Von Schlangen und Himbeerkuchen: Programmieren in der Welt von Minecraft

1 Einführung

Schreibe ein Python-Script, das dafür sorgt, dass stets Blumen auf dem Block generiert werden, auf dem Steve sich befindet. Gehe dafür wie folgt vor:

- Binde die Minecraft-Bibliothek ein, um mit dem Spiel zu kommunizieren.

  ```
  import mcpi.minecraft as minecraft
  ```

- Baue die Verbindung zum Spiel auf.

  ```
  mc = minecraft.Minecraft.create()
  ```

  Das Objekt **mc** stellt nun die Kommunikationschnittstelle mit dem Spiel dar.

- Speichere dir die Block-ID der Blume (38) in einer Variable zwischen. Das erleichtert das Programmieren und macht den Code einfacher zu lesen.

- Verwende eine **while**-Schleife für den Hauptteil deines Scripts. Da das Platzieren der Blumen solange stattfinden soll, wie das Spiel läuft, wird eine Schleife benötigt, die nur durch Beenden des Programms abgebrochen werden kann.

- Frage innerhalb der **while**-Schleife die aktuelle Position von Steve ab. Diese benötigen wir, um an genau dieser Stelle die Blumen platzieren zu können.

- Platziere zu guter letzt an Steves aktueller Position die Blumen.

2 Lava Runner

Schreibe ein Python-Script, das einen zufälligen Parcour über das Lavabecken generiert.

- Um das Lavabecken und den restlichen Rahmen zu generieren, führe das Initiliasierungsscript im Terminal aus.

  ```
  python lava_runner_initializer.py
  ```

- Binde die Minecraft-Bibliothek ein, um mit dem Spiel zu kommunizieren.

  ```
  import mcpi.minecraft as minecraft
  ```

- Öffne die Datei `lava_runner.py` mit dem Editor. Diese Datei enthält bereits Code und muss lediglich im gekennzeichneten Bereich erweitert werden.

- Parameter: Die Koordinaten x, y, z beinhalten die Position des zu setzenden Steins

- Variablen: Die Variablen x_boundary, z_boundary sind die Grenzen der Arena und markieren das Ende des Pfades

- Aufgaben:
  1. Schreibe eine Schleife, die abbricht, wenn die berechnete Steinposition größer als die x_boundary oder z_boundary ist
  2. Berechne mit der

     ```
     random.randint(lower_bound, upper_bound)
     ```

     Funktion neue x, y, z Positionen
  3. Fange mit einer If-Bedingung die Positionen ab, die vom Spieler nicht erreicht werden können
  4. ersetze die alte Steinposition durch die gerade berechnete Steinposition
  5. setze mit der `setBlock` Methode einen

     ```
     block.GLOWSTONE_BLOCK
     ```

     an die aktuelle Steinposition

3 Block Shuffel