



## Einleitung

**E**in Videospiel zu spielen macht Spaß. Aber Videospiele selbst zu programmieren ist eine kreative, fesselnde Tätigkeit, mit der du deinen eigenen Spielspaß gestalten kannst. Die kostenlose Programmierumgebung Scratch bietet dir die Möglichkeit, leicht das Programmieren zu lernen. Zwar wurde Scratch hauptsächlich für die Altersgruppe von 8 bis 16 Jahren entwickelt, doch es wird auch von anderen Personen verwendet: sowohl von jüngeren Kindern zusammen mit ihren Eltern als auch von Oberstufenschülern, die ihre erste Programmiersprache lernen.

Da man mit Scratch so viel machen kann, ist es schwer, einen Anfang zu finden. Deswegen kommt dir dieses Buch zu Hilfe. Es zeigt dir, wie du verschiedene Videospiele in Scratch erstellen kannst. Während du die Projekte aus diesem Buch nachbaust, bekommst du eine gute Vorstellung davon, wie die einzelnen Blöcke gewöhnlich zur Gestaltung von Spielen eingesetzt werden. Diese Projekte bieten dir eine solide Grundlage für deine eigenen Programme.

## Für wen ist dieses Buch gedacht?

Um dieses Buch verstehen zu können, brauchst du keinerlei Programmierkenntnisse. Die einzigen mathematischen Kenntnisse, die du benötigst, sind die Grundrechenarten Addition, Subtraktion, Multiplikation und Division. Auch wenn du eine Abneigung gegen Mathematik hast, lass dich davon nicht abhalten, programmieren zu lernen! Schließlich ist der Computer ja dazu da, dir die Rechenarbeit abzunehmen.

Jedes Programm in diesem Buch lässt sich leicht schreiben, indem du einfach den Schritt-für-Schritt-Anleitungen folgst. Dabei lernst du die einzelnen Codeblöcke und Programmiertechniken kennen, während du sie einsetzt, um Spiele zu gestalten. Egal wie dein Kenntnisstand zurzeit aussieht, es gibt keinen Grund, *nicht* sofort mit dem Lesen dieses Buches zu beginnen.

Kinder könnten den Anleitungen selbstständig folgen, aber dieses Buch ist auch für Eltern und Lehrer gedacht, die ihre Kinder oder Schüler in die Welt der Programmierung einführen wollen. Die Projekte sind ideal als Zeitvertreib fürs Wochenende oder für eine Computer-AG geeignet. Erwachsene, die dieses Buch einsetzen wollen, um andere beim Lernen zu unterstützen, müssen auch keine Softwareingenieure sein.



Das Programmieren ist aber eine *praktische* Tätigkeit, die du ebenso wie Karate oder Gitarrespielen nicht allein durch Lesen lernen kannst. Vollziehe die Anleitungen nach und schreibe die vorgestellten Spiele! Auf diese Weise lernst du viel mehr.

# Der Aufbau dieses Buches

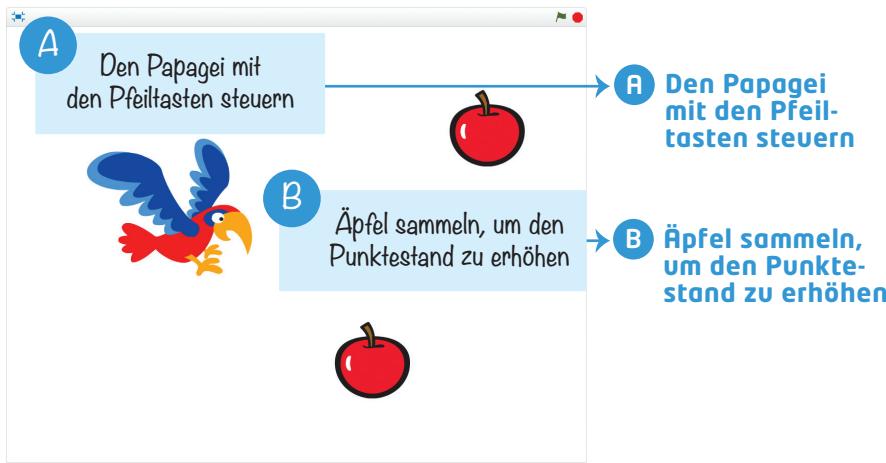
Jedes Kapitel zeigt dir ausführlich, wie du ein bestimmtes Spiel programmierst. Die verschiedenen Programmiertechniken werden dabei immer dann behandelt, wenn sie gebraucht werden. Zu Anfang erstellst du jeweils eine Skizze, die zeigt, wie das fertige Spiel aussehen soll, und planst die Hauptteile des Programms. In den folgenden Abschnitten lernst du Schritt für Schritt, wie du diese einzelnen Teile programmierst, bis du das Spiel komplett geschrieben hast. Danach kannst du das Spiel noch um besondere Funktionen oder um einen Cheat-Modus erweitern. Anhand der Wiederholungsfragen am Ende des Kapitels kannst du prüfen, ob du den Lernstoff verstanden hast.

