2. Auflage
aktualisiert & erweitert

Chris Strom

Übersetzung von Kathrin Lichtenberg

Kids programmieren 3D-Spiele mit JavaScript

Spannende 3D-Welten und coole Spielideen

o'reillys basics

O'REILLY®

2., aktualisierte und erweiterte Auflage

KIDS PROGRAMMIEREN 3D-SPIELE MIT JAVASCRIPT

Chris Strom

Deutsche Übersetzung von Kathrin Lichtenberg

O'REILLY®

Chris Strom

Lektorat: Ariane Hesse
Übersetzung: Kathrin Lichtenberg
Korrektorat: Sibylle Feldmann, richtiger-text.de
Satz: III-satz, www.drei-satz.de
Herstellung: Stefanie Weidner
Umschlaggestaltung: Michael Oréal, www.oreal.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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:
Print 978-3-96009-096-0
PDF 978-3-96010-261-8
ePub 978-3-96010-262-5
mobi 978-3-96010-263-2

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

2., aktualisierte und erweiterte Auflage 2019

Wieblinger Weg 17
69123 Heidelberg

Authorized German translation of the English edition of 3D Game Programming for Kids:
Create Interactive Worlds with JaveScript by Chris Strom Copyright © 2018 The Pragmatic Programmers, LLC.
All rights reserved. ISBN 978-1680502701

5 4 3 2 1 0

Für Elsa und alle Prinzessinnen,
die die Welt verändern werden

INHALT

DANKSAGUNG

Ohne meine wunderbare Frau Robin wäre ich nichts. Sie war die alleinige Lektorin der frühen Versionen dieses Buchs. Obwohl sie selbst genug zu tun hat, liest sie jedes Kapitel, gibt mir wertvolle Hinweise und macht Vorschläge. Sie hilft bei der Durchführung der Kid Hackathons, die zur Entwicklung dieses Buchs beitrugen (okay, sie führt sie durch). Und ja – sie ist eine fantastische Ehefrau und Mutter.

Ein riesiges Dankeschön geht auch an meine Kinder als die wichtigsten Versuchskaninchen für dieses Buch. Sie haben mich gezwungen, den langweiligen Kram zu streichen und mehr lustige und interessante Sachen aufzunehmen. Danke, Kinder.

Und natürlich geht ein großer Dank an meine technischen Gutachter. Es ist nicht leicht, ein Buch aus der Sicht eines Kindes zu bewerten, doch meine Gutachter waren dieser Aufgabe mehr als gewachsen. Danke, Ana B., Doug C., Bryson S., Cedric H., Keeley L., Paul Callaghan, Rob Donoghue, Kevin Gisi, Ron Hale-Evans, Brian Hochgurtel, Brian Hogan, Chaim Krause, Nick McGinness, James Sterrett und Jeremy Sydik.

Dieses Buch wäre ohne die großartige Arbeit von Ricardo Cabello Miguel, von allen liebevoll »Mr. doob« genannt, nicht entstanden. Ricardo ist der

wichtigste Programmierer von Three.js, der 3D-JavaScript-Bibliothek, die wir in diesem Buch benutzen. Er schrieb außerdem die ursprüngliche Implementierung des 3DE Code Editor, die wir hier verwenden. Ohne seine unglaublichen Fähigkeiten wäre dieses Buch nicht zu dem geworden, was es ist. Danke auch an Chandler Prall für seine Arbeit an der Physijs-Physik-Engine, die wir hier ausgiebig einsetzen.

Und zu guter Letzt möchte ich noch den Leuten bei *The Pragmatic Programmers* danken, dass sie an das Buch geglaubt und mir geholfen haben, sein volles Potenzial auszuschöpfen. Ein besonderer Dank geht an meine Lektorin, Adaobi. Eine zweite Auflage ist keine einfache Aufgabe, aber sie hat mich in der Spur gehalten und mir geholfen, meinen Stil zu verbessern.

EINLEITUNG

Herzlich willkommen in der Welt der Programmierung!

Ich will nicht lügen – es ist manchmal eine frustrierende Welt (ich muss wenigstens einmal in der Woche einen Wutanfall unterdrücken). Aber der ganze Frust lohnt sich. Du kannst diese Welt nach deinen Vorstellungen gestalten. Du kannst deine Welt mit anderen teilen. Du kannst Dinge bauen, die wirklich etwas bewegen.

Dieses Buch ist eine großartige Möglichkeit, um mit dem Programmieren zu beginnen. Wieso? Weil es die Meinung vertritt, dass Spielen die beste Methode darstellt, um das Programmieren zu lernen. Ja, natürlich, es gibt ein paar Kapitel, in denen Grundlagen beschrieben werden, aber dazu kommen wir erst, nachdem wir ein bisschen Spaß hatten. Wir beginnen also gleich in Kapitel 1 mit einigen ziemlich coolen 3D-Animationen.

