

# Übungsblatt 3

## Interaktive Systeme – SoSe 24

Prof. Dr. Michael Rohs, Jan Feuchter, M.Sc.

Alle Übungen, die nicht explizit als Gruppenaufgabe deklariert sind, müssen in Einzelarbeit geleistet und individuell abgegeben werden. Identische Abgaben werden als Plagiat behandelt.

Abgabe bis Montag den 22.04. um 23:59 Uhr über <https://assignments.hci.uni-hannover.de/SoSe2024/ISy>. Die Abgabe muss aus einer einzelnen zip-Datei bestehen, die alle nötigen Dateien enthält. Lösen Sie Umlaute in Dateinamen bitte auf.

Verpacken Sie Ihre Lösung (pdf-Datei und zip-Datei des exportierten Projekts) zum Hochladen in eine zip-Datei.

### Aufgabe 1: No Chrome (10 Punkte)

In der Vorlesung wurde gezeigt, dass No Chrome Interfaces versuchen Widgets zu vermeiden.

- Welche Philosophie wird mit „No Chrome“ verfolgt?
- Vergleichen Sie die Applikation OpenOffice oder Word mit dem Sublime Text Editor. Welche Widgets werden jeweils verwendet? Wie werden verschiedene Funktionen dem Nutzer zur Verfügung gestellt?
- Nennen Sie 2 Vor- und 2 Nachteile von No Chrome Interfaces. Nennen Sie aus Ihrem Alltag eine mögliche Ausprägung von No Chrome Interfaces.

## Aufgabe 2: Ladebalken (10 Punkte)

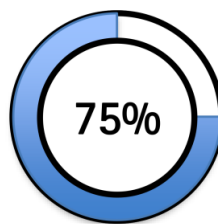
In der Vorlesung wurden Progressbars in JavaFX und deren Feedbackmethoden vorgestellt. Lesen Sie hierzu folgendes Paper:

Harrison et al. Faster Progress Bars: Manipulating Perceived Duration with Visual Augmentations. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Verfügbar unter: <https://dl.acm.org/citation.cfm?id=1753556>

Beantworten Sie folgende Fragen bezüglich des Papers:

- Beschreiben Sie kurz das Studiendesign und welche Faktoren untersucht wurden.
- Sie haben die Aufgabe einen runden Progress Indikator, wie folgt dargestellt, anhand der Ergebnisse aus dem Paper zu designen. Hierfür soll die Form nicht geändert werden. Es geht nur um die Darstellung des Progress Indikators (blaue Fläche). Wie würden Sie den Progress Indikator darstellen? Wieviel schneller (in %) können Nutzer den Ladebalken theoretisch wahrnehmen?



## Aufgabe 3: Listeners, Properties, Bindings (12 Punkte)

In dieser Aufgabe soll das Verhalten eines Dialogs zur Einstellung von Schriftarten implementiert werden. Die Struktur der UI ist vorgegeben (siehe Stud.IP: FontSelect.zip). Verwenden Sie wo immer möglich Bindings, ansonsten Listeners. Verwenden Sie im Fall von Listenern Lambda-Ausdrücke und keine anonymen Klassen. Versuchen Sie den Code so kurz wie möglich zu schreiben. Die folgenden Aspekte sollen implementiert werden. (Diese Aspekte finden sich auch als Kommentare im Quelltext des Templates.)

- a) set font to focused list item
- b) filter list as text is entered in search field (ignore case, show all fonts whose name contains the search text)
- c) update the countView to show the number of fonts remaining in the list
- d) update the percentageView to show the percentage of fonts remaining)
- e) update font size as sizeSlider is modified
- f) update underline through checkbox
- g) update strikethrough through checkbox
- h) update font color with radio buttons
- i) update font weight with radio buttons
- j) update font posture with radio buttons
- k) update font blurring as slider is modified

Die Dokumentation von JavaFX finden Sie unter: <https://openjfx.io/javadoc/11/>

Exportieren Sie Ihre Lösung als zip-Datei aus IntelliJ.