit ihren Vor- und Nachteilen beleuchtet. Zur Entscheidungserleichterung bietet folgende, daraus abgeleitete Fragenkatalog einige Orientierungshilfen.

## 4.1. Fragenkatalog zur Lösung des Umgebungsproblems

### Räumliche Mindestanforderungen

- Werden Freiräume, Oberflächen oder eine Kombination dieser benötigt?
- Wieviel davon wird benötigt?
- Wird ein Laufbereich benötigt?
- Ist der Aufwand der Herstellung der benötigten Umgebungsmerkmale angemessen?

### Freiraum

- Ist dieser Platz in der durchschnittlichen Wohnung der Zielgruppe zu finden?
- Wenn nein, ist das Spiel im freien Spielbar?
- Muss auf die körperlichen Eigenschaften der Spieler, wie unterschiedliche Sichthöhen, Rücksicht genommen werden?

### Oberflächen

- Findet diese Oberfläche auf einem durchschnittlichem Tisch Platz?
  - Wenn nein, kann das Spiel auf dem Boden gespielt werden?
  - Hat die Zielgruppe körperliche Beeinträchtigungen die sie am Spielen auf dem Boden hindern?

### Marker Tracking

- Erhält das Spiel durch die Einführung von Marker-Karten einen Mehrwert für den Spieler?
- Benötigt der Spieler die zusätzliche Kontrolle über die Platzierung der Spielinhalte?

### Dynamisch

- Wird die zusätzliche Input-Ebene durch die Bewegung der Marker-Karten benötigt?
- Wird eine ebene Oberfläche benötigt?
- Wenn ja, wie groß ist diese?

### Statisch

- Sind die Marker-Karten statisch?
- Ist das Spiel auf einer unebenen Oberfläche Spielbar?

## Vorgabe der Spielorte

- Profitiert das Spiel davon, dass der Spieler bestimmte Orte aufsuchen muss?
- Bietet das Aufsuchen eines realen Ortes einen Mehrwert für den Spieler?
- Ist das Aufsuchen der Orte für den Spieler zumutbar?
- Ist die Zielgruppe des Spiels in der Lage diese Orte aufzusuchen?
- Gibt es eine Vielzahl von Orten, damit das Spiel überall spielbar ist?
- Ist die GPS-Genauigkeit für den jeweiligen Anwendungsfall ausreichend?

## Prozedurale Generierung

- Hat das Anpassen des Spiels an die Umgebung einen Mehrwert für den Spieler?
- Ist dieser Effekt nicht durch die Anpassung der Umgebung an das Spiel zu erreichen?
- Ist die Interaktion des Spielers mit der Welt so gehalten, dass eine prozedurale Generierung der Level möglich ist?
- Ist es möglich in vielen verschiedenen Umgebungen interessante Level zu generieren?
- Wie wird mit leeren und mit Möbeln/Objekten gefüllten Räumen umgegangen?
- Muss die Umgebung bestimmte Merkmale aufweisen?
- Wie wahrscheinlich ist es, dass die Umgebungen der Spieler diese Merkmale aufweisen?

## 4.2. Ausblick

Die Blütezeit der MR-Spiele hat gerade erst begonnen, insbesondere die Entwicklung neuer AR- und MR-Brillen für den Verbrauchermarkt wird den Bereich weiter vorantreiben und neue Fragestellungen aufwerfen. Welche Auswirkungen wird eine weite Verbreitung von MR-Brillen auf das Spielverhalten der Nutzer haben? Wird es eigens für MR-Spiele gebaute Spiel-Arenen geben? Finden wir einen allgemeingültigen Weg das Umgebungsproblem für alle Arten von MR-Spielen zu lösen? All diese Fragen wird es in Zukunft zu beantworten geben.

## Literaturverzeichnis

**Barbagallo, Ralph. 2016.** So, You Wanna Make A Pokemon Go Clone? *Ralph Barbagallos self indulgent Blog*. [Online] 29. Juli 2016. [Zitat vom: 26. November 2017.] <https://ralphbarbagallo.com/2016/07/29/so-you-wanna-make-a-pokemon-go-clone/>.

—. **2015.** The Challenge of Building Augmented Reality Games In The Real World. *Ralph Barbagallo's self indulgent blog*. [Online] 17. August 2015. [Zitat vom: 26. November 2017.] <https://ralphbarbagallo.com/2015/08/17/the-challenge-of-building-augmented-reality-games-in-the-real-world/>.

