Aufgabe 1: Hopsitexte

Programmnutzung

So verwendet man das Programm im Terminal:

```
$ python3 ./hopsitexte.py
```

Das Programm kann durch Drücken von `CTRL + C` oder `CTRL + D` beendet werden, wenn man fertig mit dem Verfassen ist. Dann wird der gesamte Text ausgegeben, den man geschrieben hat.

Grundgedanken und Programmablauf

Damit man nach eigenen Wünschen einen Text verfassen kann, bei dem trotzdem nie die Spieler von der gleichen Stelle aus herausspringen, haben wir uns für einen einfachen, aber effektiven Weg entschieden: Die Nutzerin bzw. der Nutzer kann nach freien Belieben Texte formulieren, die das Programm dann auf ebendiese Fehleranfälligkeit überprüft. Es wird dann markiert, an welchem Buchstaben sich die Spieler*innen in der Position überlappen würden und es wird eine Information zur Lösung dieses Konfliktes ausgegeben.

So funktioniert der Code:

1. bis der Nutzer das Programm schließt, wiederhole:
    1. lese eine Zeile aus `stdin` ein. Füge diese an den bisher gesammelten Text an.
    2. entferne alle Zeichen, die unsere Spieler*innen überspringen würden - also alle Teile, die kein Buchstabe sind.
    3. spiele das Szenario durch:
        1. Spieler*in A startet bei 0, Spieler*in B startet bei 1.
        2. für jeden Buchstaben an der Position des Spielers/der Spielerin wird berechnet, wie weit von dort aus nun gesprungen werden muss.
        3. sollte sich an irgendeinem Zeitpunkt die Position beider Spieler gleichen, so wird das gemeldet und der Satz nicht akzeptiert.
2. ermittle die Gewinnerin bzw. den Gewinner und gebe noch einmal den gesamten Text aus.

Wichtigste Teile des Quelltextes

Dieser Teil des Codes spielt den Hopsitext durch und gibt die Position der Spieler*innen zurück:

```
def playthrough(text: str):
    """
    Spielt den bisherigen Text so weit durch, bis mindestens ein*e
Spieler*in
    den neuen Satz, an dem wir gerade sitzen, erreicht. Denn erst dann
müssen
```

```
    wir das nächstbeste Wort herausfinden.

    Gibt den Index zurück, an dem ein oder mehrere Spieler*innen angekommen
    sind, wenn sie unseren neuen Satz erreichen.
    """

    pos_a, pos_b = 0, 1
    text_end_index = len(text) - 1

    while (pos_a <= text_end_index) or (pos_b <= text_end_index):
        pos_a, pos_b = gametick(text, pos_a, pos_b)

    return (pos_a, pos_b)
```

Die Gametick-Funktion:

```
def gametick(text: str, pos_a: int, pos_b: int) -> "tuple[int, int]":
    """
    Spielt eine Runde beider Spieler im Hopsitext und gibt deren neue
Positionen zurück.
    """

    def get_new_player_pos(pos: int):
        while (pos < len(text)) and (text[pos].lower() not in
ALLOWED_CHARS):
            pos += 1  # wir überspringen alles, was kein Buchstabe ist!

        # nun geben wir die derzeitige Position plus die Wertigkeit des
derzeitigen
        # Buchstabens zurück.
        return pos + (char_value(text[pos]) if pos < len(text) else 1)

    new_pos_a = get_new_player_pos(pos_a)
    new_pos_b = get_new_player_pos(pos_b)

    if new_pos_a == new_pos_b:
        raise PlayersOnSameCharException(
            "Die Spieler sind auf dem selben Buchstaben im Text. Sie werden
\
an der selben Stelle herausspringen. Beachte, falls dir kein Fehler
auffallen \
sollte, dass unsere Spieler*innen alles, was kein Buchstabe ist,
überspringen! \
Der Fehler liegt übrigens hier - ab diesem Buchstaben haben sich die \
Positionen überlappt: \n\n"
            + part_of_text(text, new_pos_a)
        )

    return (new_pos_a, new_pos_b)
```

Und zu guter Letzt die `part_of_text`-Funktion, die uns anmerkt, wo der Fehler lag:

```
def part_of_text(text: str, pos: int, padding: int = 20) -> str:
    """
    Extrahiert einen kleinen Teil des Textes, um ihn bei Fehlern anzeigbar
zu
    machen.
    """

    left_bound: int = pos - padding
    right_bound: int = pos + padding

    if pos - padding < 0:
        left_bound = 0
    if pos + padding > (len(text) - 1):
        right_bound = len(text)

    return (
        " ... "
        + text[left_bound:right_bound]
        + " ... "
        + "\n"
        + " " * 5
        + (" " * (padding if left_bound != 0 else pos) + "^")
        + " " * 5
    )
```

Ausgaben des Proramms

```
user@testpc:~/bwinf$ python3 ./hopsitexte.py
*** Willkommen beim Hopsitext-Assistenten! ***

    Dieser Assistent hilft dir bei der Erstellung eines möglichst langen
Hopsitextes.
    Du kannst frei Sätze formulieren. Wenn es Kollisionen gibt,

    Es gilt wie immer in einer Terminalanwendung: CTRL+C oder CTRL+D
beenden das Programm.
    Drücke diese Tasten, wenn Du zufrieden bist, und dir wird dein Text
ausgegeben.

        >> Bereit? Los geht's! Drücke ENTER, um zu beginnen. <<


 --- Satz #1 ---

Formuliere einen Satz nach deinen Wünschen. Das Programm wird ihn sich
ansehen und überprüfen, ob es passieren kann, dass die Spieler*innen an der
selben Stelle landen und somit auch an derselben heraushüpfen. Dann gibt
dir das Programm Tipps, wo dieser Fehler aufgetreten ist. Dann kannst Du
```

```
dir erschließen, welches Wort zu verändern ist.
Balkone sind wunderschön.
Die Spieler sind auf dem selben Buchstaben im Text. Sie werden an der
selben Stelle herausspringen. Beachte, falls dir kein Fehler auffallen
sollte, dass unsere Spieler*innen alles, was kein Buchstabe ist,
überspringen! Der Fehler liegt übrigens hier - ab diesem Buchstaben haben
sich die Positionen überlappt:

 ... Balkonesindwunderschön ...
       ^

Bitte versuche, einen Satz zu formulieren, bei dem die Spielerpositionen ab
dieser Stelle nicht überlappen.

Okay!
 --- Satz #2 ---

Formuliere einen Satz nach deinen Wünschen. Das Programm wird ihn sich
ansehen und überprüfen, ob es passieren kann, dass die Spieler*innen an der
selben Stelle landen und somit auch an derselben heraushüpfen. Dann gibt
dir das Programm Tipps, wo dieser Fehler aufgetreten ist. Dann kannst Du
dir erschließen, welches Wort zu verändern ist.


Alles klar, Du hast das Programm verlassen. Hier dein Text:

Okay!

Spieler*in A wird gewinnen.
```