

daniel KORGEL

VIRTUAL REALITY Spiele

entwickeln mit
UNITY[®]

GRUNDLAGEN
BEISPIELPROJEKTE
TIPPS & TRICKS

HANSER



Behandelt Unity 2017

Korgel

Virtual Reality-Spiele entwickeln mit Unity®

Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



www.hanser-fachbuch.de/newsletter



Hanser Update ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



www.hanser-fachbuch.de/update



Daniel Korgel

Virtual Reality-Spiele entwickeln mit Unity®

Grundlagen, Beispielprojekte,
Tipps & Tricks

HANSER

Der Autor:

Daniel Korgel, Dortmund

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2018 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sylvia Hasselbach

Copy editing: Sandra Gottmann, Münster-Nienberge

Umschlagdesign: Marc Müller-Bremer, München, www.rebranding.de

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-45147-6

E-Book-ISBN: 978-3-446-45372-2

Inhalt

Vorwort	XV
1 Einleitung	1
1.1 Virtual Reality	2
1.2 Ziel des Buches	2
1.3 Unterstützte Virtual-Reality-Brillen	3
1.4 Unity	4
1.4.1 Die verschiedenen Versionen von Unity	5
1.5 Was brauche ich?	6
1.6 Weiterentwicklung der Technik	7
1.7 Online-Zusatzmaterial	7
2 Einführung in Virtual Reality	9
2.1 Begriffserklärungen	9
2.2 Augmented Reality, Mixed Reality und Virtual Reality	11
2.3 Virtual Reality ist tot – lang lebe Virtual Reality	12
2.4 Unterstützte Virtual-Reality-Brillen	15
2.4.1 Oculus Rift	16
2.4.2 HTC Vive	17
2.4.3 Samsung GearVR	18
2.4.4 Google Cardboard (nur teilweise unterstützt)	19
2.4.5 Google Daydream	20
2.5 Andere VR-Brillen	21
2.5.1 Microsoft Mixed Reality	21
2.5.2 PlayStation VR	22
2.5.3 Razer OSVR	23
2.6 Spiele, Anwendungen, Demos und Erfahrungen	24
2.7 Seated, Standing & Room Scale	25
2.8 „Bekannte“ Probleme	26
2.8.1 Auflösung & Fliegengitter-Effekt	27

2.8.2	Hardwareanforderungen	27
2.8.3	Kurz- und Weitsichtigkeit	28
2.8.4	Simulator Sickness	28
3	Quickstart: 3D-Entwicklung mit Unity	31
3.1	Installation	31
3.1.1	Die Unity ID	33
3.1.2	Android SDK & Java Development Kit(GearVR, GoogleVR)	34
3.2	Der erste Start	38
3.3	Ein neues Projekt anlegen	39
3.4	Die Unity-Oberfläche	40
3.4.1	Drag & Drop	41
3.4.2	Project Browser	42
3.4.3	Console	43
3.4.4	Hierarchy	44
3.4.5	Scene View	46
3.4.6	Inspector	48
3.5	Scenes	54
3.5.1	Unity in der dritten Dimension	54
3.5.2	GameObjects	55
3.5.3	Prefabs	57
3.6	Die wichtigsten Components	58
3.6.1	Camera und Audio Listener	58
3.6.2	Lights	59
3.6.3	Mesh Filter und Mesh Renderer	60
3.6.4	Collider	61
3.6.5	Rigidbody	62
3.6.6	Audio Source	63
3.6.7	Scripts/Behaviours	64
3.7	Assets	64
3.7.1	Dateien importieren	67
3.7.2	Unity Packages	68
3.7.3	Asset Store	71
3.7.4	Import Settings	72
3.8	Projektvorbereitung (GearVR, GoogleVR)	72
3.8.1	Zielplattform zu Android wechseln	73
3.8.2	Externe Tools angeben (Android SDK, JDK)	74
3.9	Test Scene erstellen und starten	76
3.9.1	Rohbau für den Raum erstellen	78
3.9.2	Scene speichern	81
3.9.3	Licht anpassen	82
3.9.4	Collider hinzufügen	83
3.9.5	Möbelstücke einräumen	84

3.9.6	Bücher hinzufügen	87
3.9.7	Weitere Details	91
3.9.8	Lichter und Lichteinstellungen optimieren	92
3.9.9	Spieler positionieren	95
3.10	VR-Unterstützung aktivieren	96
3.10.1	Unitys VR-Unterstützung aktivieren	96
3.11	Player Settings anpassen (GearVR, GoogleVR)	97
3.12	Anti-Aliasing aktivieren	99
3.13	Die Scene in VR testen	100
3.13.1	Am PC in VR testen (Oculus Rift, SteamVR)	101
3.13.2	Auf dem Smartphone in VR testen (GearVR, GoogleVR)	102
3.14	So geht es weiter	111
4	Dos and Don'ts – Game-Design für Virtual Reality	113
4.1	Beispielprojekt	114
4.2	Immersion	114
4.2.1	Immersion brechen	118
4.3	Positional Tracking	118
4.3.1	Mobile VR-Brillen	120
4.4	Locomotion	121
4.4.1	Klassische Locomotion	121
4.4.2	Teleport Locomotion	124
4.4.3	Ohne künstliche Locomotion	125
4.4.4	Andere Eingabemethoden für das Laufen	128
4.5	Lebendige Spielwelt erschaffen	129
4.6	User Interface	133
4.6.1	Distanz und Größe	136
4.7	Aufmerksamkeit lenken	137
4.7.1	Objekte untersuchen	138
4.7.2	Ereignisse nicht verpassen	141
4.8	Uncanny Valley	142
4.8.1	Auswirkung für dich	145
4.9	Sound Design und Spatial Sound	146
4.9.1	Spatial Sound	146
4.9.2	Spatial-Audio-Engines	149
5	C#-Scripte für Unity programmieren	151
5.1	C#, .Net und Mono	152
5.2	Grundlagen	152
5.2.1	C#-Syntax	152
5.2.2	Kommentare	153
5.2.3	Variablen	154
5.2.4	Methoden	161

