

Kapitel 1

Minecraft-Server

Alleine Minecraft zu spielen, kann schon jede Menge Spaß machen, noch lustiger wird es aber, wenn du dich mit anderen Spielern zusammentust, um mit ihnen oder auch gegen sie zu spielen. Dazu kannst du dir entweder einen der hunderten öffentlichen Server aussuchen, die du überall im Internet findest, oder du kannst deinen eigenen Server nutzen – dann hast du die volle Kontrolle über alle Einstellungen. Noch mehr Spaß wird dir dein eigener Server machen, wenn du im Laufe des Buches lernst, immer ausgefeiltere Plugins für ihn zu programmieren, mit denen du Minecraft nach deinen Vorstellungen erweitern kannst.

Um deinen eigenen Server zu betreiben, benötigst du neben dem normalen Minecraft-Spiel, das auch **Client** genannt wird, noch ein weiteres Programm, nämlich den **Minecraft-Server**. Den »normalen« Minecraft-Server, manchmal auch »Vanilla-Server« genannt, kannst du auf der offiziellen Minecraft Webseite www.minecraft.net herunterladen. Neben dieser Version gibt es aber auch noch zahlreiche sogenannte Mods, also Modifikationen des Original-Servers. Als Mods oder Modifikationen bezeichnet man im Zusammenhang mit Spielen Versionen eines Spiels, die in irgendeiner Form verändert, also modifiziert wurden. Diese meist von Fans entwickelten Mods bieten häufig viele zusätzliche Funktionen und Annehmlichkeiten, über die der Vanilla-Server nicht verfügt, wie zum Beispiel auch die Möglichkeit eigene Plugins zu programmieren.

Merke

Das normale Minecraft-Spiel, das du auch startest, wenn du alleine spielst, wird **Client** genannt. Das Programm, das wir in diesem Kapitel installieren werden, das du benötigst, um mit Freunden zusammen spielen zu können, heißt hingegen **Server**.

Dieses Buch ist für gleich zwei der beliebtesten Server ausgelegt. Du kannst dich entscheiden zwischen dem **CraftBukkit**-Server, häufig auch einfach nur Bukkit genannt, und dem **Spigot**-Server. Da der Spigot- auf dem Bukkit-Server aufbaut, funktionieren alle Plugins, die wir im Rahmen dieses Buches programmieren werden, auf beiden Servern. Der einzige Unterschied liegt in der Administration der Server, hier bietet Spigot mehr Möglichkeiten, ist dafür in der Bedienung aber auch etwas komplexer. Außerdem ist der Spigot-Server etwas effizienter, was bedeutet, dass er insbesondere

etwas weniger Arbeitsspeicher (RAM) benötigt. Für Anfänger, die zum ersten Mal einen eigenen Server betreiben, ist es daher ratsam zunächst auf Bukkit zu setzen; wer schon Erfahrung mit der Verwaltung eines Minecraft-Servers hat, kann sich auch an Spigot herantrauen. Ein Wechsel ist ohnehin jederzeit möglich.

1.1 Java installieren

Egal für welchen der beiden Server du dich entscheidest: Um sie später starten zu können, muss auf deinem Computer eine aktuelle Java-Version installiert sein, denn die Server sind, wie auch Minecraft selbst, in Java programmiert. Um das zu testen, kannst du unter Windows die Eingabeaufforderung öffnen, indem du den Namen einfach in das Suchfeld im Startmenü eingibst, beziehungsweise unter GNU/Linux und macOS ein Terminal öffnest. Dort gibst du dann den Befehl `java -version` ein und bestätigst deine Eingabe mit der `Enter`-Taste. Sieh die darauf folgende Ausgabe aus wie in [Abbildung 1.1](#) gezeigt, so ist Java bereits korrekt auf deinem Computer installiert und du kannst zum nächsten Abschnitt springen.

```
C:\>java -version
openjdk version "17.0.1" 2021-10-19
OpenJDK Runtime Environment (build 17.0.1+12-39)
OpenJDK 64-Bit Server VM (build 17.0.1+12-39, mixed mode, sharing)
```

Abb. 1.1: Ausgabe nach `java -version`

Bekommst du stattdessen eine Meldung angezeigt wie »Der Befehl "java" ist entweder falsch geschrieben oder konnte nicht gefunden werden.«, so ist Java noch nicht auf deinem Computer installiert. In diesem Fall kannst du unter buch.daniel-braun.com einen Link zum Download der aktuellsten Java-Version finden.

