

Hier geht es los!

Willkommen!

Danke, dass du dich für dieses Buch entschieden hast. Ich hoffe, es bietet dir abwechslungsreiche Lektüre und Spaß. Wenn du vorher noch nie programmiert hast – keine Sorge! Wir lassen es langsam angehen und beginnen ganz von vorn. Keine Expertenkenntnisse notwendig!

Alle lieben Minecraft. Ein wichtiger Grund dafür ist meiner Meinung nach, dass man an der Erstellung des Spiels teilnimmt. Du kannst Dinge erschaffen und bauen. Ob es ein kleiner Unterstand im Überlebensmodus ist oder eine umfangreiche Ziegel-Simulation deines eigenen Schlosses – du bekommst, was du erschaffst.

Manchmal reichen die in Minecraft vorhandenen Möglichkeiten aber nicht aus. Vielleicht willst du mehr tun, z. B. mit brennenden Kühen schießen oder einen Gegner in einer Kapsel aus massivem Fels einsperren. Für diese und weitere Ideen musst du das Spiel um Funktionen erweitern.

Anwendungen für deinen Computer oder dein Telefon werden in einer Textform geschrieben, die man als *Programmiersprache* bezeichnet. Programmiersprachen sind nicht so umfangreich oder schwer zu erlernen wie menschliche Sprachen, also Spanisch oder Chinesisch. Sie unterscheiden sich aber von der Sprache, die du im Alltag verwendest.

Heute werden viele verschiedene Programmiersprachen verwendet. Manche sind sehr beliebt, aber nicht sehr leistungsfähig. Andere werden nur von einer Handvoll Menschen eingesetzt und sind schwer zu meistern, leisten aber Außergewöhnliches.

Minecraft ist in der Programmiersprache Java geschrieben. Java ist verhältnismäßig leistungsfähig, besitzt jedoch auch komplexe und schwer verständliche Abschnitte. Wir konzentrieren uns auf die Grundlagen und vermeiden die schwierigen Abschnitte.

Mit diesem Buch lernst du Java schnell und einfach. Deine Kenntnisse werden ausreichen, um deine eigenen Minecraft-Plugins zu schreiben und allgemeine Aufgaben in Java zu erledigen.

Wir sehen uns an, wie du deinen eigenen Minecraft-Server und Sharing aufsetzt und cloubasierte Backups deines Codes erstellst. Dann werfen wir einen Blick auf einige fortgeschrittene Codierungstechniken.

Für wen ist dieses Buch gedacht?

Dieses Buch wendet sich an Leser ohne Programmiererfahrung, die jedoch schon ein wenig Minecraft gespielt haben. Wenn du nicht mit Minecraft vertraut bist, findest du im Internet zahlreiche Videos und Bücher für den Einstieg. Ich vermute allerdings, dass du dich recht gut mit Minecraft auskennst und so begeistert bist, dass du jetzt die Programmierung lernen möchtest.

Leser unter 8 oder 9 Jahren oder solche, die Schwierigkeiten haben, das Programmieren in Java zu verstehen, möchten vielleicht mit einer einfacheren Sprache beginnen. Scratch und Snap!¹ sind hervorragende kleine Sprachen, mit denen du die Grundlagen der Programmierung erlernen kannst. Sie zeigen dir auf optische Weise, wie die Programmierelemente zusammengehören. Wenn du den Dreh dann raus hast, kannst du wieder zur Java-Programmierung zurückkehren.

Auf jeden Fall brauchst du einen modernen Computer, der unter Windows, OS X oder Linux läuft – dann geht es los.

Erste Schritte

Minecraft ist als Client/Server-Anwendung entwickelt worden. Das heißt, es besteht aus zwei Teilen.

Zuerst ist da der Client, also die Anwendung, die auf deinem Desktop- oder Laptop-Computer läuft. Der Client berechnet (rendert) die Bilder aus der Minecraft-Welt und nimmt deine Befehle für Bewegungen und Handlungen im Spiel entgegen.

Dann gibt es den Server, der das Geschehen im Spiel überwacht, z. B. die verbundenen Spieler, ihre Inventare, was sie gebaut haben, wo sie sich befinden usw. Meist läuft der Server auf einem weit entfernten Rechner irgendwo in deinem Land. Er kann aber auch auf deinem Desktop oder Laptop laufen.

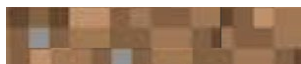
¹ <http://scratch.mit.edu>, <http://snap.berkeley.edu>

Client und Server kommunizieren über das Netzwerk miteinander, so wie du mit deinem Browser Server besuchst und Spiele spielst oder dir Katzenbilder ansiehst.

