

Tobias Hübner

GEHEIMAKTE COMPUTER

Everything they don't
teach you about tech

Retrospiele + Hacking + Künstliche Intelligenz
Quantencomputer + Tricks der IT-Branche + Coden



Kapitel 1

Das kreativste Werkzeug aller Zeiten

Im ersten Kapitel erfährst du mehr über den Aufbau der Geheimakte Computer, eine Katze wird dir das Programmieren beibringen, und am Ende findest du sogar einen triftigen Grund dafür, endlich mal wieder Marshmallows zu kaufen.

»Ich habe gerade eine neue künstliche Intelligenz ausprobiert – ChatGPT. Du glaubst nicht, was die alles kann.« So begann ein Gespräch mit meiner Frau, in der ich ihr begeistert davon berichtete, was KI mittlerweile alles kann: Hausaufgaben erledigen, Klausuren und Liebesbriefe schreiben oder ganze wissenschaftliche Arbeiten verfassen. »Na dann kann die ja auch dein Buch schreiben«, war ihre Antwort, die mich kurz schlucken ließ. Könnte ich mir die ganze Arbeit vielleicht sparen? Sind Computer jetzt so weit, dass sie uns das Schreiben abnehmen können?

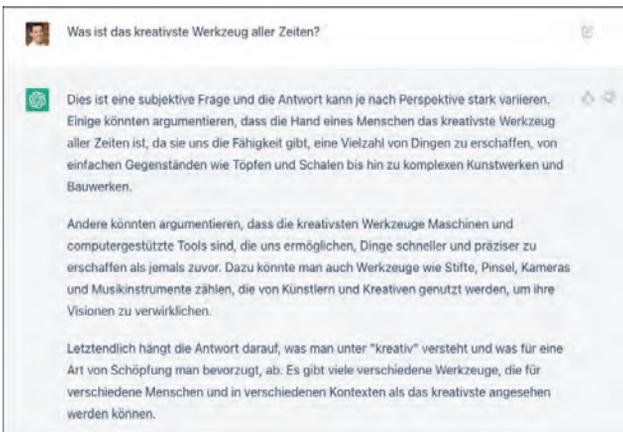


Abbildung 1.1 Hm ... keine schlechte Antwort von ChatGTP!

Aber nicht nur ich als Autor stehe vor der Aufgabe, mich mit den Möglichkeiten und Gefahren des Computers auseinanderzusetzen, sondern nahezu alle Menschen treibt

die Frage um, welchen Stellenwert sie dem Computer in ihrem Leben einräumen, und meine Vermutung ist, dass uns diese Frage in den kommenden Jahren noch deutlich mehr beschäftigen wird. Der Grund dafür ist einfach: Computer sind schon jetzt zu mächtig, um sie zu ignorieren, und diese Macht wird in den kommenden Jahren bestimmt nicht abnehmen.

Leider sind Computer aber für viele ein Buch mit sieben Siegeln und so sagenumwoben und unzugänglich wie eine Geheimakte. Dieses Buch will diese Geheimakte öffnen und dir zeigen, dass Computer zwar das komplexeste sind, was die Menschheit je geschaffen hat, aber dennoch auf verstehbaren und oft verblüffend einfachen Ideen beruht.

Da ich als Lehrer weiß, dass man mit der Vermittlung bloßer Theorie keine Herzen gewinnt, habe ich versucht, den Spaß-Faktor durch viele »Challenges« – also kleine Aufgaben, bei denen du etwas programmieren oder basteln kannst – anzuheben.

Um aber überhaupt verstehen zu können, warum Computer für mich so faszinierend sind und damit wir uns ein bisschen besser kennenlernen, lade ich dich in diesem ersten Kapitel auf eine kleine Reise in meine Kindheit ein. So kannst du vielleicht besser verstehen, was mich zum Schreiben dieses Buches veranlasst hat.

Aber vorab folgt erst einmal ein kurzer Vorgeschmack auf das, was dich in diesem Buch alles erwartet.



Abbildung 1.2 So sieht es gerade auf meinem Schreibtisch aus, während ich diese Zeilen schreibe.

1.1 So ist das Buch aufgebaut

In den zehn Kapiteln dieses Buches erfährst du nicht nur viel über die Funktionsweise von Computern (Kapitel 2), sondern auch, wie du selbst zum Entwickler eigener Hard- und Software wirst (Kapitel 5). Zusätzlich erhältst du hoffentlich interessante Einblicke in die Geschichte des Computers (Kapitel 3), der Games-Industrie (Kapitel 4) und der Hacker-Szene (Kapitel 6).

Und wenn du in den Nachrichten liest, dass der Computer (mal wieder) die Welt entweder retten oder zerstören soll – zum Beispiel durch künstliche Intelligenz (Kapitel 7), kannst du diese Heilsversprechen und Untergangsszenarien mithilfe des achten Kapitels besser einordnen.

Das neunte Kapitel zeigt dir, wie manipulativ der Computer sein kann, und im letzten Kapitel kannst du nachlesen, was dich in den kommenden Jahren so alles erwarten wird.

Damit du deine Informatik-Kenntnisse verbessern kannst, findest du in jedem Kapitel eine kleine Aufgabe, bei der es ums Programmieren geht (Coding-Challenge), und eine Bastelaufgabe (Maker-Challenge). Dazu habe ich für einige Kapitel einen Experten auf dem jeweiligen Fachgebiet befragt und das entsprechende Interview mit ins Kapitel aufgenommen – nicht zuletzt, damit außer mir auch noch jemand anders im Buch zu Wort kommt. Manchmal wird auch eine Person genauer vorgestellt, in diesem Kapitel etwa der Computer-Pädagoge Mitchel Resnick.



Abbildung 1.3 Die erste Version des Raspberry-Pi-Computers – das ideale Werkzeug, um zu verstehen, wie Computer funktionieren.

Da ich im Laufe des Buchs – zum Beispiel gleich im nächsten Absatz – immer mal wieder auf den Raspberry Pi zu sprechen komme, findest du in Anhang A eine ausführliche Anleitung, die dir zeigt, was der Pi alles kann und wie du ihn in Betrieb nimmst. Außerdem findest du in Anhang B eine kleine Einführung in die Programmiersprache Python. Aber wir wollten ja eigentlich auf eine kleine Reise in meine Kindheit gehen, also los geht's!

1.2 Vom Konsumenten zum Produzenten

Mein Interesse für den Computer wurde durch das Zocken von Computerspielen geweckt. Begonnen hat alles, als mein Vater eines Tages einen IBM XT mit 4,77 MHz in seinem Arbeitszimmer aufstellte. In einer großen Diskettenbox voller Raubkopien hatten wir bald schon die ersten Spiele, und so versammelten sich meine Kumpel Clemens, Stephan und Michael und ich immer wieder vor dem Bildschirm, und wir spielten der Reihe nach »Alley Cat«.

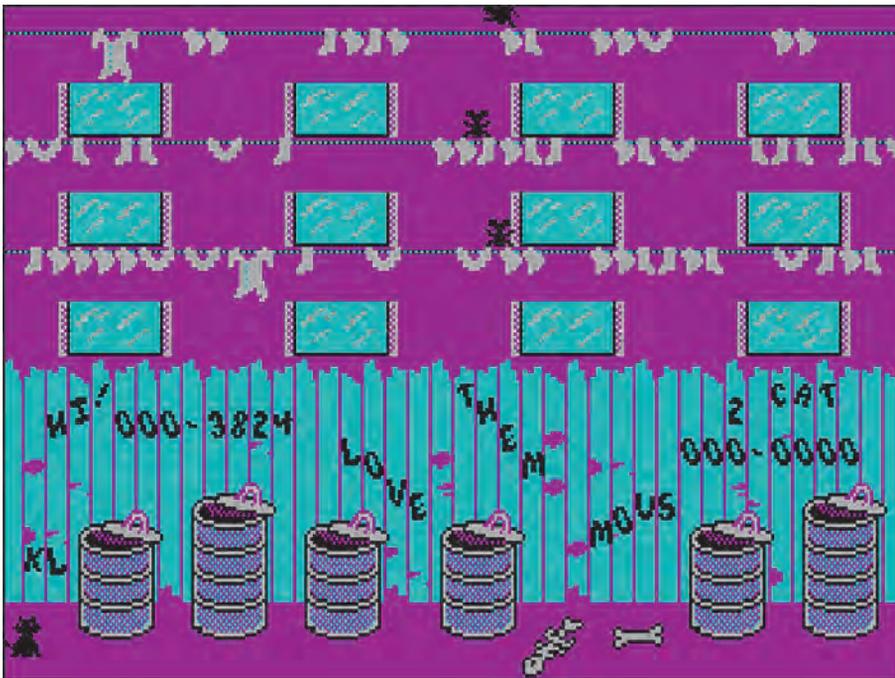


Abbildung 1.4 Mein erstes Computerspiel in Magenta und Türkis: Alley Cat

Wer den unglaublichen Nervenkitzel nachvollziehen möchte, den ich als 8-jähriger erlebte, als plötzlich ein Hund auf dem Bildschirm erschien und meiner Pixel-Katze nachjagte, kann sich kostenlos das inoffizielle Fan-Remake »Alley Cat Remeow Edition« herunterladen.

Und natürlich blieb es nicht bei diesem einen Spiel. Von Freunden und Verwandten wurde ich fortan mit immer neuem Stoff versorgt und mein acht Jahre älterer Bruder zeigte mir sogar, wie man die bockharten Adventure-Spiele von Sierra spielt, allen voran »Space Quest«, das vom Spieler das Eintippen englischer Befehle forderte. Keine leichte Aufgabe, wenn man kein Englisch kann. Aber getrieben von dem Wunsch, das Spiel zu verstehen, schlug ich Wort für Wort nach und fühlte mich dann bei jedem Fortschritt so, als hätte ich gerade einen neuen Kontinent entdeckt.

Später, als rebellischer Jugendlicher, übten dann vor allem die Spiele einen Reiz aus, die eigentlich für Erwachsene gemacht oder sogar in Deutschland verboten waren. Denn genau das waren in den 80er- und 90er-Jahren die Spiele mit der revolutionärsten Technik, der besten Grafik und vor allem der Möglichkeit, die Spielwelt nach eigenem Belieben zu verändern. Quake hieß das Spiel, mit dem meine Kumpel und ich am meisten Zeit verbrachten. Ich erinnere mich noch genau daran, wie ich bei meinem Freund Tomas am 22. Juni 1996, dem Tag, an dem die lang erwartete Shareware-Version von Quake endlich erschien, darauf wartete, dass die Daten ihren Weg über das 14,4-K-Modem auf seinen (und dann wenig später auch meinen) Rechner fanden.



Abbildung 1.5 So gruselig, dass es in Deutschland zunächst nicht offen verkauft werden durfte: Quake von ID Software

Kurze Zeit später wurde das Spiel in Deutschland indiziert, es durfte also nicht offen im Geschäft verkauft oder auch nur beworben werden. Der Reiz, ein »verbotenes« Spiel zu spielen, machte das Game für uns natürlich nur noch interessanter. Da das Spielen übers Internet damals noch in den Kinderschuhen steckte, war der Online-Mehrspielermodus am Anfang ein ziemlich frustrierendes Erlebnis – auch weil man zu Hause die Telefonleitung blockierte und für das Zocken mit den Kumpeln ziemlich viel Geld zahlte.

Also trafen wir uns am Wochenende mit unseren PCs im Partykeller meines Freundes Dominic und zockten die ganze Nacht. Als in den kommenden Jahren das Netz dank ISDN schneller wurde, gründeten wir unseren eigenen Gaming-Clan namens »Warriors on Battlefield«, und ich zog als *Sir_Weivel* in die digitale Schlacht. Zur gleichen Zeit verbrachte ich zudem immer mehr Zeit damit, das Spiel nach meinen eigenen Vorstellungen zu modifizieren. Zuerst erstellte ich einen eigenen »Skin«, also quasi ein Kostüm für meine Spielfigur, später auch eigene Level.



Abbildung 1.6 Das verrückte Team der Quake-Programmierer bei ID-Software. Die Programmierlegende John Carmack (links) arbeitete bis 2022 an VR-Headsets fürs Metaverse und widmet sich seitdem der Erforschung künstlicher Intelligenz. John Romero (Mitte) entwickelt noch immer Spiele und plant derzeit die Eröffnung eines Videospiele-Museums in Irland.

Auch wenn wir es damals nicht wussten – für meine Kumpel und mich war diese Zeit des kreativen Umgangs eine Vorbereitung auf unser späteres Leben, denn fast alle von uns haben ihr Hobby, den Computer, in der einen oder anderen Form zum Beruf gemacht.

