# CMPT 3035 Assignment 1: Widgets and Events in JavaFX

***Due: Tuesday, September 24, 11:55pm***

## Overview

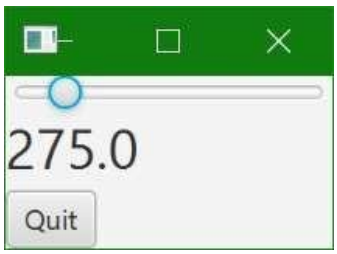
You will build a basic interface with JavaFX widgets, and will write event handlers to deal with user interaction on those widgets. You will also build a custom widget that uses JavaFX properties as its event-handling mechanism. This assignment will demonstrate your ability to use widgets and widget 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

### Resources for Part 1:

- Tutorial for the JavaFX Slider: https://docs.oracle.com/javafx/2/ui_controls/slider.htm
- Tutorial for the JavaFX VBox: https://www.tutorialspoint.com/javafx/layout_panes_vbox.htm
- 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)

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) and using a 24-point font.
- 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.

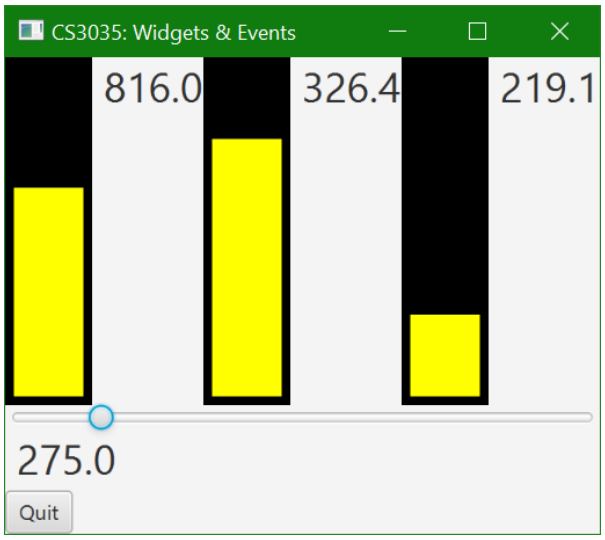
# Part 2: Adding a Custom Widget and Event Handling

## Resources for Part 2

- Tutorial for the JavaFX Canvas: https://docs.oracle.com/javafx/2/canvas/jfxpub-canvas.htm
- Tutorial for JavaFX properties: https://docs.oracle.com/javafx/2/binding/jfxpub-binding.htm

## Instructions

Build a custom widget called an BarControl, which allows the user to set a value within a bar-chart style vertical bar (three BarControl widgets are in the picture at right).




Your implementation should meet the following requirements:

## Appearance

- The widget shows a yellow rectangle on a black background (inset on all sides by 5 pixels).
- The yellow region always shows at least one pixel of yellow, even if the bar is at the minimum value (see middle widget at right).
- BarControl widgets have a fixed size of 200 pixels tall and 50 pixels wide (the widget does not need to change size for this assignment, even if the window changes size).

## Interaction

- When the user clicks or drags on the widget (in the yellow or black regions), the height of the bar will move to or follow the mouse cursor.
- The yellow bar never goes above or below the border region, regardless of where the mouse goes.
- The bar should consistently move under the mouse consistently.
- Creation of the widget. The widget's constructor should take three values:
    - The minimum value of the control (e.g., 100)
    - The maximum value of the control (e.g., 500)
    - The initial value of the bar (e.g., 250)

## Implementation

- Your widget should use its own class (BarControl) that inherits from Pane.
- Your widget should use a JavaFX Canvas to draw its graphics, or two

- *Hint*: The canvas object can be added to a StackPane layout container inside your widget (this can allow you to draw a background and a foreground separately on separate Canvas objects). Alternatively, see the canvas tutorial above for working with layers within a single Canvas object.
- *Hint*: Remember drawing starts relative to the top left corner (the origin: [0,0]), there are a number of ways to deal with this, but often subtracting drawn parts is an easy way to create the effect you want

### Event handling

- Similar to the way a JavaFX Slider works, add a JavaFX Property to your BarControl implementation. When the value of this property changes, a Listener that has been attached to the property will receive notification of the change.
- Demo program. Add three BarControl widgets (and Labels to show their values) to the interface developed in part 1 (see example above).
- Add the six new widgets (three BarControls and three Labels) into an HBox layout container.
- Add the HBox to the VBox you created earlier so that it appears above the components built for part 1.
- For BarControls 1 and 2, add Listeners to listen for changes to the property values of the BarControls, and update the text of the appropriate label whenever the properties change.
    - For BarControl 3, bind the textProperty of the third Label to the value of the third BarControl (such that it automatically updates itself without requiring a Listener). This will be a one-line statement.
        - See the tutorial above, first.
        - Also see: https://docs.oracle.com/javase/8/javafx/api/javafx/beans/binding/Bindings.html

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

# What/Where to Hand In

- Two jar files, one for each part of your assignment. Ensure that you have exported your source files in your JAR (recall that you can unzip and view the files in your jar file). See the previous assignment for instructions on how to create a JAR file.
- A readme.txt file that indicates exactly what the marker needs to do to run your code. (Systems for 3035 should never require the marker to install external libraries).
- Hand in your three files (two jar files, one for each part, and one readme.txt) to the D2L.

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 or pre-agreed arrangements.