Version 1.0, 2018
Additional note

Schüler Praktikum

Matthias Romer[1]

Zusammenfassung
Dieses Dokument soll Max helfen die Arbeitsweise eines Embedded System Engineer kennen zu lernen. Hierzu soll Max mittels einem Embedded System (Raspberry Pi) und der Programmiersprache Python selbständig aufgaben bearbeiten.

Keywords
Raspberry Pi — Python — Minecraft

[1] *Embedded Systems, Gigatronik, Ulm, Deutschland*
*__E-Mail__: Matthias.Romer@gigatronik.com

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Python 1
1.2 Raspberry Pi und Minecraft 1
2 How to start **1**
3 Programmierung mit Python **2**
3.1 Den Spieler teleportieren 2
3.2 Weiter wichtige Funktionen 3
Literatur **4**

1. Einleitung

Hier zuerst mal ein paar Basics - Was ist Python? Was ist ein Raspberry Pi und was hat das alles mit einem Spiel wie Minecraft zu tun?

1.1 Python

Wikipedia beschreibt Python wie folgt:

> Python ist eine universelle, üblicherweise interpretierte höhere Programmiersprache. Sie hat den Anspruch, einen gut lesbaren, knappen Programmierstil zu fördern. So werden beispielsweise Blöcke nicht durch geschweifte Klammern, sondern durch Einrückungen strukturiert. Wegen ihrer klaren und übersichtlichen Syntax gilt Python als einfach zu erlernen. [1]

Kurz und einfach gesagt - Python ist eine von vielen Sprachen die der Computer versteht. Dadurch ist es für dich möglich ihm Befehle zu erteilen, welche er ohne Wiederworte ausführt (solange du dich korrekt ausdrückst).

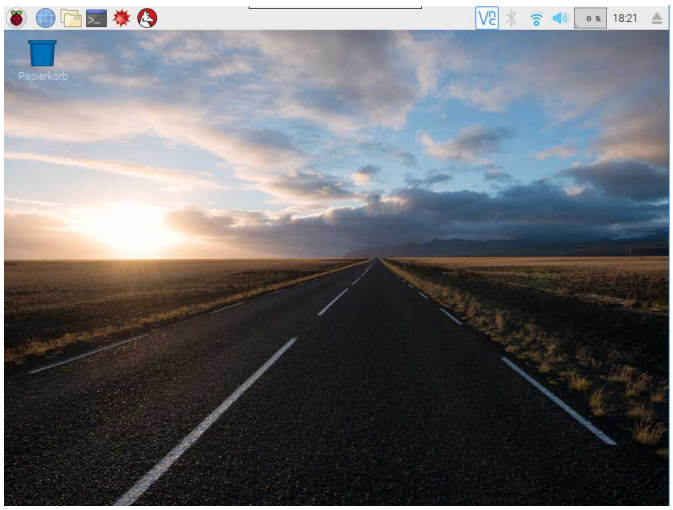
1.2 Raspberry Pi und Minecraft

Die Minecraft: Pi Edition ist eine Version von Minecraft entwickelt für den Raspberry Pi, ein sehr preisgünstiger Computer, der nur aus einer kreditkartengroßen Platine besteht. Er wird von einer Stiftung vertrieben, die das Ziel hat, jungen Menschen den Erwerb von Programmier- und Hardwarekenntnissen zu erleichtern. [2]

Es gibt schon mehrere Modelle des Raspberry Pi und für dieses Projekt kommt der Raspberry Pi 2 Modell B zum Einsatz.

2. How to start

Als erstes muss man das Raspberry Pi an einen Bildschirm und an die Stromversorgung anschließen. Hat man alles richtig angeschlossen sollte dieses Bild erscheinen:



Als nächstes wird Mincraft Pi gestartet hierzu klickt man wie im nächsten Bild dargestellt auf Mincraft Pi.

Sobald sich Mincraft Pi geöffnet hat geht man auf SStart Game”→ ”Create new”. Nun sollte sich das Spiel vollständig gestartet haben - Glückwunsch nun kann Programmiert werden.

3. Programmierung mit Python

Damit du nicht gleich so endest,

fangen wir mal recht einfach an.

3.1 Den Spieler teleportieren

Bei dieser Mission untersuchst du die Wirkungsweise von Variablen. Hierzu teleportierst du deinen Spieler mithilfe von Integern an eine andere Position. Ähm HALT STOPP was sind variablen und was ist ein Integer?