Leider wachsen manche ihr Leben lang nicht aus ihrer Konsumentenrolle heraus, und viele Menschen haben auch ein großes Interesse daran, denn Konsumenten stellen keine unbequemen Fragen und lassen sich anscheinend gerne das Geld aus der Tasche ziehen. Aber dazu später mehr.

1.3 Game Over

Irgendwann hatte dann selbst ich das Thema Computerspiele satt. Stattdessen hat mich ein anderes Unterhaltungsmedium in seinen Bann gezogen: das Buch. Bücher haben mich sogar so fasziniert, dass ich anfing, Germanistik und katholische Theologie zu studieren. Ich hatte gerade erst mein neues Zimmer im Studentenwohnheim bezogen, als ich erfuhr, dass an meiner Uni in Münster außerdem ein ganz neuer Studiengang geschaffen wurde mit dem komischen Namen »Medien und Informationstechnologien in Erziehung, Bildung und Unterricht« – kurz »MIEBU«.



medienistik ©

Hier können Sie die Seite durchsuchen!

Was ist Medienistik?

www.medienistik.de richtet sich in erster Linie an interessierte Deutschlehrerinnen und -lehrer, die mehr über die kritischen Medienpädagogik und ihre Einsatzmöglichkeiten im Unterricht erfahren möchten. Bei dem hier vorgestellten Einsatz neuer Medien soll es stets darum gehen, dass sich die Schüler Medieninhalte selbständig *aneignen*, statt sie nur passiv zu konsumieren. Nicht die Medien, sondern die Menschen, die sie nutzen, sollen im Mittelpunkt stehen, damit sie sich *aktiv und kritisch* mit der sie umgebenden Medienwelt auseinandersetzen können.

Aber auch Lehrerinnen und Lehrer anderer Fächer können in den hier angebotenen Texten, Arbeitsblättern und Handlungsimpulsen vielleicht einige Anregungen für ihren Unterricht bekommen, damit die Schülerinnen und Schüler die neuen Medien als das behandeln können, was sie sind, nämlich Hilfsmittel zur Auseinandersetzung mit ihrer Umwelt.

[Wenn Sie Hilfe bei der Navigation benötigen, klicken Sie hier.](#)

Abbildung 1.7 Die erste Version meiner Internetseite MEDIENISTIK

In zwei Jahren konnte man einen »Abschluss in MIEBU« erwerben. Obwohl ich damals noch nicht wusste, was das ist, schrieb ich mich ein und hielt bis zum Ende durch. 20 Jahre später habe ich zwar immer noch nicht herausgefunden, was ich mit dem Abschluss anfangen soll, aber so kam ich mit ganz unterschiedlichen Themen in Kontakt. Ich lernte über den Zusammenhang von Gewalt in den Medien und der Schule, Grundlagen der Filmanalyse und welchen Einfluss Medien auf Menschen haben.

Am Ende des Studiums musste ein Medienprodukt erstellt werden, und ich gestaltete eine Internetseite, der ich den unaussprechlichen Namen »MEDIENISTIK« gab – eine Verbindung der Wörter Medien und Germanistik. Ziel war es, eine Verbindung zu schaffen zwischen Technik und Literatur, Informatik und Geisteswissenschaft. Zuerst sammelte ich darauf Texte, die ich in meinem Studium geschrieben habe, später wandelte ich die Seite um in einen Blog, auf dem ich vor allem Unterrichtsmaterialien veröffentlichte.

Die Inspiration dazu kam von dem Journalisten Jeff Jarvis. Er warb schon in den Frühzeiten des Internets dafür, dass es viele Vorteile hat, wenn man Teile seines Lebens bzw. Berufslebens ins Internet stellt und mit anderen teilt. Welche Vorteile das genau sind, hat Jarvis in einem Blogeintrag beschrieben:

- ▶ **Öffentlichkeit schafft und verbessert Beziehungen.** Man muss sich mitteilen und offen sein, um Beziehungen zu knüpfen. Wenn man sich dazu entscheidet, nicht öffentlich zu sein, riskiert man, diese Beziehungen zu verlieren.
- ▶ **Öffentlichkeit ermöglicht Zusammenarbeit.** Wenn man das, woran man arbeitet, öffentlich macht, lädt man Menschen dazu ein, das Gemachte zu verbessern. Das ist auch die Philosophie der Open-Source-Software. Öffentlichkeit schafft Vertrauen. Heimlichkeit nicht.
- ▶ **Öffentlichkeit tötet den Mythos der Perfektion.** Wenn wir unseren Arbeitsprozess öffnen, zeigen wir unsere Fehler und können dem Mythos der Perfektion entkommen, der unsere Arbeitsprozesse im Industriezeitalter bestimmt.

1.4 Computerspiele – mehr als nur Zeitvertreib

Durch mein Studium fand ich auch wieder zurück zu meinem alten Hobby Computerspiele. Der Grund dafür ist leider tragisch: Am 26. April 2002 erschoss der 19-jährige Robert Steinhäuser 16 Menschen in seiner Schule und anschließend sich selbst. Wie konnte das nur passieren? Waren – so legten es einige Beiträge in Presse und Fernsehen nahe – gewalttätige Computerspiele schuld? Die Spiele, mit denen ich selbst einen nicht unerheblichen Teil meiner Jugend verbracht habe?

Im Lesesaal meines Wohnheims las ich so ziemlich jeden Artikel, der zu diesem Thema veröffentlicht wurde, und reflektierte, welchen Einfluss Computerspiele auf mein eigenes Leben hatten. Es gab in meinem Leben sicherlich Phasen, in denen ich mehr gespielt habe, als es gut für mich war. Aber selbst als sich alles in meinem Leben darum drehte, wie ich meine Freunde im Deathmatch am effektivsten erledigen kann, bin ich deswegen im Real Life nicht gewalttätiger geworden.

Ich begann daher, mich intensiv mit dem Zusammenhang zwischen virtueller und realer Gewalt zu beschäftigen, schrieb mehrere Hausarbeiten zu dem Thema und fand heraus, dass ich in Sachen Computerspiele mehr Ahnung hatte als ein Großteil der Journalistinnen und Journalisten, die über Computerspiele berichteten.

Seitdem hat mich das Thema Computerspiele nicht mehr losgelassen. Da sie – wie bei mir – oft der erste Berührungspunkt von Kindern mit dem Computer sind, eignen sie sich hervorragend dazu, die Funktionsweise von Computern zu erklären. Ich bin auch fest davon überzeugt, dass Lehrkräfte viel von Computerspiel-Designern lernen können. Was genau, das erfährst du am Ende des vierten Kapitels.

Mittlerweile schreibe ich selbst ab und zu über Computerspiele, etwa für die *Retro-Gamer*-Zeitschrift, bin zu Gast in Podcasts über Games und habe als Beta-Tester und Übersetzer beim Spiel *Nox Archaist* mitgewirkt – dem einzigen Spiel für den Apple II, das auf Steam erschienen ist.



Abbildung 1.8 Auch wenn es nicht so aussieht, ist dieses Spiel im Jahr 2020 erschienen – auf vier Floppy-Disketten!

1.5 Don't believe the hype!

Die Berichterstattung über Computerspiele und Tech-Themen allgemein ist heutzutage unvergleichlich besser als noch vor 20 Jahren. Aber vor allem bei Hype-Themen wie der künstlichen Intelligenz sollte man dennoch nicht alles glauben, was gedruckt und gesendet wird. Nicht umsonst sind kritisches Denken und so etwas wie eine »digitale Aufklärung« Kernforderungen jedes Lehrplans.

Was bei der Berichterstattung über Computer so alles schiefgehen kann, zeigt recht eindrucksvoll die Einführung des Calliope mini, eines kleinen Mikrocontrollers, der sich programmieren lässt und für den Einsatz in der Grundschule konzipiert wurde.

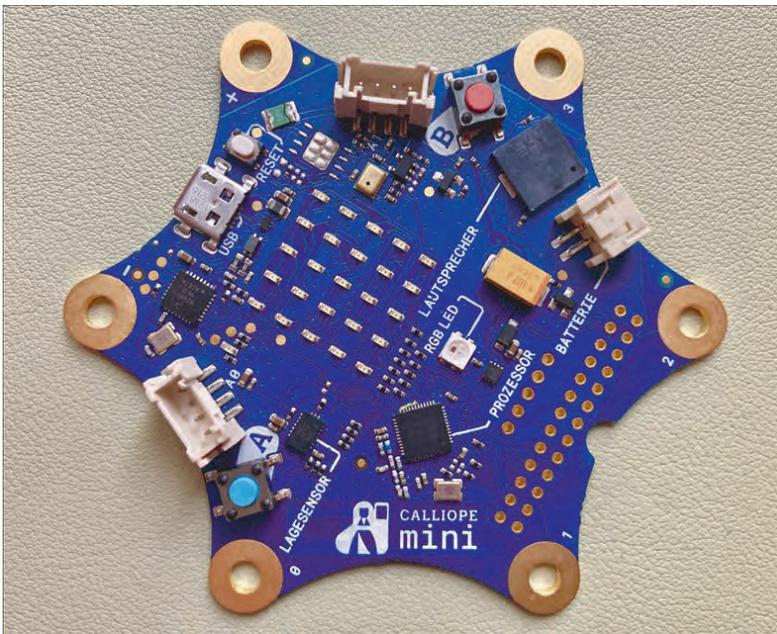


Abbildung 1.9 Der Calliope mini – kleine Platine mit großer medialer Wirkung

Da das Projekt aus Deutschland stammt und man anscheinend in Bezug auf den Einsatz von Technik in der Schule dringend auf der Suche nach positiven Nachrichten war, erschienen Ende 2016 mehrere Artikel in renommierten Zeitungen, die sich wie eine Werbebroschüre lasen – und wie in der Werbung nahm man es mit der Wahrheit nicht immer so ganz genau.

Die Wochenzeitschrift »DIE ZEIT« etwa schrieb, der Calliope mini sei »[d]ie vielleicht größtmögliche Umwälzung des deutschen Schulsystems« und in der Zeitschrift »Die Welt« stand wörtlich: »In Fachkreisen wird schon geraunt, der Calliope mini könne das

Schulsystem revolutionieren.« Selbst die Bundesregierung schrieb auf ihrer Internetseite zum Calliope mini: »Auf der handflächengroßen, sternförmigen Platine sind verschiedene Funktionen, wie beispielsweise ein Mikrofon, ein Prozessor und zwei Motoren angebracht.« Ein kurzer Blick auf die Platine reicht jedoch aus, um festzustellen, dass da weit und breit keine Motoren zu sehen sind.

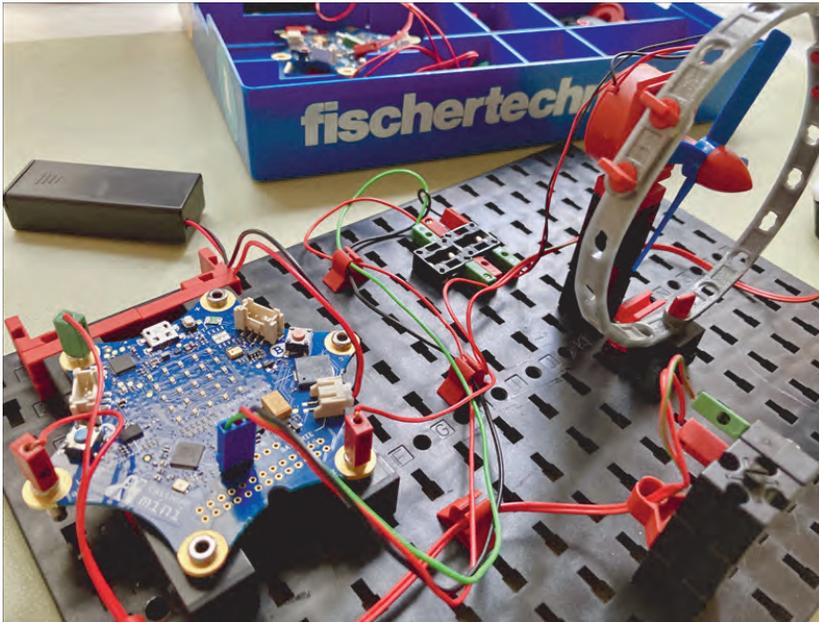


Abbildung 1.10 Der Calliope hat zwar keine Motoren auf der Platine, kann aber welche ansteuern, z. B. mit diesem Fischertechnik-Set.

Wer sich auf diese Quellen verließ, dachte wahrscheinlich wirklich, dass da etwas ganz Großes auf die Schulen zukommt, das die digitale Bildung kräftig vorantreiben wird. Schülerinnen und Schüler sowie Eltern werden jedoch wahrscheinlich wenig davon bemerkt haben, da die Heilsversprechen sich nicht erfüllt haben.

