

XenonDataGrid: An OpenSource JavaFX™ Component

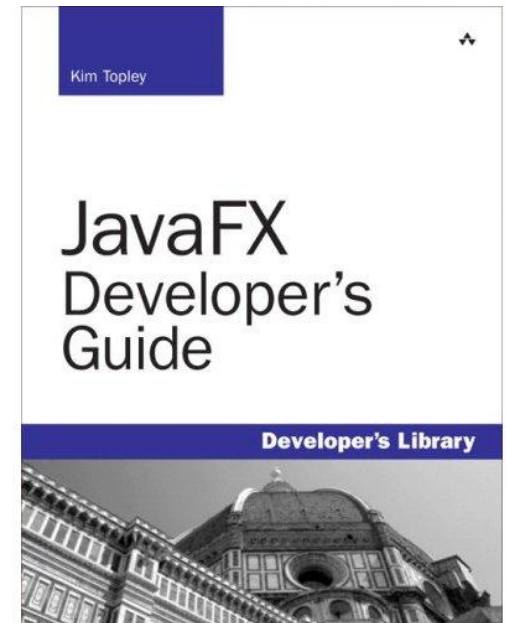
Peter Pilgrim

Java Champion,
Tech Lead, e-Business
Lloyds Banking Group



My View of the World

- Java Champion #91, SCEA 5
- Java User Group Leader
- JavaFX Blog Writer



http://jroller.com/peter_pilgrim



Overview Xenon Data Grid

Find and solve a known issue that causes others pain and grief or a certain grade of “Goodness, gracious me”





- **NATTABLE.org**
an huge influence on XDG with an excellent concept of dividing render space into regions and layers: corner, row headers, column headers and body ;-)





- **Swing JTable**

an incredibly successful table component from 1999 with limitations based on direct-mode API graphic. No easy scene graph support possible.



BLENDED
FOR THE
JAVA COMMUNITY



DEMO

Xenon Data Grid Examples



Seven Essential Recommendations

➤ 1. Location aware to the developer



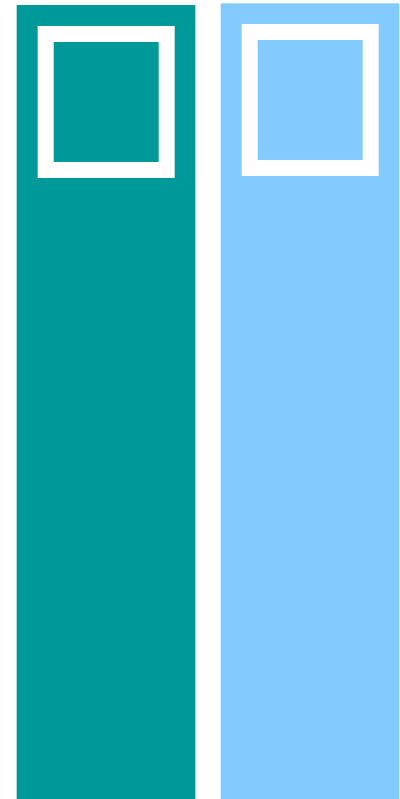
We must be
able to
determine
Parent (x,y)?



(Cell X , Cell Y), (Parent X, Parent Y)