Das wird richtig klasse!

Wie ich programmieren gelernt habe (und warum das für dich wichtig ist)

Als Kind habe ich die Programme für Computerspiele aus Büchern kopiert. Das ist schon lange her. Ich kaufte Bücher, in denen nichts weiter als die

Programme zu lesen waren, und tippte sie in den Computer ein. Als ich damit startete, hatte ich keine Ahnung, was ich eigentlich tat.

Schließlich fing ich an, bestimmte Dinge zu erkennen, und begann damit, Dinge zu ändern – zuerst nur Kleinigkeiten –, um zu sehen, was passierte. Danach kamen größere Änderungen. Schließlich wurde ich recht gut darin. Und irgendwann konnte ich meine eigenen Programme schreiben.

Ich hoffe, dass dieses Buch auch dir das ermöglicht, allerdings mit einem großen Unterschied: Ich erkläre dir, was passiert, sodass du nicht so viel raten musst.

Wie DU programmieren lernen kannst

Jeder ist anders. Jeder lernt anders.

Deshalb gibt es mindestens drei verschiedene Möglichkeiten, wie du mithilfe dieses Buchs lernen kannst:

1. Spiele mit dem coolen Kram herum und lies dann ab und zu ein Kapitel über die Grundlagen.
2. Lerne die Grundlagen kennen und mache dann was Cooles mit dem, was du weißt.
3. Tippe einfach den Code ab (wie ich es als Kind gemacht habe).

Du entscheidest, was für dich am besten ist.

Falls du zunächst spielen möchtest (die erste Möglichkeit), fängst du mit Kapitel 1 an und gehst dann der Reihe nach das restliche Buch durch. Du wirst sicher vor allem an den Spielen und Simulationen in den Projektkapiteln arbeiten – also den Kapiteln, deren Titel mit dem Wort »Projekt« beginnen –, gefolgt von gelegentlichen Kapiteln zu den grundlegenden Fertigkeiten. Wenn du dir nicht sicher bist, was am besten für dich ist, entscheide dich für diesen Weg. Ich wünschte, ich hätte auch so gelernt.

Bist du eher der Typ, der die Grundlagen verstehen möchte, bevor er sich an größere Dinge wagt (die zweite Möglichkeit), dann lies zuerst die Grundlagenkapitel – alle Kapitel ohne »Projekt« im Titel. Hier gibt es immer noch eine Menge Code und in einigen auch 3D-Spaß. Verglichen mit anderen Programmiersprachen, ist der Kern von JavaScript recht klein. Bereits durch das Lesen dieser Kapitel kannst du 80 bis 90 Prozent des Kerns von JavaScript kennenlernen. Eine andere Sache ist es, dies dann auch wirklich anwenden zu können – hier kommen die Projektkapitel ins Spiel!

Falls du nur Code schreiben möchtest, blättere zu Anhang 1, *Der Projektcode*, vor. Dort findest du den gesamten Code aller Spiele. Solltest du irgendwo hängen bleiben, schau in das Kapitel, aus dem der Code stammt.

Dort findest du dann eine genauere Erklärung. So habe ich gelernt, und im Großen und Ganzen war das okay.

Egal für welche Möglichkeit du dich entscheidest, es gibt eine wichtige Regel: *Tippe den Code immer von Hand ein*. Das geht viel langsamer, als ihn zu kopieren und einzufügen. Du wirst viel eher Fehler machen, wenn du den Code selbst abtippst.

Aber langsam vorzugehen und Fehler zu machen, ist genau der Sinn dahinter!

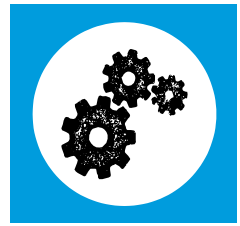
Wenn du den Code von Hand eintippst, denkst du viel mehr darüber nach, was du da gerade schreibst. Du glaubst vielleicht, dass du ebenso gut lernen kannst, wenn du dir alles durchliest und dann kopierst und einfügst, aber das stimmt auf keinen Fall. Selbst wenn du bereits seit 50 Jahren programmierst, solltest du niemals nur Code kopieren und einfügen. Beim Tippen hast du viel mehr Zeit, nachzudenken und den Code zu verstehen, den du schreibst. Das ist *viel* wichtiger, als schnell fertig zu werden.

Fehler gehören zum Programmieren dazu. In der Lage zu sein, fehlerhaften Code zu reparieren, ist genauso wichtig wie das Programmieren selbst. Also, mach ruhig Fehler. Das ist fies und frustrierend. Aber es lohnt sich.

Und damit haben wir auch schon den ersten Tipp in diesem Buch erreicht! Sehr wichtige Tipps und Hinweise sind im Buch besonders hervorgehoben. Achte darauf, denn sie können dir wirklich weiterhelfen, wenn du diese Welt erkundest.

