Subject: Introducing OSVGS
Posted by Tom1 on Tue, 27 Mar 2012 15:48:54 GMT
View Forum Message <> Reply to Message

Hi,

I have just committed the initial version of OSVGS and OSVGS_Demo in sandbox. OSVGS stands for one-stop-vector-graphics-shop. It is supposed to map a single API efficiently to various graphics backends: Draw, Painter, OpenGL and Direct2D for starters. The main idea is to get graphics dependent application code written once and then having it run on future platforms without changing the code. Just updating the OSVGS.

It could have been called TWR for the-wheel-reinvented too.

Anyone interested, please get it (SVN r.4724 or something) and test it.

- Reading OSVGS/OSVGS.h rapidly explains how it is supposed to work
- Reading OSVGS_Demo/main.cpp explains how to use it
- Reading OSVGS/*.txt explains what is the current status of the project

- Clicking on the OSVGS_Demo window's client area gives some simple benchmarking results.

- Windows 7 (or Vista SP2 +platform update for Direct2D) is required on windows to make it work.

- It should run on Linux too, but I haven't tested it in a couple of days.

- The most important thing missing here is the DrawText functionality for OpenGL implementation. I think it needs a different set of brains for decent results.

Anyway, please report any issues found. Constructive comments are most welcome -- destructive ones less so.

Best regards,

Tom

Subject: Re: Introducing OSVGS
Posted by Didier on Sat, 26 May 2012 21:19:30 GMT
View Forum Message <> Reply to Message

Hi Tom,

I just tried you're demo and it seems quite promissing.
OpenGL drawing is much faster for the drawn example.

The thing that surprises me is that nobody else answered you're post ?!?

The recent posts talking about RGBA and draw vs painter seem to show you're OSVGS philosophy is somewhat needed.

Keep-up

Subject: Re: Introducing OSVGS
Posted by Tom1 on Sun, 27 May 2012 07:11:38 GMT
View Forum Message <> Reply to Message

Hi Didier,

Thanks for your interest. (It is quite unique!)

Anyway, based on the fact that you are the first and only one to reply after two months has already passed, I'm afraid OSVGS is not going to be a "bestseller". Still, I hope this could have some effect on the direction future Upp graphics sub-systems will take.

Thanks and best regards,

Tom

Subject: Re: Introducing OSVGS
Posted by koldo on Sun, 27 May 2012 19:42:17 GMT
View Forum Message <> Reply to Message

Hello Tom

After renaming the WINXP ifdefs to avoid Direct2D I saw your package running. It seems rather interesting.

PD: It is a little bit slow to start.

Subject: Re: Introducing OSVGS
Posted by Tom1 on Mon, 28 May 2012 07:03:30 GMT
View Forum Message <> Reply to Message

Hi Koldo,

The start-up delay is caused by benchmarking the drawing performance each time the display is updated. By clicking the uncovered window area with left mouse button, you can get the performance report of the previous refresh.

The main.cpp for the OSVGS_Demo includes the TEST macro, in which int reps=2000;. Change to reps=1; and it starts reasonably fast. However, the benchmarking will not give proper results then because the timer resolution is not sufficient for such short test periods.

```
#define TEST(_function_,_result_) { \
 dword begin;\
 dword end;\
 int reps=2000;\
 dword zero=ms_timer;\
```

Thanks for your interest.

Best regards,

Tom

Subject: Re: Introducing OSVGS
Posted by koldo on Mon, 28 May 2012 09:03:28 GMT
View Forum Message <> Reply to Message

Hello Tom

One question: Why to create OSVGS instead of creating new Draw subclasses for OpenGL and Direct2D?

Now for example:
class PdfDraw : public Draw
class Painter : public Draw

Could it be this too?:
class OpenGLDraw : public Draw
class Direct2DDraw : public Draw

Subject: Re: Introducing OSVGS
Posted by Tom1 on Mon, 28 May 2012 10:06:25 GMT
View Forum Message <> Reply to Message

Hi Koldo,

There are some differences between Draw and OSVGS:

- OSVGS uses floating point coordinates whereas Draw uses ints.
- OSVGS supports wide dashed line styles (if available on the underlying platform) whereas Draw

does not.
- OSVGS supports anti-aliasing (if available on the underlying platform) whereas Draw does not.

I wrote OSVGS for Draw as an example of mapping the OSVGS API. The real thing should probably be written for Windows GDI and X11 directly (at least in part to support wide dashed line styles).

These differences rise from my personal specific needs. I also wanted to make OSVGS very simple and clear for anybody taking the first look at it, yet include enough features to render typical maps; please see OSVGS.h.

So, yes: It could have been OpenGLDraw : public Draw, ... etc. but that would not have solved my problem: Draw is too limited in a sense, and I'm not in the "inner circle" of U++ which is required to add features to Draw, so I need to work on the "outer ring" instead.

In fact, based on discussions with Mirek in this forum under a different topic a few months ago, I was left with an impression that Draw might be headed to a completely different direction: i.e. obsoleting features instead of adding more features. This was one of the reasons for me to put together this OSVGS thing.

Best regards,

Tom