**Bycer, Josh. 2016.** 3 Failings of Procedural Generated Game Design. *Gamasutra*. [Online] 8. Februar 2016. [Zitat vom: 29. November 2017.] [https://www.gamasutra.com/blogs/JoshBycer/20160802/278327/3\\_Failings\\_of\\_Procedurally\\_Generated\\_Game\\_Design.php](https://www.gamasutra.com/blogs/JoshBycer/20160802/278327/3_Failings_of_Procedurally_Generated_Game_Design.php).

**futurezone. 2017.** Neue GPS-Chips orten Smartphones ab 2018 auf 30 Zentimeter genau. *futurezone*. [Online] 23. September 2017. [Zitat vom: 26. November 2017.] <https://www.futurezone.de/digital-life/article212018823/Neue-GPS-Chips-orten-Smartphones-ab-2018-auf-30-Zentimeter-genau.html>.

**Google Developers. 2017.** Google I/O Keynote (Google I/O '17). *YouTube*. [Online] 17. Mai 2017. [Zitat vom: 28. November 2017.] <https://youtu.be/Y2VF8tmLFHw?t=6179>.

**Grabmann, Johann. 2016.** Pokémon Go: Internationaler Start & spielen in Deutschland (APK download) (Update). *johann.gr*. [Online] 9. Juli 2016. [Zitat vom: 20. November 2017.] <https://johann.gr/pokemon-go-internationaler-start-spielen-in-deutschland-apk-download/>.

**Jansen, Jonas. 2017.** Wer spielt eigentlich noch Pokémon Go? *Frankfurter Allgemeine*. [Online] 29. Juni 2017. [Zitat vom: 20. November 2017.] <http://www.faz.net/aktuell/wirtschaft/netzwirtschaft/ein-jahr-nach-dem-hype-wer-spielt-eigentlich-noch-pokemon-go-15076981.html>.

**Katzenberg, Marco. 2017.** GPS-Genauigkeit: So könnt ihr sie erhöhen (Android & iPhone). *GIGA TECH*. [Online] 4. September 2017. [Zitat vom: 26. November 2017.] <http://www.giga.de/extra/gps/tipps/gps-genauigkeit-so-koennt-ihr-sie-erhoehen-android-iphone/>.

**Lang, Ben. 2017.** The 3 Biggest Challenges Facing Augmented Reality Today. *Road to VR*. [Online] 29. September 2017. [Zitat vom: 26. November 2017.] <https://www.roadtovr.com/3-biggest-challenges-facing-augmented-reality-today/3/>.

**Lee, Jin Ha, Keating, Stephen und Windleharth, Travis. 2017.** Challenges in Preserving Augmented Reality Games: A Case Study of Ingress and Pokémon GO. *iPres 2017 - 14th International Conference on Digital Preservation*. [Online] September 2017. [Zitat vom: 26. November 2017.] <https://ipres2017.jp/wp-content/uploads/44Jin-Ha-Lee.pdf>.

**Levski, Yariv. o.J.** Markerless vs. Marker based Augmented Reality. *AppReal*. [Online] o.J. [Zitat vom: 26. November 2017.] <https://appreal-vr.com/blog/markerless-vs-marker-based-augmented-reality/>.

**Mel. 2017.** ARise: Erstes Puzzle-Spiel setzt auf Apples ARKit und verfügt zum Start über drei Level. *Appgefahren*. [Online] 22. September 2017. [Zitat vom: 24. November 2017.] <http://www.appgefahren.de/arise-erstes-puzzle-spiel-setzt-auf-apples-arkit-und-verfuegt-zum-start-ueber-drei-level-208671.html>.

**Milgram, Paul und Kishino, Fumio. 1994.** A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*. 1994, Bde. Vol E77-D, 12.

**Riedl, Mark. 2016.** Augmented Reality Games Need Artificial Intelligence. *Medium*. [Online] 17. Juni 2016. [Zitat vom: 29. November 2017.] [https://medium.com/@mark\\_riedl/augmented-reality-games-need-artificial-intelligence-b8f57ec4e5a9](https://medium.com/@mark_riedl/augmented-reality-games-need-artificial-intelligence-b8f57ec4e5a9).

