Junioraufgabe 2

Von Wem?

Vor- und Nachname: Anian Barthel

Team-Name: BitPioneer

Team-ID: 00473

Wann: 21.10.2024

Lösungsidee

Beliebig viele „Runner“ springen um die Wette, immer einer nach dem anderen. Der gewinnende Runner unterbricht und schreibt seinen Namen in die Konsole.

Umsetzung

Erst werden alle Buchstaben zu Kleinbuchstaben konvertiert und alle weiteren Zeichen, die keine Buchstaben sind, entfernt.

```
let characters: Vec<char> = text.
    to_lowercase() : String
    .chars().filter(|c| c.is_alphabetic()).collect();
```

Danach beginnt das Texthopsen.

Hierzu erstellt das Programm einen Runner pro Spieler, jeder einzelne speichert die aktuelle Position, den Spielernamen und den „sauberen“ Text.

```
let mut runners: Vec<Runner> = vec![
    Runner::new(0, "Bela", &characters),
    Runner::new(1, "Amira", &characters),
];
```

Diese Runner dürfen daraufhin nacheinander springen.

```
loop {
    for runner in &mut runners {
        // hop the winner will exit the program and print his name
        runner.hop();
    }
}
```

Der Sprung funktioniert wie folgt: Erst wird die aktuelle Position überprüft, indem der aktuelle Character aus dem Array gelesen wird. Ist der Index zu groß, wird das Programm beendet und der Spielername in die Konsole geschrieben.

```
pub fn hop(&mut self) {
    let new_pos = match &self.characters.get(self.pos) {
        Some(c) => calc_jump_width(c),
        None => {
            print!("The Winner is: {}", &self.player_name);
            exit(0);
        }
    };

    self.pos += new_pos;
}
```

Ist der Index passend, wird eine neue Position berrechet, indem der Ascii-Wert minus 96 (Wert für „a“) gerechnet wird. Ist die Zahl ein Umlaut, so kann sie nicht berrechnet werden und wird einfach durch einen match-Befehl berechnet.

```
pub fn calc_jump_width(c: &char) -> usize {  2 usages
    // umlauts are not in the ascii table so we have to handle them separately
    match c {
        'ä' => 27,
        'ö' => 28,
        'ü' => 29,
        'ß' => 30,
        _=> c.to_ascii_lowercase() as usize - 96
    }
}
```

Dann wird zur alten Position die neue addiert und der nächste „Runner“ ist am Zug.

Beispiele

1.

Text: „AbC“

1. Alles wird kleingeschrieben und alle weiteren Zeichen entfernt. Neuer Text: „abc“
2. Runner 1:
    - Position 1
    - Buchstabe „a“ ergibt die Sprungweite 1
    - Sprung an Position 1 + 1 = 2

3. Runner 2:
    - Position 2
    - Buchstabe „b“ ergibt die Sprungweite 2
    - Sprung an Position 2 + 2 = 4
    -
4. Runner 1:
    - Position 2
    - Buchstabe „b“ ergibt die Sprungweite 2
    - Sprung an Position 2 + 2 = 4
5. Runner 2:
    - Position 4
    - Die Position 4 ist nicht mehr Teil des Textes, deswegen wird das Programm beendet und schreibt den Gewinnernamen in die Konsole.

2.

Text: „%d@G&Ge.zBhtß“

1. Alles wird kleingeschrieben und alle weiteren Zeichen entfernt , neuer Text: „dggezbhtß“
2. Runner 1:
    - Position 1
    - Buchstabe „d“ ergibt die Sprungweite 4
    - Sprung an Position 1 + 4 = 5
3. Runner 2:
    - Position 2
    - Buchstabe „g“ ergibt die Sprungweite 7
    - Sprung an Position 2 + 7 = 9
4. Runner 1:
    - Position 5
    - Buchstabe „z“ ergibt die Sprungweite 26
    - Sprung an Position 5 + 26 = 31
5. Runner 2:
    - Position 9
    - Buchstabe „ß“ ergibt die Sprungweite 30
    - Sprung an Position 9 + 30 = 39
6. Runner 1:
    - Position 31
    - Die Position 31 ist nicht mehr Teil des Textes, deswegen wird das Programm beendet und schreibt den Gewinnernamen in die Konsole.

3. „hopsen1.txt“

   Bela gewinnt.

4. „hopsen2.txt“

   Bela gewinnt.

5. „hopsen3.txt“

   Bela gewinnt.

6. „hopsen4.txt“

   Amira gewinnt.

7. „hopsen5.txt“

   Bela gewinnt.

Quellcode

Siehe Ausschnitt in der Umsetzung.