

Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

Ihr mitp-Verlagsteam



Games mit Roblox

Eigene Spiele entwickeln
mit Roblox Studio

Übersetzung aus dem Amerikanischen
von Knut Lorenzen



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.

ISBN 978-3-7475-0438-3

1. Auflage 2022

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

Authorized translation from the English language edition, entitled ROBLOX DEVELOPMENT IN 24 HOURS: THE OFFICIAL ROBLOX GUIDE 1st Edition by THE ROBLOX CORPORATION published by Pearson Education, Inc, © 2021 Roblox Corporation. "Roblox," the Roblox logo, and "Powering Imagination" are among the Roblox registered and unregistered trademarks in the U.S. and other countries.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

GERMAN language edition published by MITP VERLAGS GMBH & CO. KG, Copyright © 2022.

© 2022 mitp Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt 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.

Lektorat: Janina Bahlmann

Fachkorrektur: Tanja Köhler

Sprachkorrektur: Petra Heubach-Erdmann

Covergestaltung: Chuti Prasertsith, Christian Kalkert

Satz: Petra Kleinwegen

Inhaltsverzeichnis

Vorwort	17
Über die Autorin	19
Über die weiteren Mitwirkenden	19
Über die Fachkorrektorin der deutschen Ausgabe	20
1 Was ist das Besondere an Roblox?	21
1.1 Roblox ermöglicht es, soziale Kontakte zu knüpfen	22
1.1.1 Roblox als soziale Website	23
1.1.2 Roblox als Treffpunkt für Entwickler	23
1.2 Roblox verwaltet Benutzerinhalte	24
1.2.1 Inhalte organisieren	25
1.2.2 Deine eigene Identität erschaffen	26
1.2.3 Charaktere personalisieren	27
1.3 Roblox ermöglicht schnelle Prototyperstellung und Iteration	29
1.3.1 Bereit für Anpassungen	29
1.4 Ideen einfach umsetzen	30
1.4.1 Plugins	31
1.4.2 Veröffentlichung und Updates ohne Wartezeit	31
1.5 Die Funktionen der Roblox-Engine	32
1.5.1 Vernetzung	32
1.5.2 Physik	33
1.5.3 Rendering	34
1.5.4 Plattformübergreifende Entwicklung	34
1.6 Alles kostenlos	35
1.7 Unbegrenzte Möglichkeiten	36
1.8 Finde deinen eigenen Stil	36
1.9 Zusammenfassung	38
1.9.1 Fragen und Antworten	38
1.9.2 Workshop	38
1.9.3 Aufgaben	39
2 Verwendung von Roblox Studio	41
2.1 Installation von Roblox Studio	41
2.1.1 Fehlerbehebung bei der Installation	42
2.1.2 Roblox Studio starten	43

2.2	Verwendung von Studio-Templates	44
2.2.1	Die Registerkarte All Templates	44
2.2.2	Themes	45
2.2.3	Gameplay	46
2.3	Verwendung des Game-Editors	47
2.3.1	Einrichtung des Arbeitsbereichs im Game-Editor	49
2.3.2	Verwendung des Explorer-Fensters	50
2.3.3	Erstellen eines Parts	51
2.3.4	Verwendung des Properties-Fensters	52
2.4	Verschiebung, Skalierung und Ausrichtung von Objekten	53
2.4.1	Verschieben	54
2.4.2	Skalieren	55
2.4.3	Drehen	56
2.4.4	Transformieren	57
2.5	Snapping	58
2.6	Collisions	58
2.7	Verankern	59
2.8	Speichern und Veröffentlichung deines Projekts	60
2.8.1	Speichern deines Projekts	60
2.8.2	Veröffentlichung deines Projekts	61
2.8.3	Projekt wieder öffnen	61
2.9	Spieltest	62
2.9.1	Spieltest durchführen	63
2.9.2	Spieltest beenden	63
2.10	Zusammenfassung	64
2.10.1	Fragen und Antworten	64
2.10.2	Workshop	65
2.10.3	Aufgaben	65
3	Verwendung von Parts	67
3.1	Erstellen eines Parts	67
3.2	Aussehen eines Parts ändern	67
3.2.1	Farbe	69
3.2.2	Material	69
3.2.3	Reflexionsgrad und Transparenz	71
3.3	Decals und Texturen erstellen	72
3.3.1	Decals	73
3.3.2	Texturen	76
3.4	Zusammenfassung	80
3.4.1	Fragen und Antworten	80
3.4.2	Workshop	80
3.4.3	Aufgaben	81

4	Physik-Engine	83
4.1	Verwendung von Attachments und Constraints	84
4.2	Erstellen einer Tür	86
4.3	CanCollide deaktivieren, damit ein Spieler die Tür durchschreiten kann	89
4.4	Scharniere und Federn hinzufügen	90
4.4.1	Öffnen einer Tür mit Scharnieren	90
4.4.2	Erstellen der Federn	94
4.4.3	Realistisches Verhalten von Federn	96
4.5	Verwendung eines Motors	98
4.6	Zusammenfassung	101
4.6.1	Fragen und Antworten	101
4.6.2	Workshop	101
4.6.3	Aufgaben	102
5	Landschaften gestalten	105
5.1	Der Terrain-Editor	106
5.2	Die Registerkarte Edit	109
5.2.1	Verändern des Terrains mit dem Substract-Werkzeug	111
5.2.2	Terrain anheben mit dem Grow-Werkzeug	112
5.2.3	Terrain abtragen mit dem Erode-Werkzeug	113
5.2.4	Terrain mit dem Smooth-Werkzeug glätten	113
5.2.5	Terrain mit dem Flatten-Werkzeug eibnen	114
5.2.6	Materialien ändern mit dem Paint-Werkzeug	115
5.2.7	Wasserflächen erzeugen mit dem Sea-Level-Werkzeug	117
5.3	Die Registerkarte Region	118
5.3.1	Terrain auswählen	119
5.3.2	Terrain mit dem Move-Werkzeug verschieben	120
5.3.3	Terrain mit dem Resize-Werkzeug skalieren	122
5.3.4	Verwendung der Werkzeuge Copy, Paste und Delete	123
5.3.5	Eine Fläche mit dem Fill-Werkzeug ausfüllen	123
5.4	Height Maps und Color Maps	124
5.4.1	Height Maps	124
5.4.2	Color Maps	126
5.5	Zusammenfassung	127
5.5.1	Fragen und Antworten	127
5.5.2	Workshop	128
5.5.3	Aufgaben	129

