Subject: ScrollView - a question - maybe proposal..
Posted by luoganda on Wed, 16 Jan 2019 15:09:48 GMT
View Forum Message <> Reply to Message

Ok, here is attached code from original ScrollView.
I want to scroll screen buffer left - then rewrite what's left.
Explanations are in the sample code.

File Attachments
1) ScrollView.7z, downloaded 385 times

Subject: Re: ScrollView - a question - maybe proposal..
Posted by luoganda on Thu, 17 Jan 2019 17:47:12 GMT
View Forum Message <> Reply to Message

also,
there is a small "typo" on linux platform,
ImageDraw::GetStraight() <== has no definition, while declaration is in drawHeaders

Subject: Re: ScrollView - a question - maybe proposal..
Posted by luoganda on Thu, 17 Jan 2019 18:20:01 GMT
View Forum Message <> Reply to Message

ok, i wrote a small ImageDraw helper utility,
this somehow works, but it's slowwwww..
Any improvements here or suggestions are welcome.

Also - i recommend adding GetSize() to ImageDraw, to get Width/Height without getting image itself overhead.

```
class SMXImageDraw{
 ImageDraw *id=NULL;
 public:
 ~SMXImageDraw(){Clear();}
 void Set(int w,int h){Clear();id=new ImageDraw(w,h);}
 void Clear(){if(!id)return;delete id;id=NULL;}
 void ScrollImageX(int x); //does (quick)x sanity checks
 ImageDraw &Get(){return *id;}
 bool Empty(){return id==NULL;}
};

void SMXImageDraw::ScrollImageX(int x){
 int absx=abs(x);
 if(x==0||!id/*||absx>=id->GetSize().cx*/)return; // <== here, ImageDraw::GetSize() could come
```

handy
 Image img=(Image)*id;//->GetStraight();
 if(absx>=img.GetWidth())return;
 int nx=x<0?0:x;
 id->DrawImage(nx,0,img,Rect((x<0?absx:0),0,nx+(img.GetWidth()-(x<0?0:absx)),img.GetHeight()
));
}

Subject: Re: ScrollView - a question - maybe proposal..
Posted by mirek on Tue, 22 Jan 2019 08:00:31 GMT
View Forum Message <> Reply to Message

I am about to deprecate ScrollView, with modern systems this has little advantage. Soon it will just call Refresh for backward compatibility.

Subject: Re: ScrollView - a question - maybe proposal..
Posted by Tom1 on Tue, 22 Jan 2019 14:34:51 GMT
View Forum Message <> Reply to Message

Hi,

I'm still using it in a product for scrolling a histogram plot on X11 and WinXP clients. For WinVista, Win7, Win8 and Win10 I have implemented a dual ::SetSurface() method for similar scrolling effect. (The reason for two approaches is obtaining the optimal performance on each platform.)

If you are not in hurry with it, maybe you could keep it there for a few more years... (?)

It's not a must for me though. I can use the current U++ for those old systems (still in use by our clients) until they are phased out and replaced with new ones.

Best regards,

Tom

Subject: Re: ScrollView - a question - maybe proposal..
Posted by mirek on Tue, 22 Jan 2019 14:59:41 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Tue, 22 January 2019 15:34Hi,

I'm still using it in a product for scrolling a histogram plot on X11 and WinXP clients. For WinVista, Win7, Win8 and Win10 I have implemented a dual ::SetSurface() method for similar scrolling effect. (The reason for two approaches is obtaining the optimal performance on each platform.)

If you are not in hurry with it, maybe you could keep it there for a few more years... (?)

It's not a must for me though. I can use the current U++ for those old systems (still in use by our clients) until they are phased out and replaced with new ones.

I am in no hurry That said, I think the time when scrolling the view was faster than repainting was like 10 years ago... With GTK and MacOS backends, it already is not implemented - it simply does Refresh.

Mirek

Subject: Re: ScrollView - a question - maybe proposal..
Posted by luoganda on Tue, 22 Jan 2019 17:22:26 GMT
View Forum Message <> Reply to Message

i didn't take measure for:
readrawing whole area vs just scrolling and drawing the rest, which is ideal for plotting software, but:

simple optimized BitBlt for scrolling will be probably always faster that drawing fragmented old 's' patterned pixels along y axis BitBlt is probably 2D hardware optimized func - even on most hardware, i don't know if this is true, but probably, i am not sure for SetPixel - although i don't know the workings itself in upp, how this is done. So this "point" may not apply here for optimizations. SetPixel must do any math used in BitBlt, which BitBlt can probably do only once or at least less times - for example clipping program becomes more complex for apps such as plotting, because it needs to allocate(more memory is eaten) and manage old pixels too, in this case storing and redrawing old pixels + new

Up is for windows BitBlt, but the same probably aplies for linux functions.

It would be handy that this is somehow available, if not in Ctrl itself, at least in Bazaar.
I agree that this could write someone else, not just core upp programmers..

Subject: Re: ScrollView - a question - maybe proposal..
Posted by mirek on Tue, 22 Jan 2019 18:38:55 GMT
View Forum Message <> Reply to Message

luoganda wrote on Tue, 22 January 2019 18:22i didn't take measure for:

readrawing whole area vs just scrolling and drawing the rest, which is ideal for plotting software, but:

[list type=circle]
[*] simple optimized BitBlt for scrolling will be probably always faster that drawing fragmented old 's' patterned pixels along y axis
[*] BitBlt is probably 2D hardware optimized func - even on most hardware, i don't know if this is true, but probably, i am not sure for SetPixel - although i don't know the workings itself in upp, how this is done. So this "point" may not apply here for optimizations.

There is no "2D hardware" remaining All is GPU today.

That said, you might be right it is faster, but really hard to say. With scrolling you often need to clip a lot of things anyway.

Would nice if you had a chance to benchmark...

Mirek

Subject: Re: ScrollView - a question - maybe proposal..
Posted by luoganda on Wed, 23 Jan 2019 09:40:49 GMT
View Forum Message <> Reply to Message

Quote:There is no "2D hardware" remaining Smile All is GPU today.
Hmm, i didn't know that - something enlightening everyday.

I wrote a simple smt/eeg software viewer, which is through OpenEEG, but with current software for it, it doesn't work correctly.
This is what i am using scrolling - primary, because it's less complicated to do in this way.
Here are a couple of screenshots and a program, but for program one must have smt/eeg or a serialCom simulator.
Maybe a Refresh only ver would be ok to create, and testing the two.

Also, a good tutorial about why fftw like library is needed in the first place and how to plot that eeg signals to frequencies would be more than welcome, because i am not at home with fftw thing. But this is probably out of the scope of this thread, although any explanations and/or links are welcome.
Now, i just plot raw values that i get wia serialCom - with device format taken into consideration.