Code immer eintippen – niemals kopieren und einfügen!

Gute Programmierer tippen Code immer von Hand ein – niemals kopieren sie ihn einfach und fügen ihn dann ein. Es ist viel wichtiger, Computerprogramme gut zu schreiben, als sie schnell zu schreiben. Gutes Programmieren bedeutet, über den Code nachzudenken, den du erzeugst. Es bedeutet, den Code zu verstehen. Deshalb tippe ihn ein!



Und noch eine Sache: Falls du nicht weiterkommst, kannst du immer um Hilfe bitten!

Hilfe bekommen

Jeder Programmierer braucht Hilfe. Wenn du noch nie im Leben programmiert hast, brauchst du Hilfe. Und wenn du schon seit 50 Jahren programmierst, brauchst du immer noch Hilfe.

Die zwei Regeln für das Bitten um Hilfe:

1. Versuche zuerst, es selbst herauszukriegen.
2. Fürchte dich nicht davor, um Hilfe zu bitten.

Für einen Programmierer ist es *nicht* die wichtigste Fertigkeit, eine Programmiersprache richtig gut zu kennen. Am wichtigsten ist das Lösen von Problemen. Wenn man alles über eine Programmiersprache weiß, heißt das noch lange nicht, dass man einen zu100 Prozent problemfreien Code erzeugt. Es hilft sicher, aber trotzdem wird es zu Problemen kommen. Oft sogar. Versuch deshalb, deine Probleme selbst zu verstehen. Auch wenn du ein Problem nicht lösen kannst, verbesserst du deine Fähigkeit, mit Problemen zurechtzukommen.

Apropos Fähigkeit, Probleme zu lösen: Kapitel 2, *Projekt: Einfache Formen herstellen*, ist der beste Startpunkt. In diesem Kapitel beschreibe ich häufig auftretende Fehler, wie man Probleme mit dem Codeeditor löst und wie man etwas rettet, wenn die Dinge wirklich schiefgehen.

Du darfst Kapitel 2 nicht weglassen!

Das Lösen von Problemen ist unglaublich wichtig. Deshalb musst du **Kapitel 2** unbedingt lesen.

Egal wie du lernst, die Debugging-Fertigkeiten, die in **Kapitel 2** beschrieben werden, sind ein Muss. Es mag vielleicht so aussehen, als könntest du diesen Kram überspringen, aber du würdest ja auch nicht ohne Polster Eishockey spielen oder ohne Sicherheitsgurt Auto fahren. Die Dinge scheinen ohne Schutz oder Debugging-Fertigkeiten in Ordnung zu sein. Das gilt aber nur so lange, bis etwas Schlimmes passiert.

Es ist völlig in Ordnung, wenn du ein Problem nicht selbst lösen kannst. Ich helfe dir gern. Für dieses Buch gibt es eine eigene Website, auf der du den gesamten Code sowie ein (englischsprachiges) Forum der Buchcommunity findest.[1] Falls du nicht weiterkommst, stelle deine Frage im Forum. Ich antworte fast immer innerhalb von 24 Stunden. Beschreibe aber auf jeden Fall, wie du bereits versucht hast, das Problem zu lösen – das wird sonst meine erste Frage an dich sein.

Und bitte, ich möchte, dass du Erfolg hast. Stelle Fragen. Nutze das Forum. Programmieren wir!

Was du für dieses Buch brauchst

Du brauchst den Webbrowser Google Chrome und einen relativ neuen Computer. Das war's schon!

Nicht alle Webbrowser können die coolen 3D-Spielobjekte erzeugen, die wir in diesem Buch bauen werden. Um das meiste aus dem Buch herauszuholen, solltest du auf deinem Computer den Webbrowser Google Chrome[2]

1 *https://talk.code3Dgames.com/*
2 *https://www.google.com/chrome/*

installieren. Andere Webbrowser funktionieren auch, allerdings greifen einige der Übungen in diesem Buch auf Eigenschaften und Funktionen zurück, die es nur in Google Chrome gibt.

Jeder moderne Computer (nicht älter als fünf Jahre), auf dem Google Chrome installiert ist, reicht aus. Du kannst die Fähigkeiten deines Computers testen, indem du die Get-WebGL-Site[3] besuchst. Sollte dir diese Site nicht sagen, dass dein Computer WebGL unterstützt, solltest du vielleicht besser einen anderen Computer ausprobieren.

Was ist JavaScript?

Es gibt viele, viele Programmiersprachen. Manche Programmierer führen gern lange Streitgespräche darüber, welche die beste ist, in Wahrheit aber bieten alle Sprachen einzigartige und nützliche Dinge.

