Subject: Drawing / Painter Difference,Usage,Manual
Posted by kohait00 on Thu, 24 Jun 2010 22:10:59 GMT
View Forum Message <> Reply to Message

hi all,

its quite a while now i have read about Drawing redesign down in Upp core. whats all about it? there is almost no information available about Painter compared to Draw / Drawing, again the big picture is missing..

the Draw tutorial does only explain the basics, but some expert info would be great as well . The class reference does not talk much about Painter..

There is this PaintingExamples package, with a checkbox "Paint through Painting".. what is it for. is Painter an alternative way, accelerated, favorised to Draw?

concerning the whole Painter / Drawing stuff, basic questions are (including some things i believe to have found out a bit)

1) What is Draw (interface class for performing all kind of drawing)

2) What is Drawing (a target for Draw operations for vector images received and applied by DrawingDraw??)

3) What is Image (a target for Draw operations for raster images received and applied by ImageDraw??, readonly)

4) What is ImageBuffer (a writeable Image ??)

5) What is SystemDraw (an system dependant implementation of Draw interface, related to Image / ImageBuffer, which is the base for drawing of a TopWindow

6) What is a TopWindow (a basic system dependant window unit handling the message queue for peripherial events, and managing one/the static SystemDraw??)

7) What is a Painter then and how does it fit in all that? i've seen it deriving from Draw, why?

8) What is a BufferPainter, a ImagePainter, DrawPainter, what is the difference to ImageDraw, DrawingDraw, (ImagePainter is a BufferPainter and has got ImageBuffer, DrawPainter is also an ImagePainter and has got a Draw, BufferPainter has ImageBuffer in it etc.. wooosh..a diagram is best for that..with some short descriptions what ist what for.

etc..

i know lots of questions, but this is to show where the questions of a semi experienced Upp user go wnating to dig deeper in Upp.

some ideas / the big picture would make life easier for

thanks in advance..

PS: i know documentation is hard work, coding is more fun, but great work needs some good doku..

Subject: Re: Drawing / Painter Difference,Usage,Manual
Posted by mirek on Sun, 27 Jun 2010 07:32:56 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Thu, 24 June 2010 18:10
1) What is Draw (interface class for performing all kind of drawing)

Yes. Interface class for performing simple drawing - good enough to paint all widgets. Fine enough to paint most of graphics. Does not contain advanced operations. E.g. Draw is not good enough to paint PDF or SVG.

In X11 and Win32, when Draw represents window surface, drawing operations are passed directly to host OS.

Quote:
2) What is Drawing (a target for Draw operations for vector images received and applied by DrawingDraw??)

Represents recorded sequence of Draw operations. Similar to windows metafile. DrawingDraw is Draw that creates Drawing. Draw has "DrawDrawing" operation, which is also able to preform rescaling.

Quote:
3) What is Image (a target for Draw operations for raster images received and applied by ImageDraw??, readonly)

Image is raster bitmap... a picture you can draw.

Quote:
4) What is ImageBuffer (a writeable Image ??)

Basically, yes, as Image does not have any mutating operations (except assignment).

Quote:
5) What is SystemDraw (an system dependant implementation of Draw interface, related to Image / ImageBuffer, which is the base for drawing of a TopWindow

SystemDraw is X11/Win32 implementation of Draw. Translates Draw interface to calls to GDI.

Quote:
6) What is a TopWindow (a basic system dependant window unit handling the message queue for peripherial events, and managing one/the static SystemDraw??)

TopWindow represents host platform top-level window (with window title, all platform dependent decorations etc...).

SystemDraw has to be managed in Ctrl, because there are also PopUp top-level windows...

Quote:
7) What is a Painter then and how does it fit in all that? i've seen it deriving from Draw, why?

Painter is high-level drawing system interface, implemented as BufferPainter in software. It has enough capabilities to paint SVG/PDF.

It derives from Draw basically because it can... I mean, being on higher level, it can implement simple Draw interface too. That has advantage that routines capable of drawing Draw can use Painter as target too.

[quote]
8) What is a BufferPainter
[/qute]

Implements Painter interface to render in ImageBuffer.

Quote:
, a ImagePainter,

Trivial helper class that uses ImageBuffer to produce Image as Painter rendering output.

Quote:
DrawPainter,

Another trivial class that outputs Painter rendering to Draw; to be used in Paint.

Quote:
what is the difference to ImageDraw, DrawingDraw,

ImageDraw nor DrawingDraw have Painter high-level interfaces.

Quote:
PS: i know documentation is hard work, coding is more fun, but great work needs some good doku..

That is why I hope you will provide some too

Mirek

Subject: Re: Drawing / Painter Difference,Usage,Manual
Posted by kohait00 on Sun, 27 Jun 2010 07:58:17 GMT
View Forum Message <> Reply to Message

hey mirek, thanks a lot..

this clarifies quite well the sort of questions i had. the rest can be done by digging in code.

my problem is actually, that to understand code, i need at least to understand / know the intention, big model. later infos from code diggin will fit in that.

though Upp is nice to read (i really like the code style), there are some places that are not thaaaaat easy to grasp, painter / stuff and chameleon beeing some examples for it.

but this more and more gets better.

Quote:
That is why I hope you will provide some too (doku)

how do you expect me to provide doku infos? (which format) or is that as well covered in the "how to contribute"? (i'll take a look again..)

Subject: Re: Drawing / Painter Difference,Usage,Manual
Posted by mirek on Sun, 27 Jun 2010 10:53:50 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Sun, 27 June 2010 03:58hey mirek, thanks a lot..

this clarifies quite well the sort of questions i had. the rest can be done by digging in code.

my problem is actually, that to understand code, i need at least to understand / know the intention, big model. later infos from code diggin will fit in that.

though Upp is nice to read (i really like the code style), there are some places that are not thaaaaat easy to grasp, painter / stuff and chameleon beeing some examples for it.

but this more and more gets better.

Quote:
That is why I hope you will provide some too (doku)

how do you expect me to provide doku infos? (which format) or is that as well covered in the "how to contribute"? (i'll take a look again..)

Basically, after obtaining svn rights (not sure you have them already), you can use Topic++ to write documentation (either reference of classes/methods in "src" or "big picture" in "srcdoc" and commit them into main svn tree. After that, they become available in next U++ release.

Subject: Re: Drawing / Painter Difference,Usage,Manual
Posted by kohait00 on Sun, 27 Jun 2010 11:36:36 GMT
View Forum Message <> Reply to Message

nope, have no svn rights yet