Unter GNU/Linux kannst du Java direkt über den Paketmanager deiner Wahl installieren. Unter macOS und Windows lädst du zunächst ein gepacktes Verzeichnis herunter, das nach dem Download entpackt werden muss. Dieses Verzeichnis, das je nach Version zum Beispiel den Namen `jdk-17.0.1` trägt, kopierst du unter macOS in den Ordner `/library/Java/JavaVirtualMachines/` und unter Windows in ein beliebiges Verzeichnis deiner Wahl, zum Beispiel direkt in `c:\`. Unter Windows musst du dieses Verzeichnis dann noch zur sogenannten *PATH-Variable* hinzufügen. Dazu öffnest du zunächst die erweiterten Systemeinstellungen deines Computers.

Windows XP, Vista und 7: Unter Windows XP, Vista und 7 öffnest du dafür zunächst das Startmenü und dann die **SYSTEMSTEUERUNG**. Dort wählst du aus der Kategorie **SYSTEM**

UND SICHERHEIT den Eintrag **SYSTEM** aus und im sich danach öffnenden Fenster den Eintrag **ERWEITERTE SYSTEMEINSTELLUNGEN**.

Windows 8, 10 und 11: Unter Windows 8, 10 und 11 kannst du die erweiterten Systemeinstellungen öffnen, indem du den Begriff einfach direkt in die Suche eingibst. Alternativ kannst du auch zunächst mit der rechten Maustaste auf das Windows-Logo in der unteren linken Ecke klicken und dort dann auf **SYSTEM** und in dem sich öffnenden Fenster wieder auf **ERWEITERTE SYSTEMEINSTELLUNGEN**.

Nun solltest du, unabhängig von deiner verwendeten Windows-Version, das in [Abbildung 1.2](#) gezeigte Fenster sehen.

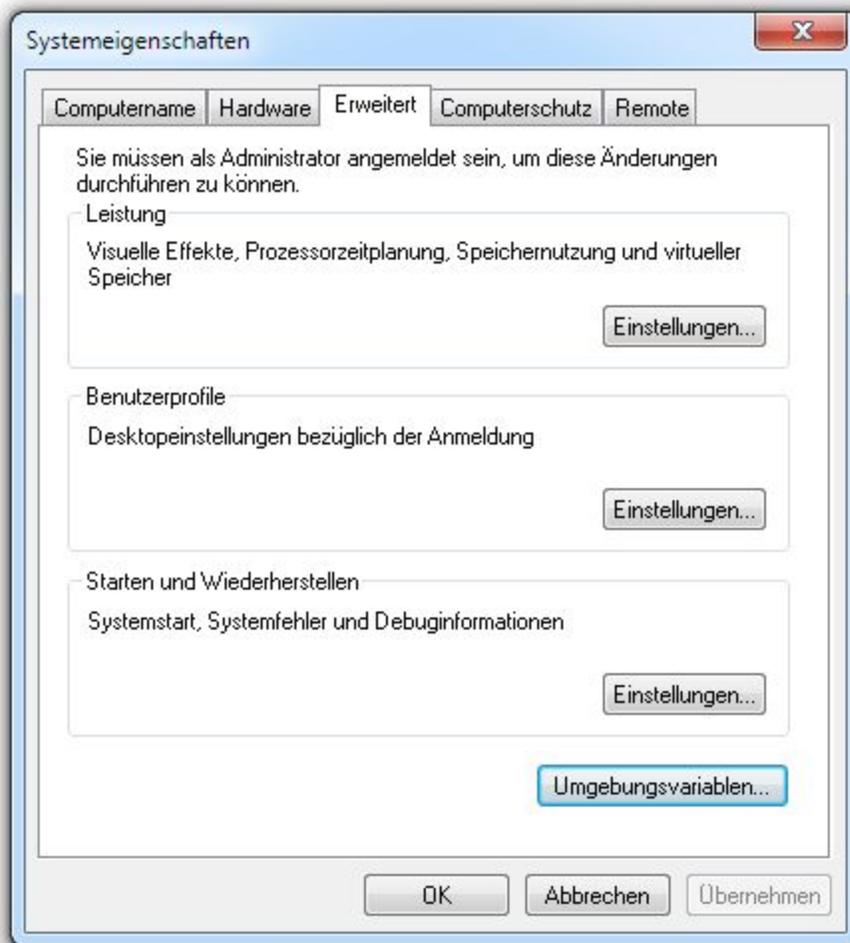


Abb. 1.2: Erweiterte Systemeinstellungen

Dort findest du in der rechten unteren Ecke einen Button mit der Beschriftung **UMGEBUNGSVARIABLEN**. Bei einem Klick darauf öffnet sich das in [Abbildung 1.3](#) gezeigte

Fenster.