Wir benutzen in diesem Buch die Programmiersprache JavaScript. Wir programmieren in JavaScript, weil es die Sprache des Webs ist. Es ist die einzige Programmiersprache, die alle Webbrowser verstehen. Wenn du in JavaScript programmieren gelernt hast, kannst du nicht nur solche Spiele herstellen, die du in diesem Buch kennenlernen wirst, sondern kannst auch alle möglichen Websites programmieren.

Was ist neu in der zweiten Auflage?

Dies ist die zweite Auflage von »Kids programmieren 3D-Spiele mit JavaScript«. Die erste Auflage war großartig. Man hat mir gesagt, ich sei voreingenommen, aber das sehe ich nicht so. Ich bin ziemlich sicher, dass die erste Auflage nahezu perfekt war.

Genau, Chris, wenn sie perfekt war, wieso bringst du dann eine zweite Auflage heraus?

Nun, zum einen ist seit der ersten Auflage des Buchs eine Menge passiert. Die Welt der Programmierung ändert sich laufend. Es kommt ständig neuer Kram heraus. Das meiste davon hilft uns nicht beim Lernen, aber manchmal gibt es tatsächlich etwas, das uns entscheidend weiterbringt. Drei Jahre nach dem Erscheinen der ersten Auflage gab es so viel hilfreiches neues Zeug, dass ich überzeugt war, ich könnte die zweite Auflage sogar noch besser machen als die erste.

Der andere Grund für eine zweite Auflage besteht darin, dass sich seit der ersten Auflage viel geändert hat. Als Programmierer darf man niemals auf-

3 *http://get.webgl.org/*

hören zu lernen. Ich programmiere jetzt seit 15 Jahren und bemühe mich darum, so viele neue Dinge zu lernen wie möglich. Ich habe außerdem hart dafür gearbeitet, ein besserer Programmierer, Lehrer und Autor zu werden. Das Lernen hilft mir sogar dabei, bessere Bücher zu schreiben!

Dies ist ein ganz anderes Buch als die erste Auflage. Ich habe einige Kapitel entfernt und ein paar völlig neue hinzugefügt. Alle verbliebenen Kapitel und der Code wurden beträchtlich verändert. Viele der Änderungen bringen uns coole neue Funktionen (Flugsteuerungen in Kapitel 5, *Funktionen: Immer und immer wieder benutzen*, oder Feuerspezialeffekte). Die neuen Funktionen sind lustig, aber sie sind nicht der Hauptgrund für die Änderungen.

Die Änderungen sind für dich. Sie machen das Buch lustiger. Sie erleichtern das Lernen. Sie können dir besser helfen, ein Programmierer zu werden.

Also, was ist neu? So ziemlich alles. Bis auf die Teile, die bereits perfekt waren.

Was dieses Buch nicht ist

Nur damit es klar ist:

Aus uns werden keine Experten in JavaScript.

Und ...

Aus uns werden keine Experten in 3D-Spiele-Programmierung.

Wir behandeln eine Menge Stoff in diesem Buch, aber wenn wir fertig sind, werden wir nicht in der Lage sein, das nächste Facebook herzustellen. Und wir werden auch nicht MarioKart nachbauen können. Für beides müssten Hunderte Programmierer Hunderte Stunden arbeiten – und bessere Kenntnisse und Fähigkeiten haben, als wir mit diesem Buch vermitteln.

Aber du wirst die wichtigsten Teile von JavaScript kennenlernen. Und du erwirbst eine Menge 3D-Fertigkeiten. Damit kannst du weitere aufregende Dinge machen – und bist dann bereit für noch größere, noch aufregendere Dinge.

Hinweise zur deutschen Fassung dieses Buchs

Du beschäftigst dich bestimmt schon eine Weile mit Computern und dem Internet und weißt deshalb, dass du dabei an der englischen Sprache nicht vorbeikommst. Das ist beim Programmieren nicht anders. So gut wie alle

Programmiersprachen basieren auf der englischen Sprache. Das gilt auch für JavaScript, das du in diesem Buch kennenlernst. Die Programmierplattform, die du benutzen wirst, um die hier gezeigten Beispiele auszuprobieren, verwendet englische Befehle. In den bereitgestellten Vorlagen für den Code, den sogenannten Templates, findest du englischsprachige Kommentare, Variablen und Funktionen, und auch die geladenen Codesammlungen oder JavaScript-Bibliotheken sind auf Englisch.

Zur besseren Orientierung haben wir im vorderen Teil des Buchs den englischen Kommentaren ihre deutschen Übersetzungen beigefügt, später haben wir dann darauf verzichtet, schließlich kennst du dich dann schon aus, und es ist nicht mehr nötig. Wenn du irgendwann anfängst, eigene Programme zu entwickeln, kannst du dir selbst Namen für deine Variablen und Funktionen ausdenken. Ob diese auf Deutsch oder auf Englisch oder auf einem ganz geheimen Code beruhen, den du dir selbst ausgedacht hast, ist ganz dir überlassen (oder hängt davon ab, was du mit deinen Mitstreitern vereinbart hast). Am besten legst du dir beim Durcharbeiten dieses Buchs ein Wörterbuch bereit, um Wörter, deren Bedeutung dir unklar ist, nachzuschlagen.