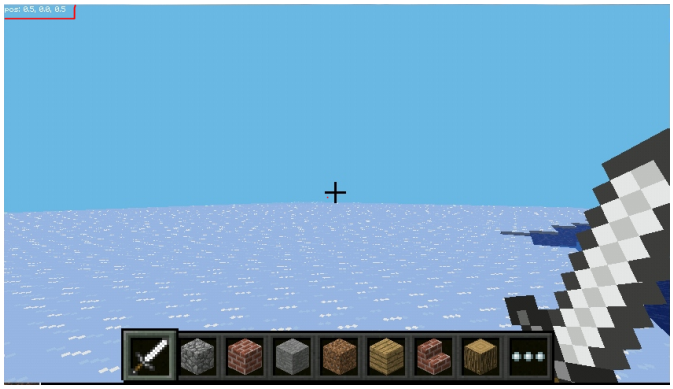
Variablen gestatten dir das Speichern von Daten, die du an späterer Stelle im Programm noch einmal brauchst. Daten können dabei alles Mögliche sein, was gespeichert werden soll: Zahlenwerte, Namen, beliebige Texte, Objektlisten usw. Hier beispielsweise ist eine Variable namens juckel, die den Zahlenwert 3 speichert:

Listing 1. Insert code directly in your document

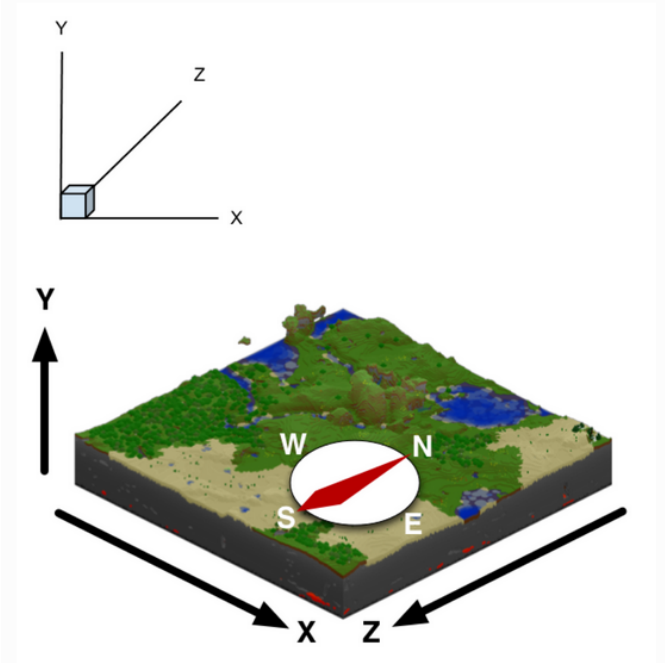
```
juckel = 3
```

Integer sind positive oder negative ganze Zahlen. Werte wie 10, 32, -6, 194689 oder -5 sind Integer, Zahlen wie 3,14 und 6,025 jedoch nicht.

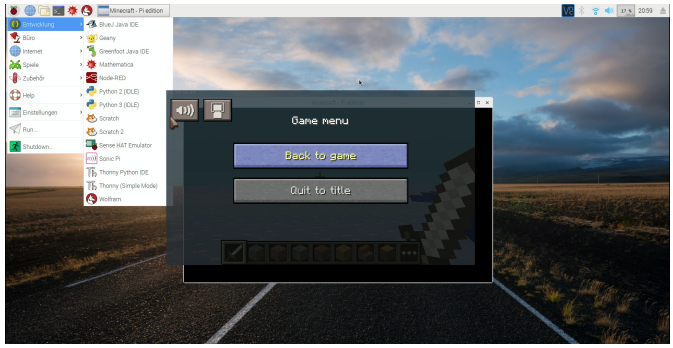
Um einen Spieler teleportieren zu wollen müssen wir zuerst verstehen wo er überhaupt steht und wo er hin soll. Im Spiel wird die Position oben links angezeigt:



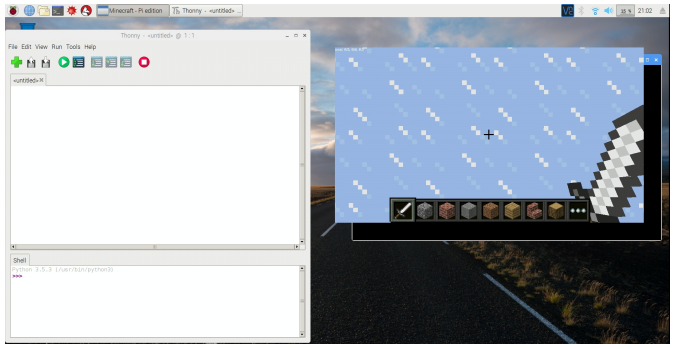
Die drei Zahlen stehen für die x-, y- und z-Achse in einem dreidimensionalem Raum. Im folgendem Bild ist dieser dargestellt. Jeder Pfeil stellt eine Richtung dar in welche die Figur sich bewegen kann.



Bewege deinen Spieler im Spiel, und achte dabei darauf, wie sich die Positionszahlen verändern. Dies sollte beim Bewegen des Spielers in Echtzeit erfolgen. Eigentlich ganz cool, oder? Allerdings benötigt man hierbei für lange Wege viel Zeit. Deswegen lassen wir ihn teleportieren. Hierzu öffnen wir im Raspberry Pi das Programm Thonny Python IDE.



