

---

Subject: Re: capsulated Scatter Vectors

Posted by [kohait00](#) on Wed, 12 May 2010 21:07:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Quote:

the old code from user point of view has less lines and follows more U++ style.

I totally agree on that one. my changes have neither performance improvements nor beauty changes. the point its all about is, to define what is the core of the Scatter control. as of my point of view it is the data, not the API (which was to be read fine but was hiding away the data too much and IMHO bloated the API a bit).

andrejs goal was to have a Scatter control with a neat API to post stuff to and forget about it. my goal is to be able to access the data of the graphs in tried and true containers' manner (Vector..) and thus public and to see the Scatter just as a wrap over the data, which, upon triggering of Refresh()/Paint() draws a representation of it. so its basicly a question of design goals.

the current solution could be improved in readability by changing the PData members' names to something more usefull, which i havent done so far (wanting to keep the visibility where the members came from and to make compare in svn easy not to change too much at once). also, the "more lines" stuff could be done better, by returning PData & instead of Scatter & AddSeries.. i will provide a cleaned up version to sea what i mean.

from code point of view, my current code leaves the control with almost only the code needed to draw the data. as api see the Vector<PData> API which is cool as well. another benefit is that functional plots are handled just the way normal plots are.. making code clearer again.

i hope this came clear..it was not my goal to enhance the api but simplify it by leaving stuff accessible instead of hiding it behind api. (this is what i learned as well that sometimes its better not to hide)

Quote:

So it would be great to let linking external data sets from:

- C arrays
- ArrayCtrl columns
- GridCtrl columns

this is a great idea.. could be done in some Additional AddSeries() functions..

BTW: OpenGL, i am starting to like it.. the only thing is to create the vertices from data, set the viewpoint and have the stuff rotate depending on mouse position as seen on a surface of a bounding sphere.