Du musst außerdem beachten, dass die Schreibweise von Dezimalzahlen anders ist als die im deutschsprachigen Raum übliche: Anstelle eines Kommas zum Abtrennen wird ein Punkt benutzt. Denke daran, das ebenfalls zu tun, da du dir ansonsten Fehler einhandelst, deren Ursache möglicherweise schwer zu finden ist.

Fangen wir an!

Genug Einleitung – starten wir mit dem Programmieren!

1 PROJEKT: EINFACHE FORMEN HERSTELLEN

WENN DU DIESES KAPITEL GELESEN HAST, DANN

- bist du in der Lage, Code zu schreiben
- weißt du, wie man 3D-Formen herstellt
- kannst du einfaches JavaScript programmieren
- du kannst deine erste Animation herstellen

In diesem Kapitel stürzen wir uns gleich in das Schreiben von Computercode. Es ist später noch genügend Zeit, JavaScript zu beschreiben – die Programmiersprache, die wir benutzen werden. Außerdem können wir später auch immer noch darüber reden, was eine Programmiersprache ist. Jetzt programmieren wir erst einmal – wir schreiben Code –, ohne uns zu sehr um die Details zu kümmern.

Der Plan ist, sich mit dem Eintippen von Code vertraut zu machen und eine Ahnung davon zu bekommen, wie es ist, zu programmieren. Ob du es glaubst oder nicht, schon allein dadurch wirst du unglaublich viel über das Programmieren lernen. Du unternimmst deine ersten Schritte in das Kreieren von 3D-Welten. Und du beginnst sogar deine erste Animation.

Das Wichtigste, das du in diesem Kapitel tun wirst, ist Spielen. Die besten Programmierer spielen mit ihrem Code. Sie experimentieren. Sie ändern Code, um zu sehen, was passiert. Sie basteln herum, um Grenzen zu überschreiten. Sie machen Dinge kaputt. Wirklich!

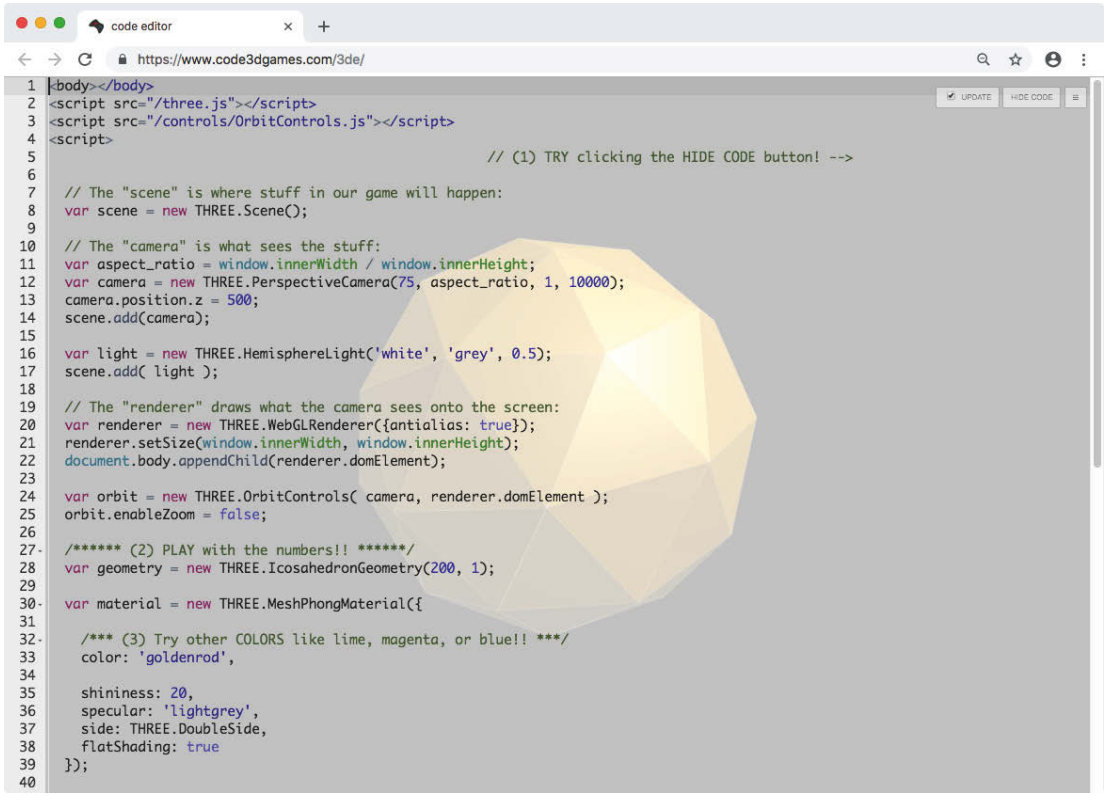
Fangen wir deshalb an, gemeinsam etwas kaputtzumachen – und zu erschaffen.