Das ist auch kein Wunder, denn der Calliope mini war eigentlich nichts Neues, sondern in weiten Teilen eine Kopie einer schon zwei Jahre zuvor in England veröffentlichten Platine namens micro:bit, die von der BBC in Kooperation mit Microsoft, Samsung und über 20 weiteren Partnern entwickelt wurde.

Stellt man die beiden Geräte gegenüber, wird das relativ schnell deutlich. Sie verfügen beide über 5×5 rote LEDs, zwei Knöpfe, verschiedene Sensoren und Anschlüsse für weitere Hardware. Und selbst der micro:bit war nicht die erste Platine dieser Art. Bereits Anfang 2015 erschien der CodeBug, der ganz ähnliche Features bietet.

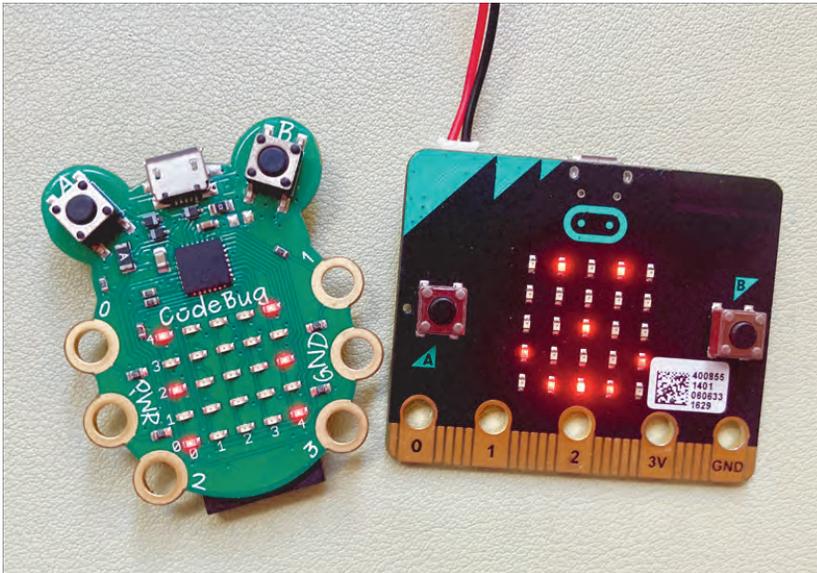


Abbildung 1.11 die Vorbilder des Calliope mini: CodeBug und micro:bit

1.6 Programming-Challenge: Scratch

Jetzt ist es aber wirklich Zeit für die erste Programming-Challenge. Bist du bereit? Dann schnapp dir deinen Raspberry Pi oder Computer und wir programmieren zusammen ein wenig.

Wie? Du hast noch nie programmiert? Kein Problem! Für unsere erste Übung verwenden wir die Programmiersprache *Scratch*, die bereits in Grundschulen dafür verwendet wird, Kindern die Funktionsweise von Computern zu demonstrieren.

Scratch ist die wohl am einfachsten zu erlernende Programmiersprache, mit der sich in wenigen Minuten und ohne Vorkenntnisse ein funktionsfähiges Computerprogramm erstellen lässt. Gehe einfach im Browser deiner Wahl auf die Seite scratch.mit.edu, und schon kann es losgehen. Wenn du einen *Raspberry Pi* nutzt, findest du eine Offline-Variante von Scratch im Ordner *Programmieren* im Startmenü.

Auch wenn Scratch sehr leicht zu erlernen ist, handelt es sich dennoch um eine »richtige« Programmiersprache, mit der sich auch komplexere Projekte realisieren lassen. Außerdem lässt sich eine Vielzahl unterschiedlicher Hardware mit Scratch ansteuern, etwa LEGO-Roboter oder Mikrocontroller wie der micro:bit.

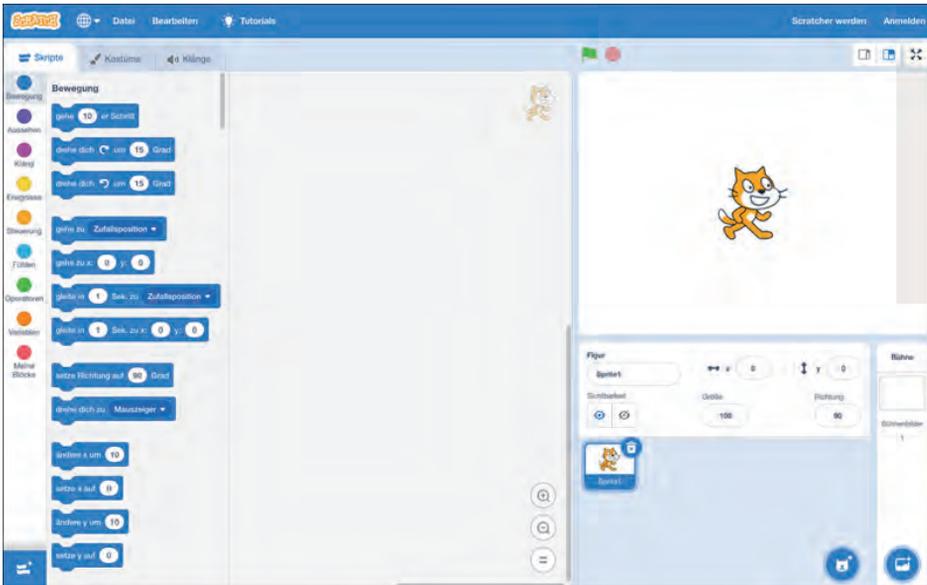


Abbildung 1.12 Der Startbildschirm von Scratch

Scratch ist in vielen verschiedenen Sprachen erhältlich. Um sie auszuwählen, klicke auf das Welt-Symbol in der oberen linken Ecke und wähle die Sprache deiner Wahl aus.

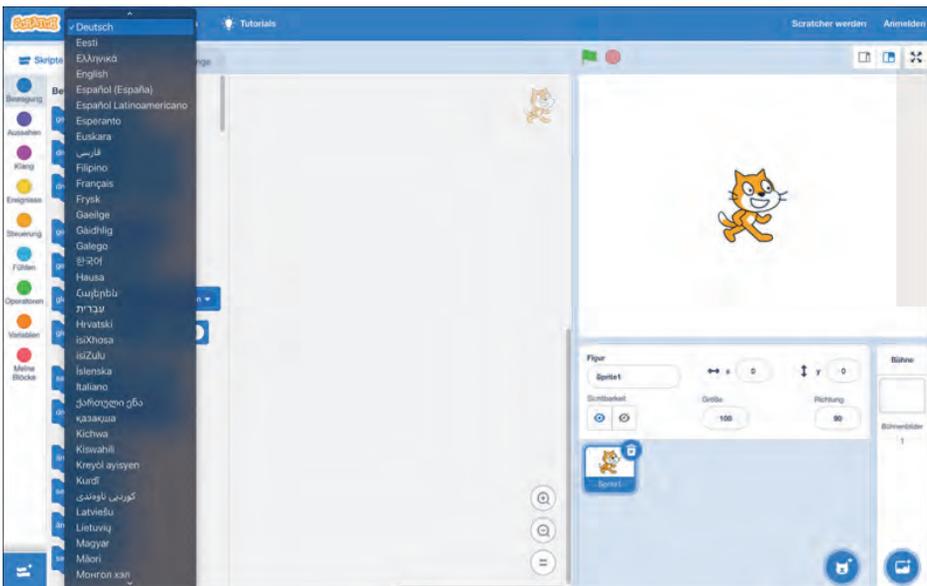


Abbildung 1.13 Scratch bietet viele verschiedene Sprachen zur Auswahl.

Mache dich nun mit der Oberfläche von Scratch vertraut. Oben rechts findest du eine Vorschau deines Programms. Dort siehst du einen weißen Hintergrund und in der Mitte eine Comic-Katze, die du mit der Maus hin- und herbewegen kannst, wenn du die linke Maustaste gedrückt hältst.

Unter der Vorschau siehst du alle Figuren und Hintergründe, die in deinem Programm bislang enthalten sind, also ein weißer Hintergrund (ganz rechts unter der Überschrift BÜHNE) und die Katze. Über die blauen Knöpfe unten rechts kannst du weitere Figuren und Bühnen hinzufügen. Dabei hast du die Möglichkeit, zufällig eine Figur oder eine Bühne einzufügen oder sie selbst in einem kleinen integrierten Zeichenprogramm zu erstellen. Darüber hinaus kannst du auch bereits vorhandene Bilddateien von deinem Computer hochladen.

Info-Box: Programmieren ohne Anmeldung

Scratch lässt sich kostenlos und ohne Anmeldung nutzen. Wenn du jedoch dein Programm speichern möchtest, solltest du oben rechts auf SCRATCHER WERDEN klicken und einen Benutzernamen sowie ein Passwort auswählen. Du benötigst zudem eine Mail-Adresse, um dich anmelden zu können.

Wenn du deine echte Mail-Adresse nicht preisgeben möchtest, verwende einfach eine Wegwerfadresse, die du z. B. bei www.muellmail.com kostenlos und mit einem Mausklick erstellen kannst.

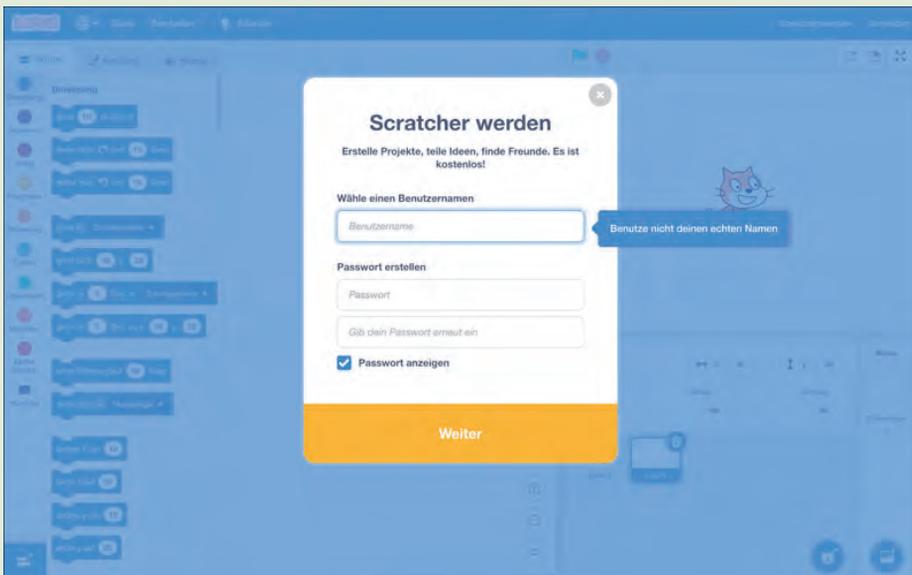


Abbildung 1.14 Der Anmelde-Bildschirm von Scratch

Der wichtigste Bereich von Scratch findet sich jedoch auf der linken Seite. Hier siehst du am Rand verschiedenfarbige Kreise, die dich nach einem Klick zu Programmblöcken der jeweiligen Kategorie führen. Klickst du etwa auf den ersten Kreis BEWEGUNG, erscheinen rechts daneben Blöcke, die deine Figur nach oben, unten, links und rechts bewegen, sie drehen und noch einiges mehr. Klickst du hingegen auf AUSSEHEN, findest du Blöcke, die das Aussehen der Figur ändern, also z. B. Sprechblasen anzeigen oder die Größe und Farbe der Figur ändern.

Die wichtigste und zugleich komplizierteste Kategorie ist jedoch STEUERUNG. Hier findest du Programmblöcke, die für die logische Verknüpfung der anderen Blöcke zuständig sind. Mithilfe des Blocks WENN TASTE ... GEDRÜCKT WIRD kannst du z. B. eine Bewegungssteuerung für deine Figur programmieren.

Ziehe dazu diesen Programmblock nach rechts in das leere Fenster mit der halb durchsichtigen Katze oben rechts. Wenn du mehrere Figuren in deinem Programm hast, zeigt dieses durchsichtige Symbol an, welche Figur (oder welche Bühne) du gerade programmierst. Wähle nun statt der Leertaste die Pfeil-nach-oben-Taste aus, indem du auf das kleine weiße Dreieck klickst.

Als Nächstes benötigen wir den Block ÄNDERE Y UM ... aus der Kategorie BEWEGUNG. Ziehe ihn unter den anderen Block. Wenn du nun mit der rechten Maustaste auf die beiden Blöcke klickst, kannst du den Punkt DUPLIZIEREN wählen und nun programmieren, dass sich die Figur nach unten bewegen soll, wenn du die -Taste drückst. Wie das genau aussieht und wie du auch die Bewegung nach links und rechts hinzufügst, siehst du im folgenden Bild.

