Subject: SQLCommander, MySQL and other questions...[NEEDS SPLITTING...]
Posted by pivica on Mon, 21 Nov 2005 19:54:55 GMT
View Forum Message <> Reply to Message

1. I tried today SQLCommander. I wonted to connect to MySQL but when I compiled it and started there wasn't MySQL support. Looked the code and find next

```
#ifndef flagNOMYSQL
  #define HAVE_MYSQL
#endif
```

but I wasn't define anywhere flagNOMYSQL nor anywhere in the code I found a place where that flag is defined. How to compile SQLCommander with MySQL support?

2. Also doing digging in uppsrc (amassing what already exist there:) and find intersting package Geom with some subpackages. But I can't figure it out what exactly it is for (there are 2D and 3D vector support, and lots of stuff) so could anyone give some explanation. Also I am very interesting in 2D and 3D vector geometry - is U++ have already some ctrl for this (besaide OpenGL ctrl)?

3. Started to write topic for refence\CalbackArgTarget and next line is not completely clear to me

```
CallbackArgTarget<int> result;
```

Now for this to work type has to be Null able. Thats probably mean that int is actually not a C++ keyword or maybe it is??; is it some kind of class? Searched uppsrc and found next

```
NTL_MOVEABLE(int)
```

hmmm, don't know what this is exactly - are now int wrapped in some kind of Movable and Null type?

4. I think that I found little error in Topic::src::Callbacks::CallbackArgTarget::IsNullInstance(). It says

```
Return value...True if there is not Null in output value.
```

maybe it should says without 'not'

```
Return value...True if there is Null in output value.
```

That's it for now.

Ivica

Subject: Re: Couple of questions
Posted by mirek on Mon, 21 Nov 2005 20:45:57 GMT
View Forum Message <> Reply to Message

I hope Tom will answer 1 and 2.

3: Null for int is defined as INT_MIN. Maybe somewhat weird, but simple and works actually very well. (similarly, for double Null is defined as very large negative number like 1.0E-300). NTL_MOVEABLE is macro that marks a type Moveable (can be stored in Vector) - note that you cannot use Moveable template for 'int'.

Mirek

Subject: Re: Couple of questions
Posted by pivica on Mon, 21 Nov 2005 22:09:17 GMT
View Forum Message <> Reply to Message

Quote:note that you cannot use Moveable template for 'int'.

I stil don't understand what NTL_MOVEABLE(int) is. If int can't be use as a Moveable type, what then this line in uppsrc/Core/Topt.h on line 231 means - that int is Moveable or not?

Ivica

Subject: Re: Couple of questions
Posted by mirek on Mon, 21 Nov 2005 22:50:32 GMT
View Forum Message <> Reply to Message

Well, fundamental types are moveable by detail. Anyway, moveable types have to be marked somehow and this has to be done for 'int' as well.

You can mark newly created classes using

class Foo : Moveable<Foo>

but this is of course not possible for built-in types. Thus NTL_MOVEABLE macro that makes the same thing for type that already exists.

Actually, this is even documented here:

http://upp.sourceforge.net/srcdoc$Core$Moveable$en-us.html

it is one of the oldest documentation pieces in existence (I would no speak about std::string today...)

Subject: Re: Couple of questions
Posted by rylek on Thu, 24 Nov 2005 10:54:56 GMT
View Forum Message <> Reply to Message

To use SQLCommander with MySQL, you must select the appropriate main package configuration. To build against the mysqlclient library, it is necessary to use MT mode.

As concerns Geom, this is more or less a randomly selected collection of routines I needed over the years to implement analytic and computational geometry, mostly 2D and a little 3D stuff. For instance:

*) a bunch of simple analytic geometry functions over Pointf & Rectf (vector length, rotation around a given point, point on quadratic Bezier curve, polar-carthesian system conversion, vector product, dot product, rectangle diagonal)

*) 3x2 Matrixf object used for 2D affine transforms

*) distance calculations (point from line, point from circular arc, point from circle)

*) rectangular crossing calculations for line, circle, circular arc

*) polygon intersection check

*) line clipping

*) polygon splitting algorithm. This routine converts a potentially complex polygon with holes into a series of simple polygons without holes which can be easily rasterized or drawn.

*) linear matrix solver using Gauss elimination with full pivotage

*) a simple (quadratic time) Delaunay triangulation generator (volunteers are very welcome to implement a linear-logarithmic algorithm)

*) Pointf3, Plane3, Matrix3 and Box3 are used to implement simple 3D analytic geometry routines. A Camera object can be used to calculate projective projection matrix based on intuitive parameters (camera location, viewing direction, up direction, viewing angle). A simple OpenGL example (examples/OpenGL) demonstrates the use of Camera in the context of a very simple 3D terrain viewer application.

Regards

Tomas

Subject: Re: Couple of questions
Posted by pivica on Thu, 24 Nov 2005 16:42:55 GMT
View Forum Message <> Reply to Message

Thanks for answering.

Quote:As concerns Geom, this is more or less a randomly selected collection of routines I needed over the years to implement analytic and computational geometry, mostly 2D and a little 3D stuff. For instance:

It's seem to me that it's possible, with a help of Geom package, to make some kind of Ctrl for drawing and viewing 2D vector objects. Am I right?