6	Beleuchtung	131
6.1	Eigenschaften der Beleuchtung	132
6.1.1	Eigenschaften des Erscheinungsbilds	134
6.1.2	Die Eigenschaften Data und Exposure	136
6.2	Beleuchtungseffekte	137
6.2.1	SpotLight, PointLight und SurfaceLight	140
6.3	Zusammenfassung	143
6.3.1	Fragen und Antworten	143
6.3.2	Workshop	143
6.3.3	Aufgaben	144
7	Atmosphäre	147
7.1	Eigenschaften der Atmosphäre	148
7.1.1	Density	149
7.1.2	Offset	150
7.1.3	Haze	151
7.1.4	Color	152
7.1.5	Glare	153
7.1.6	Decay	154
7.2	Anpassen der Skybox	156
7.2.1	Erstellen einer Skybox	156
7.2.2	Anpassen der Himmelskörper	159
7.2.3	Anpassen der Beleuchtungsfarben	160
7.3	Zusammenfassung	162
7.3.1	Fragen und Antworten	162
7.3.2	Workshop	162
7.3.3	Aufgabe	163
8	Effekte	165
8.1	Partikel	165
8.1.1	Anpassen der Partikel	167
8.1.2	Ändern der Farbe von Partikeln	168
8.1.3	Eigenschaften eines ParticleEmitters	169
8.2	Beams	170
8.2.1	Krümmung	172
8.2.2	Segments	174
8.2.3	Width	175
8.2.4	Lichtern mit einem Beam einen Strahleneffekt hinzufügen	176

8.3	Zusammenfassung	177
8.3.1	Fragen und Antworten	177
8.3.2	Workshop	178
8.3.3	Aufgaben	178
9	Objekte importieren	183
9.1	Einfügen und Hochladen kostenloser Modelle	183
9.1.1	Hochladen des Modells bei Roblox	185
9.1.2	Zugriff auf Modelle	187
9.1.3	Einfügen kostenloser Modelle	187
9.2	Importieren mit MeshParts und Asset Manager	189
9.2.1	Mehrere Meshes auf einmal importieren mit dem Asset Manager	192
9.3	Importieren von Texturen	194
9.3.1	Importieren von Decals mit dem Asset Manager	196
9.4	Sounds importieren	196
9.5	Zusammenfassung	197
9.5.1	Fragen und Antworten	198
9.5.2	Workshop	198
9.5.3	Aufgaben	199
10	Spielstruktur und Zusammenarbeit	201
10.1	Einem Spiel Places hinzufügen	201
10.2	In Roblox Studio zusammenarbeiten	203
10.2.1	Zusammenarbeit bei Gruppenspielen	204
10.2.2	Konfiguration der Rollen	204
10.2.3	Zuweisung der Rollen	205
10.2.4	Team Create aktivieren	205
10.2.5	Hinzufügen und Verwalten von Benutzern in Team Create	206
10.2.6	Zugriff auf die Team-Create-Sitzung	208
10.2.7	Verwendung von Roblox Studio Chat	209
10.2.8	Team Create deaktivieren	209
10.3	Erstellen von Roblox-Packages und darauf zugreifen in Roblox Studio	210
10.3.1	Objekte in Packages konvertieren	210
10.3.2	Zugriff auf die Package-Toolbox	212
10.3.3	Zugriff auf Packages im Asset Manager	212
10.3.4	Aktualisieren eines Packages	213
10.3.5	Eine größere Anzahl von Packages aktualisieren	214

10.4	Zusammenfassung	215
10.4.1	Fragen und Antworten	216
10.4.2	Workshops	216
10.4.3	Aufgaben	216
11	Lua – ein Überblick	219
11.1	Der Coding Workspace	220
11.1.1	Erstellen des ersten Scripts	220
11.2	Eigenschaften ändern mithilfe von Variablen	222
11.2.1	Überblick über Variablen	222
11.2.2	Verwendung von Variablen	223
11.2.3	Erstelle eine halb durchsichtige Bombe	223
11.3	Kommentare zum Code hinzufügen	224
11.4	Funktionen und Events	226
11.4.1	Erstellen einer Funktion	226
11.4.2	Verwendung einer Funktion, um eine Bombe explodieren zu lassen	227
11.4.3	Der Einsatz von Events	228
11.4.4	Verwendung eines Events, um einen Part explodieren zu lassen, wenn er berührt wird	228
11.5	Verwendung bedingter Anweisungen	229
11.6	Arrays und Dictionaries	230
11.7	Verwendung von Schleifen	231
11.8	Gültigkeitsbereiche	234
11.9	Benutzerdefinierte Events	235
11.10	Debugging des Codes	237
11.10.1	Debugging mit Strings	237
11.10.2	Lua-Debugger	237
11.10.3	Log-Dateien	238
11.11	Zu einem besseren Spieleentwickler werden	239
11.12	Zusammenfassung	240
11.12.1	Fragen und Antworten	240
11.12.2	Workshop	241
11.12.3	Aufgaben	241
12	Kollisionen, Humanoide, Punktzahl	243
12.1	Kollisionen	243
12.1.1	CollisionFidelity	243
12.1.2	Anzeige und Verbesserung der Kollisionsgeometrie	244
12.1.3	Collision Groups Editor	245
12.1.4	Direkte Verwendung des Collision Groups Editors	246
12.1.5	Verwendung des Collision Group Editors per Script	247

12.2	Kollisionserkennung	249
	12.2.1 Verwendung von Touched	249
	12.2.2 debounce	250
12.3	Humanoid	253
	12.3.1 Humanoid innerhalb der Hierarchie	253
	12.3.2 Eigenschaften, Funktionen und Events	254
12.4	Zusammenfassung	262
	12.4.1 Fragen und Antworten	262
	12.4.2 Workshop	263
	12.4.3 Aufgaben	263
13	Interaktion mit der Benutzeroberfläche (GUI)	265
13.1	Erstellen von GUIs	266
	13.1.1 PlayerGui	266
	13.1.2 SurfaceGui	270
13.2	Grundlegende GUI-Elemente	274
13.3	Programmierung interaktiver GUIs	274
13.4	Tweening	277
13.5	Layouts	278
13.6	Erstellen einer GUI mit Countdown	282
13.7	Zusammenfassung	283
	13.7.1 Fragen und Antworten	283
	13.7.2 Workshop	284
	13.7.3 Aufgaben	285
14	Programmierung von Animationen	287
14.1	Position und Rotation	287
	14.1.1 Bewegen eines Objekts von Punkt A nach Punkt B	289
	14.1.2 Rotation von Parts mit CFrames	291
14.2	Ruckelfreies Bewegen von Objekten mit Tween	295
	14.2.1 Tweening zwischen zwei Punkten	296
	14.2.2 EasingStyle und EasingDirection	297
14.3	Bewegen des gesamten Modells	298
14.4	Zusammenfassung	300
	14.4.1 Fragen und Antworten	300
	14.4.2 Workshop	300
	14.4.3 Aufgaben	301