5.2.5	Klassen	164
5.2.6	Dynamische Datentypen	165
5.2.7	Statische Methoden und statische Variablen	167
5.2.8	Vererbung	168
5.2.9	Namespaces	171
5.2.10	Lebenszyklus von Variablen	172
5.2.11	Sichtbarkeiten	173
5.2.12	Ausnahmen – „Exceptions“	174
5.3	Erster Test	175
5.3.1	Microsoft Visual Studio 2017	175
5.3.2	Script, Behaviour, Component und ihr Aufbau	179
5.3.3	Debug-Ausgaben in die Console	180
5.3.4	Eigenes Script testen	181
5.3.5	Notation von Methoden und Variablen in diesem Buch	183
5.4	Erweiterte Grundlagen	183
5.4.1	Bedingungen und Verzweigungen	183
5.4.2	Schleifen	186
5.4.3	Eigenschaften	191
5.4.4	Dynamische Listen	192
5.5	Unity-spezifische Grundlagen	196
5.5.1	Zufallswerte	196
5.5.2	Editor-Eigenschaften	197
5.5.3	MonoBehaviour Lebenszyklus-Methoden	200
5.5.4	MonoBehaviour-Event-Methoden	202
5.5.5	Auf GameObjects zugreifen	203
5.5.6	GameObjects via Code erstellen	206
5.5.7	GameObjects zerstören	207
5.5.8	Transforms: GameObjects bewegen	207
5.5.9	Auf andere Components zugreifen	209
5.5.10	Components zu GameObject hinzufügen	212
5.5.11	Components entfernen	213
5.5.12	Parallele Methoden mit Coroutines	213
5.5.13	Scripted-Event-Beispiel	214
5.5.14	Methoden verzögert aufrufen mit Invoke	215
5.6	Problemlösung	216
5.6.1	Wort im Code ist rot unterstrichen	216
5.6.2	Fehler in der Console im Editor	217
5.6.3	Fehler in der Console, während das Spiel läuft	217
5.6.4	„Can't add script“-Meldung oder Script ist nicht im „Add Component“-Menü vorhanden	218
5.6.5	Sonstige Fehler	219
6	Erweiterte Unity-Einführung	221
6.1	Licht, Schatten und Lightmapping	221

6.1.1	Global Illumination	222
6.1.2	Farbräume	232
6.1.3	Environment Light	233
6.1.4	Lichter in der Scene	234
6.1.5	Emissive Materials	240
6.1.6	Light Probes	240
6.1.7	Reflexionen	243
6.1.8	Lighting-Fenster	246
6.1.9	Light Explorer	248
6.2	Shader	249
6.2.1	Der Standard Shader	250
6.2.2	Mobile Shader	255
6.3	Audio	258
6.3.1	AudioSources	259
6.3.2	AudioSources via Script steuern	262
6.3.3	Temporäre AudioSource via Script	264
6.3.4	Reverb Zones	264
6.3.5	Audio Mixer	265
6.4	Tastatur und Gamepad-Eingaben lesen	269
6.4.1	Der Input-Manager	270
6.4.2	Virtuelle Tasten und Achsen	270
6.4.3	Konfiguration einer Achse	271
6.4.4	Gamepad- und Tastatureingaben erfassen	273
6.5	Physik	275
6.5.1	Collider	275
6.5.2	Rigidbody	278
6.5.3	Kräfte und Beschleunigung	280
6.5.4	Physics Joints - Verknüpfungen	281
6.5.5	Kollisionen und Trigger erkennen	285
6.5.6	Raycasts	291
6.5.7	Character Controller	292
6.6	Partikeleffekte	295
6.6.1	Partikelsysteme	296
6.6.2	Particle-Effect-Steuerung	297
6.6.3	Module des Particle Systems	298
6.6.4	Particle Effect Editor	305
6.6.5	Particle Systems via Script steuern	305
6.6.6	Standard-Partikeleffekte	306
6.7	Landschaften mit Terrains erstellen	306
6.8	Animation System	309
6.8.1	Animation Workflow	310
6.8.2	Humanoide Animationen und Avatare	312
6.8.3	Modelle & Animationen importieren	313
6.8.4	Animation Controller	317

6.9	Wegfindung	321
6.9.1	Wegfindungssystem einrichten	321
6.9.2	Grundlagen	323
6.9.3	NavMeshs berechnen	325
6.9.4	NavMeshs verbinden	326
6.9.5	Dynamische Hindernisse	326
6.9.6	NavMeshAgent erstellen	327
6.9.7	NavMeshAgent über Script steuern	327
6.9.8	Erweiterte Beispiele	328
6.10	Savegames - Daten speichern und laden	329
6.11	Performance optimieren	330
6.11.1	Batching	330
6.11.2	Performance im Auge behalten	331
6.11.3	VR-Optimierungs-Checkliste und Grenzen	335
6.12	User Interface - Quickstart	337
6.12.1	User Interface im World Space erstellen	337
6.12.2	User Interface in VR bedienen	342
7	Die Virtual-Reality-SDKs	345
7.1	Unity-interne VR-Unterstützung	346
7.1.1	Unitys interne VR-Unterstützung aktivieren	346
7.1.2	Interne VR-Unterstützung in der Scripting-API	348
7.1.3	InputTracking	348
7.1.4	VRDevice	349
7.1.5	VRSettings	350
7.2	Oculus Integration for Unity (Rift und Gear)	352
7.2.1	Download und Import	352
7.2.2	Prefabs	354
7.2.3	Gamepad-, Headset- und Touch-Eingaben lesen	361
7.2.4	Haptisches Feedback	363
7.2.5	Oculus Avatar SDK	365
7.2.6	Mehr Infos und Hilfe	367
7.3	SteamVR (u. a. HTC Vive)	368
7.3.1	Download und Import	369
7.3.2	Prefabs	371
7.3.3	Eingaben der Controller lesen	378
7.3.4	Haptisches Feedback	379
7.3.5	Interaktionsbeispiel aus The Lab (u. a. Longbow, Teleport)	380
7.3.6	Mehr Infos und Hilfe	380
7.4	GoogleVR SDK	381
7.4.1	Download und Import	381
7.4.2	Prefabs	382
7.4.3	Eingaben des Daydream Controllers lesen	386
7.4.4	Erweiterte Editorfunktionen fürs Testen	387

