Exercise sheet 2

Interactive Systems - SoSe 24

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

All exercises that are not explicitly declared as group assignments must be completed individually and handed in individually. Identical submissions will be treated as plagiarism.

Submission by Monday, April 15 at 11:59 pm via https://assignments.hci.uni-hannover.de/SoSe2024/ISy. The submission must consist of a single zip file containing all necessary files. Please remove umlauts from file names.

Pack your solution (PDF file for tasks 1, 2 and 3, as well as your exported project as a zip file for task 4) into a zip file for uploading. Use the export functionality from IntelliJ, for example, to ensure that all required files are included in your submission.

Task 1: Mouse with scroll wheel (20 points)

Many computer mice have a so-called scroll wheel (see illustration). The scroll wheel can be rotated forwards and backwards and pressed down (like a button).

a) Specify the states and actions for the scroll wheel. Ignore states in which simul- taneous actions are possible (e.g. pressing the wheel and scrolling down at the same time).

b) Draw the state transition diagram for the scroll wheel. (Only for

the scroll wheel in each case, not for the entire mouse).

Task 2: Instrumental Interaction (18 points)

The concept of "Instrumental Interaction" was presented in the lecture. The corresponding paper can be found online[1]. Briefly describe the properties "degree of indirection", "degree of integration" and "degree of compatibility" for the following GUI elements:

a) Menu

b) Dialog box

c) "Handles" for graphical editing of objects in presentation or drawing programs

Task 3: GUI design for older people (11 points)

Buttons can be automatically laid out using the layout panels discussed. The distances between two buttons can be freely varied. You now have the task of adapting the size of the buttons and their spacing for a graphical interface on a touchscreen. The user group should consist of older people. To do this, read the linked paper[2] and answer the following questions:

a) What size of a single button would you choose if a fast response speed is required? What size of several buttons arranged next to each other would you choose if the display size is limited?

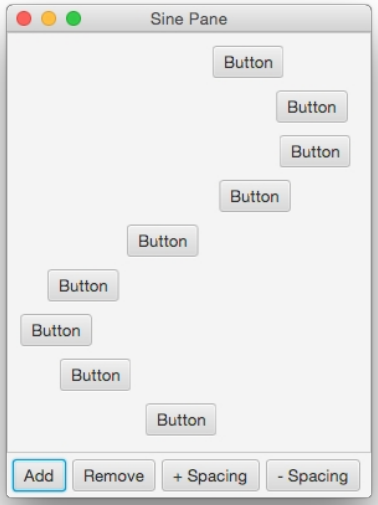
b) What spacing would you choose between the buttons if several buttons are to be arranged next to each other? What happens if the spacing is too large?

c) Give at least two examples of graphical user interfaces that are used by all age groups from the age of majority to the elderly. Which age group should mainly be considered in the design?

[1] Michel Beaudouin-Lafon. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. Proceedings of CHI 2000, pp. 446-453.
https://www.lri.fr/~mbl/papers/CHI2000/

2Xia Jin, Zhao n Plocher, Thomas n Kiff, Liana. (2007). Touch Screen User Interfaces for Older Adults: Button Size and Spacing. Conference: Universal Access in Human Computer Interaction. Coping with Diversity, 4th International Conference on Universal Access in Human-Computer Interaction, UAHCI 2007. 933-941.
https://link.springer.com/chapter/10.1007/978-3-540-73279-2_104

Task 4: Layout Panes (optional, 20 bonus points)



To solve this task (and some future tasks), the OpenJDK, JavaFX and IntelliJ IDEA are required as a development environment. Install these components on your computer. As a rule, you can also obtain the other required components via the IDE.

On Stud.IP you will find a template (SineLayout.zip) for IntelliJ IDEA. In the source code, the sections to be edited are marked with "@todo". Write your answers to the subtasks as comments in the source code.

The SinePane class for the layout of nodes is derived from javafx.scene.layout.Pane and is intended to implement the following layout strategy:

The nodes are scaled to their preferred size and arranged from top to bottom. The x-positions are selected so that the centers of the nodes lie on a sine curve that runs vertically. A selectable spacing can be set between the nodes. There is also padding at the edges of the SinePane. The nodes should not fall below their preferred size.

The size can be scaled down.

a) Briefly explain what happens when the Add button is clicked.

b) Briefly explain why the setSpacing method calls the requestLayout method. What happens if requestLayout is not used here?

c) Explain the method double waveX(double y). How is the x-position calculated from the y-position passed as a parameter?

d) Describe the functions of the computePrefHeight and computePrefWidth methods.

e) Describe in the comment of the method computeMaxWidth what would happen if max width and preferred width have the same value.

f) Complete the documentation of the functions maxPrefWidthChildren, sumPrefHeightChildren, sumSpacing.

g) Complete the implementation of layoutChildren. The nodes are first scaled to their preferred sizes using autosize and should now be positioned as shown in the figure above. (Any layout attributes of the nodes should be ignored).

The documentation for the Pane class (base class for layout panes) can be found at:
https://openjfx.io/javadoc/11/

Export your solution as a zip file from IntelliJ and hand it in.