

Junioraufgabe 2: Hopsitexte

Team-ID: 00578

Team-Name: Frederik Hamann

Bearbeiter/-innen dieser Aufgabe:
Frederik Hamann

18. November 2024

Inhaltsverzeichnis

1	Lösungsidee	1
2	Umsetzung	1
3	Beispiele	2
4	Quellcode	2

1 Lösungsidee

Tatsächlich habe ich erst, nachdem ich die Aufgabe 1 aus dem Aufgabenblatt bearbeitet habe, festgestellt, dass ich die Junioraufgaben auch bearbeiten darf, daher habe ich diese Aufgabe erst nach dem Bearbeiten von Aufgabe 1 gelöst, wodurch ich im Prinzip das Programm aus Aufgabe 1 für die Aufgabe wieder verwendet habe und nur in Teilaspekten gekürzt habe und den Algorithmus welcher in Aufgabe 1 von mir verwendet wird nun so nutze, dass er anstatt den Abstand von Hopser 1 und 2 zu berechnen, er untersucht ob Hopser 1 oder 2 weiter gekommen ist.

Algorithm 1 check_win()

```
global winner
while true do
    check_hopsi(hopser_1)
    check_hopsi(hopser_2)
    if check_hopsi(0) > check_hopsi(1) then
        winner ← "Hopser 1 gewinnt!"
    else if check_hopsi(1) > check_hopsi(0) then
        winner ← "Hopser 2 gewinnt!"
    else
        winner ← "Unentschieden!"
    end if
end while
```

2 Umsetzung

Es wird mithilfe der Python-Bibliothek „tkbootstrap“ ein GUI erstellt, welches ein Textfeld und eine Radialanzeige enthält. Das Textfeld dient der Eingabe des Hopsitextes, während die Radialanzeige zur

Darstellung des Abstands zwischen den Endpositionen genutzt wird. Es hat einen Anzeigebereich von 0 bis 29. Der Anzeigebereich wurde auf 29 berechnet, da der höchste Sprungwert bei 30 liegt und der Abstand somit nur bei max. 29 liegen kann, da der 2. Hopser einen Buchstaben nach dem 1. Texthopser startet und somit in einem Fall, wo auf ein „ß“ 29 mal „a“ folgen würde, der Abstand 29 betrüge (s. Abbildung 3).

3 Beispiele

Genügend Beispiele einbinden! Die Beispiele von der BwInf-Webseite sollten hier diskutiert werden, aber auch eigene Beispiele sind sehr gut, besonders wenn sie Spezialfälle abdecken. Aber bitte nicht 30 Seiten Programmausgabe hier einfügen!

4 Quellcode