Und ordnen den Programme wie folgt auf dem Bildschirm an:



Nun schreiben wir in das linke Fenster unser erstes Programm. Dies sieht wie folgt aus:

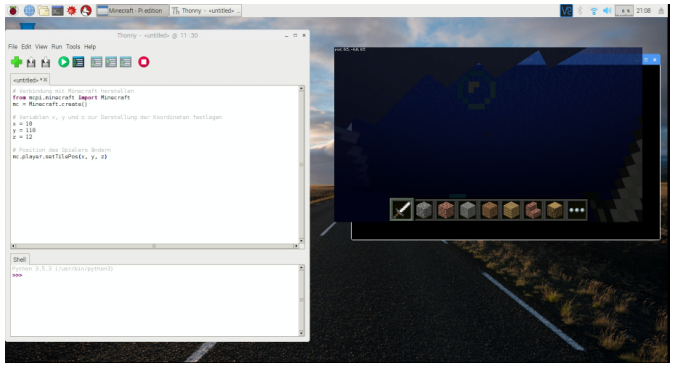
Listing 2. Insert code directly in your document

```
# Verbindung mit Minecraft herstellen
from mcpi.minecraft import Minecraft
mc = Minecraft.create()

# Variablen x, y und z zur Darstellung
# der Koordinaten festlegen
x = 10
y = 110
z = 12

# Position des Spielers aendern
mc.player.setTilePos(x, y, z)
```

Wichtig ist hierbei damit die Zahlen Variablen x, y und z nicht größer als 127 und nicht kleiner als -127 sind. Nun sollte es wie folgt aussehen:



Drücken wir nun den grünen Kreis mit dem weißen Pfeil wird unser Programm ausgeführt und unserer Spieler wird auf die eingestellt Position gestellt. Es kann sein damit zuerst die Aufforderung kommt zum Speichern. Hier kannst du kreativ sein und ein Name vergeben.

Natürlich kannst du auch anstelle von Integer-Variablen auch Float-Variablen vergeben. Float-Variablen sind Zahlen mit Nachkommastellen, wie z.B. 1.34 (Wichtig für das Komma ein Punkt setzen, da man das so in England macht und beim Programmieren ist Englisch die Hauptsprache). Dann sieht unsere obiges Beispiel mit juckel so aus:

Listing 3. Insert code directly in your document

```
juckel = 1.34
```

Jetzt ist juckel eine Float-Variable. Mit Float-Variablen kann man sich punktgenau teleportieren, was natürlich auch mit negative Zahlen geht.

Listing 4. Insert code directly in your document

```
juckel = -1.34
```

3.2 Weiter wichtige Funktionen

Mit dieser Funktion kannst du an Position x, y, z ein Block entfernen.

Listing 5. Insert code directly in your document

```
world.getBlock(x, y, z)
```

Mit dieser Funktion kannst du an Position x, y, z ein Block setzen.

Listing 6. Insert code directly in your document

```
world.setBlock(x, y, z, block_type)
```

Wichtig ist damit du für block_type einen der folgenden Block Namen einsetzt: AIR, STONE, GRASS, DIRT, COBBLESTONE, WOOD_PLANKS, SAPLING, BEDROCK, WATER_FLOWING, WATER, WATER_STATIONARY, LAVA_FLOWING, LAVA, LAVA_STATIONARY, SAND, GRAVEL, GOLD_ORE, IRON_ORE, COAL_ORE, WOOD, LEAVES, GLASS, LAPIS_LAZULI_ORE, LAPIS_LAZULI_BLOCK, SANDSTONE, BED, COBWEB, GRASS_TALL, WOOL, FLOWER_YELLOW, FLOWER_CYAN, MUSHROOM_BROWN, MUSHROOM_RED, GOLD_BLOCK, IRON_BLOCK, STONE_SLAB_DOUBLE, STONE_SLAB, BRICK_BLOCK, TNT, BOOKSHELF, MOSS_STONE, OBSIDIAN, TORCH, FIRE, STAIRS_WOOD, CHEST, DIAMOND_ORE, DIAMOND_BLOCK, CRAFTING_TABLE, FARMLAND, FURNACE_INACTIVE, FURNACE_ACTIVE, DOOR_WOOD, LADDER, RAIL, STAIRS_COBBLESTONE, DOOR_IRON, REDSTONE_ORE, SNOW, ICE, SNOW_BLOCK, CACTUS, CLAY, SUGAR_CANE, FENCE, GLOWSTONE_BLOCK, BEDROCK_INVISIBLE, STONE_BRICK, GLASS_PANE, MELON, FENCE_GATE, GLOWING_OBSIDIAN und NETHER_REACTOR_CORE.

Mit dieser Funktion kannst du gleich mehrere Block auf einmal setzen. Hier Baut das Programm dir alles zu von x1 bis x2 und von y1 bis y2 und von z1 bis z2.

Listing 7. Insert code directly in your document

```
world.setBlocks(x1, y1, z1, x2, y2, z2, block_type)
```

Diese Funktion gibt dir den höchsten Block auf der Stelle x, z zurück.

Listing 8. Insert code directly in your document

```
world.getHeight(x, z)
```

Mit dieser Funktion kannst du Nachrichten schreiben im Spiel.

Listing 9. Insert code directly in your document

```
world.postToChat("Message")
```

Literatur

[1] Python (programmiersprache), Apr 2018.

[2] Mincraft pi edition.