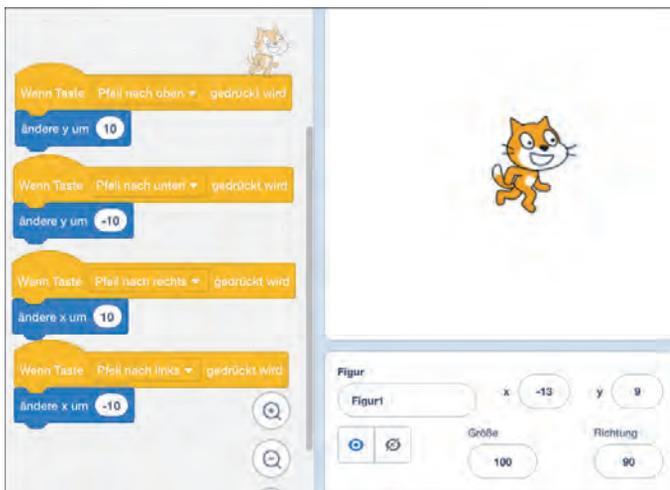


Abbildung 1.15 Mit diesem Programm kannst du deine Katze mit den Pfeiltasten steuern.

Figuren in Scratch können »Kostüme« anziehen, um z. B. verschiedene Gesichtsausdrücke darzustellen oder eine kleine Animation zu erstellen. Wenn du auf den Reiter KOSTÜME oben rechts klickst, siehst du, dass die Katze standardmäßig bereits zwei Kostüme besitzt, mit denen sich eine einfache Animation programmieren lässt.

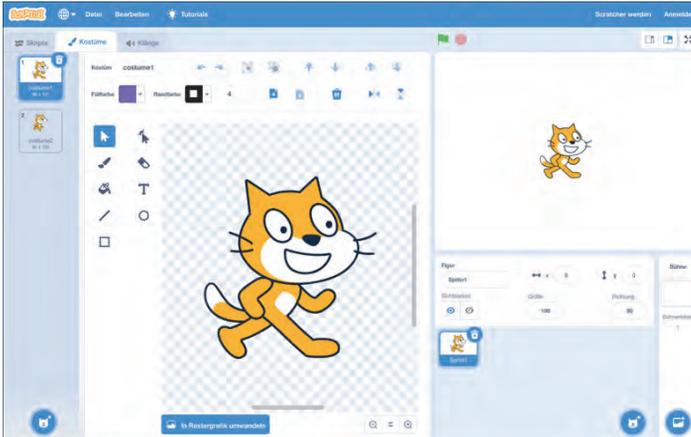


Abbildung 1.16 Oben links werden die beiden Kostüme der Katzenfigur angezeigt.

Um deine Figur zu animieren, musst du deinen vier Programmblöcken nur einen einzigen Befehl hinzufügen, nämlich WECHSLE ZUM NÄCHSTEN KOSTÜM aus der Kategorie AUSSEHEN. Dein Programmcode sollte dann wie folgt aussehen:

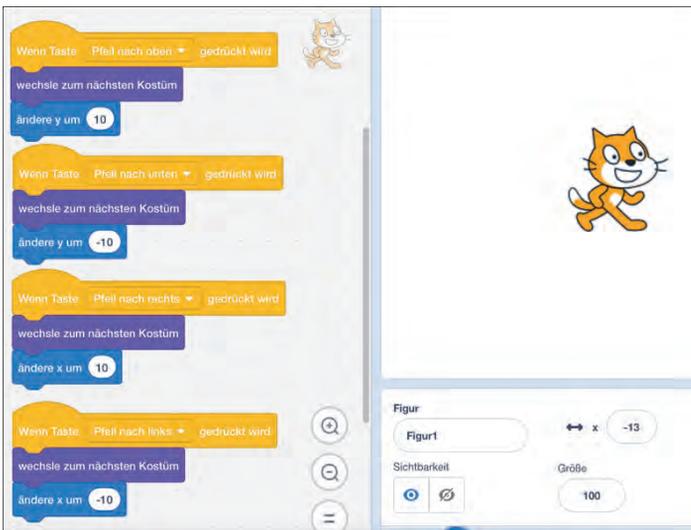


Abbildung 1.17 Nun bewegt sich die Katze mit einer kleinen Animation.

Wie du siehst, ist der Einstieg ins Programmieren mit Scratch kinderleicht. Da man mit Scratch keinen »falschen« Code schreiben kann, der das Programm abstürzen lässt, kannst du einfach ein bisschen ausprobieren und versuchen, welche Blöcke untereinander passen und was sie bewirken.

Durch einen Klick auf den Knopf TUTORIALS am oberen Bildschirmrand gelangst du zu vielen Anleitungen, die dir weitere Funktionen von Scratch genauer erklären.

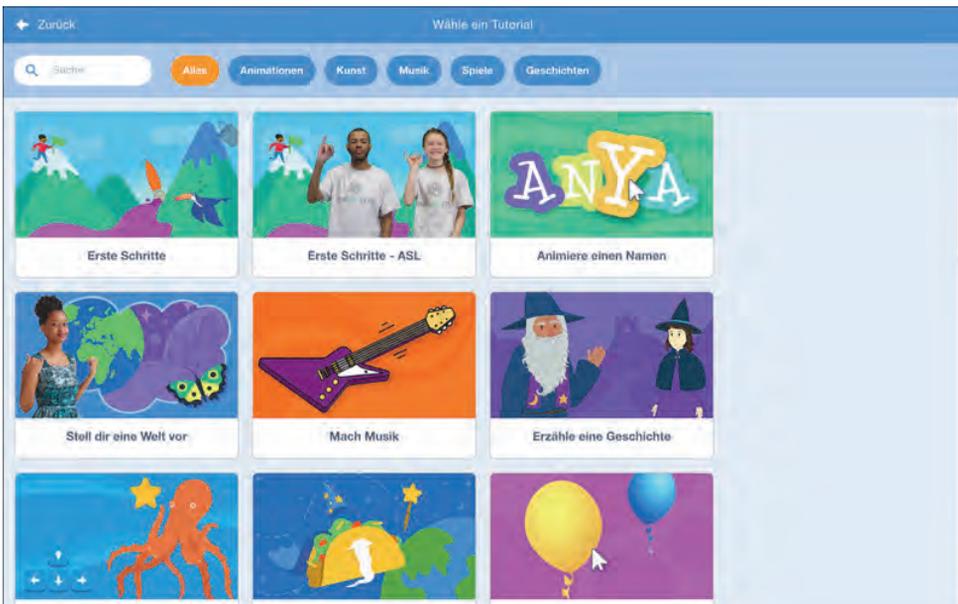


Abbildung 1.18 Scratch bietet viele Tutorials, mit denen du die Programmiersprache erlernen kannst.

1.7 Im Porträt: Mitchel Resnick

Und – gefällt dir Scratch? Wenn ja, interessiert dich vielleicht auch, wie diese Programmiersprache für Kinder überhaupt entstanden ist und welche Idee dahintersteckt.

Der 1956 geborene Mitchel Resnick hat den wohl coolsten Job-Titel aller Zeiten: LEGO-Professor. Noch dazu arbeitet er an einem der (zumindest für Nerds jeden Alters) coolsten Orte überhaupt, dem *MIT Media Lab*. Laut Resnick wird Technik bei Kindern meistens dazu eingesetzt, entweder zu unterhalten oder zu belehren. In seiner Forschungsgruppe am MIT Media Lab arbeitet man hingegen daran, Technologien zu erschaffen, die Kinder selbst kontrollieren und gestalten können, um etwa eigene Geschichte zu erzählen, Roboter zu bauen oder Animationen zu erstellen.



Abbildung 1.19 Alle lieben Toy Story? Nein! Mitchel Resnick stört an dem Film, dass der gute Junge brav mit Pferd und Cowboy spielt (links im Bild) und der Bösewicht derjenige ist, der kreativ neue Dinge baut (siehe rechts). Lernen entsteht für ihn nämlich gerade im spielerischen Umbau der Welt um uns herum. (Foto: Jan-Felix Müller)

Einer der kreativsten Orte der Welt ist für Resnick der Kindergarten, wo Kinder auf spielerische Art und Weise ihre Welt gestalten, indem sie basteln, bauen, malen, und zwar dann, wenn sie Lust dazu haben. Die von ihm gegründete Gruppe am MIT trägt daher auch den passenden Namen »Lifelong Kindergarten«. Wenn man das von Resnick initiierte Lernlabor am MIT besucht, sieht es dort tatsächlich aus wie in einem Kindergarten für Erwachsene. Überall finden sich verschiedenste Materialien zum Basteln – sowohl digitale wie Mini-Computer und Sensoren – als auch »analoge« wie Kleber, Schere und Papier. Alle diese Materialien werden dazu genutzt, Prototypen zu basteln und Dinge auszuprobieren, die dann vielleicht eines Tages zu einem marktreifen Produkt werden. An diesem Ort ist übrigens auch die berühmte Guitar-Hero-Serie entstanden. Wenn du mehr darüber erfahren möchtest, schau doch einmal im Kapitel über Computerspiele nach.

Und nicht zufällig finden sich überall LEGO-Steine, denn es besteht eine enge Kooperation zwischen dem MIT und dem Spielzeugkonzern aus Dänemark. 1998 wurde im Royal College of Art in London das erste Produkt dieser Kooperation vorgestellt: *LEGO Mindstorms*. Hauptbestandteil von LEGO Mindstorms ist ein kleiner Computer namens »RCX«, der in einem LEGO-Block untergebracht und mit einem 16-MHz-Prozessor und 32 KB RAM ausgestattet ist. Er wurde im Set mit zwei Motoren, zwei Berührungssensoren

ren sowie einem Licht-Sensor verkauft. Die Datenübertragung erfolgte bei diesem Modell mittels Infrarot. Funfact: Die Übertragungsgeschwindigkeit stieg merklich an, wenn man das Licht ausschaltete.



Abbildung 1.20 Gitar Hero Live war das letzte Spiel der »Guitar Hero«-Spielserie.

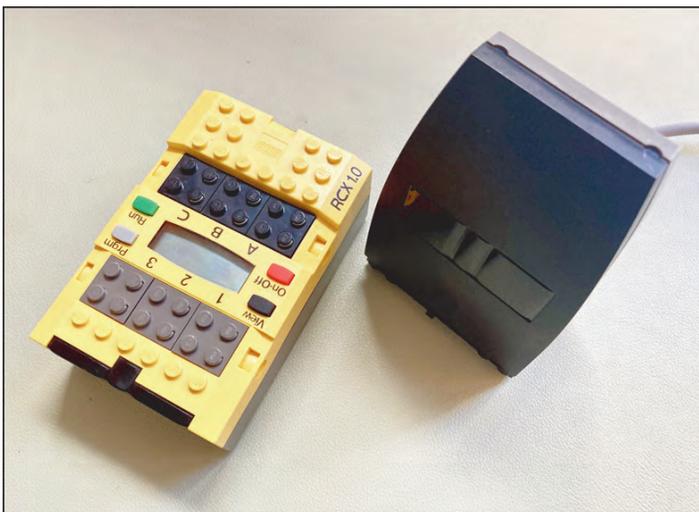


Abbildung 1.21 Der RCX 1.0 war das erste Produkt aus der LEGO-Mindstorms-Serie. Die Datenübertragung erfolgte per Infrarot-Schnittstelle.

Der Nachfolger des RCX-Blocks erschien 2006 und nannte sich Mindstorms NXT. Das Set kam mit drei Motoren, einem Lichtsensor, einem Soundsensor, einem Ultraschallsensor und einem Touch-Sensor. 2013 erschien dann die nächste Generation der programmierbaren LEGO-Steine: EV3. Im Gegensatz zu den Vorgängermodellen ist hier also ein »richtiger« Computer verbaut, der einige Zeit braucht, bis er einsatzbereit ist. Dafür bietet er aber auch mehr Funktionen, darunter Bluetooth, WiFi und einen SD-Kartenslot. 2020 erschien mit dem Robot Inventor das vorerst letzte Mindstorms-Produkt von LEGO.