8	Beispielprojekte – Einleitung	391
9	Bow and Arrow Castle Defense – GoogleVR mit Controller	393
9.1	Der Start	394
9.2	Die Kernmechanik – Bogenschießen	396
9.2.1	Scene vorbereiten	396
9.2.2	Einfaches Scene-Design	396
9.2.3	Spieler konfigurieren und Daydream Controller einbinden	399
9.2.4	Schusskraft für den Bogen	401
9.2.5	Ein Pfeil für den Bogen	403
9.2.6	Pfeile nachladen	406
9.2.7	Dynamische Schusskraft	408
9.2.8	Flugkurve Vorschau – Add-on	410
9.2.9	Schaden und zerstörbare Ziele	413
9.3	Game- & Level-Design	414
9.4	Game Manager	416
9.5	Gegner – Orks!	421
9.5.1	NavMesh erstellen	421
9.5.2	Ork-Animator Controller	421
9.5.3	Die künstliche Intelligenz	423
9.5.4	Der Ork	424
9.5.5	Prefab erstellen und platzieren	426
9.6	Abschluss	426
9.6.1	Umsetzung dieses Beispiels für andere VR-Brillen	427
10	Space Battle – Oculus Rift & SteamVR	429
10.1	Der Start	430
10.1.1	Scene vorbereiten	431
10.2	Die Kernmechanik – ein Raumschiff fliegen	434
10.2.1	Spieler-Raumschiff erstellen	435
10.2.2	Raumschiff-Steuerung	438
10.2.3	Starfield/Weltraumstaub	445
10.2.4	Schadenssystem & Plasmakanone	448
10.3	Feinde und Verbündete	456
10.3.1	KI-Raumschiff	456
10.3.2	Die künstliche Intelligenz	458
10.4	Abschluss	467
10.4.1	Umsetzung dieses Beispiels für andere VR-Brillen	468
11	Dungeon Crawler – GearVR	469
11.1	Der Start	470
11.1.1	Osig und Unity2017.1.0f3-Patch einfügen	472
11.2	Einen Dungeon bauen	472

11.2.1 Die Bauteile	473
11.2.2 Level-Design	473
11.2.3 Audio: Environment-Loop	476
11.3 Occlusion Culling	476
11.4 Kernmechanik – Laufen, Kämpfen, Interagieren	478
11.4.1 Spieler erstellen	479
11.4.2 Bewegungen entlang des Gitters	481
11.4.3 Interaktive und lebendige Objekte	484
11.4.4 Die Spielerfunktionen	486
11.4.5 Spielereingaben lesen	491
11.4.6 Player fertigstellen	494
11.5 Gegner: Goblins	495
11.5.1 Die künstliche Intelligenz	496
11.5.2 Das GameObject und Prefab	498
11.5.3 Bevölkerung des Dungeons mit Goblins	499
11.6 Pickups: Kristalle	500
11.7 Dungeon-Ausgang	502
11.8 Interaktive Hindernisse	504
11.8.1 Falle: Stacheln	504
11.8.2 Rätsel: Schalter und Tore	506
11.9 Abschluss	509
11.9.1 Umsetzung dieses Beispiels für andere VR-Brillen	510
12 Sci-Fi-Stealth-Shooter – Oculus Rift mit Oculus Touch und SteamVR	511
12.1 Der Start	512
12.2 Trainings-Areal anlegen	514
12.3 Player-GameObject erstellen	516
12.3.1 Oculus SDK: Player-GameObject erstellen	516
12.3.2 SteamVR SDK: Player-GameObject erstellen	517
12.3.3 Character Controller	518
12.3.4 Locomotion – klassisch und teleportieren	521
12.3.5 Eingaben lesen und Methoden aufrufen	525
12.3.6 Nach Gegenständen greifen	530
12.4 Lebensenergie verwalten	540
12.4.1 Lebensenergie des Spielers und Game Over	541
12.5 Laserpistole	542
12.5.1 Das GameObject vorbereiten	542
12.5.2 Die Laserpistolen-Funktion	543
12.5.3 Der Laserstrahl	545
12.6 Roboter-Gegner	547
12.6.1 EnemyAi – wie funktioniert die KI?	547
12.6.2 Roboter-GameObject anlegen	548

12.6.3 Trigger Collider ignorieren	550
12.7 Level erstellen	551
12.7.1 Level-Ende	552
12.8 Abschluss	553
13 VR-Spiele teilen, präsentieren und Feedback erhalten	555
13.1 Spieleidee entwickeln	555
13.2 VR-Spiele Builds erstellen und teilen	556
13.3 Feedback erhalten und Spiel bewerben	558
13.3.1 Tipps für das Präsentieren von Virtual Reality	559
13.4 Spiel veröffentlichen	560
13.5 Kostenlose Assets	561
14 Schlusswort	563
Index	565

Vorwort

Was war der größte Traum deiner Kindheit? Mit bezahlbaren Virtual-Reality-Brillen, deren Fokus auf immersivem Entertainment liegt, ist für mich mein Kindheitstraum wahr geworden. Lange war es nur eine Fantasie aus Sci-Fi-Büchern und eine Technologie, die Anfang der 90er gescheitert war. Doch als ich 2012 das erste Mal von der *Oculus Rift* hörte, schien der Traum plötzlich nicht mehr so fern und ich verbrachte viel Zeit damit, mich immer intensiver mit dem Thema zu beschäftigen. Schnell reichte mir das nicht mehr: Ich wollte andere ebenfalls für das Thema begeistern und helfen, das Thema leichter zugänglich zu machen. Deswegen habe ich 2012 *Bloculus.de* gegründet, wo ich mehrmals die Woche über die aktuellen Neuigkeiten und Spekulationen der Szene schrieb. Heute, wo jede IT-News-Seite über das Thema berichtet, ist mein Blog nicht mehr vonnöten, um Neuigkeiten zu verbreiten. Deshalb begann ich, mehr und mehr aus der Sicht eines VR-Entwicklers zu schreiben und so meinen Lesern eine alternative Perspektive zu bieten.

Ich hatte es mir schon häufig vorgestellt, aber nie hätte ich gedacht, dass ich mal wirklich ein Buch schreibe. Umso erfreuter war ich über den Anruf von Sylvia Hasselbach vom Hanser Verlag, mit der Frage, ob ich Interesse hätte, ein einsteigerfreundliches Buch über Virtual-Reality-Entwicklung zu schreiben.

Es hat Spaß gemacht, all dieses Wissen niederzuschreiben, und der Gedanke, es mit vielen Menschen zu teilen, war sehr motivierend. Aber es war nicht immer leicht, manche Wochen waren sehr anstrengend. Umso dankbarer bin ich für jede Unterstützung, die ich erfahren habe. Danke an alle Freunde, Bekannte und meine Familie, die mich bei diesem Buch, auf die eine oder andere Weise, unterstützt haben – und wenn es nur war, dass sie dafür Verständnis hatten, dass ich mich wenig gemeldet habe, zu spät kam und zu früh wieder ging. Mein ganz besonderer Dank geht an Kathrin Zenses, die mich stets unterstützt hat und mir mit viel Verständnis begegnet ist. Das alles ist nicht selbstverständlich, danke an euch!

