Subject: Re: geom package?
Posted by rylek on Mon, 28 May 2007 21:29:44 GMT
View Forum Message <> Reply to Message

Hello Gertwin!

Sorry for the missing documentation. In fact I started to write the Geom-related packages first as a supplement to Mirek's basics in Core (which in fact comprise little more than a 2D floating-point vector and rectangle) for use in my own applications. Over the time, some of the concepts turn out to be more viable and reusable, some do not.

For practical reasons it would be very hard for us to make a clean cut between "nice" and "dirty" stuff; at the time being it might be cleanest if I agreed with Mirek on making a new set of packages which would contain only the 'clean' stuff, whereas the 'dirty' stuff would be moved into a kind of TGeom and perhaps in time removed from the uppsrc tree altogether.

The trouble is that Mirek almost never uses floating point geometry; he simply doesn't often write the kind of software needing this and I'm not sure how eager he would be to assist in filtering the portions of the above packages for 'common use' when he almost seems to take pride in avoiding them.

To make the long story short: take the geometry-related packages with a grain of salt. As you've already noticed, so far there's no documentation to them; I can gradually start creating small example programs demonstrating various aspects of these modules but it's going to take time.

The individual modules have the following purposes:

Geom - various 2D and 3D analytic geometry calculations, linear equation set solver, Delaunay triangulator, generic polygon rasterizer

Geom/Coords - utilities for geographic coordinate transform calculations; several coordinate systems have already been implemented, unfortunately (for evident reasons) mostly systems used in the middle Europe.

Geom/Draw - linear & bilinear image warping (imgtran), floating-point based drawing tools (plotter), generic patterned line drawing (pathDraw)

Geom/Ctrl - Ctrl-related stuff, most important is PlotterCtrl which is a zoomable view to a physical floating-point-based world

The most important classes to start with (for the purposes you mentioned):

1) Plotter - a Draw combined with a transform matrix (2x3), clipping regions and some other info. Used to transform between logical and device coordinate space and to draw stuff based on logical floating-point coordinates.

2) PathTool, AreaTool, MarkTool, TextTool - tools for drawing lines, areas, markers and text based on logical floating-point coordinates (use a Plotter to transform the logical coordinates into

device coordinate space).

3) PlotterCtrl - basically a zoomable view into a floating-point image, map or chart. At the very least, you just override the Plot method (a Plotter-based floating-point equivalent of Ctrl::Paint), set up the logical extent (SetExtent) and you're all set. You can also set the direction of positive X- and Y- axis, set up fixed or free aspect ratio (free aspect ratio is used for views with heterogeneous X- and Y- coordinates).

4) PlotterDragDrop - by deriving from this class you can define various drag & drop modes for the PlotterCtrl view. The Geom/Ctrl package implements three standard drag & drop modes - the zoom in mode, zoom out mode and pan mode. In a similar way, you can implement arbitrary additional editation, selection or drawing modes.

Regards

Tomas