

Aufgabenblatt 8

Aufgabe 26: (mar, kov)

Eine Methode, um sinnfreie Texte zu erstellen, die auf den ersten Blick echt wirken, basiert auf Markov-Ketten. Dabei wird das nächste Wort im generierten Text so ausgewählt, dass es zu dem (den n) vorhergehenden Wörtern passt.

Programmieren Sie einen Markov Text-Generator!

Für jede der folgenden Einzelaufgaben gibt es einen Punkt.

- Schreiben Sie eine Funktion `process_text(filename)` → dict, die eine Textdatei einliest und ein dictionary, das als Schlüssel jedes Wort des Textes enthält, zurückgibt. Der Wert zu jedem Schlüssel soll ein set sein, das die Wörter enthält, die im Text nach dem „Schlüssel“-Wort vorkommen.
- Schreiben Sie eine Funktion `markov_gen(dict, n)` → str, die aus diesem dictionary einen Text mit n Wörtern erstellt. Beginnen Sie mit einem zufälligen Wort, und wählen sie das jeweils nächste Wort aus dem zugehörigen set zufällig aus. Dabei können Sie die Funktion `choice` aus dem Modul `random` verwenden.

Als Textvorlage können sie den Werther von der Vorlesungsseite oder jeden beliebigen anderen Text verwenden.

Aufgabe 27: Zahlenraten

Wir haben in der Vorlesung kurz das Spiel *Zahlenraten* angesprochen: Ein Mitspieler denkt sich eine Zahl zwischen 1 und 100. Die andere Mitspielerin muss die Zahl erraten, indem sie Fragen der Art „Ist Deine Zahl größer als x ?“ stellt.

Programmieren Sie dieses Spiel aus den verschiedenen Perspektiven!

Für jede der folgenden Einzelaufgaben gibt es einen Punkt.

- Der menschliche Spieler denkt sich eine Zahl, der Computer muss raten.
- Der Computer wählt eine Zahl, der Mensch muss raten.
- Verändern Sie Ihren Code so, dass statt 1 und 100 beliebigen Zahlen als Grenzen vorgegeben werden können. Diese Grenzen sind beiden Spielern bekannt.

Aufgabe 28: Zusatzaufgabe: Schildkröten malen besser

Mit dem Modul `turtle` können Sie ganz einfach Zeichnungen erstellen. Sie müssen dazu `tkinter` nachinstallieren. Der folgende Code malt das Haus vom Nikolaus:

```
1 #!/bin/env python
2
3 import turtle as t
4
5 t.Screen()
6 s = 100
7 points = [(s,0), (s,s), (0,s), (0,0), (s,s), (s/2.,2.*s), (0,s), (s,0)]
8 for p in points: t.goto(p)
9 t.mainloop()
```

Auf der Vorlesungsseite habe ich noch zwei weitere Beispiele hochgeladen. Außerdem können Sie sich für folgende Aufgabe auf <https://docs.python.org/3.5/library/turtle.html> inspirieren lassen und indem Sie folgendes aufrufen und staunen:

```
$ python -m turtledemo
```

Vorsicht: Einige der demos sind animiert. Wenn Sie photosensibel sind, bitte vorsichtig sein! Vor allem `round_dance`!

Die Aufgabe: Erschaffen Sie ein Kunstwerk mit `turtle`!

Das schönste erhält einen Extrapunkt!