EditEngine on the Overlay

It's a long way to Tipperary

CIB SOFTWARE GMBH
LIBREOFFICE CONFERENCE ROME
OCTOBER 11TH, 2017

Armin.Le.Grand@cib.de

Agenda

> Background/Where do we come from?

> First big change – CTL/RTL/TTB

> Second big change - Primitives

> Remaining Problems

> Suddenly – a performance Bug (OMG)

> Third big change - Overlay

> Future: Where could we go…?

> Did you know about TextOnCurve…?

Background – Where do we come from?

> Where is it used?

- For interactive Editing: DrawObject Text, MultiLineEdit, TableCell (Calc), all Apps

- For ModelDataChange in core: for all kinds of Text manipulation

> Why does it exist?

- Why not use Writer as TextEdit plugin – early tries, HelpSystem, MemLoad, Performance

> Basic technic for rendering:

- A vcl::Window (!OutDev) plugin/proxy, all msg passing methods overloaded, direct rendering and layouting

- Works on OutlinerParaObject as data object, operator==, Redlining, Text as String per Paragraph, ParaPortions, ItemSets

First big change – CTL/RTL/TTB

> Massive adoptions, need to write Right-ToLeft, Anchor Top-Right

> TopToBottom: Similar, but another coordinate System again

> All changes executed through multiple complex layers (also Mouse, Selection, …)

> Result is working CTL/RTL/TTB

> Result is also a non-obvious mix of CoordinateSystem

> The question to ask yourself when you have to touch it: "In which coordinate system am I right now?"

> Would have been better to use linear Transformations/homogen Matrices

Second big change - Primitives

> EditMode: Keep direct rendering

> Layout/Core mode:

   - Decompose to TextPrimitives (Simple/Complex)

   - Sequence of Primitves, deep Structure (Line/Paragraph, ...)

   - Reusable, buffered decompose due to operator== in OutlinerParaObject (before: every paint of every text in every object → setup Outliner, let it layout and paint)

   - Layouted on Top-Left Point(0.0, 0.0)

   - Result is Transformable, so it could follow ObjectTransformations the first time ever (Mirroring, Rotation, Shear, ...)

Remaining Problems

> Acceptable Solution, working for quite some time unchanged

> 'Fake' grow/shrink of edited object, may lead to problems

> Still based on direct paint, vcl::Window, OutputDevice

> PrimitiveDecomposer and DirectRendering in parallel, potential differences (see URLs in TextEdit)

> 'Exceptional' State in Apps' EditViews, Overlay disabled (Window, direct), hack to paint at least to OutputDevice

> Heavy EditMode Setup/Shutdown (ever looked at BeginTextEdit...?)

> XOR-Paints for Selections

> The last part in Draw/Impress drawing directly to OutputDevice

Suddenly – a performance Bug (OMG)

> In direct paint/edit mode, EditEngine uses a 'guessed' Color (´handed over at setup) to paint areas getting visible

> This was removed – sure by good intention, but with triggering complete Repaint of the Background when changes happened (every character)

> Usually fast enough, but not with expensive Backgrounds. Remember: Overlay functionality limited in that mode...

> What to do...

  – Bring back the 'guessed' color..?

  – Or ... take the chance, do it right, go one step further?

Third big change - Overlay

> Existing Decompose is very stable

> We have working Overlay in all Applications

> Implement an OverlayObject, connect to EditEngine, do not render directly anymore

> Get Primitive Decomposition for visualization (in Overlay)

> At long last: transparent Selection visualization in user-defined color – as done in Writer for some time (on Overlay, too)

> Get away from that 'exceptional' paint mode (not completely – MultiLineEdit, Cells)

> Performance Problem is gone, no more Background repaint needed, completely System-Independent

Future: Where could we go…?

> Rework/Remove RTL/CTL/TTB, just use Transformations

> Simpler adaption to run embedded – vcl::Window binding, proxy functionality

> Line/Paragraph buffering of Primitives in OutlinerParaObject

> Use in MultiLineEdit, TableCells → get rid of direct paint comletely

> Support InterActionHandles and TextEdit at the same time

> Support TextEdit in transformed mode – do not snap back to centered, non-transformed – do real WYSIWIG

Did you know about TextOnCurve

> Combined with EditMode and Transformation, real WYSIWIG would be nice to have…