Human-Computer Interaction Group

Leibniz Universität Hannover

Mobile Interaction
Summer 2024

Prof. Dr. Michael Rohs, Shashank Ahire, M.Sc.

Assignment 7

All exercises that are not explicitly declared as group tasks must be done individually and handed in individually. Identical submissions are treated as plagiarism. Plagiarism may lead to loss of exam bonus points.

You can submit the solution to this task in English or German until Wednesday, May 29, at 23:59 via https://assignments.hci.uni-hannover.de. Create a pdf file that contains the text and images of your solution, name it "Assignment-07-<Firstname>-<Lastname>.pdf", and save it together with the exported project (Android Studio: File → Export → Export to Zip File) in a single zip file. Your submission must consist of a single zip file containing all necessary files. The name of the .zip file, as well as the names of the contained files, **must not contain any umlauts**. Therefore, please resolve umlauts in file names.

Android Phones: Please let us know if you need an Android phone for the assignments. We can lend a limited number of simple Android phones for the duration of the semester. These must only be used for the assignments.

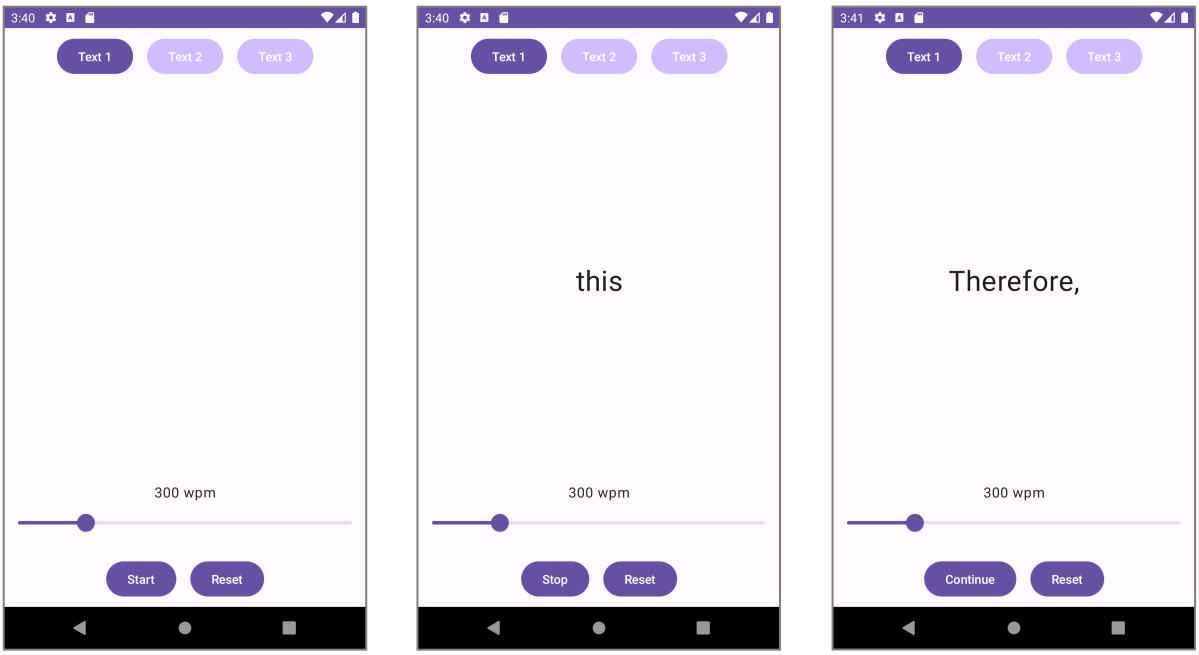
Exercise 1: RSVP App (15 points)

Rapid Serial Visual Presentation (RSVP) is a technique for displaying text on small displays. By displaying individual words at the same place on the display, unnecessary eye movements are avoided. Implement an Android app that allows you to test RSVP.

As a template use MI-Assignment-07-RSVPAppTemplate.zip. A screenshot of the app is given below. The buttons at the top let you select one of the texts. The slider controls the presentation rate between 200 wpm and 700 wpm. The two buttons at the bottom allow you to start, stop, continue, and reset the presentation.