Programmieren mit dem 3DE Code Editor

In diesem Buch nutzen wir den 3DE Code Editor – kurz 3DE – zum Programmieren. Zu den wichtigsten Eigenschaften des 3DE gehören:

- Er läuft direkt in deinem Browser.
- Er erlaubt uns, den Programmcode direkt einzutippen und sofort die Ergebnisse zu sehen.
- Deine Arbeit wird automatisch beim Eintippen gespeichert – du kannst sie aber auch mithilfe eines Befehls aus einem Menü speichern.
- Du kannst Projekte zur Benutzung auf anderen Sites herunterladen (darüber werden wir in Kapitel 20, *Code in das Web bekommen*, sprechen).
- Du kannst alle Projekte exportieren, um sie als Backup zu sichern oder um sie auf einen anderen Computer zu laden.
- Der 3DE Code Editor läuft offline – nach dem ersten Besuch wird er in deinem Browser gespeichert, sodass du auch dann arbeiten kannst, wenn du nicht mit dem Internet verbunden bist.

Abbildung 1-1 ▼
Der 3DE Code Editor in Google Chrome



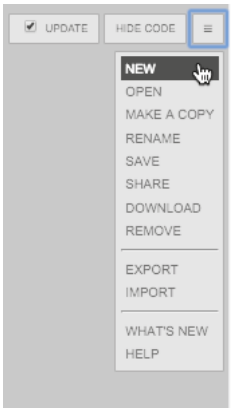
Denk dran, du musst für dieses Buch den Browser Google Chrome benutzen. Die meisten Übungen funktionieren zwar auch in anderen Browsern, dennoch ist es am einfachsten, für alle unsere Projekte bei einem einzigen Browser zu bleiben – vor allem wenn du im Forum dieses Buchs Fragen stellen möchtest.[1]

Um zu beginnen, öffne mit Chrome den 3DE Code Editor[2]. Das sollte dann etwa so aussehen wie in Abbildung 1-1.

Das sich drehende Ding mit den vielen Seitenflächen ist ein Beispiel für die Dinge, die wir in diesem Buch machen wollen. In diesem Kapitel legen wir ein neues Projekt namens **Formen** an.

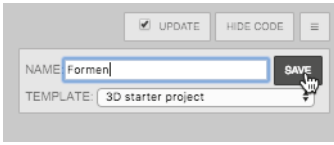
Erzeuge dein erstes Programmierprojekt

Um ein neues Projekt im 3DE Code Editor anzulegen, klicken wir auf den Menübutton (den Button mit den drei waagerechten Strichen) in der oberen rechten Ecke des Fensters und wählen NEW aus dem Menü.



◀ Abbildung 1-2
Das geöffnete Menü

Tippe den Namen des Projekts, `Formen`, in das Textfeld ein und klicke dann auf SAVE. Das Template (ein Template ist eine Vorlage) lässt du einfach auf `3D starter project` stehen.

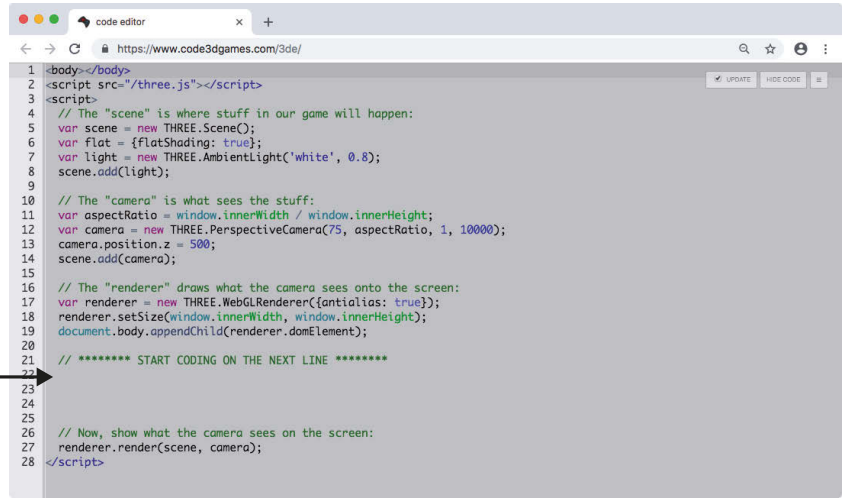


◀ Abbildung 1-3
Ein neues Projekt wird angelegt.