Virtual Reality dringt immer mehr in den Massenmarkt und immer häufiger hört man die Worte: „*Jetzt stell dir das mal in Virtual Reality vor ...*“ Mit diesem Buch möchte ich das Entwickeln für Virtual Reality jedem zugänglich machen, damit auf diesen Satz immer die Aussage folgt: „*Das probiere ich mal aus!*“

Dortmund, September 2017

Daniel Korgel

1

Einleitung

Mit einer selbst gebauten und von Klebeband zusammengehaltenen Virtual-Reality-Brille überzeugte der 19-jährige Palmer Luckey 2011 einige wichtige Personen aus der Videospiel-Branche und erweckte so Virtual Reality aus dem Winterschlaf. Zu den überzeugten Personen gehörte unter anderem auch *Doom*-Erfinder *John Carmack*, was Luckey viel Vertrauen von Kollegen und Fans eingebracht hat. 2012 erhielt Luckey über die Crowdfunding-Plattform *Kickstarter* fast 2,5 Millionen US Dollar, um seine Virtual-Reality-Brille **Oculus Rift** zu finanzieren.

Heute, mehr als fünf Jahre später, wurde Palmer Luckeys Firma *Oculus* von Facebook gekauft. Neben Oculus bieten einige weitere bekannte Technikfirmen, darunter *Sony*, *HTC*, *Samsung*, *Microsoft* und *Google*, ihre eigenen Virtual-Reality-Brillen für Computer, Spielkonsolen und Smartphones an. Erstmals ist es jedem möglich, gute Virtual-Reality-Hardware, zu einem bezahlbaren Preis, zu kaufen. Virtual Reality wird das nächste große Ding in der Unterhaltungsindustrie.

Für viele Gamer und Technikfans wird mit bezahlbarer Virtual-Reality-Hardware ein Traum wahr. Aber einigen reicht das reine Erleben von Virtual Reality nicht: Die völlig neuen Möglichkeiten wecken den Wunsch, eigene virtuelle Welten zu erschaffen – und das genau zur richtigen Zeit: Virtual Reality ist ein so junges Thema, dass die App Stores noch nicht überfüllt sind mit Kopien von bekannten Spielen. Wenn du jetzt ein innovatives VR-Spiel veröffentlichst, ist die Chance, dass du dich von der Masse der Spiele-Apps abhebst, deutlich höher als zum Beispiel mit einem 2D-Spiel für Smartphones oder den PC.

Durch moderne 3D-Engines wie Unity und Unreal ist dies heutzutage sogar ein relativ leicht realisierbarer Traum. Vor einigen Jahren hättest du noch bei null anfangen und zunächst eine eigene Grafikengine schreiben müssen. Dank dieser Engines, welche auch von großen Studios genutzt werden, kannst du dich heute vollkommen auf das eigentliche Spiel konzentrieren.

Möchte man allerdings ein gutes Virtual-Reality-Spiel veröffentlichen, muss man auch einiges beachten: Nicht jedes existierende 2D-Videospiel kann 1:1 in Virtual Reality realisiert werden. Viele jahrelang genutzte Techniken müssen neu überdacht werden, damit sie in Virtual Reality die gewünschte Wirkung erzielen. Dazu gehören zum Beispiel das Storytelling und die Art und Weise, wie man mit der virtuellen Welt interagiert. Auch etablierte Spielmechaniken, die du in jedem PC- und Konsolenspiel findest, müssen neu überdacht

werden: Klebt die Waffe zum Beispiel, wie in Egoshootern üblich, an deinem Kopf, fühlt sich das in Virtual Reality komisch an.

Bei diesem immer noch jungen Thema gilt es also, ausgetretene Pfade zu verlassen und neue Wege zu ergründen.

■ 1.1 Virtual Reality

Vergleichst du die neusten technologischen Entwicklungen mit dem, was schon vor Jahrzehnten in Science-Fiction beschrieben wurde, stellst du einige auffällige Parallelen fest: Seien es automatische Schiebetüren oder Mobiltelefone, sie existierten zuerst in Science-Fiction-Büchern und -Serien, teilweise lange bevor die Technik überhaupt denkbar war. Es war also nur eine Frage der Zeit, bis auch massenmarkttaugliche Virtual-Reality-Hardware den Sprung aus Büchern wie *Neuromancer* oder *Snow Crash* in die Realität schafft.

Doch wie sieht Virtual Reality derzeit aus? Nun, wir sind noch nicht so weit wie in dem Film *Matrix*, wo ein Anschluss im Nacken des Helden die Schnittstelle zu einer virtuellen Welt darstellt. In erster Linie werden heutzutage *Virtual-Reality-Headsets* verwendet, welche wie eine große Brille aufgesetzt werden. Virtual-Reality-Headsets werden deshalb häufig auch „Brillen“ genannt. In der *Brille* werden für dein linkes und dein rechtes Auge zwei getrennte Bilder angezeigt und wie bei einem 3D-Film entsteht hierdurch ein 3D-Effekt. Durch spezielle Linsen ist das Blickfeld so groß, dass es so wirkt, als würde dich die virtuelle Welt vollkommen umgeben. Dieser Eindruck entsteht zusätzlich auch dadurch, dass du dich in der virtuellen Welt umschauchen kannst, indem du dich tatsächlich mit dem Kopf umsiehst. Anders als bei traditionellen Videospiele werden kein Gamepad und keine Maus hierfür benötigt.

Dieses Gefühl, wirklich in der Welt zu sein, unterscheidet Virtual Reality von einem gewöhnlichen Monitor. Es ist der Grund, warum diese Plattform, trotz der anfangs noch nicht optimalen Hardware, schon früh sehr viele Anhänger hatte. Auf keiner anderen Plattform lassen sich Ängste, Emotionen und Abenteuerlust so auf den Spieler übertragen wie in Virtual Reality.

■ 1.2 Ziel des Buches

Mithilfe dieses Buches wirst du deine ersten Virtual-Reality-Spiele mit der Unity-Engine entwickeln und dabei lernen, wie du zukünftig eigene Ideen völlig ohne fremde Hilfe umsetzen kannst. In diesem Buch wird dir nicht nur die Handhabung von Unity und diversen VR-Entwicklerwerkzeugen erklärt, sondern auch, was du bei der Entwicklung für ein Virtual-Reality-Headset beachten musst. Zum Beispiel erhältst du auch die Antworten auf Fragen wie „Welche Effekte sind besonders beeindruckend?“, „Wie designe ich ein Virtual-Reality-taugliches Menü?“ und „Was sollte ich unbedingt vermeiden?“.

Bei Virtual Reality und auch in der Softwareentwicklung allgemein ist es meiner Ansicht nach sehr wichtig, dass du nicht nur über die Dinge liest, sondern sie auch selber ausprobierst. Deswegen wirst du in den späteren Kapiteln sowohl positive als auch negative Beispiele selber auf deiner eigenen Brille ausprobieren können. Häufig versteht man erst durch Ausprobieren der Beispiele, warum sie in Virtual Reality gut funktionieren, oder warum nicht.