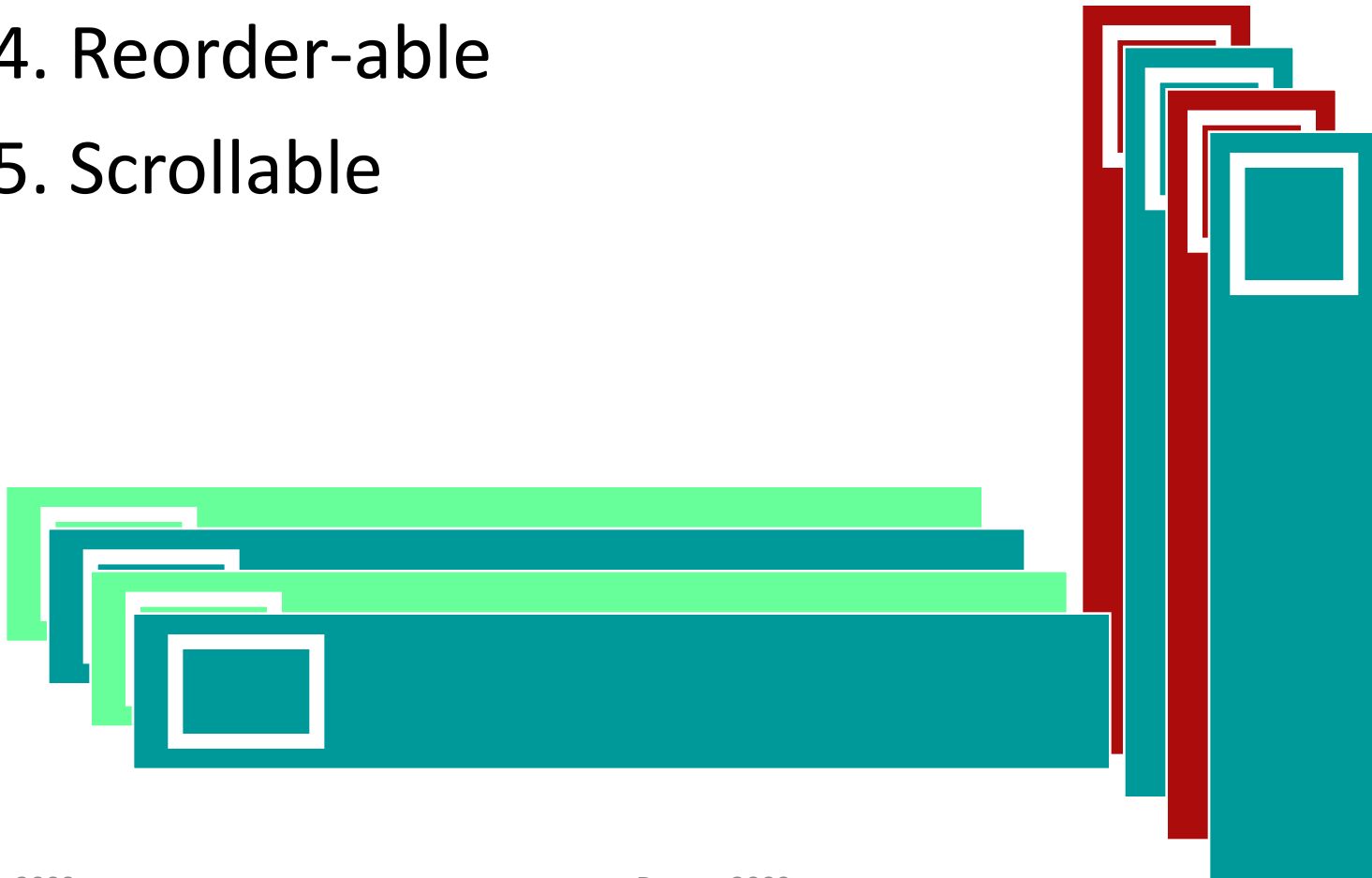
Seven Essential Recommendations (contd)

- 2. Row Headers
- 3. Column Headers



Seven Essential Recommendations (contd)