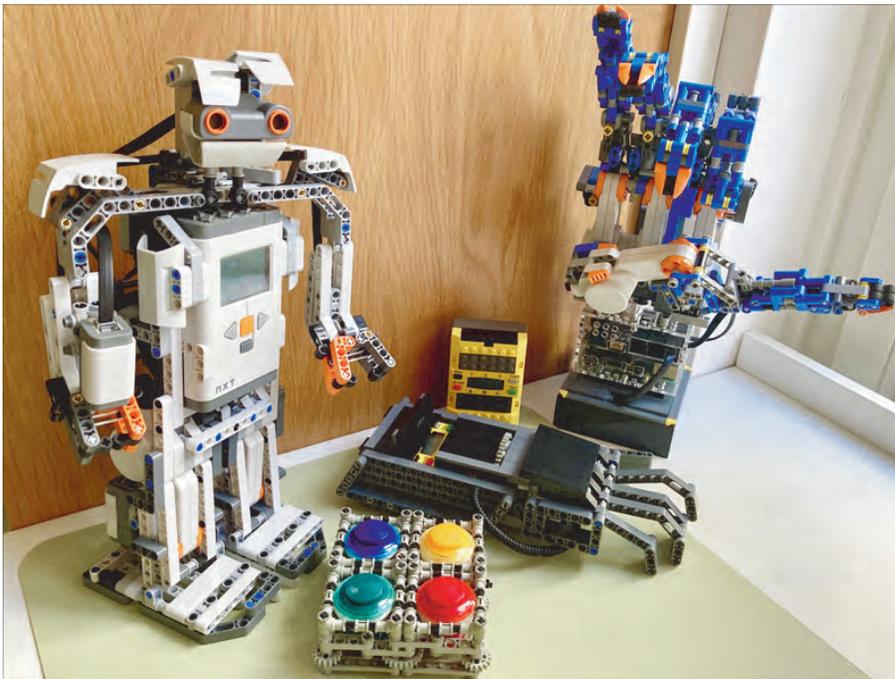


Abbildung 1.22 LEGO-Mindstorms-Konstruktionen mit dem RCX- und NXT-System

Durch die Kooperation mit LEGO entstanden aber nicht nur interessante Produkte, sondern auch wegweisende Lerntheorien, z. B. in dem Text »LEGO/Logo: Learning Through and About Design« von Mitchel Resnick und Stephen Ocko, der im Buch »Constructionism« erschienen ist. Bereits 1991 dachten sie darüber nach, wie man Schülerinnen und Schülern die Bedeutung von »Design« beibringen könne. In dem Artikel schreiben sie, dass Design-Aktivitäten mit Schülerinnen und Schülern dann am besten funktionieren, wenn sie die Freiheit haben, selbst Produkte zu erstellen, die für sie (oder andere) bedeutsam sind. Denn dann steckt in den Produkten eine besondere Sorgfalt und Interesse, das in der Schule selten zu finden ist. Wenn Schülerinnen und Schüler diese Frei-

heit haben, probieren sie auch mehr aus und verinnerlichen die mathematischen und wissenschaftlichen Konzepte, die ihrem Produkt zugrunde liegen.

Resnick und Ocko fassten am Ende des Artikels ihr pädagogisches Programm in wenigen Leitlinien zusammen:

- ▶ Kindern die Kontrolle übergeben
- ▶ mehrere Wege zum Lernziel ermöglichen
- ▶ einen Sinn für Gemeinschaft wecken

Auch über ein Vierteljahrhundert später haben diese Ideen nichts von ihrer Aktualität verloren, und Mitch Resnick, übrigens heute noch Leiter der »Lifelong Kindergarten«-Gruppe am MIT, entwarf einige Jahre später die bald schon erfolgreichste Programmiersprache für Kinder: Scratch.

1.8 Maker-Challenge: Marshmallow-Challenge

Puh – nach dem Einstieg in die Programmierung hast du jetzt vielleicht Lust, ein wenig in der realen Welt aktiv zu werden. Jedes Kapitel in diesem Buch enthält eine sogenannte »Maker-Challenge« – eine Anregung, selbst etwas zu bauen oder zu basteln.

Die erste davon ist die sogenannte »Marshmallow-Challenge«. Du kannst sie alleine ausprobieren, viel mehr Spaß macht sie allerdings in einer großen Gruppe, etwa einer Schulklasse. Vielleicht kannst du ja deine Lehrkraft dazu überreden, diese Maker-Challenge mit der ganzen Klasse zu machen.

Die Regeln sind einfach: Vierer-Teams haben 18 Minuten Zeit, um aus einem Meter Klebeband, einem Meter Schnur und 20 ungekochten Spaghetti-Nudeln einen Turm zu bauen, auf dessen Spitze ein Marshmallow befestigt ist. Anschließend wird gemessen, in welcher Höhe sich das Marshmallow befindet. Das Team, dessen Marshmallow am weitesten von der Tischoberfläche entfernt ist, hat gewonnen.

Tom Wujec, der Erfinder der Marshmallow-Challenge, erklärt in seinem TED-Vortrag, wie er über 70 unterschiedlichsten Teams diese Aufgabe gestellt hat und welche Lehren er daraus gezogen hat. Beispielsweise schnitten Kindergarten-Kinder mit am besten ab, da sie weniger darum kämpften, wer die Leitung in der Gruppe hat, und verschiedenste Prototypen bauen und testen, statt die eine perfekte Lösung zu finden.

Eine weitere interessante Erkenntnis ist, dass eine hohe Belohnung für den Gewinner nicht dazu führt, dass die einzelnen Teams besser abschneiden – im Gegenteil! Als ein Preis in Höhe von 10.000 US-\$ ausgesetzt wurde, schaffte es keines der zehn teilnehmenden Teams, einen tragfähigen Turm zu bauen.

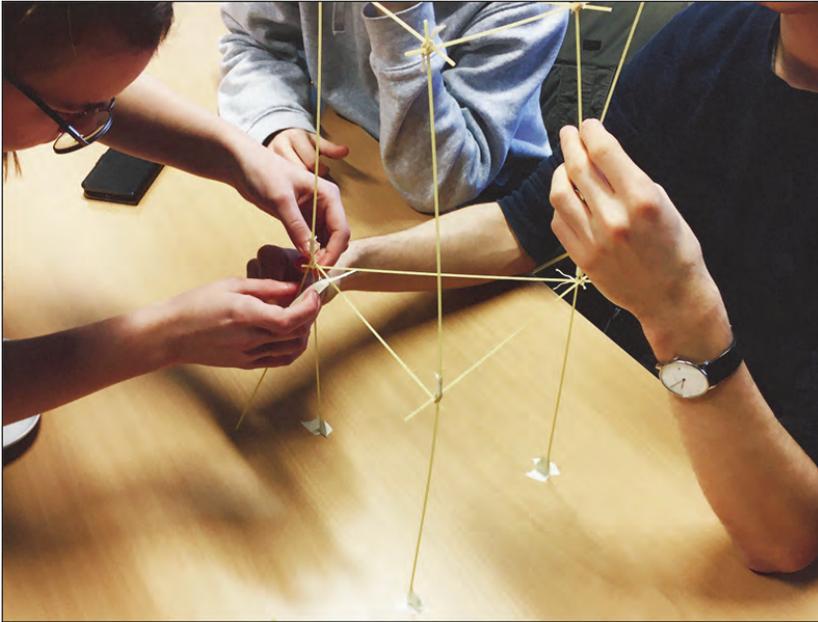


Abbildung 1.23 Konzentriertes Bauen mit Nudeln bei der Marshmallow-Challenge

Wenn du selbst in deiner Klasse die Challenge machen möchtest, kannst du im Anschluss an die Bauphase (und Siegerehrung) einige Fragen mit den Teams diskutieren, z. B.:

- ▶ Gab es jemanden, der im Team die Führung übernommen hat? Wenn ja, wie wurde diese Person auserwählt?
- ▶ Haben sich alle im Team zu gleichen Teilen an der Lösung des Problems beteiligt?
- ▶ Was hätte man rückblickend besser machen können?
- ▶ War der Zeitdruck hilfreich oder hinderlich?

Kapitel 5

Just for fun: Gaming-Basteleien

Ah, endlich ein Kapitel, das sich von vorne bis hinten nur dem Basteln mit Elektronik widmet. Wenn du also schon immer von deinem eigenen Controller oder der selbst gebauten Spielkonsole geträumt hast, dann schnapp dir die Werkzeugkiste und lies schnell weiter.

Zocken macht Spaß, aber noch mehr Spaß macht es, sich seine eigene Gaming-Hardware zu basteln. Wäre es nicht cool, wenn du nicht nur deinen eigenen Controller, sondern auch eine eigene Konsole basteln könntest?

In diesem Kapitel zeige ich dir einige meiner Controller- und Konsolen-Basteleien und erkläre dir, wie sie funktionieren, wie schwierig das Projekt ist und welche Teile und Werkzeuge du dafür benötigst. So kannst du dir das Projekt raussuchen, das dich am meisten anspricht.

Viele Projekte basieren auf dem Raspberry Pi. Der Mini-Computer, der ab 15 € (für den Pi Zero W) erhältlich ist, hat genügend Power zum Emulieren alter Spielkonsolen und kann zudem für viele weitere Gaming-Projekte, etwa das Programmieren von Minecraft, verwendet werden.

In vielen Workshops habe ich die Erfahrung gemacht, dass das Bauen eines eigenen Controllers nicht nur ein motivierender und einfacher Einstieg ins Basteln mit elektronischen Bauteilen ist, sondern auch das kreative Denken fördert. Wenn man verstanden hat, wie einfach das Bauen eines Controllers ist, entstehen nämlich fast automatisch Ideen für eigene Projekte im Kopf, etwa der Bau einer interaktiven Landkarte mit Knöpfen und LEDs für die nächste Präsentation in der Schule oder der Bau eines eigenen digitalen Musikinstruments.

5.1 Die Raspberry-Pi-Konsole

Benötigte Hardware:

- ▶ Raspberry Pi

Die einfachste Art und Weise, sich eine kleine Konsole zu basteln, ist die Installation eines entsprechenden Betriebssystems auf dem Raspberry Pi. Am einfachsten funktioniert es mit dem System *Recalbox*.

Die Installation ist so einfach wie möglich gehalten. Unter der Kategorie DOWNLOAD auf der Seite www.recalbox.com kann man die Hardware-Plattform auswählen.

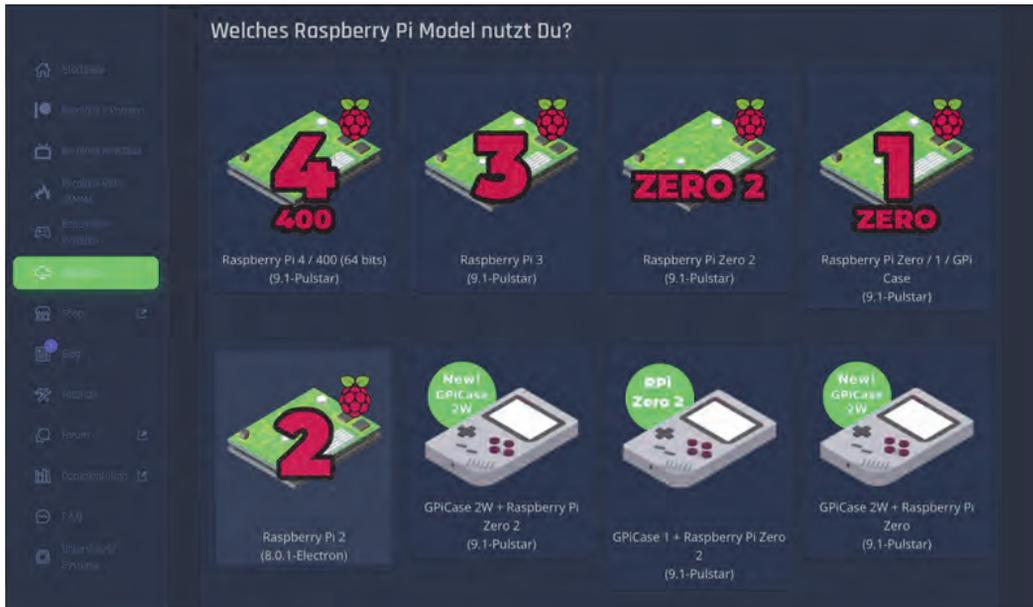


Abbildung 5.1 Das Download-Menü auf der Recalbox-Website

Nach einem Klick auf RASPBERRY PI kann man sein Pi-Modell und das Betriebssystem des aktuellen Rechners auswählen. Mithilfe des anschließend herunterladbaren Programms »Raspberry Pi Imager« kann man dann ganz einfach Recalbox auf eine Micro-SD-Karte aufspielen.

Ein angeschlossener Game-Controller wird in der Regel automatisch erkannt. Wenn du die Knöpfe anders belegen möchtest, schließe einfach eine Tastatur an den Raspberry Pi an, drücke die -Taste und gehe zum Menüpunkt CONTROLLER SETTINGS. Bestätige mit der Taste  auf der Tastatur, und anschließend kannst du alle Funktionen des Controllers neu belegen.

