

## Lösungsidee:

Man braucht eine Struktur mit Koordinaten, auf die man gut zugreifen kann. Hierbei gibt es verschiedene Möglichkeiten, eine wäre es, viele Objekte zu erstellen mit Attributen für die X/Y-Koordinaten. Ich habe mich aber für ein zweidimensionales Array entschieden, dies hat nämlich den Vorteil, dass es X und Y Koordinaten beinhalten (als Zeilen und Spalten).

```
breite:15, hoehe 7
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[Finished in 194ms]
```

Jede Null hat eine (X|Y) Koordinate und kann mit der Syntax `array[y][x]` angesteuert werden.

Mit dem Random Modul von Python, bekommt das Wort was eingesetzt werden soll eine zufällige (X|Y) Koordinate. Diese Koordinate kann man als Anker bezeichnen, denn von diesem Punkt wird das Wort dann eingesetzt.

Vertikal:  $(x_{\text{konstant}} | y_{\text{Ankerpunkt}}) \dots (x_{\text{konstant}} | y_{\text{Ankerpunkt}} + \text{Wortlänge} - 1)$

Horizontal:  $(x_{\text{Ankerpunkt}} | y_{\text{konstant}}) \dots (x_{\text{Ankerpunkt}} + \text{Wortlänge} - 1 | y_{\text{konstant}})$

```
breite:15, hoehe 7
['0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']
['0', '0', '0', 'I', 'n', 'f', 'o', 'r', 'm', 'a', 't', 'i', 'k', '0', '0']
['0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']
['0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']
['0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']
['0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']
['0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']
[Finished in 197ms]
```

*Ankerpunkt* (red arrow pointing to 'I' at row 1, column 4)

*Wortlänge = 10* (red bracket under the word 'Informatik' from column 4 to column 13)

(4|2), (5|2), (6|2), ..., (13|2) bevor das Wort eingesetzt wird, wird geschaut, ob sich auf diesen Koordinaten schon etwas befindet. Wenn nein, dann kann es eingesetzt werden, wenn ja, dann wird ein neuer Ankerpunkt zugewiesen. So lange, bis eine freie Stelle gefunden wurde oder die Anzahl von verfügbaren Versuchen erreicht ist. Der Ankerpunkt bei den vertikalen Worten „Informatik“ kann einen maximalen Wert von  $(x_{\text{breite-wortlänge}} | y_{\text{hoehe}})$  erreichen. Wenn alle Wörter eingesetzt wurden, werden die Nullen dann durch wählbare Symbole ausgetauscht.

```
BWINF\wortsuche>python wortsuche.py worte4.txt SHJKIAPONQMKSL --h --v
```

`python wortsuche.py test_datei.txt einsetz_symbole --h (horizontal) --v (vertikal)`