15	Sounds und Musik	303
15.1	Einen Soundtrack erstellen	303
15.2	Importieren von Musik und Sounddateien	305
15.3	Umgebungsgeräusche hinzufügen	306
15.4	Sounds via Code abspielen	308
15.5	Gruppieren von Sounds	309
15.6	Zusammenfassung	311
	15.6.1 Fragen und Antworten	311
	15.6.2 Workshop	312
	15.6.3 Aufgaben	313
16	Animation-Editor	315
16.1	Einführung in den Animation-Editor	316
	16.1.1 Anforderungen an das Modell	316
	16.1.2 Öffnen des Animation-Editors	317
16.2	Erstellen von Posen	318
16.3	Speichern und Exportieren von Animationen	322
16.4	Easing	323
16.5	Inverse Kinematik	324
	16.5.1 IK aktivieren	324
	16.5.2 Parts anheften	326
16.6	Einstellungen für Animationen	327
	16.6.1 Wiederholungen	327
	16.6.2 Priorität	328
16.7	Animation-Events	328
	16.7.1 Events hinzufügen	329
	16.7.2 Verschieben und Löschen von Events	330
	16.7.3 Events klonen	330
	16.7.4 Implementierung von Events in Scripts	330
	16.7.5 Standardanimationen ersetzen	332
16.8	Zusammenfassung	333
	16.8.1 Fragen und Antworten	334
	16.8.2 Workshop	334
	16.8.3 Aufgaben	335
17	Kämpfe, Teleportation und Datenspeicher	337
17.1	Einführung in Tools	337
	17.1.1 Grundlagen	338
	17.1.2 Ein Tool erstellen	339
	17.1.3 Tool-Handle	339
	17.1.4 Tool-Ausrichtung	341

17.2	Teleportation	347
17.2.1	Teleportation innerhalb eines Place	348
17.2.2	Teleportieren zwischen verschiedenen Places	350
17.2.3	Spieluniversen	350
17.3	TeleportService	351
17.3.1	Funktionen	351
17.3.2	Abrufen der PlaceID	352
17.3.3	Beispiel: Client	352
17.3.4	Beispiel: Server	353
17.4	Verwendung des dauerhaften Datenspeichers	356
17.4.1	Unterstützte Datentypen und Beschränkungen	356
17.5	Funktionen des Datenspeichers	360
17.5.1	UpdateAsync() und SetAsync()	362
17.6	Fehlerbehandlung	362
17.6.1	Was ist ein pcall?	362
17.6.2	Schutz vor Datenverlust	363
17.7	Zusammenfassung	363
17.7.1	Fragen und Antworten	363
17.7.2	Workshop	364
17.7.3	Aufgaben	364
18	Mehrspieler-Code und das Client-Server-Modell	367
18.1	Das Client-Server-Modell	367
18.1.1	Scripts und LocalScripts	368
18.1.2	Replikation	368
18.2	Was sind RemoteFunctions und RemoteEvents?	369
18.2.1	Verwendung von RemoteEvent und RemoteFunction	370
18.2.2	Erstellen eines RemoteEvents	371
18.3	Serverseitige Validierung	374
18.4	Teams	375
18.4.1	Hinzufügen von Teams	375
18.4.2	Automatische Zuordnung von Spielern zu einem Team	376
18.4.3	Manuelle Zuordnung von Spielern zu einem Team	377
18.5	Network-Ownership	378
18.6	Zusammenfassung	378
18.6.1	Fragen und Antworten	379
18.6.2	Workshop	379
18.6.3	Aufgaben	380

19	Module-Scripts	383
19.1	Kurz vorgestellt: Das Module-Script	383
19.1.1	Aufbau eines Module-Scripts	384
19.1.2	Hinzufügen von Code, der überall verwendet werden kann	385
19.1.3	Verwendung eines Module-Scripts	385
19.2	Clientseitige und serverseitige Module-Scripts	387
19.3	Der Einsatz von Module-Scripts: Game-Loop	389
19.3.1	Einstellungen	390
19.3.2	Erstellen wiederverwendbarer Funktionen für Spielrunden	391
19.3.3	Die Game-Loop	391
19.4	Zusammenfassung	394
19.4.1	Fragen und Antworten	394
19.4.2	Workshop	394
19.4.3	Aufgaben	395
20	Programmierung von Kamerabewegungen	397
20.1	Einführung in die Verwendung von Kameras	397
20.1.1	Kameraeigenschaften	399
20.1.2	Handhabung der Kamera	400
20.2	Programmierung einer Kamerabewegung	401
20.3	Verwendung von Render-Step	402
20.4	Versetzen der Kamera	403
20.4.1	Dauerhafte Verknüpfung mit dem Render-Step	406
20.4.2	deltaTime	408
20.5	Zusammenfassung	409
20.5.1	Fragen und Antworten	409
20.5.2	Workshop	409
20.5.3	Aufgaben	410
21	Plattformübergreifende Entwicklung	413
21.1	Verbesserung der Performance	413
21.1.1	Speicherbedarf	413
21.1.2	Optimierung	414
21.1.3	Vereinfachung des physikalischen Verhaltens	416
21.1.4	Inhalte streamen	416
21.1.5	Diverse weitere Optimierungen	417
21.2	Verbesserung der Scripts	418
21.2.1	Zuweisung des Parents bei Objekten	418
21.2.2	Blindes Vertrauen in Server/Client	419
21.2.3	Schleifen sparsam verwenden	419

21.3	Kompatibilität mit Mobilgeräten	420
21.3.1	Erscheinungsbild	420
21.3.2	Steuerung	421
21.3.3	Simulation von Mobilgeräten	422
21.4	Spielkonsolen und VR	424
21.4.1	Xbox-Richtlinien	424
21.4.2	VR Best Practices	425
21.5	Zusammenfassung	425
21.5.1	Fragen und Antworten	426
21.5.2	Workshop	426
21.5.3	Aufgaben	427
22	Globale Community	429
22.1	Einführung in Lokalisierung	429
22.1.1	Texte für die Übersetzung erfassen	429
22.1.2	Übersetzung der erfassten Texte	431
22.1.3	Einsetzen der Übersetzung	432
22.2	Globale Regelkonformität	433
22.3	Datenschutzgesetze: DSGVO und CCPA	434
22.3.1	Allgemeine Richtlinien	435
22.3.2	Löschen von Spielerdaten	435
22.4	Zusammenfassung	437
22.4.1	Fragen und Antworten	437
22.4.2	Workshop	438
22.4.3	Aufgaben	439
23	Monetarisierung	441
23.1	Game Pass: Einmalige Käufe	441
23.2	Game Passes im Spiel verkaufen	443
23.2.1	Game-Pass-Vorteile aktivieren	444
23.3	Developer Products: Consumables	445
23.4	Roblox Premium	448
23.5	Developer Exchange: Verdiane richtiges Geld mit deinem Spiel ...	450
23.6	Zusammenfassung	452
23.6.1	Fragen und Antworten	452
23.6.2	Workshop	453
23.6.3	Aufgaben	454
24	Spieler auf das Spiel aufmerksam machen	457
24.1	Icons, Vorschaubilder und Trailer	457
24.2	Updates	461

