Verwenden von Bibliotheken: Matplotlib

Claas de Boer, Tilman Hinnerichs

17. Dezember 2020

Python-Grundlagen

Rückblick

Aufgaben

1. Erweitere die Spielerklasse, sodass ein Spieler mehrere Würfel besitzen kann. Wenn er würfelt, so soll er mit allen Würfeln nacheinander würfeln und eine Ergebnisliste zurückbekommen.
2. Modelliert das Würfelspiel als eigene Klasse.

↪ Lösung in Snippets

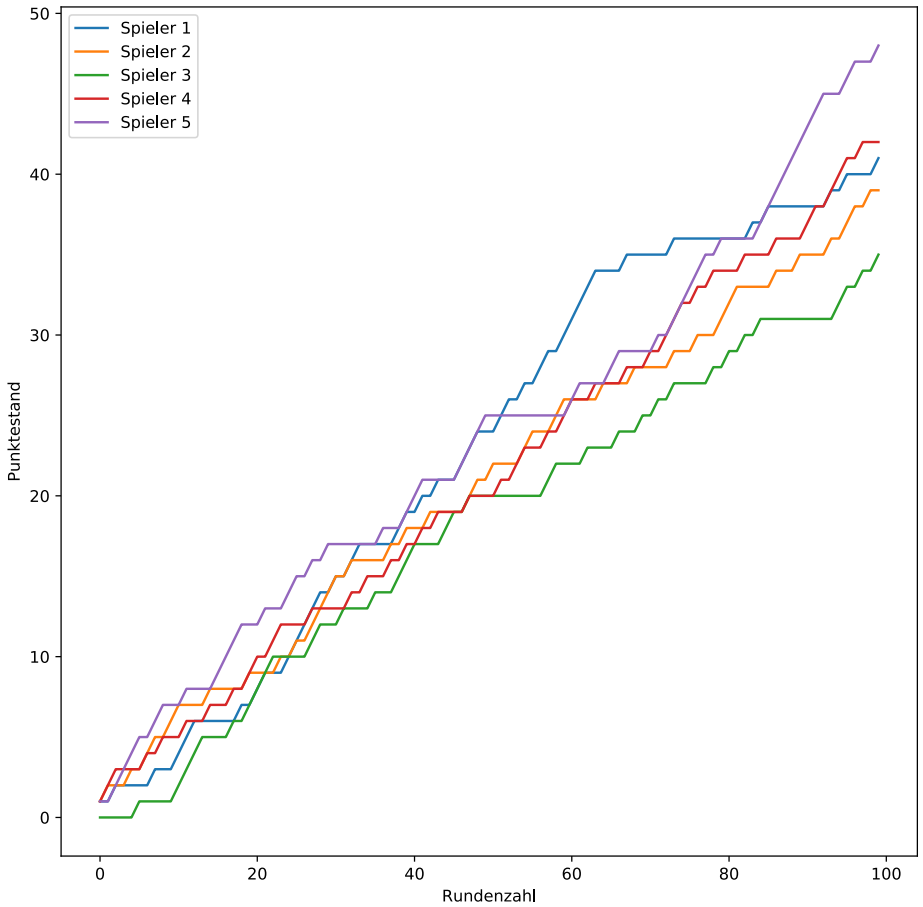
Module einbinden

```
1  from random import randint
2  print(randint(1, 10))
3
4  from wuerfel import *
5  d20 = Wuerfel(20)
6  print(d20.wuerfeln())
7
8  import spieler
9  spieler = spieler.Spieler("Arndt", d20)
10 print(spieler.name)
11
```

matplotlib

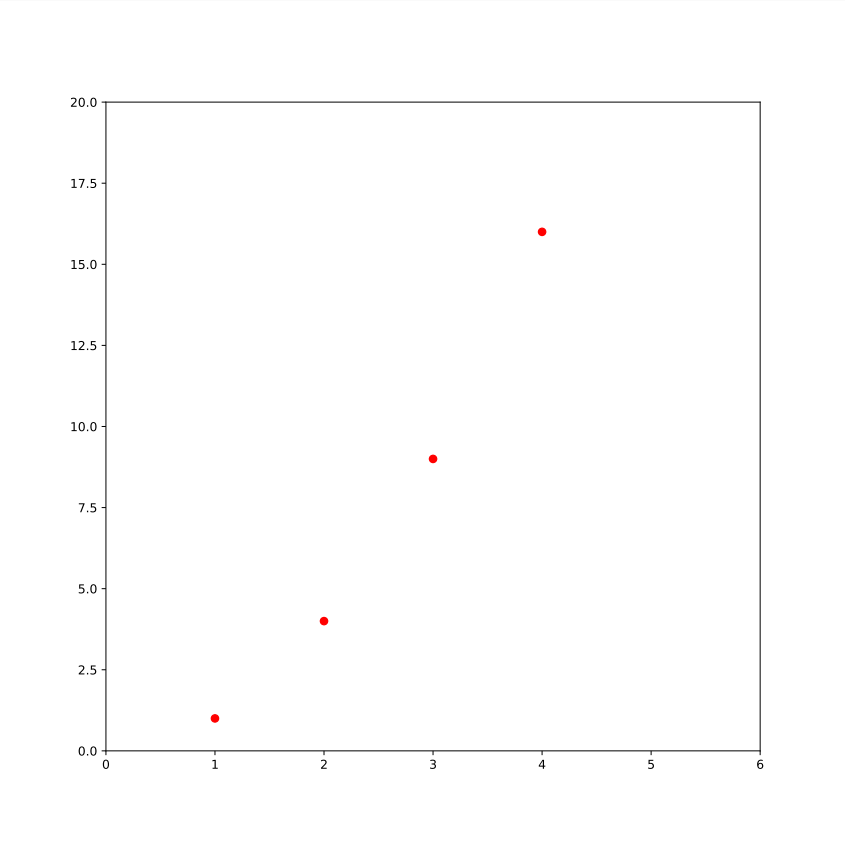
Motivation

- Mit Python berechnete Daten plotten:



Einfaches Beispiel

```
1 from matplotlib import pyplot
2 pyplot.plot([1,2,3,4], [1,4,9,16], 'ro')
3 pyplot.axis([0, 6, 0, 20])
4 pyplot.show()
5
```



Matplotlib instalieren

Linux (Ubuntu):

```
pip3 install --user matplotlib

```

Windows:

```
python.exe -m pip install matplotlib

```

Matplotlib Tutorial:
`https://matplotlib.org/users/pyplot_tutorial.html`

Aufgaben

Aufgaben

1. Erstelle ein Diagramm, in dem der Entstand eines Spiels nach der Simulation (`.simuliere()`) für jeden Spieler dargestellt ist.
2. Erstelle ein Diagramm, in dem der Punktestand aller Spieler im zeitlichen Verlauf dargestellt ist (s. Beispiel in dieser Präsentation)
3. Füge einen schummelnden Spieler zu deinem Spiel hinzu:
   3.1 Mehr Würfel
   3.2 Andere Seitenzahl
   3.3 seid kreativ . . .

   Wie wirkt sich das Schummeln bei den verschiedenen Wertungsmodi aus?
   (`"wuerfelsumme"`, `"wuerfelmaximum"`, `"paschzahl"`)
   Dies wird bei höherer Rundenzahl möglicherweise deutlicher.