

Junioraufgabe 2

Von Wem?

Vor- und Nachname: Anian Barthel

Team-Name: BitPioneer

Team-ID: 00473

Wann: 21.10.2024

Lösungsidee

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

Umsetzung

Erst entfernt das Programm alle nicht alphabetischen Buchstaben.

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

Danach beginnt das Texthopsen.

Dann erstellt das Programm 2 Runner Objekte, diese speichern die aktuelle Position, Spielernamen und den „sauberen“ Text.

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

Diese Runner Objekte dürfen dann 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 in dem 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 berechnet in dem die ascii Zahl - 96 machen ist die Zahl ein Umlaut kann sie nicht berechnet werden und wird einfach durch ein match ausgegeben.

```
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 zu der alten Position die neue hinzugefügt und der Zyklus beginnt erneut.

Beispiele

1.

Text: „abc“

1. Runner 1: startet bei 1 („a“), und a ist im Array also wird $2(1 + \text{Wert von „a“})$ als neue Position gespeichert.
2. Runner 2: startet bei 2 („b“), und speichert $4(2 + 2(\text{Wert von b}))$ als neue Position.
3. Runner 1: ist wieder dran und 2 ist im Array vorhanden also wird eine neue Position $4(2 + \text{„b“})$ gespeichert.

4. Runner 2: 4 ist nicht im Array, deswegen unterbricht der Runner das Programm und schreibt seinen Namen in die Konsole.

2.

Text: „a b,c.p%z“

1. Es werden erst alle analphabetischen Buchstaben entfernt, das ergibt „abcpz“.
2. Runner 1: startet bei 1 („a“), und a ist im Array also wird $2(1 + 1)$ (Wert von „a“) als neue Position gespeichert.
3. Runner 2: startet bei 2 („b“), und speichert $4(2 + 2)$ (Wert von „b“) als neue Position.
4. Runner 1: ist wieder dran und speichert 4 als neue Position.
5. Runner 2: Index 4 ist im Array also wird als neue Position 19 ($4 + 15$)
6. Runner 1: Index 4 ist im Array also wird als neue Position 19 ($4 + 15$)
7. Runner 2: ist wieder dran und 19 ist nicht im Array also beendet Runner 2 das Programm und schreibt seinen Namen in die Konsole.