Der Fokus dieses Buches liegt darauf, dir zu zeigen, wie du ohne große Umwege eine gute Virtual-Reality-Erfahrung entwickeln kannst. Du wirst in diesem Buch Beschreibungen zu allen Funktionen finden, die du benötigst, um ein gutes Virtual Reality-Spiel zu entwickeln. Lange Erklärungen zu jedem einzelnen Menüpunkt in Unity wirst du in diesem Buch jedoch nicht finden.

Nach einer kurzen Einführung in Virtual Reality und das wichtigste Vokabular, beginnen wir auch schon mit dem Quickstart-Kapitel, welches dir den schnellsten Weg von einem neuen Unity-Projekt zu einer selbst erstellten Virtual-Reality-Erfahrung zeigt. Danach sehen wir uns auch schon die positiven und negativen Beispiele für die VR-Entwicklung an. In den darauffolgenden Kapiteln werde ich dir sowohl einen tieferen Einblick in Unity und der Programmiersprache C# geben als auch die Entwicklerwerkzeuge (*SDKs*) für die VR-Brillen vorstellen. An dieser Stelle hast du dann ausreichend Wissen und Erfahrung gesammelt, um die Beispielprojekte anzugehen und deine ersten eigenen Virtual-Reality-Spiele zu entwickeln.

■ 1.3 Unterstützte Virtual-Reality-Brillen

In diesem Buch beschreibe ich dir nicht, wie man nur für eine einzige Virtual-Reality-Brille entwickelt, sondern wie man für alle von Unity unterstützten VR-Brillen entwickelt. Dabei werde ich dir natürlich nicht nur die Funktionen beschreiben, die alle Brillen gemeinsam haben, sondern auch auf die individuellen Details jeder Brille eingehen. Dies hat den Vorteil, dass du dir nicht eine bestimmte Brille kaufen musst, um diesem Buch folgen zu können, sondern dass du einfach die Brille verwenden kannst, die du am besten findest.

Unterstützte Virtual-Reality-Brillen:

- Oculus Rift
- Oculus Rift mit Oculus Touch
- Steam VR (z. B. HTC Vive)
- Samsung GearVR
- Google VR: Daydream (Hinweis: Cardboard wird teilweise unterstützt)¹

¹ Grundsätzlich kannst du dem Buch auch mit einer für Google Cardboard kompatiblen Brille folgen, indem du dich einfach an die Schritte für „Google VR: Daydream“ hältst. Es wird jedoch nicht explizit auf die Eigenheiten der Cardboard-Brille eingegangen und es existiert kein Beispielprojekt gezielt für diese Brille. Die meisten Tipps und Tricks und die Einführung in Unity sind jedoch universal für jede Brille anwendbar.

Da diese Brillen alle von Unity offiziell unterstützt werden, können sie ohne spezielle Anpassungen direkt in Unity verwendet werden. Wenn möglich, beschreibe ich daher alle Inhalte dieses Buches so, dass du sie mit jedem der Headsets nachvollziehen kannst. Das gilt vor allem für die Grundlagen und die Einführung in die VR-Entwicklung.

Im Laufe des Buches zeige ich dir jedoch auch, wie du die speziellen Eigenschaften jeder der VR-Brillen verwenden kannst. Da manche der Brillen individuelle Besonderheiten haben (Hand Controller, Touch Pads etc.), sind manche Abschnitte deswegen nur für einzelne der unterstützten Geräte relevant. Es wird jedoch keine bestimmte Virtual-Reality-Brille in diesem Buch bevorzugt, sodass du mit jeder Brille auf deine Kosten kommst.

Ich werde mich in diesem Buch zudem nicht nur auf die Unity-interne Unterstützung für die VR-Brillen beschränken, da diese nur eine Schnittmenge der Funktionen aller Brillen abdeckt. Ich werde dir deshalb auch die einzelnen Entwicklungswerkzeuge (*SDKs*) für die VR-Brillen vorstellen. Spätestens hier lernst du dann die Fähigkeiten von jeder unterstützten VR-Brille vollständig kennen und erfährst, wie du sie verwenden kannst.

Die vier vollkommen unterschiedlichen Beispielprojekte am Ende des Buches fassen vieles, was du davor gelernt hast, nochmals in praktischen Beispielen zusammen. Jedes Beispielprojekt ist für eine der unterstützten VR-Brillen-Arten optimiert und wird gezielt für diese VR-Brillen erklärt. Auf diese Weise kann das Projekt die Brille vollständig ausnutzen. Nachdem du aber das Projekt für deine eigene VR-Brille erfolgreich realisiert hast, kannst du versuchen, eines der anderen Projekte ebenfalls für deine Brille umzusetzen. Hier musst du dann selbstständig überlegen, wie du das Spieleprinzip am besten für deine VR-Brille umsetzen kannst. Du findest am Ende der Kapitel zusätzlich aber auch ein paar Tipps und Tricks für diese Aufgabe.

■ 1.4 Unity

Unity ist eine Multiplattform-3D- und -2D-Engine, welche von Unity Technologies aus Dänemark entwickelt wird. Multiplattform bedeutet, du kannst mit ihr nicht nur für eine Plattform, zum Beispiel Windows-Computer, entwickeln, sondern auch für Mac, Linux, Smartphones, Browser und Videospielekonsolen. Unity verwendet hauptsächlich die einsteigerfreundliche Programmiersprache *C-Sharp (C#)*, welche der Sprache *Java* stark ähnelt. Falls du also schon mal mit Java gearbeitet hast, wird dir der Umstieg sehr leichtfallen. Dieses Buch setzt jedoch keine Programmierkenntnisse voraus.

Wir verwenden Unity, da es eine große Anzahl an verschiedenen Virtual-Reality-Brillen unterstützt, ohne dass man die Brillen selber integrieren muss. Außerdem ist es mit Unity möglich, ohne große Umwege, schnell erste Ergebnisse zu erzielen.

Unity ist schon seit Jahren in der Indie-Szene sehr beliebt und hat eine große, aktive und hilfsbereite Community. Wenn du also nicht weiterweißt, findest du wahrscheinlich schon nach kurzem Suchen eine hilfreiche Antwort. Solltest du keine passenden Antworten finden, kannst du entweder im offiziellen *Unity-Forum* oder auch auf Seiten wie *Stack Overflow* ein Posting erstellen und erhältst meist eine passende Antwort von anderen Entwicklern.

