

CMPT 381 Assignment 1: Widgets and Events in JavaFX

Due: Friday, January 20, 11:59pm

Overview

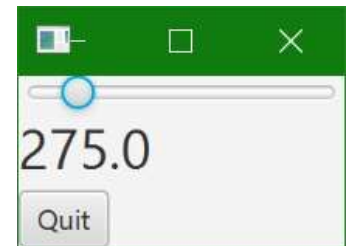
You will build a basic interface with JavaFX widgets, and will write event handlers to deal with user interaction on those widgets. You will also build a custom widget that uses JavaFX properties as its event-handling mechanism. This assignment will demonstrate your ability to use widgets and widget APIs, to handle basic user-interface events at the application level, and to implement simple 2D graphics for a custom widget.

Part 1: An interface with basic widgets and events

Build a simple interface with a Slider, a Label, and a Button (see picture at right).

Your system should include:

- A Slider that allows the user to set a value in the range 150.0 to 1000.0. The slider's handle should be set to an initial value of 275.0.
- A Label that displays the current value of the slider, formatted to have one decimal place (e.g., 275.0).
- A Button that displays the text "Quit". When the button is pressed, the program prints "Goodbye!" to the console, and then exits.
- The three widgets should be organized vertically using a VBox layout container
- Your interface should be built using code, not FXML.



Resources for part 1:

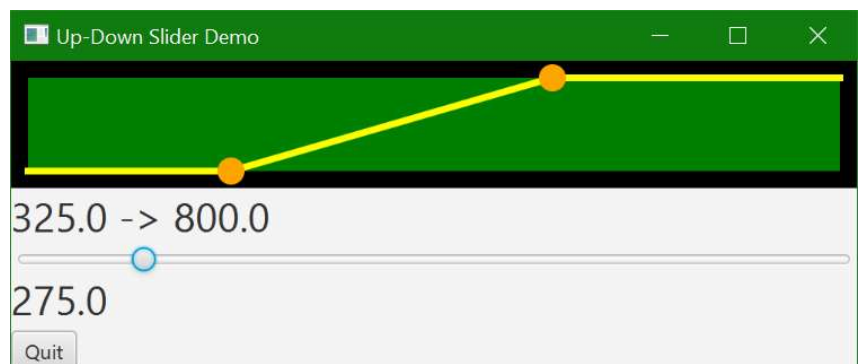
- Tutorial for the JavaFX Slider: https://docs.oracle.com/javafx/2/ui_controls/slider.htm
- Tutorial for the JavaFX VBox: <http://www.javafx-tutorials.com/tutorials/hboxvbox/>
- Tutorial for JavaFX event handling: <https://docs.oracle.com/javafx/2/events/jfxpub-events.htm> (note that this tutorial does not cover events for Slider widgets)

Part 2: Adding a custom widget and event handling

Build a custom widget called an UpDownSlider, which allows the user to set two values within a range (picture below). This widget could be used, for example, to set parameters for a transfer function.

Your implementation should meet the following requirements:

- Appearance. The widget shows a green rectangle on a black background. On the top and bottom edges of the green rectangle there are circular orange handles that indicate the upper and lower values of the slider. The handles turn red when the user clicks on them (i.e., when dragging). The widget draws a yellow line from the lower left of the



- green rectangle, to the lower handle, to the upper handle, and finally to the top right of the green rectangle.
- Size of the widget. The widget will have a fixed size of 500 pixels wide and 75 pixels tall. (note: the widget does not need to change size for this assignment)
- Interaction. The user can drag either of the two handles left and right. The upper handle cannot move further left than the lower handle, and cannot move past the top right of the green rectangle. The lower handle cannot move further right than the upper handle, and cannot move past the lower left of the green rectangle.
- Creation of the widget. The widget's constructor should take four values:
 - The minimum value of the slider (e.g., 0)
 - The maximum value of the slider (e.g., 1200)
 - The initial value of the lower handle (e.g., 325)
 - The initial value of the upper handle (e.g., 800)
- Implementation:
 - Your widget should use its own class (UpDownSlider) that inherits from Pane.
 - Your widget should use a JavaFX Canvas to draw its graphics
 - The canvas object should be added to a StackPane layout container (note that this container is different from any layout container in demo programs that use your UpDownSlider widget)
 - Set reasonable values for the amount that the green rectangle is inset inside the black background, the width of the yellow line, and the radius of the handles (note that the full circle of the handle should always be visible in the widget)
- Event handling. Similar to the way a JavaFX Slider works, add JavaFX Properties to your UpDownSlider implementation – one for the upper value, and one for the lower value. When the values of these properties change, a ChangeListener that has been attached to the property will receive notification of the change.
- Add your UpDownSlider to the interface developed in part 1 (place the UpDownSlider at the top of the window).
- Add a second Label to the interface (below the UpDownSlider); this label will show the two values of the UpDownSlider (separated by “->”, and formatted to have one decimal place)
- Add a ChangeListener to the interface to listen for changes to the properties of the UpDownSlider, and update the text of the label whenever the properties change.

Resources for part 2:

- Tutorial for JavaFX properties: <https://docs.oracle.com/javafx/2/binding/jfxpub-binding.htm> (note: the sections on binding are not required)
- Tutorial for the JavaFX Canvas: <https://docs.oracle.com/javafx/2/canvas/jfxpub-canvas.htm>

This assignment is to be done individually; each student will hand in an assignment.

What to hand in

- Java: a zip file of your NetBeans project folder. Note that you do not have to provide separate projects for part 1 and part 2 – you can just hand in the complete system built for part 2.
- A readme.txt file that indicates exactly what the marker needs to do to run your code. (Systems for 381 should never require the marker to install external libraries).

Where to hand in

Hand in your two files (one zip and one readme.txt) to the link on the course Moodle.

Evaluation

Marks will be given for producing a system that meets the requirements above, and compiles and runs without errors. Note that no late assignments will be allowed, and no extensions will be given, without medical reasons.