Inhaltsverzeichnis

24.3	Anzeigen und Benachrichtigungen	461
24.3.1	Sponsoranzeigen	461
24.3.2	Benutzeranzeigen	464
24.3.3	Benachrichtigungen	466
24.4	Analytics	468
24.5	Zusammenfassung	469
24.5.1	Fragen und Antworten	469
24.5.2	Workshop	469
24.5.3	Aufgaben	470
A	Lua-Scripting	473
A.1	Änderung von Eigenschaften (Datentyp und Enumerationen)	473
A.2	Bedingte Anweisungen und Verzweigungen	474
A.3	Ausbau der Lua-Kenntnisse	476
B	Eigenschaften und Funktionen von Humanoid	477
	Stichwortverzeichnis	481



Vorwort

Stelle dir ein virtuelles Universum vor, das von einer weltweiten Community entwickelt wurde, der Künstler, Programmierer, Geschichtenerzähler und die verschiedensten anderen Leute angehören. In diesem Traum würden Menschen aus allen Teilen der Welt zusammenkommen, um Millionen Erlebnisse zu erschaffen und mit ihren Freunden zu teilen und um voneinander zu lernen. Es wäre ein Universum, das auf der Vorstellungskraft beruht, in dem alles möglich wäre, unabhängig vom verwendeten Gerät, dem Aufenthaltsort oder der Zeit. Und wenn ich nun feststelle, dass es diese digitale Utopie schon seit mehr als einem Jahrzehnt gibt?

Als Erik Cassel und ich 2004 Roblox mitgründeten, wollten wir einen immersiven, dreidimensionalen, physikalisch simulierten Raum für mehrere Spieler erschaffen, mit dem sich jeder verbinden kann, um zusammen mit anderen Spaß zu haben. In den Anfangstagen von Roblox waren wir davon fasziniert, was die Leute alles anstellten. Sie wollten ihr eigenes Restaurant managen, eine Naturkatastrophe überleben oder erfahren, wie es ist, ein Vogel zu sein. Wenn ich nun, siebzehn Jahre später, in die Zukunft blicke, ist es offensichtlich, dass diese Plattform noch viel mehr zu bieten hat.

Roblox stellt eine neue Kategorie für Erlebnisse mit anderen Menschen dar, in der die Grenzen zwischen Spiel, sozialem Netzwerk und Medien verschwimmen. Unser Team hat festgestellt, dass sich die Millionen täglichen Roblox-User nicht nur zum Spielen anmelden, sondern sich zusammenfinden, um eine Community aufzubauen, Geschichten zu erzählen und sowohl mit Freunden als auch mit Fremden etwas zu erleben.

Wir fahren damit fort, unsere Plattform weiterzuentwickeln, die es Milliarden von Benutzern ermöglicht, Erfahrungen mit anderen zu teilen – es gab nie einen besseren Zeitpunkt, einer weltweiten Community kreativer Menschen beizutreten, die so erstaunliche Beiträge zu unserer Plattform leisten. 3-D-Erlebnisse zu entwickeln, macht nicht nur Spaß, sondern bietet auch die Möglichkeit, Fähigkeiten und Kenntnisse zu sammeln, die für eine Laufbahn in den Bereichen Informatik, Design, Kunst und vielen anderen erforderlich sind. Viele der besten Entwickler unserer Plattform haben das Geld, das sie mit Roblox verdient haben, verwendet, um ihr Studium zu finanzieren, ein eigenes Studio für Spieleentwicklung zu gründen oder um eine Anzahlung für das Haus ihrer Eltern zu leisten.

Ich bin davon überzeugt, dass Roblox letzten Endes zur Entstehung des *Metaverse* führt, einer allumfassenden digitalen Realität, die durch unsere physische ergänzt wird. Wir sehen den Tag kommen, an dem die Menschen Roblox nicht nur aufsuchen, um zu spielen und Kontakte zu knüpfen, sondern auch, um geschäftliche Treffen abzuhalten oder die Schule zu besuchen. Die Möglichkeiten, die das *Metaverse* bietet, nehmen mit jedem Tag zu. Das gilt auch für den Bedarf an innovativen und kreativen Entwicklern, die für die Erlebnisse sorgen, von denen wir in der Science-Fiction seit Jahren träumen.

Ich lade dich herzlich dazu ein, an der Welt von Roblox teilzunehmen, nicht nur als Spieler, sondern auch als Entwickler. Zu lernen, Spiele und immersive 3-D-Erlebnisse zu entwerfen, trägt dazu bei, weltweit Millionen von Menschen durch die Spiele zu verbinden und eine Community aufzubauen, die nicht durch Grenzen, Sprachen oder Geografie definiert ist. Wenn du am Programmieren, an Game Design oder der immersiven 3-D-Welt von Roblox interessiert bist, solltest du dieses Buch lesen und deine verrücktesten und kreativsten Ideen verfolgen. Das *Metaverse* braucht Entwickler wie dich.

Deine Fantasie wartet schon!

*David »Builderman« Baszucki
Gründer und Geschäftsführer
Roblox Corporation*

Über die Autorin



Genevieve Johnson ist als Senior Instructional Designer bei Roblox tätig, der weltweit größten sozialen Spiele-Plattform mit benutzergenerierten Inhalten. Sie betreut die Erstellung von Bildungsinhalten und berät Pädagogen rund um den Globus dabei, wie man Roblox in STEAM-basierten Lernprogrammen einsetzen kann. Ihre Arbeit ermöglicht es Schülern, Laufbahnen als Unternehmer, Ingenieur und Designer einzuschlagen. Bevor sie zu Roblox kam, war Johnson für bildungsrelevante Inhalte bei ID Tech verantwortlich, einem technischen Ausbildungsprogramm, das jährlich mehr als 50.000 Schülerinnen und Schüler zwischen 6 und 18 Jahren nutzen. Bei ID Tech war sie an der Einrichtung eines erfolgreichen STEAM-Programms nur für Mädchen beteiligt. Ihr Team hat Ausbildungsinhalte für mehr als 60 technologiebezogene Kurse für eine Vielfalt von Themen entwickelt, von der Programmierung über Robotik bis hin zum Game Design.

Über die weiteren Mitwirkenden

Ashan Sarwar ist ein Roblox-Entwickler, der seit 2013 Roblox Studio verwendet. Von ihm ist *LastShot*, ein Shooter, der auf Roblox verfügbar ist.

Raymond Zeng ist ein Roblox-Entwickler, der das Programmieren liebt und andere Programmierer auf jedem Niveau unterrichtet. Unter der Bezeichnung *MacAndSwiss* betreibt er einen YouTube-Kanal, auf dem er Lua unterrichtet, über Neuigkeiten zu Roblox berichtet und seine Programmierprojekte vorstellt.

Theo Docking ist seit vier Jahren als Spieleprogrammierer tätig. Er mag es, an spannenden Projekten zu arbeiten, die Roblox richtig ausreizen und dabei interessante Menschen zu treffen. Außerdem liebt er es, mit Roblox' Physik-Engine zu experimentieren und Code für NFCs, Autos und mehr zu schreiben. Wenn er nicht programmiert, zeichnet er Pläne für neue Spiele oder spielt *Ultimate Driving*, um sich zu entspannen.