1 *https://talk.code3Dgames.com/*
2 *http://code3Dgames.com/3de*

Wenn der 3DE ein neues 3D-Projekt öffnet, gibt es in der Datei schon eine Menge Code. Wir schauen uns diesen Code später in Kapitel 9, *Was ist das alles für ein Code?*, genauer an. Im Moment wollen wir jedoch unser Programmierabenteuer auf Zeile 22 beginnen. Suche nach der Zeile, auf der `START CODING ON THE NEXT LINE` steht.

Abbildung 1-4 ▶
Hier legst du nun los!



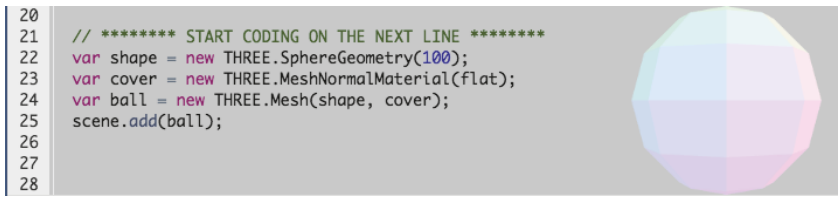

```
<body></body>
<script src="/three.js"></script>
<script>
  // The "scene" is where stuff in our game will happen:
  var scene = new THREE.Scene();
  var flat = {flatShading: true};
  var light = new THREE.AmbientLight('white', 0.8);
  scene.add(light);

  // The "camera" is what sees the stuff:
  var aspectRatio = window.innerWidth / window.innerHeight;
  var camera = new THREE.PerspectiveCamera(75, aspectRatio, 1, 10000);
  camera.position.z = 500;
  scene.add(camera);

  // The "renderer" draws what the camera sees onto the screen:
  var renderer = new THREE.WebGLRenderer({antialias: true});
  renderer.setSize(window.innerWidth, window.innerHeight);
  document.body.appendChild(renderer.domElement);

  // ******** START CODING ON THE NEXT LINE ********



  // Now, show what the camera sees on the screen:
  renderer.render(scene, camera);
</script>
```

Tippe in Zeile 22 Folgendes ein:

```
var shape = new THREE.SphereGeometry(100);
var cover = new THREE.MeshNormalMaterial(flat);
var ball = new THREE.Mesh(shape, cover);
scene.add(ball);
```

Sobald du damit fertig bist, solltest du etwas Cooles sehen:

Abbildung 1-5 ▶
Hier hat sich schon was getan.



```
  // ******** START CODING ON THE NEXT LINE ********
  var shape = new THREE.SphereGeometry(100);
  var cover = new THREE.MeshNormalMaterial(flat);
  var ball = new THREE.Mesh(shape, cover);
  scene.add(ball);
```

Der Ball, den wir eingetippt – der Ball also, den wir *programmiert* – haben, ist im 3DE aufgetaucht. Herzlichen Glückwunsch! Du hast gerade dein erstes JavaScript-Programm geschrieben!

Falls es nicht funktioniert hat, überprüfe noch einmal, ob du alles genau so eingegeben hast, wie es hier gezeigt wurde. Achte vor allem auf die Groß- und Kleinschreibung. Damit gibt es besonders häufig Probleme. Funktioniert es dann immer noch nicht, blättere weiter bis Kapitel 2, *Debugging: Code re-*

parieren, wenn etwas schiefgeht. Dort findest du Schritte zur Fehlersuche. Und falls das immer noch nichts nützt, wende dich an das Buch-Forum.

Werfen wir nun einmal einen Blick auf die 3D-Programmierung, die wir gerade durchgeführt haben. 3D-Dinge bestehen aus zwei Teilen: der Form und etwas, das diese Form bedeckt. Die Kombination aus beidem, der Form und ihrer Umhüllung, trägt in der 3D-Programmierung einen besonderen Namen: *Mesh* (Gitter oder auch Gewebe).

»Mesh« ist ein schickes Wort für ein 3D-Ding. Meshes brauchen *Formen* (manchmal als *Geometrie* bezeichnet) und etwas, um sie zu umhüllen (sogenanntes *Material*). Wir schauen uns in diesem Kapitel verschiedene Formen an. Zu unterschiedlichen Umhüllungen für unsere Formen kommen wir erst später.

Sobald wir ein Mesh haben, fügen wir es der Szene hinzu. Die Szene ist die Stelle in der 3D-Programmierung, an der gezaubert wird. Es ist die Welt, in der alles passiert. In diesem Fall ist es der Ort, an dem unser Ball herumlungert und auf Freunde wartet. Fügen wir der Szene einige weitere Formen hinzu, damit der Ball nicht so allein ist.

Formen mit JavaScript herstellen

Wir haben eine Form gesehen: die Kugel. 3D-Programmierern stehen aber eine Menge Formen zur Verfügung. Formen können einfach sein – Würfel, Pyramiden, Kegel und Kugeln –, sie können aber auch komplexer sein, wie Gesichter oder Autos. In diesem Buch bleiben wir bei einfachen Formen. Wenn wir so etwas wie Bäume bauen, kombinieren wir einfache Formen wie Kugeln und Zylinder miteinander.