Bei vielen Hobby-Entwicklern sehr beliebt ist der sogenannte *Unity Asset Store*. Dieser Online-Shop ist in die Engine integriert und erlaubt es dir, schnell und einfach für die Unity-Engine optimierte Modelle, Plug-ins und mehr herunterzuladen. Es gibt viele kostenlose Inhalte, die du in deinem Projekt verwenden kannst, ohne dass du etwas dafür bezahlen musst. Dies ist besonders für Hobby-Projekte sehr hilfreich, da du dich so auf deine Leidenschaft konzentrieren kannst. Programmierst du lieber, kannst du dir 3D-Modelle herunterladen – bist du eher ein Designer, lädst du dir eine fertige Spiellogik und passt diese lediglich deinen Vorstellungen an.

1.4.1 Die verschiedenen Versionen von Unity

Unity wird mit drei unterschiedlichen Lizenzen „Personal“, „Plus“ und „Pro“ angeboten. Die vollkommen **kostenlose** „Personal“-Version wird für den Anfang wahrscheinlich genau die richtige für dich sein. Du kannst sie für Projekte, mit denen du kein Geld verdienst, frei verwenden. Die kostenlose Version bietet zudem bereits den vollen Umfang an Funktionen auf allen Plattformen. Die einzige für den Anfänger wahrnehmbare Einschränkung ist, dass bei jedem Start des Spiels das Unity-Logo während des Ladevorgangs angezeigt wird. Planst du deine Anwendung zu verkaufen oder Werbung in ihr zu schalten, kannst du die Personal-Version weiterhin verwenden, solange du weniger als 100.000 \$ pro Jahr damit verdienst. Sobald du so viel Geld mit deinem Spiel pro Jahr verdienst, macht es dir aber wahrscheinlich auch nichts aus, zu einer der beiden größeren Lizenzen zu wechseln.

Die Bezahlversionen sind als monatliche Abos für 35 \$ („Plus“) oder 125 \$ („Pro“) verfügbar und werden für dich erst wirklich interessant, wenn du sehr große Projekte angeht oder mehr als die erwähnten 100.000 \$ mit deinem Spiel verdienst. Es ist übrigens kein Problem, ein Projekt mit der kostenlosen Version zu starten und später zu einer der kostenpflichtigen Versionen zu wechseln, sobald du die Grenze erreichst.

Detaillierte Informationen zu den Unterschieden der einzelnen Lizenzen findest du auf der offiziellen Webseite von Unity:



Alle aktuellen Unity-Lizenzen im Überblick

Auf der offiziellen Unity-Webseite: <https://store.unity.com/>



Möchtest du ein Onlinespiel entwickeln, gibt es bei der Personal-Version die Einschränkung, dass maximal 20 Spieler gleichzeitig spielen können. Solltest du nur deswegen ein Upgrade in Erwägung ziehen, solltest du vorher einmal im Unity Asset Store nach alternativen Netzwerk-Add-ons suchen.

Wenn du nach Informationen über Unity-Lizenzen im Internet suchst, wirst du wahrscheinlich noch häufig auf veraltete Infos stoßen: Früher bot Unity Technologies eine kostenlose *Personal Edition* und eine kostenpflichtige *Professional Edition* an. Die Personal Edition war in ihrem Funktionsumfang stärker eingeschränkt. Zudem musste man für jede Plattform

(also PC, Android, iOS etc.) eine eigene Lizenz kaufen, um alle Funktionen auf der jeweiligen Plattform zu erhalten. Wie erwähnt, hast du aber mit dem neuen Lizenzmodell auch in der kostenlosen Version Zugang zu allen Funktionen auf allen Plattformen. Anders als heute konnte die alte Professional-Edition zu einem festen Preis und ohne Abo gekauft werden, dies ist mit dem neuen Lizenzmodell nicht mehr möglich.

■ 1.5 Was brauche ich?

Um diesem Buch folgen zu können, benötigst du eine Virtual-Reality-Brille sowie die entsprechende Hardware, um sie verwenden zu können. Außerdem benötigst du einen Computer oder Laptop, auf dem du Unity installieren kannst.

Folgst du dem Buch mit einer MobileVR-Brille wie *Google Daydream* oder *Samsung GearVR*, muss der Computer nicht *VR-Ready* sein, da du deine Spiele in erster Linie auf deinem Smartphone ausführen wirst.

Wenn du eine Oculus Rift- oder SteamVR-Brille verwendest, reicht ein Computer, der den Mindestanforderungen der jeweiligen VR-Brillen entspricht, auch zum Entwickeln aus.

Alle Abbildungen und Anleitungen in diesem Buch wurden auf einem Computer mit Windows 10 (64 Bit) erstellt.

Ich gehe in diesem Buch davon aus, dass du bereits erste Erfahrungen mit deiner Virtual-Reality-Brille gesammelt hast. Das bedeutet, du hast sie bereits vollständig eingerichtet und ggf. auch schon ein paar Spiele oder Anwendungen ausprobiert.

Du benötigst für dieses Buch kein Vorwissen über Unity, Softwareentwicklung oder Virtual Reality. Da es Unity nur in englischer Sprache gibt, wird es dir leichter fallen, damit zu arbeiten, wenn du einfache Ausdrücke und Sätze auf Englisch verstehst. Dies wird dir auch beim Lernen von Unitys Programmiersprache *C#* helfen, da diese ebenfalls englische Ausdrücke verwendet. Ich werde dir in diesem Buch aber ohnehin alle wichtigen Funktionen erklären, weshalb du auch mit weniger guten Englisch-Kenntnissen folgen kannst. In meiner Schulzeit hat das Entwickeln von Spielen nicht nur mein Englisch, sondern auch mein Interesse am Englisch-Unterricht deutlich verbessert.

Für alles, was in diesem Buch beschrieben wird, kannst du immer die kostenlosen Versionen der Programme verwenden. Du musst dir also keine zusätzliche Software kaufen.

Die meisten Inhalte dieses Buches können auf jedes Virtual-Reality-Headset übertragen werden, auch wenn die VR-Brille in diesem Buch nicht direkt beschrieben wird.

■ 1.6 Weiterentwicklung der Technik

Sowohl Unity als auch die beschriebenen Entwicklungswerkzeuge entwickeln sich stetig weiter. Aus diesem Grund kann es durchaus vorkommen, dass ein Menü oder ein Dialog in der neusten Version der Software nicht exakt genauso aussieht wie auf den Abbildungen im Buch. In den allermeisten Fällen ändern sich aber nur Kleinigkeiten und das Gesamtbild bleibt gleich, sodass du die gesuchte Funktion trotzdem schnell finden solltest.

