

# Aufgabe 1: Hopsitexte

Team-ID: 00178

Team-Name: Team-Name

Bearbeiter/-innen dieser Aufgabe:  
Matthew Greiner

October 31, 2024

## Contents

<b>1</b>	<b>Lösungsidee / Ansatz</b>	<b>1</b>
<b>2</b>	<b>Umsetzung</b>	<b>2</b>
<b>3</b>	<b>Beispiele</b>	<b>2</b>
<b>4</b>	<b>Quellcode</b>	<b>2</b>

## 1 Lösungsidee / Ansatz

In dieser Aufgabe geht es darum, ein Programm zu erstellen, das Zara hilft, deutsche Hopsitexte zu erstellen. Um ein Programm zu gestalten, damit Zara besonders gut damit arbeiten kann, müssen meiner Meinung nach folgende Kriterien für das Programm erfüllt sein:

1. Das Programm muss ein graphisches Nutzerinterface mit einem Textfeld haben, in der Zara ihren Text tippen kann.
2. Es muss anzeigen, ob der eingegebene Text ein "Hopsitext" ist oder nicht.
3. Die Stellen, an denen die Sprünge stattgefunden haben, müssen wie in den Beispieltexten, rot für den ersten Spieler und blau für den zweiten Spieler, gefärbt sein.
4. Wenn der Text kein Hopsitext ist, muss die Stelle, an dem sich die Spieler treffen, farblich markiert sein, damit Zara merkt, dass sie Änderungen vornehmen muss.
5. Da es ein deutscher Text sein soll, müssen Umlaute wie "ä, ö, ü" und "ß" auch unterstützt werden.
6. Die Ausgabe des Programmes und die farbliche Markierungen des Eingabetextes muss dynamisch und automatisch mit dem Tippen von Zara geschehen und Zara kann an beliebigen Stellen im Eingabetext Änderungen vornehmen (Quality of life).

Auf der linken Seite des Programmes soll Zara ihren Text tippen können und auf der rechten Seite soll dieser Text, an den Sprüngen der Spieler farblich markiert - wie im Beispieltext, angezeigt werden. So sieht Zara beim tippen, ob ihre aktuelle Eingabe ein Hopsitext ist oder nicht. Zusätzlich soll oberhalb des Textfeldes deutlich angezeigt werden, ob es sich um einen Hopsitext handelt. Wenn diese Kriterien erfüllt sind, hat Zara ein Programm, in der sie einen Text schreiben kann, das ihr beim tippen anzeigt, ob ihre aktuelle Eingabe ein Hopsitext ist oder nicht. Falls es kein Hopsitext ist, sieht Zara das sofort und kann ihren Text bzw. einfach die letzten Worte mit dem Sprung, ändern.

## 2 Umsetzung

Um dieses Programm zu erstellen, habe ich JavaFX als GUI Library verwenden, was mir grundlegende Bausteine für das Interface zur Verfügung stellt.

Zu Beginn erstelle ich ein Textfeld auf der linken Seite des Programms, eine formatierte Textanzeige auf der rechten Seite, in der die Hopsisprünge markiert werden und ein Text mit dem Ergebnis ganz oben. Das Textfeld bekommt einen Listener, der jedes mal gerufen wird, wenn sich der Text im Textfeld ändert. So wird jedes mal das Ergebnis und der formatierte Text auf der rechten Seite geupdated, sobald sich die Texteingabe ändert.

```

1      private TextArea textArea = new TextArea();
      private TextFlow textFlow = new TextFlow();
3      private Text output = new Text("Ist dein Text ein Hopsitext? Tippe in das Textfeld");

5      @Override
      public void start(Stage stage) throws IOException {
9          // Listens for changes in the TextArea
          textArea.textProperty()
              .addListener(
11                 (ObservableValue<? extends String> observ, String old, String newVal) -> {
                    // Called every time the text is changed
                    processText(newVal);
13                 });

15         // Give the TextArea the left side and colored text on the right side
        SplitPane splitPane = new SplitPane();
17         splitPane.getItems().addAll(textArea, textFlow);

19         // Put the result on top
        BorderPane layout = new BorderPane();
21         layout.setCenter(splitPane);
        layout.setTop(output);
23     }

```

Um zu markieren, wo die Sprünge eines Spielers stattgefunden haben, muss der eingegebene Text geparkt werden. Dabei wird berechnet wieviele Zeichen übersprungen werden müssen, um zum nächsten Zeichen zu kommen, dass markiert werden muss. Dieser Prozess wird für die zwei Spieler separat gemacht und am Ende in einen farblich markierten Text zusammengefügt.

## Edgecases

1. Text beginnt mit nicht alphabet Zeichen.
2. Kein Text
3. Kurzer Text (paar wenige Zeichen, z.B. 1)
- 4.

Hier wird kurz erläutert, wie die Lösungsidee im Programm tatsächlich umgesetzt wurde. Hier können auch Implementierungsdetails erwähnt werden.

## 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

Unwichtige Teile des Programms sollen hier nicht abgedruckt werden. Dieser Teil sollte nicht mehr als 2-3 Seiten umfassen, maximal 10.