JAVAFX CHARTS: IMPLEMENTATION OF MISSING FEATURES

G. Kruk, O. Alves, L. Molinari, CERN, Geneva, Switzerland

Introduction

JavaFX, the GUI toolkit included in the standard JDK, provides charting components with commonly used chart types, a simple API and wide customization possibilities via CSS. Nevertheless, while the offered functionality is easy to use and of high quality, it lacks a number of features that are crucial for scientific or controls GUIs. Examples are the possibility to zoom and pan the chart content, superposition of different plot types, data annotations, decorations or a logarithmic axis. The standard charts also show performance limitations when exposed to large data sets or high update rates.

Some of these features are present in the enhancements list in OpenJFX but up to Java 10 (including) there are no plans to address them by the JavaFX team.

JavaFX Chart Extensions

The central entity of the extension package is XYChartPane class, overlaying instances of XYChart (see Fig. 1) and managing nodes belonging to custom chart plugins. The main (base) chart must be specified at the construction of XYChartPane but the additional charts, drawn on top of each other, can be added and removed at any moment via exposed observable list of overlay charts.

NumericAxis and LogarithmicAxis

The JavaFX NumberAxis does not allow adding zoom capabilities to XYCharts therefore we implemented NumericAxis that provides the necessary enhancements. In addition, it can be further customized via dedicated properties:

- autoRangePadding – fraction of the range to be applied as padding on both sides of the axis.
- autoRangeRounding – determines if the automatically calculated range should be extended to the next major tick value.
- tickUnitSupplier – strategy (TickUnitSupplier) responsible for calculation of major tick units.

We implemented also LogarithmicAxis with a configurable logarithm base and minor tick count (see Fig. 3).

HeatMapChart

HeatMapChart is a specialised chart that uses colours to represent data values contained in a matrix. At CERN, it is typically used by applications displaying beam images like on Figure 4.

Chart Plugins

The XYChartPane provides an extension point via XYChartPlugin class that allows to interact with the content of displayed charts and to add graphical elements that are drawn on top of them (see Fig. 2).

At the moment the package provides the following plugins:

- ChartOverlay - allows adding to the chart area any Node at an arbitrary position.
- CrosshairIndicator – cross following the mouse cursor and displaying its coordinates.
- DataPointTooltip – tooltip label with coordinates of the hovered data point
- Zoomer – zooms the visible area to the drawn rectangle drawn with mouse cursor.
- Panner – allows dragging the visible chart area.
- XValueIndicator, YValueIndicator – line indicating specified X or Y value, with an optional text label.
- XRangelIndicator, YRangelIndicator – rectangle indicating range between specified X or Y values, with an optional text label.

Plotting Large Data Sets

The JavaFX charting package performs well with series containing up to a few thousands data points. However drawing series containing tens of thousands points takes it several seconds, efficiently blocking the FX thread and making the application unresponsive.

To address this issue, we developed DataReducingObservableList, a specialized implementation of the ObservableList interface, performing data reduction to the specified number of most significant points. It is a wrapper over a source list (containing all points) that triggers execution of the reduction algorithm on every change of the source data or X-axis range, exposing to the chart

Conclusion

The implemented extension fulfills the substantial set of features missing in the JavaFX charting package, enabling its usage for all controls applications. All added components follow JavaFX design principles and API style, making their usage simple and intuitive. The application of data reduction algorithm addresses the performance issues, allowing visualisation of large data sets flawlessly.