Joshua Wood hat Roblox 2013 entdeckt und ein Jahr später angefangen, seine eigenen Spiele zu erstellen. Er ist der Entwickler von *Game Dev Life*, das mehr als eine Million mal gespielt wurde. Zudem ist er Entwickler von *DoubleJGames*.

Swathi Sutrave ist ein selbsternannter Technik-Geek. Sie war als Expertin für verschiedene Programmiersprachen, unter anderem auch Lua, für Unternehmen, Start-ups und Universitäten tätig.

Henry Chang ist ein Computergrafikdesigner, der sich verschiedener Medien bedient, unter anderem zwei- und dreidimensionaler Grafiken und Animationen. Er ist Autodidakt und hat verschiedene Möglichkeiten interaktiver Medien ausgelotet. Weitere Informationen sind unter <https://www.henrytcgweb.com> zu finden.

Über die Fachkorrektorin der deutschen Ausgabe



Tanja Köhler hat Architektur studiert und war für verschiedene Architekturbüros im In- und Ausland tätig. Als 3D-Artistin vertiefte sie ihre 3D-Software-Kenntnisse in der Computerspiele-Branche bei Opus Studio Inc. (Tokio) und Bluebyte/Ubisoft. Unter anderem visualisierte sie die Gebäude für *Die Siedler VI – Aufstieg eines Königreiches*. Heute lebt und arbeitet sie als freie Architektin bei München und ist Lehrbeauftragte an der TU München. Dort unterrichtet sie Studierende der Architekturinformatik und Games Engineering. Zudem betreibt sie ein Studio für Programmierung und Computerspiele-Design für Kinder und Jugendliche; weitere Informationen unter www.nextlevel-gauting.de.

Was ist das Besondere an Roblox?

Die Themen in diesem Kapitel:

- Wie Roblox soziale Kontakte ermöglicht
- Verwaltung von Benutzerinhalten in Roblox
- Schnelle Prototyperstellung und Iteration
- Die Bestandteile der Roblox-Engine

Willkommen zu Roblox! Bei dieser kostenlosen 3-D-Online-Plattform zur Erstellung von Spielen sind dir nur durch deine Vorstellungskraft Grenzen gesetzt. Jeder kann in dieser dynamischen und verknüpften sozialen Umgebung Spiele erschaffen und spielen. Es stehen Millionen verschiedene Welten zur Auswahl, die alle von Benutzern erstellt wurden, die jeweils ein einzigartiges Gameplay haben, ein individuelles Design besitzen und über eigenständige Communities verfügen. In diesem Abschnitt wirst du mehr über die Features erfahren, die Roblox dir bietet, um damit anzufangen, dein eigenes handwerklich gut gemachtes Spiel zu erstellen. Die Kenntnis der Stärken von Roblox wird dir dafür den richtigen Weg weisen.

Roblox ist eine vollständig eigenständige Plattform. Bei anderen Engines benötigt man als Entwickler Engine-Code, eine Moderation für die Nutzer und mehr. Roblox übernimmt all diese Aufgaben für dich. Die Infrastruktur erledigt die unangenehmen Arbeiten wie den Betrieb der Server und die Vernetzung mehrerer Spieler. Dadurch hast du mehr Zeit, um deine Ideen umzusetzen. Zum Spielen und Erstellen von Spielen benötigst du nur ein Benutzerkonto.

Die Roblox-Plattform wird täglich von Millionen Besuchern genutzt und übernimmt für sie alle die Moderation. Sie unterstützt plattformübergreifendes Spielen und verfügt über eine fest eingebaute Währung namens *Robux*. Auf diese Weise können sich die Entwickler auf das Wichtigste konzentrieren, nämlich die Umsetzung neuer, fesselnder Spielerlebnisse für Benutzer rund um den Globus (Abbildung 1.1).



Abb. 1.1: Willkommen bei Roblox!

Zur Veröffentlichung von Spielen benötigst du keine jahrelange Programmiererfahrung, keine teuren Softwarelizenzen und du musst auch kein kompliziertes und umständliches Einreichungsverfahren durchlaufen. Alles, was du brauchst, sind ein PC oder Mac, eine stabile Internetverbindung, Roblox Studio sowie etwas Fantasie und schon kann dein Spiel in wenigen Minuten veröffentlicht werden – völlig kostenlos.

Wenn die Anzahl deiner Spiele und ihrer Spieler zunimmt, kannst du die mit den Einkäufen von Spielern verdienten Robux in echtes Geld umtauschen.

1.1 Roblox ermöglicht es, soziale Kontakte zu knüpfen

Bei Roblox wird Wert auf soziale Kontakte gelegt. Die Palette der Spiele reicht von äußerst detailreichen gemütlichen Welten, in denen man sich gerne mit anderen trifft, bis hin zu Spielen, bei denen man es vor Spannung kaum aushält.

Und weil Roblox plattformübergreifend funktioniert, kann jeder mit einem PC, einem Mac oder einem Mobilgerät mitmachen. Niemand wird ausgeschlossen. Um soziale Kontakte zu fördern, bietet Roblox die Möglichkeit, Freunde hinzuzufügen, miteinander zu chatten und sich auszutauschen. Die Spieler können gemeinsam Welten erkunden, Rätsel lösen und sogar Auftritte ihrer Lieblingsmusiker ansehen.

1.1.1 Roblox als soziale Website

Roblox-Benutzer können Gruppen anlegen, um sich untereinander zu organisieren. Jede Gruppe verfügt über eine eigene Seite (Abbildung 1.2), auf der sie ihre Spiele bereitstellen, virtuelle Artikel verkaufen und Verbindung zu anderen Gruppen herstellen kann, wobei sie immer eigenständig bleibt. Zudem ist es üblich, dass Gruppen als eine Art Treffpunkt verwendet werden, über den verschiedene Teams ihre Ressourcen für die Entwicklung miteinander teilen können.

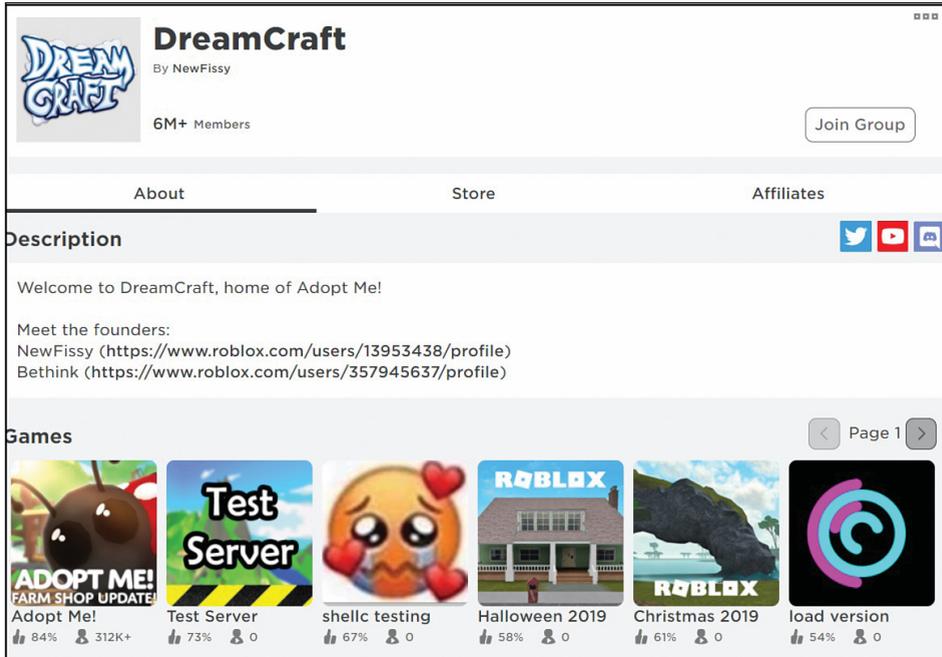


Abb. 1.2: Die Seite einer Gruppe, die Spiele entwickelt, mit Links zu externen sozialen Medien