Alte Spiele (ROMS) aus dem Internet

Mit Recalbox lassen sich theoretisch Tausende alter Spiele für beliebte Konsolen wie das NES, die Playstation 1 oder den Game Boy emulieren. Diese Emulation alter Hardware ist

auch zu 100 % legal. Etwas anders sieht es bei den Spielen aus. Es ist zwar gesetzlich erlaubt, ein Spiel, das man besitzt, auf einem Emulator zu spielen, aber deswegen darf man nicht automatisch die entsprechende ROM-Datei aus dem Internet laden.

Recalbox bietet aus diesem Grund viele vorinstallierte Spiele für unterschiedliche Systeme, die frei von Copyright sind. So kannst du auch ohne illegale ROMs viel Spaß mit deiner selbst gebauten Retro-Konsole haben.

Wenn du möchtest, kannst du dir nun noch ein hübsches Gehäuse für deine Raspberry Pi-Konsole bauen. Hier drei Anregungen:

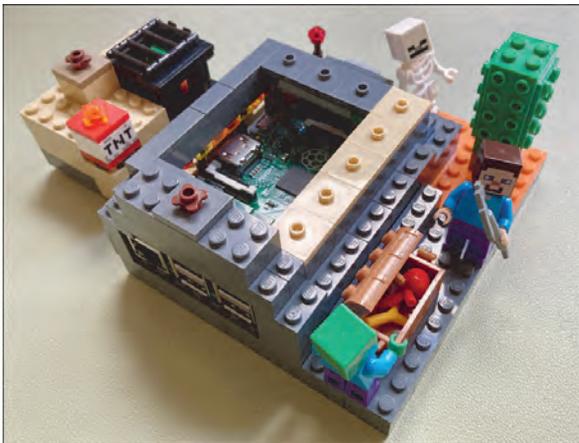


Abbildung 5.2 Ein Pi-Gehäuse aus LEGO im Minecraft-Style. Die Bauanleitung findest du unter: <https://medienistik.wordpress.com/2016/10/04/minecraft-lego/>.



Abbildung 5.3 Ein weiteres Gehäuse für den Raspberry Pi aus LEGO, das an ein Nintendo-Entertainment-System erinnert



Abbildung 5.4 Ein Raspberry Pi-Gehäuse aus Holz, hergestellt mit einem Lasercutter und anschließend angemalt

5.2 Der Ich-mach-alles-zum-Controller-Controller

Benötigte Materialien:

- Makey-Makey-Platine

Um zu verstehen, wie ein Gaming-Eingabegerät funktioniert, genügt ein Blick ins Innere. Unter dem Steuerkreuz und den Knöpfen befinden sich zwei voneinander getrennte Flächen aus Metall. Wird der Knopf gedrückt, senkt sich ein Stück leitendes Gummi auf beide Flächen und verbindet sie.

Wenn du einen Knopf auf einem Controller drückst, schließt du also einen Stromkreis, wodurch das entsprechende Signal an die Konsole oder den Computer weitergeleitet werden kann.

Dieses Wissen genügt schon, um dir deinen eigenen Controller zu bauen. Um den Aufwand so gering wie möglich zu halten, verwenden wir für das erste Projekt einen Controller-Bausatz namens *Makey Makey*.

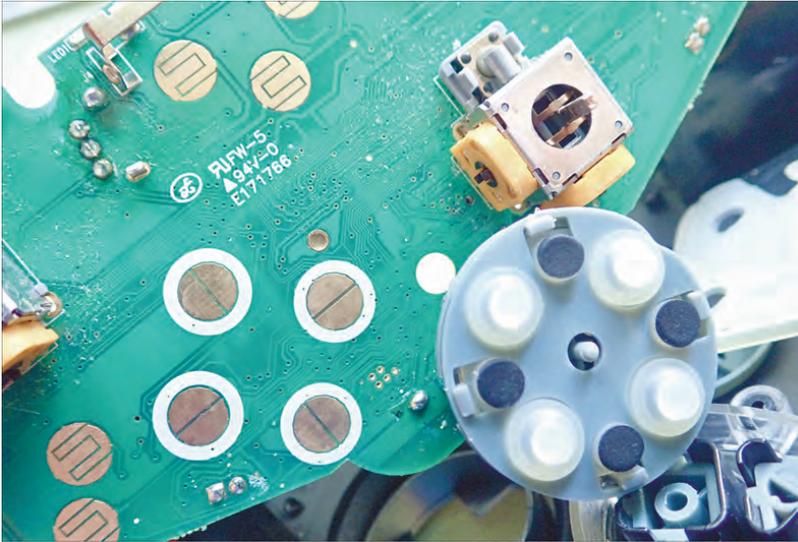


Abbildung 5.5 Das Innere eines Game-Controllers: Oben rechts sieht man den Analog-Stick; die runden goldenen Stellen auf der Platine sind die Kontakte, die bei einem Knopfdruck durch leitendes schwarzes Gummi auf der Unterseite des D-Pads (siehe unten rechts im Bild) verbunden werden.

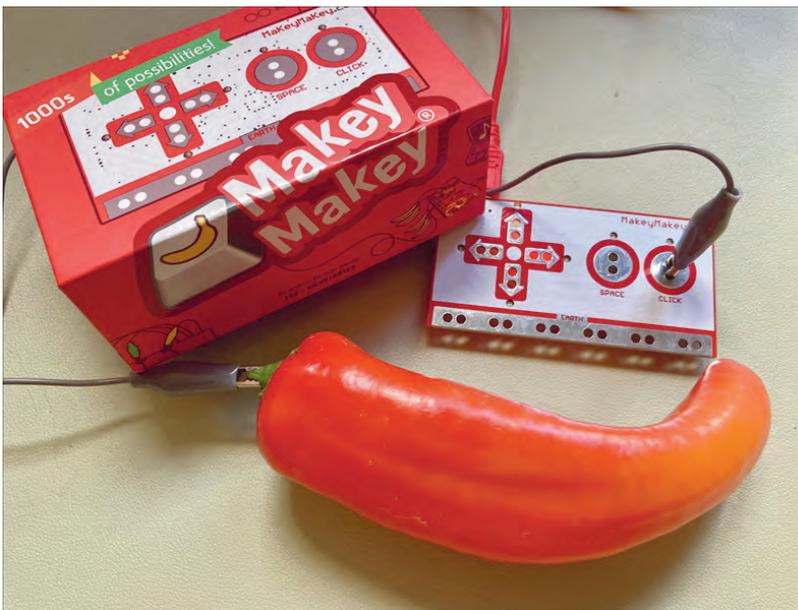


Abbildung 5.6 Eine Paprika zum Controller machen? Kein Problem mit dem Makey Makey.

Der Makey Makey besteht aus einer Platine, die so ähnlich aussieht wie das, was in einem einfachen Gamepad steckt. Auf der Vorderseite befinden sich eine Art Steuerkreuz, zwei Knöpfe und eine Reihe von Löchern auf der Unterseite.

Um den Makey Makey in Betrieb zu nehmen, steckst du ihn einfach mit dem mitgelieferten USB-Kabel an einen PC an. Hältst du nun einen Finger auf die untere Leiste des Makey Makey und berührst mit einem anderen eine der silbernen Flächen darüber, leuchtet eine grüne LED, die anzeigt, dass der Makey Makey einen Tastendruck registriert hat.

OK – bis hierhin ist der Makey Makey also das unpraktischste Gamepad aller Zeiten. Seine Stärke liegt aber darin, dass du mithilfe der beigelegten Krokodilklemmen jeden beliebigen leitenden Gegenstand an die Platine anschließen und ihn sozusagen zu einer Taste machen kannst. Du kannst also – das Lieblingsbeispiel der Erfinder des Makey Makey – ein Bananenklavier bauen oder einfach einen Controller mit dem Bleistift zeichnen und ihn dann tatsächlich verwenden, schließlich leitet Graphit ebenfalls Strom. Einfacher kann man sich keinen Controller bauen. Wenn dir übrigens die Anschlüsse an der Vorderseite des Makey Makey nicht ausreichen, findest du auf der Rückseite der Platine noch weitere Anschlussmöglichkeiten, um beispielsweise den Mauscursor zu steuern oder weitere Tasten zu verwenden.

5.3 Der One-Button-Ü-Ei-Controller

Diese Anleitung zeigt dir, wie du einen einfachen Gamecontroller mit nur einer Taste sowie einer LED bastelst und an den Raspberry Pi anschließt.

Benötigte Materialien:

- ▶ ein Raspberry Pi (jedes Modell)
- ▶ zwei Mikrotaster
- ▶ eine Reißzwecke
- ▶ ein Stift
- ▶ einige Jumper-Kabel (4 × Stecker-Buchse und 4 × Buchse-Buchse)
- ▶ zwei Überraschungseier
- ▶ ggf. zwei LEDs sowie zwei Widerstände (220 Ohm)

Elektronische Tasten gibt es in einer schier unendlichen Vielfalt. Hier nur eine kleine Auswahl:

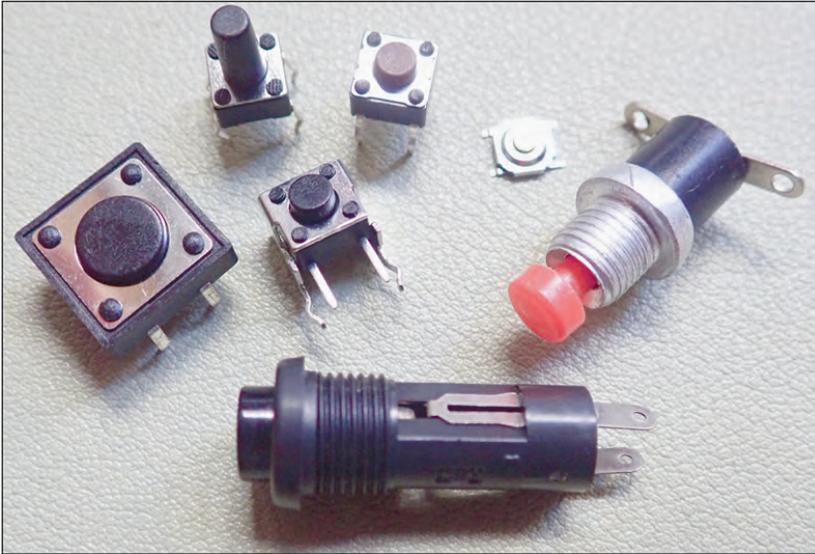


Abbildung 5.7 Elektronische Tasten gibt es in den unterschiedlichsten Varianten zu kaufen.

Für unser Bastelprojekt verwenden wir eine Taste, bei der die Pins (wie beim Knopf ganz links) nach unten zeigen. In einer kleinen Coding-Challenge zeige ich dir außerdem, wie du den Joystick testen und dir dein eigenes 2-Spieler-Reaktionsspiel für zwei selbst gebaute Controller programmieren kannst. Um deinen ersten Controller mit einem Überraschungsei zu bauen, folge einfach dieser kleinen Anleitung:

1. Als Erstes musst du mit der Reißzwecke einige Löcher in die Oberseite des Ü-Eis machen, und zwar so, dass der Knopf und die LED hineinpassen. Halte dazu die Bauteile einfach an, dann findest du leicht den richtigen Abstand. Alternativ kannst du die Beine des Tasters auch in ein Stempelkissen und anschließend auf das Ei drücken, dann hast du vier kleine Punkte, die du mit der Reißzwecke durchstechen kannst.



Abbildung 5.8 Das Ü-Ei sollte nun ungefähr so aussehen.

2. Stecke nun den Taster und die LED in die Löcher.

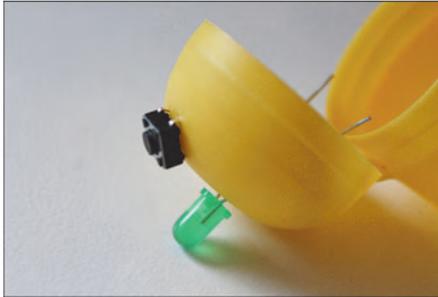


Abbildung 5.9 Sollte die Taste nicht halten, kannst du sie auch mit ein wenig Kleber am Ü-Ei befestigen.

3. Mache jetzt mit einem Stift ein Loch in die Unterseite des Ü-Eis. Du kannst vorher mit der Reißzwecke einige Löcher ins Plastik machen, dann geht es einfacher.

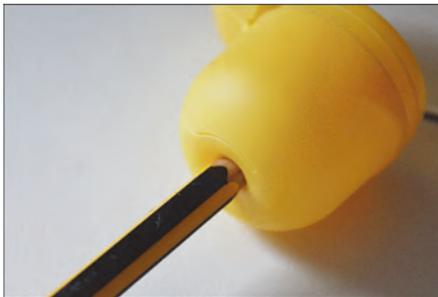


Abbildung 5.10 Durch dieses Loch werden später die Kabel geführt.

