
Subject: Re: New graph packages
Posted by [Didier](#) on Sun, 04 Mar 2012 12:48:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Koldo,

I think DataSource used in ScatterDraw should go template:

The only reason for this is to enable high performance if needed:

```
class DataSource {
public:
    typedef double (DataSource::*Getdatafun)(int id);

    DataSource() : isParam(false) {}
    virtual ~DataSource() {};
    virtual double z(int id) {return Null;};
    virtual double y(int id) {return Null;};
    virtual double x(int id) {return Null;};
    virtual int GetCount() {return Null;};
    bool IsParam() {return isParam;};

    virtual double MinX() {return Min(&DataSource::x);}
    virtual double MinY() {return Min(&DataSource::y);}
    virtual double MinZ() {return Min(&DataSource::z);}

    virtual double MaxX() {return Max(&DataSource::x);}
    virtual double MaxY() {return Max(&DataSource::y);}
    virtual double MaxZ() {return Max(&DataSource::z);}

    virtual double AvgX() {return Avg(&DataSource::x);}
    virtual double AvgY() {return Avg(&DataSource::y);}
    virtual double AvgZ() {return Avg(&DataSource::z);}
```

Class DataSource is a pure virtual class

==> so all method calls need to go by the virtual table ==> poor performance

This is specially true for the x,y,z methods which get called for each point drawn.

If Scatter Draw had was defined the following way:

```
template<class DATASOURCE = DataSource>
class ScatterDraw {
public:
    ...
```

The following high performance trivial class could be used instead:

```

template<int NBPOINTS>
class DataSource {
private:
    double _x[NBPOINTS];
    double _y[NBPOINTS];
    double _z[NBPOINTS];
public:

    DataSource() {}
    inline double z(int id) {return _x[id];}
    inline double y(int id) {return _y[id];}
    inline double x(int id) {return _z[id];}
    inline int GetCount() {return NBPOINTS;}
    inline bool IsParam() {return false;}

    inline double MinX() {return ....;}
    inline double MinY() {return ....;}
    inline double MinZ() {return ....;}

    inline double MaxX() {return ....;}
    inline double MaxY() {return ....;}
    inline double MaxZ() {return ....;}

    inline double AvgX() {return ....;}
    inline double AvgY() {return ....;}
    inline double AvgZ() {return ....;}
};

```

The drawback to this is that most of ScatterDraw should go in the header
 But after all that is not such a big issue.

NB: I know that most of the performance is due to drawing speed but all enhancements are
 welcome I think

What do you think ?