Die Beispielprojekte in diesem Buch wurden mit *Unity Version 2017.1.0f3* erstellt. Um garantieren zu können, dass alles korrekt funktioniert, empfehle ich dir, diese Version ebenfalls zu verwenden. Wenn du einmal in Unity eingearbeitet bist, wird es dir leichter fallen, Neuerungen und Unterschiede zu erkennen und das Projekt entsprechend anzupassen.

Dasselbe gilt für alle verwendeten Entwicklerwerkzeuge und *Unity Packages*. Ich habe dir auf der begleitenden Webseite die Versionen der *Packages* bereitgestellt, sodass du bei dir alles exakt, wie im Buch, nachbauen kannst.

Du erfährst in diesem Buch natürlich trotzdem, wo du immer die neusten Updates herbekommst, falls du lieber mit diesen arbeiten möchtest. Ich empfehle dir, wenn du einmal mit einer Version von Unity und den Entwicklungswerkzeugen angefangen hast, bei diesen Versionen zu bleiben, bis du dich sicher im Umgang mit ihnen fühlst. Durch häufiges Updates können auch Probleme entstehen, welche dir eventuell nicht sofort auffallen, wenn du noch nicht sicher im Umgang mit der Software bist.

Mit neuen Updates kommen auch neue Funktionen in die Engine. Deswegen kann es natürlich auch vorkommen, dass ich mich über die Abwesenheit einzelner Funktionen beschwere, obwohl sie in der neusten Version bereits vorhanden sind. Wenn dies passiert, kannst du selber entscheiden, ob du lieber den Anweisungen im Buch folgst oder die neue Funktion ausprobierst.

■ 1.7 Online-Zusatzmaterial

Auf der Webseite zum Buch habe ich einen passwortgeschützten Bereich für dich vorbereitet. Dort findest du Zusatzmaterialien, die den Inhalt des Buches ergänzen und aktuell halten sollen. Außerdem findest du dort auch Downloads zu allen Materialien, die du für die Beispielprojekte in diesem Buch benötigst.



Zusatzmaterialien online:

Adresse: <http://www.VRSpieleEntwickeln.de/zusatz/>

Passwort: **VRUnity!**

Auf der Seite findest du unter anderem:

- Downloads zu den verwendeten Programmen und Tools in der Version, wie sie hier im Buch verwendet werden
- Grafiken, Modelle, Sounds etc. und jeweils ein passendes Beispielprojekt für folgende Spiele:
 - ein mittelalterliches Pfeil-und-Bogen-Spiel
 - einen Dungeon Crawler
 - eine Weltraumschlacht mit Raumschiffen
 - einen Sci-Fi-Stealth-Shooter

2

Einführung in Virtual Reality

Bevor es nun richtig losgeht, möchte ich dir zunächst ein wenig Hintergrundwissen geben, damit wir eine gemeinsame Basis haben, auf der wir aufbauen können. Das hat den Vorteil, dass ich in den kommenden Kapiteln nicht immer wieder weit ausholen muss. Besonders wichtig ist natürlich die Erklärung häufig verwendeter Begriffe. Außerdem möchte ich dir ein wenig Hintergrundwissen zur Geschichte von Virtual Reality und zu den einzelnen Headsets, mit denen wir uns in diesem Buch beschäftigen, geben. Zusätzlich werfen wir noch einen Blick über den Tellerrand auf VR-Headsets, mit denen wir uns in diesem Buch nicht gezielt beschäftigen.

■ 2.1 Begriffserklärungen

Ich möchte damit anfangen, dir ein paar Begriffe zu erklären, die dir im Umgang mit Virtual Reality und auch in der Spieleentwicklung immer wieder unterkommen werden. Du kannst in diesem Kapitel nachschlagen, wenn ich ein Wort benutze, unter dem du dir nichts vorstellen kannst.

Wort	Bedeutung
Assets	Das sind alle Medien oder auch „Ressourcen“ in deinem Projekt: Texturen, Sounds, 3D-Modelle, Materialien, Skripte etc.
Field of View, FoV	Beschreibt das Blickfeld in der Brille. Hierbei ist zu beachten, dass es sich meistens nicht auf das horizontale Blickfeld bezieht, sondern auf das diagonale. Zwei Headsets mit dem gleichen Field of View können daher trotzdem ein unterschiedliches Blickfeld in der Brille besitzen. Das eine Headset kann zum Beispiel ein besonders breites und das andere ein besonders hohes Blickfeld haben.
Frames	Ein Frame ist ein einzelnes auf dem Monitor oder Headset dargestelltes Bild, das von der Grafikkarte berechnet wurde.
Frames per Second, FPS	„Bilder pro Sekunde“, beschreibt die Anzahl an Bildern pro Sekunde, welche von der Grafikkarte an das jeweilige Ausgabemedium gesendet wird.

Wort	Bedeutung
Head-Mounted-Display	Wörtlich übersetzt ein „am Kopf befestigtes Display“. Diese Kategorie umfasst Virtual-Reality-Brillen, schließt aber auch Augmented-Reality-Brillen wie Microsoft <i>HoloLens</i> mit ein.
Head-Tracking	Bezeichnet die Positions- und Rotationserkennung des Kopfes im Raum.
Immersion	Beschreibt die Eintauchtiefe in die virtuelle Welt.
Locomotion	Beschreibt die Art und Weise der Fortbewegung, zum Beispiel normales „Laufen“ oder „Teleportieren“.
Low Persistence	Eine Technologie, die es ermöglicht, auch in der Bewegung scharfe Bilder in einem VR-Headset anzuzeigen. Hierbei wird das Bild nicht dauerhaft auf dem Display angezeigt, sondern blitzt z. B. 60- oder 90-mal pro Sekunde kurz auf, wenn der Frame gerade frisch vom Grafikprozessor berechnet wurde.
Motion-Controller	Das sind Eingabegeräte, die der Anwender für gewöhnlich in der Hand hält und deren Position und Rotation erkannt werden können.
Positional-Tracking	Bezieht sich im entsprechenden Kontext nur auf die Positionserkennung und nicht auf die Rotationserkennung.
Presence	Beschreibt den Zustand, in welchem die Immersion so hoch ist, dass man vergisst, dass die dargestellte Welt nicht real ist. Häufig verwendet man den Begriff „Micro Presence“, um Situationen zu beschreiben, wo dieser Zustand für einen sehr kurzen Moment erreicht wird.
Recentern	Das Zurücksetzen der virtuellen Position und Rotation in die Ausgangsstellung. Hierdurch wird bestimmt, welche reale Blickrichtung mit welcher virtuellen Blickrichtung übereinstimmt.
SDK	Steht für <i>Software Development Kit</i> . Das sind Entwicklungswerkzeuge, die meistens von dem Hersteller einer Hardware oder Software für externe Entwickler (dich) zur Verfügung gestellt werden; zum Beispiel das „Oculus SDK“ zum Entwickeln für Oculus-Brillen.
Thumbstick	Eine alternative Bezeichnung für den Analog-Stick, den man auf den meisten aktuellen Gamepads findet.
Tracking	Beschreibt ganz allgemein Positions- und Rotationserkennung von Gegenständen im 3D-Raum. Der Begriff kann sich also z. B. sowohl auf die Headsets als auch auf die Controller beziehen.
Virtual-Reality-Headset	Ein anderes Wort für Virtual-Reality-Brille
VR	Kurz für Virtual Reality