1.1.2 Roblox als Treffpunkt für Entwickler

Roblox bietet Entwicklern die Möglichkeit, sich miteinander in Verbindung zu setzen. Das Roblox-Entwicklerforum ist ein Online-Forum, in dem Roblox-Entwickler verschiedene Themen diskutieren und Fähigkeiten untereinander austauschen können.¹ Um dem Entwicklerforum beizutreten, musst du es nur regelmäßig besuchen und die dortigen Inhalte lesen. Nachdem du dort genügend Zeit mit dem Herumstöbern und dem Lesen der Ressourcen verbracht hast, wirst du automatisch vom Besucher (*visitor*) zum Mitglied (*member*) oder sogar zum erfahrenen Entwickler (*community sage*) hochgestuft und kannst dann eigene Beiträge schreiben.

¹ Im Entwicklerforum werden alle Beiträge auf Englisch veröffentlicht.

Kapitel 1

Was ist das Besondere an Roblox?

Es gibt ein Offline-Treffen, das nur für eingeladene Entwickler zugänglich ist: die *Roblox Developers Conference*. Auf dieser Konferenz halten ausgewählte Entwickler und Mitglieder des offiziellen Roblox-Teams Vorträge und präsentieren ihre Fortschritte, die sie mit der Plattform erzielt haben. Du bist auf deinem Weg zum Entwickler also nicht allein!

Die *Annual Bloxy Awards* sind ein jährlich stattfindendes live übertragenes Event, bei dem die Spiele im Vordergrund stehen und das den sozialen Charakter von Roblox hervorhebt. Bei diesem Ereignis werden die besten Roblox-Entwickler durch Abstimmung der Community nominiert und mit einem einzigartigen virtuellen Pokal ausgezeichnet. Weitere Informationen zu diesen Veranstaltungen kannst du im offiziellen Roblox-Blog (Abbildung 1.3) unter <https://blog.roblox.com> oder im Entwicklerforum nachlesen.

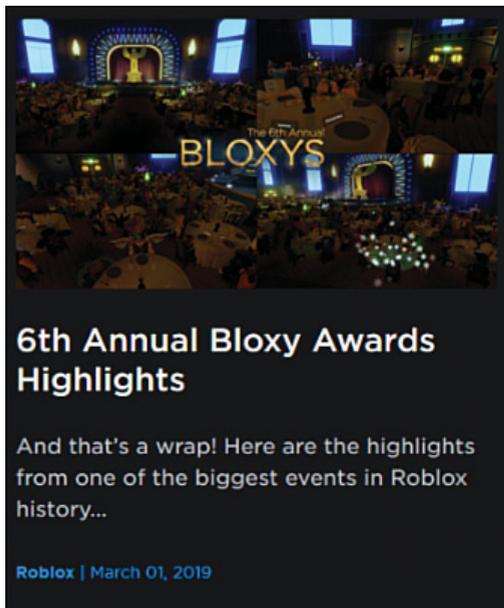


Abb. 1.3: Ein Blog-Eintrag zu den 6. Annual Bloxy Awards 2019

1.2 Roblox verwaltet Benutzerinhalte

Was die Inhalte betrifft, die Spieler entwerfen dürfen, ist Roblox ziemlich freigiebig. Es steht dir also frei, deine wildesten Fantasien umzusetzen. Fast alles bei Roblox ist an ein Benutzerkonto gebunden: Spiele, Zubehör, Plugins und Medieninhalte von Spielen sind nur einige der Dinge, die Benutzer hochladen können. Den Benutzern steht es fast vollständig frei, zu entscheiden, was sie in ihren Spielen verwenden möchten.

Alle bei Roblox hochgeladenen Inhalte werden moderiert

Alle Inhalte bei Roblox werden von Moderatoren geprüft, bevor sie Spielern angezeigt werden dürfen. Zudem können Benutzer darauf hinweisen, wenn ihnen etwas unangemessen erscheint, damit es markiert und gelöscht werden kann. Das betrifft nicht nur deine Spiele und deren Inhalte, sondern auch dein Benutzerkonto. Du solltest dir Roblox' Vorschriften und die Nutzungsbedingungen unter <https://en.help.roblox.com/hc/de/articles/115004647846> durchlesen, um weitere Informationen zur Moderation zu erhalten.

1.2.1 Inhalte organisieren

Für bestimmte Inhalte und Produkte, die auf die Website hochgeladen werden, gibt Roblox die Organisationsstruktur vor. Von dir auf der *Create*-Seite (Abbildung 1.4) erstellte Dinge und alles, was du dir beschafft hast, gehören zu deinem Inventar (*inventory*).

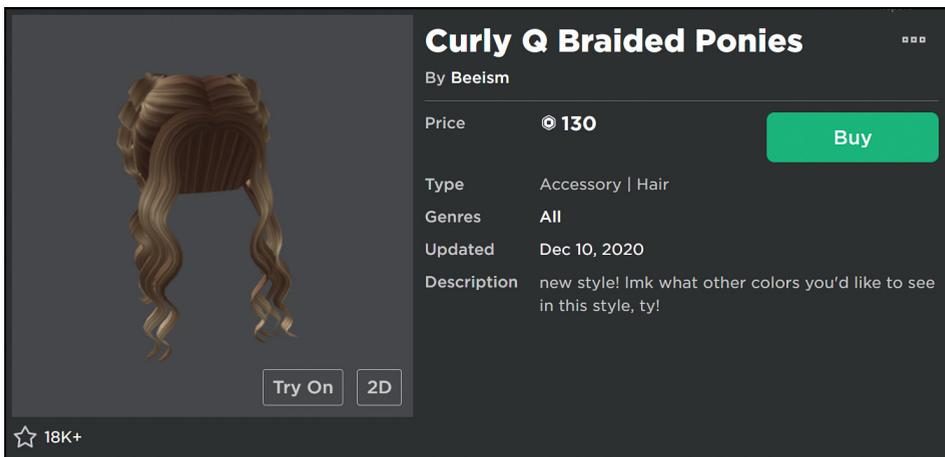


Abb. 1.4: Die Create-Seite mit Entwicklerressourcen nebst Links zu Roblox Studio, dem offiziellen Entwickler-Wiki und dem Entwicklerforum (Community)

Von der *Create*-Seite aus kannst du auch auf die *Asset-Library* (Objekt-Bibliothek, Abbildung 1.5) zugreifen, die beispielsweise Modelle, Decals, Audio, Meshes und Plugins bereithält.