Kugeln herstellen

Bälle werden in der Geometrie und der 3D-Programmierung als *Kugeln* (oder mathematisch korrekt als *Sphären*) bezeichnet. Es gibt zwei Möglichkeiten, die Form einer Kugel in JavaScript zu kontrollieren.

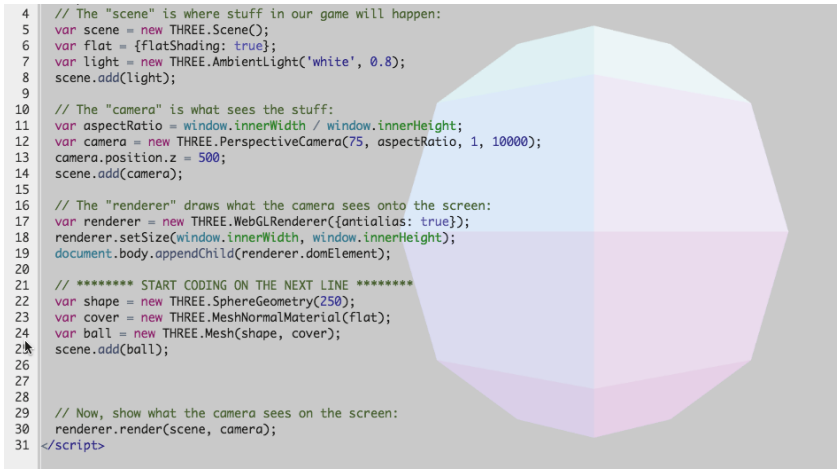
Größe: SphereGeometry(100)

Zunächst einmal können wir eine Kugel kontrollieren, indem wir beschreiben, wie groß sie ist. Als wir `new THREE.SphereGeometry(100)` sagten, haben wir damit einen Ball geschaffen, dessen Radius `100` ist. Was passiert, wenn du den Radius auf `250` änderst?

```
var shape = new THREE.SphereGeometry(250);
var cover = new THREE.MeshNormalMaterial(flat);
var ball = new THREE.Mesh(shape, cover);
scene.add(ball);
```

Das sollte die Kugel deutlich größer machen:

Abbildung 1-6 ▶
Ups, die Kugel ist gewachsen!



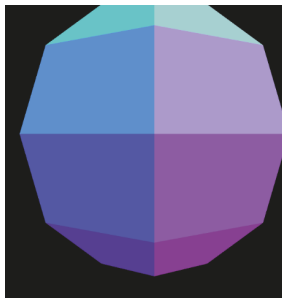
```
// The "scene" is where stuff in our game will happen:
var scene = new THREE.Scene();
var flat = {flatShading: true};
var light = new THREE.AmbientLight('white', 0.8);
scene.add(light);

// The "camera" is what sees the stuff:
var aspectRatio = window.innerWidth / window.innerHeight;
var camera = new THREE.PerspectiveCamera(75, aspectRatio, 1, 10000);
camera.position.z = 500;
scene.add(camera);

// The "renderer" draws what the camera sees onto the screen:
var renderer = new THREE.WebGLRenderer({antialias: true});
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

// ******** START CODING ON THE NEXT LINE ********
var shape = new THREE.SphereGeometry(250);
var cover = new THREE.MeshNormalMaterial(flat);
var ball = new THREE.Mesh(shape, cover);
scene.add(ball);



// Now, show what the camera sees on the screen:
renderer.render(scene, camera);
</script>
```

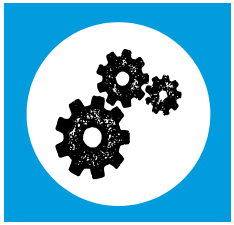
Was passiert, wenn du die 250 zu 10 änderst? Wie du sicher erraten hast, wird sie viel kleiner. Das ist also eine Möglichkeit, um die Form einer Kugel zu kontrollieren. Welche andere Möglichkeit hast du?

Nicht klobig: SphereGeometry(100, 20, 15)

Wenn du auf den HIDE CODE-Button im 3DE klickst, bemerkst du sicher, dass unsere Kugel eigentlich kein wirklich glatter Ball ist:

Abbildung 1-7 ▶
Eigentlich keine Kugel, sondern ein kugelförmiges Gebilde aus Flächen





Du kannst den Code ganz leicht ein- und ausblenden

Wenn du in der oberen rechten Ecke des 3DE-Fensters auf den weißen HIDE CODE-Button klickst, siehst du nur den Spielbereich und die Objekte im Spiel. So wirst du in späteren Kapiteln auch die Spiele bedienen. Um den Code wieder hervorzuzaubern, klickst du auf den weißen SHOW CODE-Button im 3DE Code Editor.