4. Nimm nun den Widerstand und wickle ihn um das kürzere Bein der LED. Das geht ein bisschen leichter, wenn du ihn vorher durch das Loch oben steckst.



Abbildung 5.11 Das kürzere Bein einer LED ist der Minuspol (Kathode). Das lange Bein (Anode) ist demnach der Pluspol.

5. Ziehe nun die vier Jumperkabel durch das untere Loch, sodass sich die vier Buchsen außerhalb des Ü-Eis und zwei Buchsen und zwei Stecker in der Innenseite befinden. Verbinde die beiden Buchsen mit dem langen Bein der LED und dem losen Ende des Widerstands. Drücke dann die beiden Stecker so durch die Löcher beim Taster, dass sie zwei nebeneinanderliegende Beine berühren.

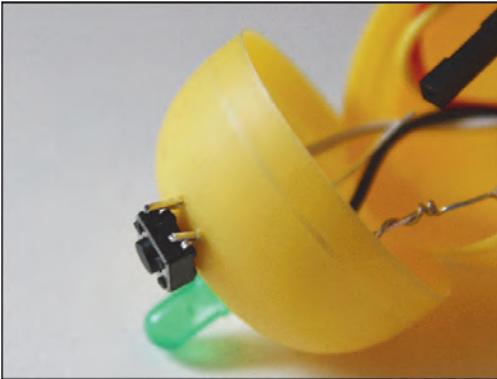


Abbildung 5.12 Damit ein Stromkreis entsteht, müssen die beiden Steckerenden der Jumperkabel zwei nebeneinanderliegende Beine des Tasters berühren.

6. Nun kannst du das Ü-Ei schließen und die vier losen Enden der Jumperkabel an den Raspberry Pi anschließen.

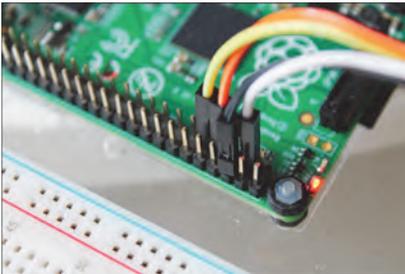


Abbildung 5.13 Im Bild führen das schwarze und das weiße Jumperkabel zum Taster. Das rote Kabel führt zum langen Bein der LED, das gelbe zum Widerstand, der um das kurze Bein der LED gewickelt ist.

5.4 Programming-Challenge

Nun kannst du auf dem Raspberry Pi ein paar kleine Python-Programme schreiben, die die Funktionsfähigkeit deines Controllers testen. Dazu kannst du das Programm *IDLE3* verwenden, das du im Startmenü des Raspberry Pi im Ordner *Entwicklung* findest. Soll-

test du noch mehr Hilfe zur Programmierung mit Python benötigen, lies einfach das Kapitel »Einstieg in Python« in Anhang B dieses Buches.

Das erste Programm überprüft, ob die LED funktioniert:

```
from gpiozero import LED
from time import sleep

led1 = LED(4)
while True:
    led1.on()
    sleep(1)
    led1.off()
    sleep(1)
```

Wenn die LED blinkt, kannst du mit dem folgenden Programm testen, ob auch der Taster funktioniert:

```
from gpiozero import Button, LED

led1 = LED(4)
Spieler1 = Button(2)
while True:
    if Spieler1.is_pressed():
        led1.on()
    else:
        led1.off()
```

Nun kannst du einen zweiten Controller bauen und ihn an den Raspberry Pi anschließen. Hier siehst du eine Skizze, die dir zeigt, an welche Pins du die Kabel anschließen musst:

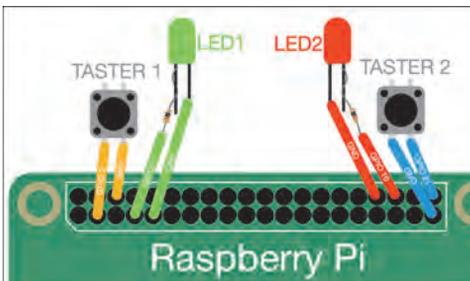


Abbildung 5.14 An diese Pins müssen die beiden Ü-Ei-Controller angeschlossen werden. Der Knick im weißen Kasten um die GPIO-Pins markiert übrigens auf jedem Raspberry Pi den ersten Pin und hilft dir dabei, die Kabel richtig anzuschließen.

Wenn du beide Controller angeschlossen hast, kannst du nun das *Reaktionsspiel* programmieren. Wenn das Programm startet, fangen beide LEDs an zu leuchten. Sobald sie ausgehen, müssen beide Spieler so schnell wie möglich ihren Knopf drücken. Wer schneller ist, gewinnt das Spiel.

```
from gpiozero import Button, LED
from time import sleep
import random
led1 = LED(4)
led2 = LED(16)
Spieler1 = Button(2)
Spieler2 = Button(21)
Zeit = random.uniform(5,10)
led1.on()
led2.on()
sleep(Zeit)
led1.off()
led2.off()

while True:
    if Spieler1.is_pressed:
        print("Spieler 1 gewinnt!")
        break
    if Spieler2.is_pressed:
        print("Spieler 2 gewinnt!")
        break
```

Und hier ist noch lange nicht Schluss. Wenn du willst, kannst du deinen Controller noch etwas ausbauen und hübsch gestalten. Hier eine Variante mit drei Knöpfen und einem etwas hübscheren Outfit:



Abbildung 5.15 Ein Ü-Ei-Controller für Spiele, die mehr als eine Taste benötigen.

Und Knöpfe lassen sich natürlich auch überall einbauen, etwa in dieses Gehäuse aus LEGO:



Abbildung 5.16 Vier Knöpfe in einem Gehäuse aus LEGO Technic – die Kabel lassen sich an den GPIO-Port anschließen. Die Ground-Anschlüsse (also jeweils eines der beiden Anschlusskabel) beliebig vieler Knöpfe lassen sich übrigens zusammenlöten, um Kabel zu sparen.

5.5 Der IKEA-Joystick

Nun kannst du dich daran wagen, einen richtigen Controller zu bauen, der aus mehr als nur einer Taste besteht. Dafür benötigst du folgende Teile:

- ▶ eine IKEA-Kiste vom Typ GLIS (oder ein anderes Gehäuse)
- ▶ einen Arcade-Joystick, vier Arcade-Buttons (oder mehr) und einen USB-Encoder samt Kabeln (Tipp: Alle diese Teile sind oft in einem Set namens »Arcade DIY Kit« im Internet erhältlich.)
- ▶ eine Bohrmaschine mit Stufenbohrer; diese Art Bohrer ermöglicht es, verschieden große Löcher zu bohren



Abbildung 5.17 »Arcade DIY Sets« enthalten alles, was man für den Bau eines (bzw. in diesem Fall von zwei) Joysticks benötigt.

Überleg dir zunächst, wie viele Knöpfe du in deinem Joystick haben möchtest. Zum Spielen alter Konsolen-Klassiker reichen oftmals vier Knöpfe aus. Zum Bohren der großen Löcher verwendest du am besten einen Stufenbohrer. Achte aber darauf, dass du die Löcher nicht zu groß bohrst.



Abbildung 5.18 Eine IKEA-GLIS-Kiste, die noch nicht weiß, dass sie gleich ein Joystick wird.

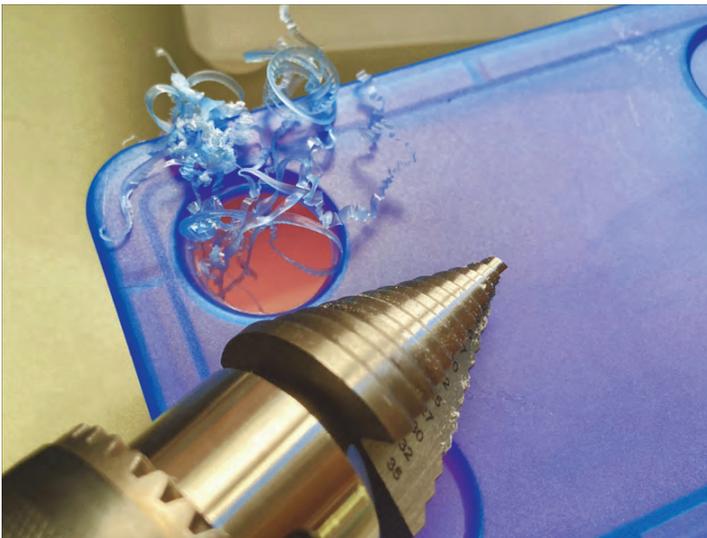


Abbildung 5.19 Dank des Stufenbohrers weiß sie es jetzt!

Schraube anschließend die große Metallplatte auf der Oberseite des Joysticks ab und verwende sie als Schablone, um die Löcher für den Joystick zu bohren (siehe Abbildung 5.20). Zusätzlich zu dem großen Loch in der Mitte benötigst du vier kleine Löcher zum Befestigen der Schrauben. Verwende hierzu einfach einen kleinen Bohrer. Bohre außerdem ein Loch in die Unterseite des Gehäuses, aus dem du später das USB-Kabel herausführen kannst.

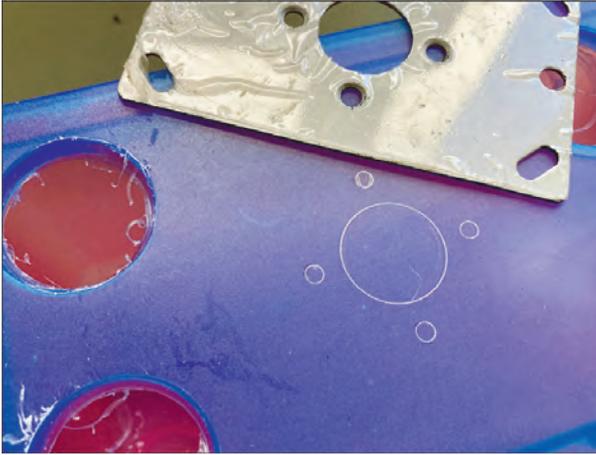


Abbildung 5.20 Schraubst du die Metallplatte vom Joystick ab, hast du eine Schablone, mit der du einzeichnen kannst, wo du als Nächstes bohren musst.

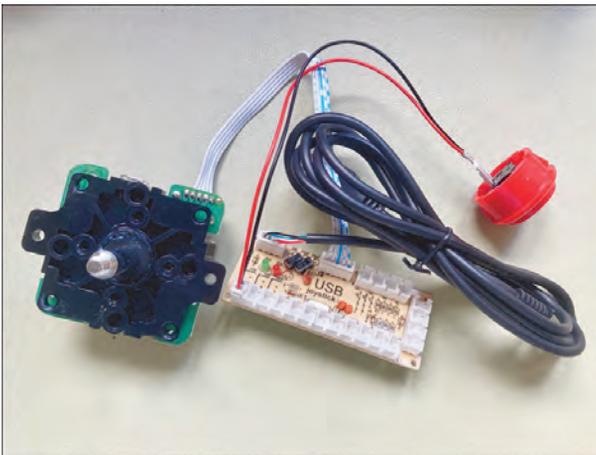


Abbildung 5.21 Die Verkabelung des Joysticks ist einfach. Schließe dazu einfach das mitgelieferte weiße Kabel an Joystick und Platine an. In die unteren Pins kannst du die Kabel für bis zu zwölf Knöpfe einstecken.



Abbildung 5.22 Wenn du alle Löcher gebohrt hast, kannst du den Joystick festschrauben.

Schraube nun den Joystick fest, setze die Knöpfe ein und verbinde beides mit dem USB-Encoder, und fertig ist dein selbst gebauter Controller!



Abbildung 5.23 So sieht der fertige Joystick aus.



Abbildung 5.24 Du kannst den Joystick natürlich auch in ganz andere Gehäuse einbauen, etwa diese Kiste aus Plexiglas ...



Abbildung 5.25 ... oder in eine alte Weinkiste aus Holz, wie dieses Kind, das seinen eigenen Joystick in einem Workshop gebaut hat.