**Spacey, John. 2016.** Augmented Reality vs Augmented Virtuality. *Simplicable*. [Online] 27. Dezember 2016. [Zitat vom: 20. November 2017.] <https://simplicable.com/new/augmented-reality-vs-augmented-virtuality>.

**Statistisches Bundesamt Deutschland. 2017.** *Statistisches Jahrbuch Deutschland 2017*. Berlin : Statistisches Bundesamt Deutschland, 2017. 978-3-8246-1057-0.

**Wohnung.com. o.J.** Wohnzimmer: Diese Größe brauchen Sie. *Wohnung.com*. [Online] o.J. [Zitat vom: 28. November 2017.] <http://www.wohnung.com/ratgeber/416/wohnzimmer-diese-groesse-brauchen-sie>.

## Spieleverzeichnis

**Pokémon GO.** Niantic Inc.. Niantic Inc., Nintendo Co. Ltd., The Pokémon Company, 2016.

**ARise.** Climax Studios Ltd.. Climax Studios Ltd., 2017.

**The Machines.** Directive Games Limited. Directive Games Limited, 2017.

**Robo Raid.** Microsoft Corporations. Microsoft Corporation, 2016.

**Pulzar.** XDEV Studio Europe. Sony Computer Entertainment, 2012.

**Genesis Augmented Reality.** Genesis Augmented Reality. Genesis Augmented Reality, 2016.

**Young Conker.** Microsoft Corporations. Microsoft Corporation, 2016.

## Abbildungsverzeichnis

**Abbildung 1:** Realität-Virtualität Kontinuum nach Milgram. Quelle: Valoriani, Matteo. 2016.

Etna dev. Introduction to Mixed Reality with HoloLens. Slideshare.net [Online] 24. November 2016. <https://de.slideshare.net/MatteoValoriani/etna-dev-2016-introduction-to-mixed-.....> 1

**Abbildung 2:** Platzierung einer Marker-Karte in dem PS Vita Spiel Pulzar um den Laser durch einen Spiegel in Richtung Raketenbasis abzulenken. Quelle: Vincent, Ryan. 2012.

<https://futurefive.co.nz/story/game-review-tabletop-tanks-and-pulzar-on-ps-vita/> ..... 7

**Abbildung 3:** Beispiel für die Bewegung des Spielercharakters auf einer unebenen Oberfläche, in dem Spiel Genesis, außerhalb der weißen Marker-Karte. Quelle: Selbst erstellte Grafik ..... 7

**Abbildung 4:** Beispiel für das Einfügen von virtuellen Objekten in einem leeren Raum um ihn für die Level Generierung interessanter zu gestalten. Dabei zeigt Teil A ein mögliches Level in einem leeren Raum und Teil B den selben Raum mit einer Erweiterung durch virtuelle Objekte um die möglichen generierten Level zu verbessern. Quelle: Selbst erstellte Grafik 12

**Abbildung 5:** Beispiel für das Einfügen von virtuellen Objekten um die räumliche Komplexität der Spielumgebung zu verringern. Dabei zeigt Teil A ein mögliches Level in einer komplexen Umgebung und Teil B dieselbe Umgebung, aber vereinfacht durch die Vereinheitlichung von Oberflächen durch virtuelle Objekte. Quelle: Selbst erstellte Grafik. 12



## Tabellenverzeichnis

<b>Tabelle 1:</b> Zusammenfassung und Gegenüberstellung aller genannten Vor- und Nachteile von Mindestanforderungen in Form von Freiräumen und Oberflächen. Quelle: Selbst erstellte Grafik .....	5
<b>Tabelle 2:</b> Zusammenfassung und Gegenüberstellung aller genannten Vor- und Nachteile von Marker basierten AR-Spielen in Form von statischem und dynamischem Marker Tracking. Quelle: Selbst erstellte Grafik .....	8
<b>Tabelle 3:</b> Zusammenfassung aller genannten Vor- und Nachteile von vorgegebenen Spielorten. Quelle: Selbst erstellte Grafik.....	10
<b>Tabelle 4:</b> Zusammenfassung aller genannten Vor- und Nachteile von prozeduraler Generierung zur Anpassung des Spiels an die physische Umgebung des Spielers. Quelle: Selbst erstellte Grafik.....	13