Minecraft mit Python beeinflussen

Das Spiel Minecraft gibt es in verschiedenen Versionen. Auf dem Raspberry Pi läuft eine ältere Version die für den Raspberry Pi angepasst wurde. Es gibt in ihr nur den Modus, in dem man die Welt auch mit mehreren Spielern verändern kann, aber z. B. nicht gegeneinander kämpfen kann. Dafür gibt es die Möglichkeit mit Hilfe von Python direkt Einfluss auf ein laufendes Spiel zu nehmen.

1 Grundlegendes

Um von Python in ein Minecraft-Spiel eingreifen zu können, muss zuerst in Python die passenden Pakete importiert werden. Mit dessen Hilfe lässt sich eine Verbindung zum aktuellen Minecraft-Spiel erstellen, das auf dem Raspberry Pi läuft, auf dem auch Python ausgeführt wird:

```
from mcpi.minecraft import Minecraft
import mcpi.block as block
mc = Minecraft.create()
```

Mit dem dann erzeugten Minecraft-Objekt, das über `mc` zu erreichen ist, lassen sich weitere Dinge ansteuern, die dann auch über die Python-Konsole direkt nutzbar sind.

2 Steine setzen

Am einfachsten lässt sich die Minecraft-Welt dadurch beeinflussen, dass man an bestimmte Positionen Steine setzt. Es können sowohl einzelne Steine, als auch mehrere Steine als Block gesetzt werden. Einen einzelnen Stein setzt man mit Hilfe von `setBlock(args)` der mehrere Argumente benötigt. Die ersten drei Argumente sind für die x-, y- und z-Koordinaten. Das darauf folgende Argument ist der Blocktyp, mit dem man angibt, welche Steinsorte gesetzt werden soll. Bei einigen Steinen lässt sich noch optional ein weiteres Argument übergeben, was z. B. beim Treppenstein die Ausrichtung ist.

Um einen ganzen Block zu setzten wird die Methode `setBlock(args)` benutzt. Sie erwartet bei den ersten sechs Argumenten die Koordinaten von den zwei Ecken, die den Block begrenzen. Das folgende Argument gibt wieder den Typ an und als letztes Argument kann wieder optional die weitere Einstellungen, die vom gewählten Steintyp abhängig sind, gesetzt werden.

Im folgenden Beispiel wird dadurch ein Dreckstein an der Position (0,20,0) gesetzt und ein Block aus 9 Holzsteinen im Bereich von (-1,-1,-1) bis (1,1,1) gesetzt.

```
mc.setBlock(0, 20, 0, block.DIRT.id)
mc.setBlocks(-1, -1, -1, 1, 1, 1, block.WOOD.id)
```

Eine Auswahl an möglichen Steintypen sind: AIR, STONE, DIRT, WATER, ICE und CACTUS.

3 Aktuelle Postion – Wo bin ich?

Um im seinen eigenen Sichtbereich die Blöcke zu setzten, sollte man aber auch die Postion des Spielers wissen. Diese wird im Spiel immer in der oberen linken Ecke angezeigt. Sie lässt sich aber auch mit Hilfe von `x,y,z=mc.player.getPos()` auslesen. Dabei werden in x, y, und z die jeweilige Koordinate gespeichert, die dann weiter genutzt werden kann.

Aufgabe 1

Platziere in der Nähe der Position der Spielfigur verschiedenen einzelne Steine und Blöcke. Setze damit ein Muster zusammen, wie z. B. eine Pyramide.

Aufgabe 2

Grabe im Spiel einen Tunnel in den Boden oder in einen Berg. Lasse dann durch Python eine große Halle um deine Figur herum entstehen.

4 Analyse der Umgebung

Der ideale Baumeister geht auf seine Umgebung ein. So sollte z. B. ein Haus nicht in der Luft schweben, sondern auf dem Boden aufsetzten, auch wenn dieses in Minecraft möglich ist. Die Untersuchung der Umgebung ist mit `blockType = mc.getBlock(x,y,z)` möglich. Diese Methode liefert den Typ des Blocks zurück. So wird im folgenden Beispiel mögliches Wasser an der übergebenen Position durch Eis ersetzt.

```
if mc.getBlock(3, 5, 9) == block.WATER.id :
    mc.setBlock(3, 5, 9, block.ICE.id)
```

Aufgabe 3

Begebe dich in die Nähe eines Sandstrands und ersetze in einem kleinen Bereich (10 x 10 x 10 Steine) alle Sandsteine durch Granit.

Aufgabe 4

Erstelle an einem Berg oder Hügel eine 10 x 10 Platzform. Sorge dabei dafür, dass sie an allen Punkten auf oder über dem aktuellen Bodenlevel liegt. Da sie nicht schweben soll, muss ggf. automatisch bis zum Boden aufgefüllt werden.

5 Mehrspielermodus

Wenn man mit mehreren Spielern an einem Spiel teilnimmt ist es wichtig, dass man sich auch mit dem Spiel der Person direkt verbindet, auf dessen Raspberry das Spiel gestartet wurde. Ansonsten bekommen alle anderen von den eigenen Aktionen nichts mit. Dazu muss man beim Erzeugen des Minecraft-Objekts die entsprechende IP-Adresse des Servers angeben. Dazu wird im Kopf des Programms die entsprechende Zeile durch `mc = Minecraft.create(<Adresse des Servers>)` ersetzt. Hier lässt sich einfach die IP-Adresse eingeben, die man in Minecraft im Mehrspielermodus bei der Auswahl des Spieles angezeigt bekommt.

Jeder Spieler wird im Spiel mit einer eigenen ID gelistet. Eine Liste aller Spieler-IDs kann man mit `entityIds = mc.getPlayerEntityIds()` erhalten. Diese kann man auch nutzen, um dann die Position eines Spieler unter Angabe seiner ID zu erhalten. Dieses geht mit `x,y,z = mc.entity.getPos(entityId)`. Das folgende Beispiel gibt die IDs aller Spieler und deren Position aus:

```
playerIds = mc.getPlayerEntityIds()
print("Spieler-IDs:", playerIds)
for id in playerIds
    x, y, z = mc.entity.getPos(id)
    print("Spieler", id, "bei␣x:", x, "y:", y, "z:", z)
```

6 Weitere Informationen

Eine Referenz zu vielen weiteren möglichen Befehlen ist auf der Seite `http://www.stuffaboutcode.com/p/minecraft-api-reference.html` zu finden.