■ 2.2 Augmented Reality, Mixed Reality und Virtual Reality

Wenn du gerade erst anfängst, dich mit dem Thema zu beschäftigen, stößt du im Zusammenhang mit Virtual Reality wahrscheinlich auch häufig auf den Begriff „*Augmented Reality*“. Augmented Reality ist eine vollkommen eigenständige Technologie mit individuellen Einsatzmöglichkeiten und hat mit Virtual Reality nicht direkt zu tun. Zusätzlich gibt es noch den Begriff „*Mixed Reality*“, welcher mehr eine Oberkategorie als eine bestimmte Technologie beschreibt.



Bild 2.1 Mixed Reality

Bild 2.1 zeigt dir eine Grafik zur groben Orientierung. Bei den einzelnen Begriffen ist nicht definiert, ob man sich die jeweilige Realität über eine Brille (Head Mounted Display) oder zum Beispiel auf einem Smartphone-Bildschirm ansieht. Innerhalb der einzelnen Kategorien werden für die jeweiligen Darstellungsformen dann weitere Unterscheidungen vorgenommen.

Augmented Reality beschreibt eine Technologie, welche die echte Welt mit virtuellen Informationen erweitert. Das bedeutet zum Beispiel, dass du dir eine reale Speisekarte in einem Restaurant ansiehst, und wenn du mit deinem Finger auf eines der Menüs zeigst, erscheint eine dreidimensionale Darstellung des Menüs vor dir. Diese Darstellung kann entweder über AR-Brille, ein Smartphone, welches das Kamerabild augmentiert, oder auch über ein Hologramm erzeugt werden.

Virtual Reality beschreibt eine Technologie, bei der die Welt vollständig virtuell ist und mit der auf eine realistische Art und Weise interagiert werden kann. Du wärst also in einem virtuellen Restaurant und bestellst von einer virtuellen Speisekarte ein virtuelles Essen, ohne in der Realität etwas zu erhalten. Auch Virtual Reality muss nicht zwingend mit einer VR-Brille angesehen werden: Sogenannte „*CAVEs*“ sind spezielle Räume, bei denen die Wände (aber nicht zwingend alle Wände) durch mehrere Beamer angestrahlt werden und den Anwender auf diese Weise in die virtuelle Welt versetzen.

Bei **Augmented Virtuality** wird die Darstellung einer virtuellen Welt mit Elementen aus der echten Welt überlagert. Du sitzt also zum Beispiel in einem virtuellen Restaurant, siehst und isst aber ein reales Essen, das auch wirklich vor dir steht. Von der Realität siehst du allerdings nur das Essen und deine Hände, nicht aber den Raum, in dem du dich tatsächlich befindest. Falls du von diesem Begriff noch nie etwas gehört hast, liegt das höchstwahrscheinlich daran, dass Augmented Virtuality häufig auch einfach als „Virtual Reality“ bezeichnet wird.

Bei den Begriffen geht es also immer darum, was das Ausgangsmedium ist: Bei Augmented Reality bist du von der realen Welt umgeben, die teilweise mit virtuellen Elementen angereichert wird. Bei Virtual Reality bist du von der virtuellen Welt umgeben und es werden höchstens einzelne Gegenstände aus der realen Welt dargestellt. Die Grenzen zwischen den einzelnen Arten sind jedoch fließend und rein technisch kann man noch viele weitere Unterscheidungen vornehmen. Man findet also durchaus Fälle, die nicht so einfach auf den Strahl von Bild 2.1 einsortiert werden können.

■ 2.3 Virtual Reality ist tot – lang lebe Virtual Reality

In den letzten Jahren hat Virtual Reality begonnen, richtig durchzustarten, doch Virtual-Reality-Hardware existiert schon viel länger. Es gab sogar schon eine richtige Welle an Virtual-Reality-Headsets. Vielleicht kennst du Nintendos berühmt-berüchtigten Virtual Boy? Dieser war nur die Spitze des Eisberges an Virtual-Reality-Headsets in den späten 80ern und 90ern. In diesem Kapitel möchte ich dir gerne ein paar Informationen zur ersten großen Virtual-Reality-Welle mitgeben.

Der Begriff „Virtual Reality“ wurde 1982 durch Damien Brodericks Roman *The Judas Mandala* geprägt, das Konzept einer Virtual-Reality-Brille existierte, wurde jedoch schon 1934 erstmals beschrieben. Noch bevor es die ersten Computer gab, beschrieb Stanley G. Weinbaum in seiner Kurzgeschichte *Pygmalion's Spectacles* ein auf Brillen basierendes System mit holografischen Aufnahmen von fiktiven Ereignissen.

Mit dem Aufkommen von Arcade-Hallen, Heimcomputern und den ersten flachen LCD-Bildschirmen begann auch die Forschung an Virtual-Reality-Headsets für Heimanwender. 1980 forschte Atari zum Beispiel an einem Virtual-Reality-Arcade-Automaten, musste die Forschung 1983 allerdings wegen des Video-Game-Crashes¹ einstellen. 1985 entwickelte die NASA eine Virtual-Reality-Workstation namens *VIEW*. Die *Virtual Environment Workstation* umfasste sogar schon einen Datenhandschuh, mit dem virtuelle Objekte gegriffen werden konnten.



Mehr Details zur NASA Virtual-Reality-Workstation findest du hier:

https://www.nasa.gov/ames/spinoff/new_continent_of_ideas/

¹ Zwischen 1983 und 1985 brach der amerikanische Videospiegelmarkt zusammen und viele amerikanische Firmen gingen bankrott oder wendeten sich von der Videospiegelindustrie ab. Als Symbol des Crashes wird häufig der Niedergang des zuvor langjährigen Marktführers Atari verwendet. Erst das 1985 in den USA erscheinende Nintendo Entertainment System belebte den amerikanischen Videospiegelmarkt wieder. In den folgenden Jahren dominierten jedoch japanische Hersteller wie Nintendo, Sega und Sony den weltweiten Videospiegelmarkt.