Kapitel 1

Was ist das Besondere an Roblox?

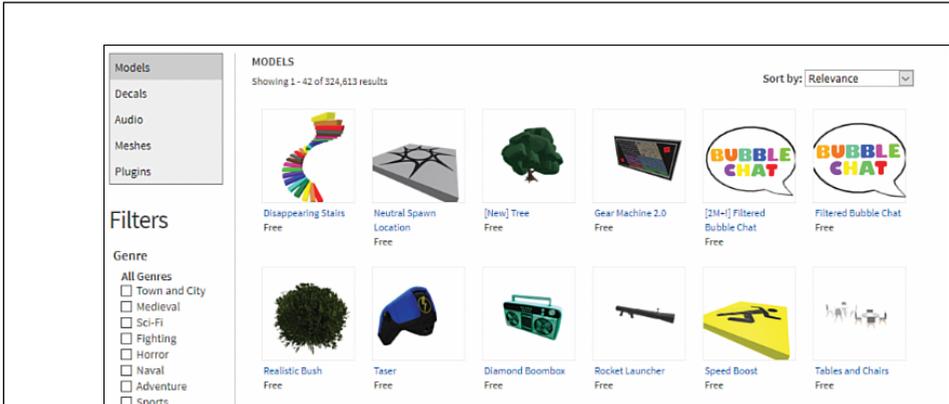


Abb. 1.5: Die Asset-Library, in der Assets angezeigt werden

1.2.2 Deine eigene Identität erschaffen

Vergiss nicht, dass Roblox eine soziale Website ist. Die Entwickler können Bilder erstellen und sie hochladen, beispielsweise Icons, Miniaturbilder und sogar Werbebanner, um sie auf der Website anzuzeigen (Abbildung 1.6). Du kannst ein Bildbearbeitungsprogramm deiner Wahl verwenden, um eigene Bilder zu erstellen und sie hochladen. Dieses Feature verleiht Gruppen, Spielen und Charakteren bei Roblox eine ganz individuelle Identität.

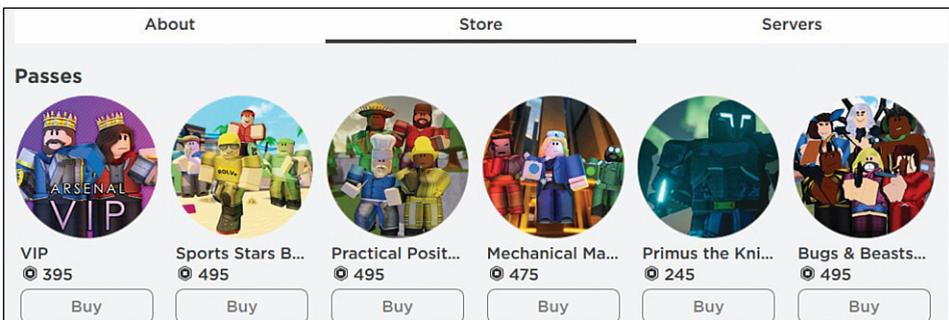


Abb. 1.6: Von Benutzern auf die Registerkarte STORE des Spieleanbieters Arsenal für Mikrotransaktionen hochgeladene Bilder (Arsenal von ROLVe Community)

1.2.3 Charaktere personalisieren

Im Avatar-Shop, der auch als *Catalog* (Abbildung 1.7) bezeichnet wird, können Benutzer virtuelle Objekte für ihre Avatare erwerben, beispielsweise Hüte, Köpfe, Ausrüstungsgegenstände, Zubehör und weitere Dinge. Eigentlich ist der offizielle Roblox-Account für den Avatar-Shop verantwortlich, die Erstellung von Pullis, T-Shirts und Hosen lag aber schon immer in den Händen der Community. Sie wird als wesentlicher Bestandteil von Fashion-Design- und auf Individualisierung spezialisierter Gruppen betrachtet. Und da die Inhalte des Avatar-Shops Objekte mit eigenen IDs sind, können sie von Entwicklern mit dem Studio in ihre Spiele eingebunden werden.

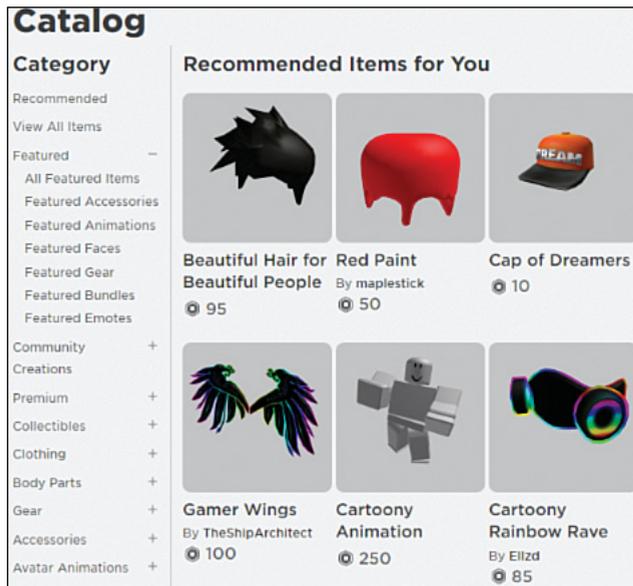


Abb. 1.7: Kategorien und einige Objekte im Catalog des Avatar-Shops

Abbildung 1.8 zeigt das Inventar eines Spielers mit Teilen des Catalogs, der Objekte enthält, die er erstellt hat und die ihm gehören, wie beispielsweise Gegenstände aus der Asset-Library und dem Avatar-Shop.

Kapitel 1

Was ist das Besondere an Roblox?