- 4. Reorder-able
- 5. Scrollable



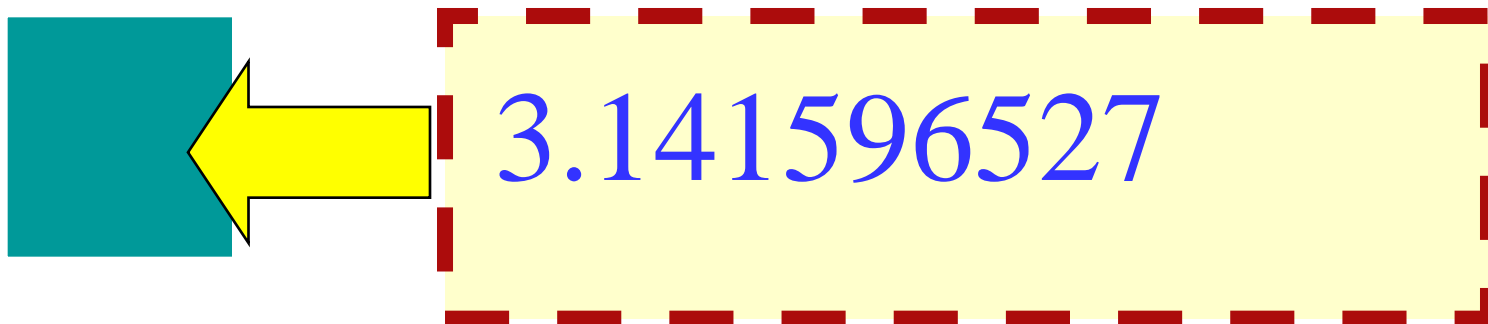
Seven Essential Recommendations (contd)

- 6. Mapping model data to render cells:
- Given a table model with (D_x, D_y) on the table UI component T , we should be able to find a Renderable Cell UI component R with (R_x, R_y)



Seven Essential Recommendations (contd)

- 7. Bidirectional
- Users tend to enter bulk data with a table and then submit or save. A good design allows the underlying data model to reflect the Renderer model.



BLENDED
FOR THE
JAVA COMMUNITY



DEMO

Final Xenon Data Grid Examples

BLENDED
FOR THE
JAVA
COMMUNITY

The Strategy Pattern is “King”

- Plug-in layout strategy allow new forms of grid cell physically
- Plug-in cell selection strategy for mouse driven event
- Plug-in keyboard navigation strategy
- (Soon) plug-in type conversion strategy

MultiLayer, Layers and Renderer Cells

- MultiLayer contains named Layers
- Layers control “regions” of cells
- “Renderer cells” name given to JavaFX type that wrap a scene graph node
- “Painter cells” name given to cell that render to an `javafx.reflect.FXType` (e.g. String, Integer or Float)

Table Model

- GetData(row, column): Object
- SetData(row, column, value: Object)
- IsEditable(row, column): Boolean
- getType(row, column): FXType
- **Extended Table Model** available for storing data cells that include *extra* attributes

Code 1

```
XenonDataGrid {  
    masterLayer: MultiGridLayer {  
        cornerLayer: GridLayer { /*..*/ }  
        rowHeaderLayer: GridLayer { /*..*/ }  
        columnHeaderLayer: GridLayer { /*..*/ }  
        bodyLayer: GridLayer { /*..*/ }  
    }  
}
```

Code 2

```

XenonDataGrid {
  masterLayer: MultiGridLayer {
    /* ...*/
    bodyLayer: GridLayer {
      cellFactory: function(row,column):
GridCellRenderer { XenonGridCellRenderer{
      backgroundFill: Color.LIGHTGREEN } }
      tableModel: TableModel {
        rows: 100 column 10 data: data }
    }
  }
}

```

Xenon Data Grid Summary #1

- Designed for flexibility and efficiency
- **Table Model**
- Layout, Selection, Navigation **plug-in strategies**
- **Layers, Regions** and **Multi-Layer** (Corner, Column Headers, Row Header and Body)
- Just added **Cell Factory** in GridLayer.fx

Xenon Data Grid Summary #2

- Open source part of the Nelson FX
- XDG is an **upper Alpha-ware** state
- Expect a Nelson FX M4 release soon
- Expect more **Strategies** and **Adaptor** classes to make your programming lives easier
- Opine, Contribute and Accelerate

Thanks For Your Attention!

- Search Vimeo for Screencast on the XDG
- Audioboo
http://audioboo.fm/profile/peter_pilgrim
- Missing JavaFX table component

http://jroller.com/peter_pilgrim