Ivica

Subject: Re: Couple of questions
Posted by rylek on Fri, 25 Nov 2005 09:53:25 GMT
View Forum Message <> Reply to Message

Most of the functions by themselves are purely mathematical calculations. It is up to you to choose when to call them and what to do with their results. Of course, as the functions are mostly concerned with 2D & 3D analytic geometry and transforms, they can be used very easily to implement graphics.

The subpackage Geom/Draw is designed to do this. A class named Plotter combines a Draw object with a coordinate transform (defined by a Matrixf) and so it can be used to implement drawing in a logical coordinate space while generating the visual output in another coordinate system bound to the logical coordinate system by an affine transform (a combination of shifting, rotating, scaling and skewing). There are specific classes named PathTool, AreaTool, MarkTool and TextTool which use Plotter to draw the respective objects (defined by their logical coordinates) on a Plotter object.

The subpackage Geom/Ctrl goes even further to define PlotterCtrl, a generic editor object used to display, zoom & pan and edit an arbitrary geometric object (no matter whether vector- or raster-oriented) with a certain logical extent within a simple Ctrl view. The control implements everything concerning scaling and panning, all you have to do to get something working is to derive from the class, override the virtual method Plot(Plotter&) and use it to draw your graphics. You also have to SetExtent to tell the control how big your logical object is.

The PlotterCtrl class also implements a very generic drag & drop mechanism, enabling you to easily define your own custom drag & drop algorithms. In my various commercial applications (cited on our homepage), mainly WinZPV and HydroCheck, this is used to edit charts, to visually define intervals for interpolation, to draw cross sections etc.

A simple example, examples/OpenGL, demonstrates the use of Matrixf3 and Camera to draw 3D vector graphics. This is a very rudimentary 3D terrain viewer, which draws a shaded image of the terrain surface (using OpenGL's Z buffer) and over it the linear mesh of terrain polygon edges (using Draw::DrawLine). This demonstrates correctness and match of the two different computation mechanisms.

Subject: Re: Couple of questions
Posted by pivica on Fri, 25 Nov 2005 16:05:32 GMT
View Forum Message <> Reply to Message

Rylek many thanks for your explanation of Geom package, which by the way looking very good. Expect more questions from me about this topic when I cover all the basics of U++:)

I am still amaze what you two guys created. U++ really rocks. Just that documentation problem, eh;))

Subject: Re: Couple of questions and PLOTTING
Posted by amea on Mon, 05 Dec 2005 13:22:40 GMT
View Forum Message <> Reply to Message

Hi,

Any hints on how to implement basic plotting/charting using these classes? I have been looking around for some tutorials on how to create basic line graphs or charts from a data set. I know there are libraries available, but they interface to python etc.. I also know of GD and imagemagick. But I don't know if they're any use from UPP?

Is there a buried class to control things like axes and labeling?

Thanks.

Subject: Re: Couple of questions and PLOTTING
Posted by fudadmin on Mon, 05 Dec 2005 14:02:58 GMT
View Forum Message <> Reply to Message

amea wrote on Mon, 05 December 2005 08:22Hi,

Any hints on how to implement basic plotting/charting using these classes? I have been looking around for some tutorials on how to create basic line graphs or charts from a data set. I know there are libraries available, but they interface to python etc.. I also know of GD and imagemagick. But I don't know if they're any use from UPP?

Is there a buried class to control things like axes and labeling?

Thanks.

I am very much interested in this as well. I'm working on a similar project. Would you accept a co-operation?
But, of course, we need much help from Myrek and Tom...

Subject: Re: Couple of questions and PLOTTING
Posted by amea on Mon, 05 Dec 2005 22:30:12 GMT
View Forum Message <> Reply to Message

Yep,

But I give advance warning that my C++ skills are very weak at present. Taught it many years ago, but have not used for any real work!

The project I'm working on requires pulling data from a database (SQLite, at present, which I'm setting up at the moment) and doing some basic statistical analysis and plotting. I have not finalised the statistics yet (once again, revising), but there will be some option to choose from a small selection of tests based on DB queries. Data entry and reporting are there as usual!

I know there are great OSS tools for doing this stuff (like the R statistics project) but they are overkill and the application must be self-contained, have a small footprint (hence no .NET, Java) and be relatively 'cheap' for the end users. I explored Python (Reportlab, matplotlib, great database support, wxWidgets (messy) --- poor deployment options, and relatively large footprint). I also seriously considered a desktop browser app, but don't think the users would accept it! Looked at Ruby On Rails and even Tcl with its server and Tcl plugin --- very cool deployment mechanism - metakit and starkits! But development of language and libraries a bit slow! Main reason for above was wish to create cross-platform app.

Anyway, sorry about that, it's been a frustrating month!

I only discovered the UPP project a few days ago, I only wish I'd found it a month and a half ago! It really is one of the best OSS projects I've seen in some time. The GUI support and the ide are superb.

BTW looking at gnuplot, it's at version 4.00, and it's come along way. Very good core system and interfaces to c/c++!

amea

Subject: Re: Couple of questions and PLOTTING
Posted by fudadmin on Thu, 08 Dec 2005 01:34:56 GMT
View Forum Message <> Reply to Message

amea wrote on Mon, 05 December 2005 17:30
Yep,
[...]
But I give advance warning that my C++ skills are very weak at present. Taught it many years ago, but have not used for any real work!
[...]

amea, I've sent you a private message. Don't worry about your C++ skills. I'm ready to help you.

I'd like to know your intentions.