The texts are stored as string resources with identifiers "text1", "text2", and "text3" in res/values/strings.xml. The user interface is defined in composable function Screen in the MainActivity. The function wpmToMs converts the given words-per-minute rate to milliseconds per word. At the top of the Screen function you find a few variables that represent the dynamic state of the user interface: `running` indicates whether or not the RSVP animation is currently active; `ticks` is a counter that is incremented according to the selected WPM rate; `textId` is the currently selected string resource; `words` is the list of strings based on the currently selected text; `word` is the current word (or the empty string if RSVP is not running); and `wpm` is the current output rate.

Your task will be to implement the user interface according to the descriptions given in the subtasks and according to the screenshots.



a) Create a Column layout for the three text selection buttons, the animated RSVP text, the WPM slider and text, and the control buttons. For the Column layout use Modifier.fillMaxWidth(). Implement the text selection as a row of buttons. Indicate the currently selected text by changing the color like this:
colors = ButtonDefaults.buttonColors(if (textId == textIds[i]) Purple40 else Purple80)
The color constants are defined in ui.theme.Color. Use a padding of 8.dp for the buttons. **(3 points)**

b) Implement the animated RSVP text as a Text (font size 32.sp) inside a Box (fillMaxWidth). You should see the RSVP animation running at 300 wpm. Make sure that the shown text corresponds to the selected text. **(2 points)**

c) Implement the WPM rate control using a Text and a Slider with a valueRange of 200f to 700f. Link the slider to the wpm state variable (value, onValueChange). You should now be able to control the RSVP speed. **(3 points)**

d) Implement the control buttons. These should behave as follows: At the start position of a text, the left button says "Start". Clicking it starts RSVP. When RSVP is active, it says "Stop". When stopped, but not at the start position, it says "Continue". Another click continues the animation. The right button always resets the current text to the start position. Test that you can properly start, pause, continue, and reset the animation. Moreover, the text should only play once, not continuously as in the template. When a text has played to the end, reset ticks to -1 and set running to false. **(3 points)**

e) To make the layout fill the screen, you may surround the animated RSVP text part with two spacers: `Spacer(Modifier.weight(1.0f))`. This fills the unused space so that the buttons are located at the top and bottom of the screen, respectively. Check that the layout stays consistent when you rotate the device. **(1 point)**

f) The playback should pause a little longer after sentences. For simplicity, treat a '.' as an end-of-sentence marker. Modify the LaunchedEffect block so that it pauses for 750 ms at the end of a sentence (i.e., when a word ends with a dot). Explain in a comment how this naive approach of determining the end of a sentence could fail. **(3 points)**

Exercise 2: RSVP Evaluation (12 points)

The goal of this exercise is to evaluate RSVP with yourself as the test person. The main task is to determine a comfortable speed in words per minute using the app developed in exercise 1.

Copy three articles of interest by title from any news page as content into your app. Avoid reading the articles before. The articles should have a length of at least 750 words each. Then read the articles using the app and adjust the playback speed while reading. The speed should be as fast as possible, but as slow as necessary. Read the articles one after the other, paying attention to the following aspects:

- **1st article:** Find the wpm rate at which you can still fully understand the content and are able to reproduce it.
- **2nd article:** Find the wpm rate at which you can still understand the whole content.
- **3rd article:** Find the wpm rate at which you are able to grasp the main points of the article.

a) Note the wpm rate for each condition. Include the used articles into your implementation, which is part of your submission (export zip file from Android Studio). **(6 points)**

b) Assume that you are a user interface designer who has to set an RSVP speed for your users, which your users cannot change. Based on your experience in (a) what speed would you set? Justify your choice. **(3 points)**

c) Based on your experience in (a) rate the following statements on a Likert scale of 1 to 7, where 1 means "no agreement" and 7 means "full agreement". Briefly justify your rating. We will discuss the results in the exercise session **(3 points)**:
   1. "I can grasp content faster with RSVP than with traditional reading methods."
   2. "Reading at a high speed with RSVP is more strenuous than reading by other common methods such as scrolling."
   3. "Reading with RSVP is a good choice for very small devices, like smartwatches."

Hint: Long news articles can be found on the Web pages of major publishers. Look out for multi-page articles, they almost always contain over 1000 words.