
Subject: [GLCtrl] Adding an Initilisation function to GLPaint

Posted by [Xemuth](#) on Tue, 03 Mar 2020 10:32:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello community,

When we have to deal with OpenGL, it is often interesting to load a certain amount of data (Texture / Vertex / Shaders) before rendering a scene.

Nowaday, to load thoses things, I must declare a boolean in my derived class of GLCtrl and use it to perform many load at the first loop in GLPaint, as Follow :

```
bool isLoaded = false;
virtual void GLPaint() {
    if(!isLoaded){
        /* Loading Few things ! */
        isLoaded=true;
    }

    /* Code to perform draw */
    Refresh();
}
```

Even if this way of working is functional, Having to write many and many code and routine of loading in GLPaint() is annoying and don't help in code readability.

That's why I started to dig out the GLCtrl to implement a virtual function called Initialisation() wich can be overwritten. The purpose of it is to be called right after OpenGL context Initialisation. My first way of doing this was Adding the call of function here (GLCtrl.cpp line 33):

```
void GLCtrl::Init()
{
    Transparent();
#ifdef PLATFORM_WIN32
    pane.ctrl = this;
    Add(pane.SizePos());
#endif
    restore_gl_viewport__ = SetCurrentViewport;
    Initialisation(); //--> I call Initialisation Here
}
```

But (It seems) due to Ctrl creation, the function wich is called at this moment is the non overwritten Initialisation() from GLCtrl and not the one I have overwritten in my derived class from GLCtrl.

So my next try was to implement it here (Win32GLCtrl.cpp) :void GLCtrl::GLPane::State(int reason)

```
{
    DHCtrl::State(reason);

    if(reason == OPEN) {
```

```
HWND hwnd = GetHWND();
CreateContext();
HDC hDC = GetDC(hwnd);
if(!SetPixelFormat(hDC, s_pixelFormatID, &s_pfd))
    return;
ReleaseDC(hwnd, hDC);
static_cast<GLCtrl*>(parent)->Initialisation();//--> I call Initialisation Here
}
```

This time, it call the overwritted Initialisation() but OpenGL Context is not yet well initialised, consequence is every call to OpenGL api (all call to allocate some memory in vram) fail !

So far I dig, I didn't find the proper way of adding this function, so I need your help ! If someone have an idea...

Thanks in advance.
Best regard

Xemuth

Subject: Re: [GLCtrl] Adding an Initilisation function to GLPaint
Posted by [mirek](#) on Wed, 04 Mar 2020 12:06:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

I think the problem is that context is created but not associated with current thread.

Try with ExecuteGL method...

That said, doing this in GLPaint actually seems fine to me.

Mirek

Subject: Re: [GLCtrl] Adding an Initilisation function to GLPaint
Posted by [Xemuth](#) on Wed, 04 Mar 2020 12:28:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Wed, 04 March 2020 13:06 I think the problem is that context is created but not associated with current thread.

Try with ExecuteGL method...

Hello Mirek, you are right, doing this kind of thing work :

```
void GLCtrl::GLPane::ExecuteGL(HDC hDC, Event<> paint, bool swap_buffers)
{
    static bool HaveBeenInitialised = false;
    wglMakeCurrent(hDC, s_openGLContext);
    if(!HaveBeenInitialised){
        static_cast<GLCtrl*>(parent)->Initialisation();//--> I call Initialisation Here
        HaveBeenInitialised = true;
    }
    paint();
    if(swap_buffers)
        SwapBuffers(hDC);
    else
        glFlush();
    wglMakeCurrent(NULL, NULL);
}
```

mirek wrote on Wed, 04 March 2020 13:06 That said, doing this in GLPaint actually seems fine to me.

Yes, it work aswell but the idea of having a visible boolean just to do some action at the first loop of drawing is disturbing me :twisted: (the fix I did in ExecuteGL(...) is basicly the same things but user don't see it :d Is it possible to add it to the GLCtrl code or you prefer stay like that ?

Thanks in advance
Best regard !

Subject: Re: [GLCtrl] Adding an Initilisation function to GLPaint
Posted by [mirek](#) on Fri, 06 Mar 2020 09:28:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Xemuth wrote on Wed, 04 March 2020 13:28 mirek wrote on Wed, 04 March 2020 13:06 I think the problem is that context is created but not associated with current thread.

Try with ExecuteGL method...

Hello Mirek, you are right, doing this kind of thing work :

```
void GLCtrl::GLPane::ExecuteGL(HDC hDC, Event<> paint, bool swap_buffers)
{
    static bool HaveBeenInitialised = false;
    wglMakeCurrent(hDC, s_openGLContext);
    if(!HaveBeenInitialised){
        static_cast<GLCtrl*>(parent)->Initialisation();//--> I call Initialisation Here
        HaveBeenInitialised = true;
    }
}
```

```
}  
paint();  
if(swap_buffers)  
    SwapBuffers(hDC);  
else  
    glFlush();  
wglMakeCurrent(NULL, NULL);  
  
}
```

mirek wrote on Wed, 04 March 2020 13:06 That said, doing this in GLPaint actually seems fine to me.

Yes, it work aswell but the idea of having a visible boolean just to do some action at the first loop of drawing is disturbing me :twisted: (the fix I did in ExecuteGL(...) is basicly the same things but user don't see it :d Is it possible to add it to the GLCtrl code or you prefer stay like that ?

Thanks in advance
Best regard !

For I prefer it to stay as it is....

Subject: Re: [GLCtrl] Adding an Initilisation function to GLPaint
Posted by [koldo](#) on Fri, 06 Mar 2020 13:20:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Do not worry Xemuth. With your control totally developed, GLPaint will be scarcely used.