- ▶ **Kapitel 1: Erste Schritte mit Scratch** Hier erfährst du, wie du die Scratch-Website erreichst, und lernst den Scratch-Editor kennen.
- ▶ **Kapitel 2: Regenbogenlinien im Weltraum** In diesem Kapitel erstellst du ein animiertes Kunstprojekt mithilfe einfacher Codeblöcke und mehrerer Figuren. Dabei lernst du auch etwas über Gradangaben für Richtungen.
- ▶ **Kapitel 3: Ein Labyrinthspiel** Hier erstellst du ein Labyrinthspiel, in dem der Spieler die Katze mithilfe der Tastatur steuert und durch acht verschiedene Labyrinthe lenkt.
- ▶ **Kapitel 4: Basketball mit Schwerkraft** Dieses Kapitel zeigt dir, wie du ein Basketballspiel schreibst, in dem sich die springende Katze und die fallenden Bälle realistisch wie unter Schwerkraefteinfluss bewegen.
- ▶ **Kapitel 5: Ein Brick-Breaker-Spiel mit Schliff** Hier lernst du einfache Techniken kennen, um aus einem einfachen Brick-Breaker-Spiel ein professionell aussehendes, spannendes Spiel mit Animationen, Geräuschen usw. zu machen.
- ▶ **Kapitel 6: Snaaaaaake!** In diesem Kapitel geht es um das klassische Computerspiel, in dem du eine Schlange über den Bildschirm lenbst, die ständig wächst. Hier erfährst du, wie du die Klonfunktionen von Scratch nutzen kannst, um den Schlangenkörper immer länger zu machen.
- ▶ **Kapitel 7: Obstschlitzer** Hier schreibst du eine eigene Version des erfolgreichen Smartphone-Spiels *Fruit Ninja*, in dem der Spieler durch die Luft fliegende Früchte zerschneiden muss.
- ▶ **Kapitel 8: Asteroidenknacker** In diesem Kapitel programmierst du eine eigene Version des klassischen Weltraum-Schießspiels *Asteroids*, wobei du die Objekte im Spiel diesmal nicht nur mit der Tastatur, sondern auch mit der Maus steuerst.
- ▶ **Kapitel 9: Ein anspruchsvolles Jump-&-Run-Spiel** Hier greifen wir auf viele der Techniken aus den vorherigen Kapiteln zurück, um ein Jump-&-Run-Spiel mit Geh- und Springanimationen, Plattformen und KI-gesteuerten Gegnern zu schreiben.

## Die Projekte durcharbeiten

Am Anfang aller Projekte in diesem Buch steht eine Skizze des Spiels, das wir erstellen wollen. Die Beschriftungen in dieser Skizze beschreiben die Spielfunktionen, die wir mit dem Code für das Spiel gestalten wollen.

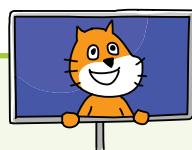
Um uns nicht zu verzetteln, gehen wir die Spielfunktionen eine nach der anderen an. Die blauen ABC-Überschriften im Buch entsprechen dabei den in der Skizze genannten Funktionen.



Wenn du eine umfangreiche Aufgabe in mehrere kleine Teilaufgaben zerlegst, wirkt sie nicht so überwältigend. Vor allem kannst du auf diese Weise gezielt über die Lösungen nachdenken. Wenn du eine einfache Version deines Spiels zum Laufen gebracht hast, kannst du weitere Funktionen, Cheat-Modi usw. hinzufügen. Ich rate dir, immer mit einer einfachen Skizze anzufangen – auch wenn du später deine eigenen Spiele entwickelst.

### Zeit zu speichern!

Überall in diesem Buch wirst du die Kästen mit der Überschrift »Zeit zu speichern!« sehen. Da du die Programme Schritt für Schritt erstellst, solltest du immer wieder innehalten und das Programm in dem Zustand ausführen, den du schon erreicht hast, auch wenn es noch nicht fertig ist. Dadurch kannst du überprüfen, ob du so weit alles richtig gemacht hast, und Fehler frühzeitig erkennen. Diese Kästen sollen dich auch daran erinnern, deine bisherige Arbeit an dem Programm zu sichern, indem du in der Menüleiste auf **Datei ▶ Speichern** klickst.



## Zusatzmaterial auf der Begleitwebsite

In der Scratch-Umgebung werden viele Bilder bereitgestellt, aber für die Projekte in diesem Buch benötigst du noch einige zusätzliche Dateien. Sie befinden sich alle in einer ZIP-Datei, die du von <https://www.nostarch.com/scratchplayground/> herunterladen kannst. Dabei handelt es sich um die englischsprachige Webseite der amerikanischen Ausgabe dieses Buchs, *Scratch Programming Playground*. Um die enthaltenen Dateien nutzen zu können, musst du die ZIP-Datei auf deiner Festplatte auspacken.

In der ZIP-Datei findest du die Bilddateien, die in den Projekten in diesem Buch verwendet werden, sowie die Skelettdateien für die einzelnen Projekte. In den Skelettdateien sind alle vorbereitenden Schritte bereits durchgeführt, sodass du nur noch die Codeblöcke hinzufügen musst. Falls du Schwierigkeiten dabei haben solltest, ein Programm bis zum Abschluss zu führen, kannst du versuchen, von der Skelettdatei auszugehen, anstatt mit einem ganz neuen, leeren Projekt zu beginnen. Die Verwendung der Skelettdateien ist auch gut für Lehrer geeignet, die mehrere Schüler unterrichten, aber nur begrenzte Zeit zur Verfügung haben.



## Korrekturen und Ergänzungen

Wir haben uns größte Mühe gegeben, dieses Buch fehlerfrei zu halten. Korrekturen und Ergänzungen findest du auf [www.dpunkt.de/scratch](http://www.dpunkt.de/scratch).