Um die Funktionen von Minecraft zu verändern, musst du das Java-Programm auf dem Server ändern oder erweitern. Wie das geht, lernst du in diesem kleinen Buch. Du erfährst hier, wie du Java-Programme mithilfe von Java-Anweisungen schreibst (was wir *Quellcode* oder einfach *Code* oder *Programm* nennen), um Plugins für den Minecraft-Server zu entwickeln. Ein Plugin ist einfach ein kleiner Code, den du einem bestehenden Programm hinzufügst.²

Bevor wir mit Plugins beginnen, musst du zum Testen einen lokalen Minecraft-Server einrichten und die Sprache Java und einige andere Anwendungen installieren. Das machen wir in den ersten beiden Kapiteln. Das Installieren ist nicht besonders aufregend. Es kann sogar sehr langweilig sein. Ich versuche daher, diese langweilige Phase so kurz wie möglich zu halten.

Damit du einen besseren Überblick über deine Fortschritte bekommst, zeigt ein Infokasten am Ende jedes Kapitels an, wie weit du vorangekommen bist. Du beginnst sozusagen mit einem leeren Acker ...



... und hast am Ende eine grüne Wiese:



Einige Kapitel lassen sich schneller als andere durcharbeiten, und einige bringen dich auch weiter als andere. Letztendlich machst du aber immer Fortschritte.

Ein Sprung ins kalte Wasser

Da ich auf so wenigen Seiten so viele Sachen unterbringen muss, zeige ich dir zuerst einige Dinge, die du manchmal einfach umsetzen musst, und erläutere die Details ein wenig später. Das kann sich wie ein Sprung ins kalte Wasser anfühlen. Wenn du etwas als unlogisch empfindest, mach dir keine allzu großen Sorgen. Spring einfach hinein. Die Erklärung folgt danach.

² Manche Anwender schreiben Minecraft-Mods, die im Grafik-Client laufen, aber wir behandeln die hier nicht. Mods stehen meist mit dem Laden neuer Skins in Verbindung. Unsere Entwicklung von Plugins ist näher an dem, was Profi-Programmierer tun, und ändert das Spiel an sich und nicht nur das Aussehen.

In vielen Fällen kannst du etwas erfolgreich einsetzen, ohne genau zu wissen, wie die Sache funktioniert. Ich kann die Schreibtischlampe einschalten und nutzen, ohne zu verstehen, wie Strom erzeugt wird. Ich kann sogar eine eigene Schreibtischlampe bauen, ohne zu wissen, wie ein Kraftwerk funktioniert. Ich muss nur wissen, wie die Teile zusammengesetzt werden.

Wir konzentrieren uns hier so weit wie möglich auf dieses Zusammenbauen der Teile.

Wo bekommst du Hilfe?

Für Fragen, Updates und Tipps zu diesem Buch gibt es eine englischsprachige Website (das ist die Website zum US-amerikanischen Original, von dem dieses Buch eine Übersetzung ist). Gehe einfach zu <http://pragprog.com/book/ahmine2> und klicke auf *Discuss*.

Auf eben dieser Seite findest du einen Link zum Quellcode. Von dort kannst du alle Listings dieses Buchs herunterladen.³

Bitte lade den Quellcode jetzt auf deinen Desktop herunter. Den Desktop wirst du für die meisten Arbeiten verwenden, dazu aber später mehr. Beginne einfach mit dem Download.

Ich warte auf dich.

Der Download besteht aus einem Archiv, das mit *zip* erstellt wurde. Du musst es auf deinem Desktop auspacken. Dazu kannst du an der Befehlszeile `unzip` eingeben (bei OS X oder Linux). Unter Windows verwendest du *WinZip* oder das kostenlose *7-Zip*.⁴

Alles installiert und ausgepackt? Prima!

Wenn wir weitermachen, lernst du neue Tools und ihre Verwendung kennen. Den Überblick über das Gelernte erhältst du durch den Infokasten am Ende jedes Kapitels. Am Ende des Buchs kannst du dann eigenen Code erstellen und deine eigenen Plugins von Grund auf schreiben.

³ Der genaue Link ist: https://pragprog.com/titles/ahmine2/source_code/

⁴ Verfügbar unter <http://www.winzip.com> oder <http://7-zip.org>

Schreibweisen

Code oder Befehle werden in diesem Buch wie folgt dargestellt:

\$ I've typed all of this as an example for you.

Code-Elemente oder Befehle, die du direkt eintippen musst, formatiere ich fett:

\$ you type this part here (but not the dollar-sign prompt)

Alles, was folgt und nicht mehr fett formatiert ist, gehört dann zur Ausgabe, die der Befehl auslöst.

Text in *kursiv* dient als Platzhalter. Du gibst ihn nicht direkt ein. Etwas wie

```
me.sendMessage( string msg );
```

bedeutet, dass du den kursiven Teil ersetzt, wie hier:

```
me.sendMessage("Creepers are coming.");
```

Sehen wir uns an, wie das funktioniert.