Inhalt

| | |
|---|-----------|
| Über dieses Buch | 11 |
| 1 Das kreativste Werkzeug aller Zeiten | 15 |
| 1.1 So ist das Buch aufgebaut | 17 |
| 1.2 Vom Konsumenten zum Produzenten | 18 |
| 1.3 Game Over | 21 |
| 1.4 Computerspiele – mehr als nur Zeitvertreib | 22 |
| 1.5 Don't believe the hype! | 24 |
| 1.6 Programming-Challenge: Scratch | 26 |
| 1.7 Im Porträt: Mitchel Resnick | 31 |
| 1.8 Maker-Challenge: Marshmallow-Challenge | 35 |
| 2 Komplex, aber im Prinzip einfach: So funktioniert ein Computer | 37 |
| 2.1 Der Computer für die Jugend | 38 |
| 2.2 So funktioniert der Binärcode | 40 |
| 2.3 Binärcode – eine uralte Idee | 42 |
| 2.4 Im Porträt: Konrad Zuse | 45 |
| 2.5 Die wichtigste Erfindung der letzten 100 Jahre | 47 |
| 2.6 Wie Transistoren rechnen | 48 |
| 2.7 Von NAND zu Tetris | 53 |
| 2.8 Maker-Challenge: ein Logikgatter bauen | 54 |
| 2.9 Computer ohne Strom | 59 |
| 2.10 Coding-Challenge | 62 |
| 3 Alte Technik neu entdecken | 67 |
| 3.1 Im Porträt: Steve Wozniak | 68 |

- 3.2 Der erste Nerd-Stammtisch der Welt** 70
- 3.3 Der Beginn des PC-Zeitalters** 71
- 3.4 Erweiterungen** 75
- 3.5 Coding-Challenge: einen virtuellen Apple II erkunden** 76
 - 3.5.1 Bedientipps 77
 - 3.5.2 Den Apple II mit BASIC programmieren 79
 - 3.5.3 Die Programmiersprache LOGO auf dem Apple II 79
 - 3.5.4 Weitere Programme für den Apple II 81
 - 3.5.5 Tetris 2 (apple II/disc images/games/t/tetris 2.dsk) 81
 - 3.5.6 PICK A DILLY PAIR
(apple II/disc images/games/p/pick-a-dilly-pair.dsk) 82
 - 3.5.7 Still Alive Demo
(apple II/disc images/mockingboard/still_alive.dsk) 82
 - 3.5.8 Oregon Trail
(apple II/disc images/games/o/oregon trail_1.dsk) 82
- 3.6 Im Interview: Prof. Dr. Dr. Stefan Höltgen** 83
- 3.7 Maker-Challenge: Mach flott den Schrott** 87
 - 3.7.1 Monitor defekt, Computer in Ordnung 87
 - 3.7.2 Ein neues Betriebssystem für alte Laptops 90
- 3.8 Alte Tablets und Smartphones wiederbeleben** 92
 - 3.8.1 Ein altes Tablet als Spielkonsole 92
 - 3.8.2 Spaß mit der Kamera 93
 - 3.8.3 Bilderrahmen und Second Screen 93
- 3.9 Retro-Maker-Challenge: Diskettenbox basteln** 93

4 Games – besser als ihr Ruf 97

- 4.1 Vom Spiel zum Musikinstrument** 98
- 4.2 Toshio Iwai und Tod Machover – die Games-Musiker** 100
- 4.3 Die Geburt der Spieleindustrie** 103
- 4.4 Rollenspiele, aber analog statt digital** 105
- 4.5 Meilensteine der Ultima-Serie** 106
- 4.6 Gaming-Challenge: Ultima IV spielen** 109
- 4.7 Interview mit Richard Garriott de Cayeux** 113

| | | |
|----------|--|-----|
| 4.8 | Coding-Challenge: Zeichnen mit Vektoren | 117 |
| 4.9 | Maker-Challenge: ein Flexahedron bauen | 119 |
| 4.10 | Schule als Computerspiel | 125 |
| 4.11 | Was Schulen von Games lernen können | 126 |
| 5 | Just for fun: Gaming-Basteleien | 129 |
| 5.1 | Die Raspberry-Pi-Konsole | 129 |
| 5.2 | Der Ich-mach-alles-zum-Controller-Controller | 132 |
| 5.3 | Der One-Button-Ü-Ei-Controller | 134 |
| 5.4 | Programming-Challenge | 137 |
| 5.5 | Der IKEA-Joystick | 140 |
| 5.6 | Deine eigene (tragbare) Konsole | 145 |
| 5.7 | Bonus-Tipp: alte Konsolen lackieren und umbauen | 150 |
| 6 | Die dunkle Seite: Hackertricks und wie man sich davor schützt | 155 |
| 6.1 | Die Anfänge des Hackings – noch ohne Computer | 157 |
| 6.2 | Hacker als Superhelden | 159 |
| 6.3 | Die Tools der Hacker | 161 |
| 6.3.1 | Das Windows-Anmeldekennwort zurücksetzen | 161 |
| 6.3.2 | Auf dem Rechner gespeicherte Passwörter auslesen | 161 |
| 6.3.3 | Eine mit Passwort gesicherte ZIP-Datei öffnen | 162 |
| 6.3.4 | WLAN-Passwörter knacken | 162 |
| 6.3.5 | Einen Zugang zum Computer finden | 163 |
| 6.3.6 | Das Internet der Dinge durchsuchen | 163 |
| 6.4 | Fiese Hacking-Hardware für die USB-Schnittstelle | 163 |
| 6.4.1 | Wenn der Blitz im USB-Port einschlägt | 164 |
| 6.4.2 | Böses Quietscheentchen | 164 |
| 6.4.3 | Kabel-Klon | 165 |
| 6.4.4 | Eine bombige Überraschung | 165 |
| 6.4.5 | Fazit | 165 |

- 6.5 **Maker-Challenge: Setz dem TV-Terror ein Ende** 166
- 6.6 **Den Raspberry in den besten Hacking-Computer (oder den besten zu hackenden Computer) verwandeln** 169
- 6.7 **Erste Schritte in Kali Linux auf dem Raspberry Pi (oder VirtualBox)** 170
- 6.8 **Dein erster Hack** 172
- 6.9 **Die unsichtbaren Daten in der Luft** 174
- 6.10 **Social Engineering – Menschen hacken** 176
- 6.11 **Interview mit dem IT-Security-Experten Friedhelm Düsterhöft** 177
- 6.12 **Coding-Challenge: Minecraft auf dem Raspberry Pi hacken** 180

7 The end of the world as we know it! Wie gefährlich ist künstliche Intelligenz? 185

- 7.1 **Der Traum vom künstlichen Menschen** 187
- 7.2 **Coding-Challenge: einen einfachen Chatbot programmieren** 188
- 7.3 **Die Anfänge der KI-Forschung** 189
- 7.4 **Was ist Intelligenz?** 191
- 7.5 **Künstliche Intelligenz – anders, als viele denken** 193
- 7.6 **KI verstehen** 194
- 7.7 **Überwachtes Lernen** 196
- 7.8 **Unüberwachtes Lernen** 197
- 7.9 **Verstärkendes Lernen** 199
- 7.10 **Maker-Challenge: ein selbstfahrendes Auto bauen** 200
- 7.11 **Interview mit der KI-Expertin Viktoriya Olari** 202
- 7.12 **KI-Spielzeug** 206
 - 7.12.1 Tamagotchi 206
 - 7.12.2 Furby 207
 - 7.12.3 Cozmo 209
 - 7.12.4 XBOX 360 Kinect 210
- 7.13 **KI-Ethik** 211

8 Computerpropheten: Evangelisten und Apokalyptiker 215

| | | |
|------|--|-----|
| 8.1 | Das Ende ist nah | 215 |
| 8.2 | Manfred Spitzer | 217 |
| 8.3 | Das ist nachgewiesen. Dazu gibt es gute Untersuchungen | 218 |
| 8.4 | Ja, aber | 219 |
| 8.5 | Auf anderem Gebiet wesentlich kompetenter | 220 |
| 8.6 | Joseph Weizenbaum | 221 |
| 8.7 | Maker-Challenge: einen besseren Chatbot programmieren | 223 |
| 8.8 | Vom Priester der Informatik zum Ketzer | 226 |
| 8.9 | Ray Kurzweil | 228 |
| 8.10 | Erfinder der Zukunft | 229 |
| 8.11 | Ewiges Leben | 230 |
| 8.12 | Eine neue Religion | 231 |
| 8.13 | Was ist der Mensch? | 232 |
| 8.14 | Und jetzt? | 233 |
| 8.15 | Maker-Challenge | 234 |
| 8.16 | Idee Nr. 1: einen LED-Throwie bauen | 235 |
| 8.17 | Idee Nr. 2: ein LEGO-Set beleuchten | 237 |

9 Die fieseren Tricks der Tech-Industrie 239

| | | |
|-------|---|-----|
| 9.1 | Die Zerstörung einer ganzen Generation? | 240 |
| 9.2 | Das Handbuch der fieseren Tricks | 243 |
| 9.2.1 | Der Kontext zählt | 245 |
| 9.2.2 | Du bist fast am Ziel | 246 |
| 9.2.3 | Oh nein, gleich ist alles weg! | 246 |
| 9.3 | Maker-Challenge | 247 |
| 9.4 | Das Gegengift | 248 |
| 9.5 | FIFA und die Lootbox | 249 |

- 9.6 Illegales Glücksspiel? 251
- 9.7 Das unmoralische Diablo 254
- 9.8 Die ersten zehn Minuten 256
- 9.9 Willkommen an Board! 257
- 9.10 Keine Lootboxen? 259
- 9.11 Programming-Challenge: ein eigenes Spiel programmieren 260

10 Die Zukunft des Computers 265

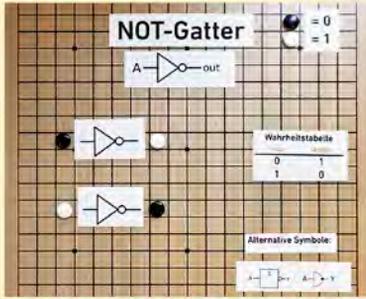
- 10.1 Die Rückkehr des Analoges 267
- 10.2 Ungewöhnliche Rechner 269
- 10.3 Ein Quantensprung in der Computertechnik 270
- 10.4 Quantencomputer 273
- 10.5 Wie funktioniert ein Quantencomputer? 274
- 10.6 Die Welt als Simulation 276
- 10.7 Virtual Reality 277
- 10.8 Deine eigene Coding-Challenge 280
- 10.9 Dein eigenes Making-Projekt 283
 - 10.9.1 Einen Ort zum Arbeiten suchen 283
 - 10.9.2 Experte werden 284
 - 10.9.3 Gleichgesinnte suchen 284

Anhang

- A Der kleine Raspberry-Pi-Guide 287
- B Einstieg in Python 307
- Index 311

Nützliches Insiderwissen und jede Menge Bastelspaß

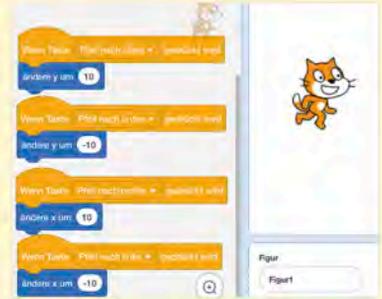
Dieses Buch bietet Dir ganz viel Wissen, das du so nirgendwo findest: spannende Geschichten über den Computer sowie brandaktuelle Infos und Tipps zu Hacking, Digitalisierung und künstlicher Intelligenz. Nimm die Herausforderungen an, und los geht's mit Coden und Basteln.



So tickt dein Computer



Baue ein Auto und vieles mehr



Spiele und programmiere

Vom Apple II zum Quantencomputer

Entdecke die Welt des Computers! Du erfährst alles über die Meilensteine der Computer- und Games-Entwicklung. Wie funktioniert ein Computer? Was ist ein Comptometer? Und wie sieht der Computer der Zukunft aus? Die Antworten gibt es hier!

Wichtige IT-Themen unterhaltsam erklärt

Lerne, wie du dich vor Hackern schützt, den Tricks der Games-Industrie entkommst und dich unabhängiger von Social Media machst. Diskutiere mit, wenn es um KI und Digitalisierung geht – im Buch findest du die Ansichten von Experten dazu.

Coden und basteln – dein Abenteuer beginnt

Das Buch steckt voller Coding- und Maker-Challenges: Programmiere ein Spiel, baue ein selbstfahrendes Auto, hacke Minecraft oder bastle eine Spielekonsole mit dem Raspberry Pi.



Tobias Hübner ist Lehrer, Autor, Dozent und IT-Trainer. Er setzt sich für digitale Bildung ein und wurde für seine kreativen Ideen mehrfach ausgezeichnet. Viele seiner Bastelprojekte erstellt er mit dem Raspberry Pi.

Aus dem Inhalt

Nützliches Wissen

- So funktioniert ein Computer
- Alte Technik neu entdecken
- Meilensteine der Games-Entwicklung
- Von Hackern und ihren Tricks
- Wie gefährlich ist KI?
- Digitalisierung: Pro & Contra
- Die Tricks der Tech-Industrie

Spannende Projekte

- Maker- und Coding-Challenges
- Basteln, coden und hacken mit dem Raspberry Pi
- Games spielen und programmieren

Specials

- Interviews mit IT-Experten
- Kompakter Raspberry-Pi-Guide
- Kurze Einführung in Python