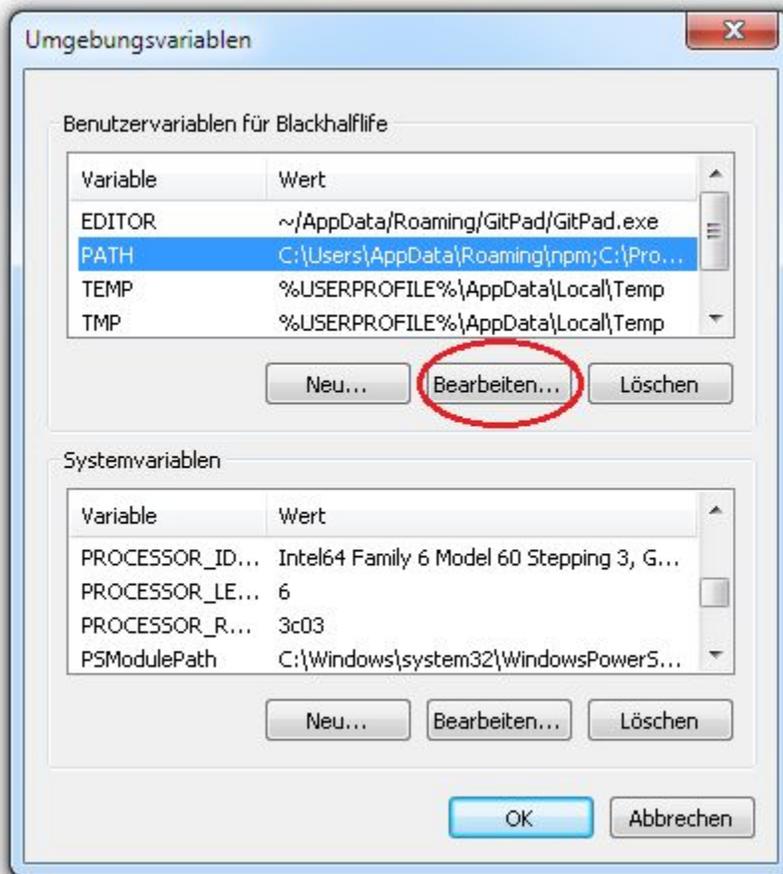


Abb. 1.3: Umgebungsvariablen

Dann wählst du, wie in [Abbildung 1.3](#) gezeigt, den Eintrag **PATH** aus und klickst anschließend auf **BEARBEITEN**. Sollte der Eintrag nicht vorhanden sein, so kannst du direkt zum nächsten Absatz springen. Danach öffnet sich ein langes Textfeld, in dem es schon zahlreiche Einträge gibt, die auf keinen Fall geändert werden dürfen. Stattdessen solltest du am Ende, abgetrennt durch ein Semikolon, den Pfad angeben, an den du zuvor das entpackte Verzeichnis kopiert hast, gefolgt von `\bin\`. Also zum Beispiel:

```
C:\jdk-17.0.1\bin\;
```

Je nachdem, welche Java-Version du installiert hast, kann der Pfad aber, insbesondere bei der Versionsnummer, leicht abweichen. Daher solltest du unbedingt darauf achten, den tatsächlichen Installationspfad zu nutzen. Danach musst du die Änderungen nur noch mit **OK** und **ÜBERNEHMEN** bestätigen.

Sollte es bei dir noch keinen Eintrag mit dem Namen **PATH** geben, so kannst du diesen ganz einfach selbst anlegen. Dazu klickst du statt auf **BEARBEITEN** einfach auf **NEU**. Im Fenster, das sich daraufhin öffnet, gibst du als **NAME DER VARIABLEN** das Wort **PATH** ein und als **WERT DER VARIABLEN** den Pfad zur Installation, beendet durch ein Semikolon, und bestätigst deine Eingabe mit **OK**. Anschließend sollte der Befehl `java -version` dann in der Eingabeaufforderung funktionieren.

Hinweis

Sollte beim späteren starten des Servers die in [Abbildung 1.4](#) gezeigte Fehlermeldung erscheinen, so ist die auf deinem Computer installierte Java-Version veraltet. Bitte führe die oben beschriebenen Schritte dann mit der neusten Java-Version durch.

```
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.UnsupportedClassVersionError: org/bukkit/craftbukkit/
bootstrap/Main has been compiled by a more recent version of the Java Runtime (class file
version 60.0), this version of the Java Runtime only recognizes class file versions up to
52.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(Unknown Source)
    at java.security.SecureClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.access$100(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.launcher.LauncherHelper.checkAndLoadMain(Unknown Source)
```

Abb. 1.4: Fehlermeldung nach dem Starten des Servers bei veralteter Java-Version

1.2 Installation

An dieser Stelle musst du dich nun entscheiden, welchen Server du zum Testen deiner Plugins verwenden möchtest. Wenn du dich für den Bukkit-Server entscheidest, kannst du in [Abschnitt 1.2.1](#) weiterlesen, möchtest du lieber den Spigot-Server verwenden, dann kannst du direkt zu [Abschnitt 1.2.2](#) springen.

1.2.1 CraftBukkit