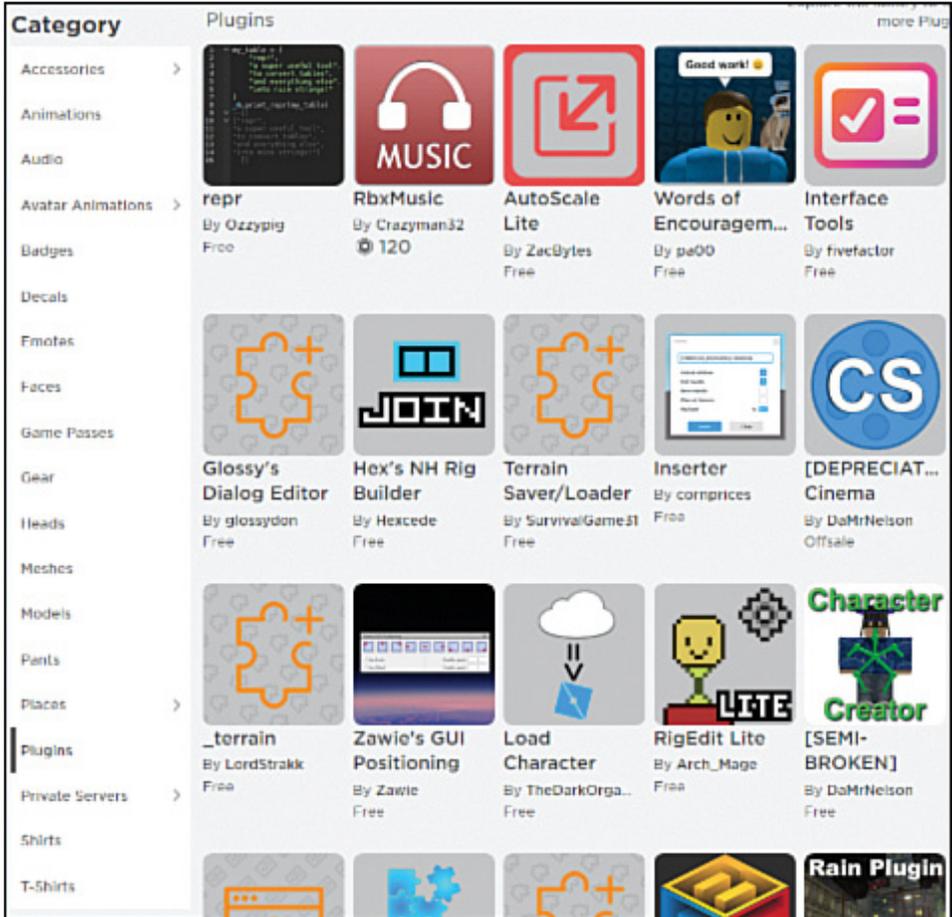


Abb. 1.8: Das Inventar eines Spielers

Seit August 2019 haben ausgewählte Spieler die Möglichkeit, ihre eigenen Hüte in der Kategorie für Zubehör hochzuladen. Sie sind die ersten Benutzer, die eigene Hüte hochladen können. (Im Laufe der Zeit wird dieses Feature weiteren Benutzern zur Verfügung gestellt.) Bei manchen Hüten ist zum Kauf eine Premium-Mitgliedschaft erforderlich.

Roblox bietet zwei offizielle Avatare (sogenannte »Rigs«), die ursprüngliche namens R6 und die neuere R16 Rthro. Beide können Objekte aus dem Avatar-Shop verwenden. Benutzerdefinierte Rigs können mithilfe von Modellen hochgeladen werden, allerdings kann es erforderlich sein, sie mit Roblox Studio zu bearbeiten, damit sie vollständig kompatibel zu deinem Spiel sind.

1.3 Roblox ermöglicht schnelle Prototyperstellung und Iteration

Roblox ist eine flexible Engine, die dir möglichst viel Zeit für das Umsetzen deiner Ideen gibt. Roblox-Spiele verwenden eine Programmiersprache namens *Lua*, die nicht kompiliert werden muss. Du kannst in Sekundenschnelle vom Programmieren zum Testen wechseln. Darüber hinaus verfügt Roblox über eine Befehlsleiste, die Fehlermeldungen ausgibt und die während des Spiels angezeigt werden kann, was bei der Fehlersuche sehr hilfreich ist. Wenn du ein neues Spiel mit Roblox Studio erstellst, stehen dir Kontrollsysteme für Spieler, Ausrüstung, Animationen, Bewegungssteuerung, Beleuchtung, Multiplayermodus und bestimmte Features der Benutzeroberfläche zur Verfügung. Roblox Studio stellt diese Werkzeuge bereit, um damit zu experimentieren, zudem gibt es einige externe Werkzeuge und weitere nützliche Software. Dein Roblox-Spiel kann darauf aufbauen oder völlig neu beginnen.

1.3.1 Bereit für Anpassungen

Roblox Lua ermöglicht es, schon vorhandene Eigenschaften anzupassen. Die Eigenschaften legen fest, wie Objekte aussehen und funktionieren. Solche Eigenschaften sind bei vielen Klassenobjekten vorhanden. Zu den üblichen Eigenschaften gehören beispielsweise die Form, die Farbe oder auch das Material, aus dem ein Objekt besteht.

Betrachte beispielsweise diese einfache Form, die als *Part* (Bauteil) bezeichnet wird. In Abbildung 1.9 sind die Eigenschaften dargestellt, unter anderem die Farbe (*Medium stone grey*, mittelgrau) und das Material (*Plastic*, Plastik).

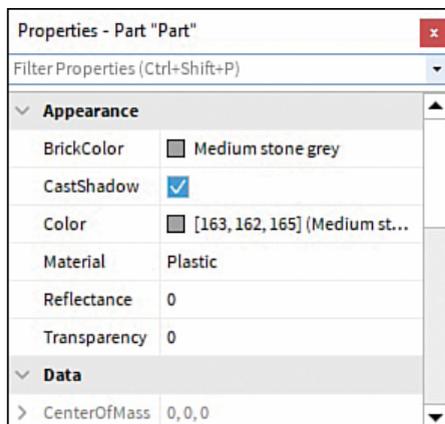


Abb. 1.9: Einige der Part-Eigenschaften in Roblox Studio

Die Werte der Eigenschaften, die nicht ausgegraut sind, können durch den Programmcode oder vom Benutzer geändert werden. Du kannst nicht nur die Eigenschaften von 3-D-Objekten ändern, sondern auch von allen anderen Objekten, die du im Studio platzieren und verwenden kannst, wie beispielsweise `PARTICLEEMITTER` (Partikelerzeuger) oder `USERINTERFACEFRAMES` (benutzerdefinierte Rahmen). Wenn du weißt, welche Eigenschaften in Roblox verwendet werden können, fällt es dir leichter, dir komplexere Spiele auszudenken. In der Einführung in Kapitel 2, Verwendung von Studio, gehen wir näher auf Parts und Eigenschaften ein.

1.4 Ideen einfach umsetzen

Es gibt kostenlose Objekte, die dir dabei helfen, mit der Toolbox loszulegen. Sie werden gestreamt und in Echtzeit geladen, damit du sie nicht installieren musst. Du kannst sie mit anderen veränderlichen einfachen Bauteilen kombinieren, beispielsweise mit Blöcken und Kugeln, um neue Ideen auszuprobieren oder um zu experimentieren (Abbildung 1.10).



Abb. 1.10: Eine noch blockartige Testumgebung aus Parts, kostenlosen Objekten und veränderten Einstellungen

Wenn du nicht nur Blöcke verwenden möchtest, kannst du Umgebungen mit dem Terrain-Editor (Abbildung 1.11) von Roblox gestalten, der auch Begrünung ermöglicht.