Subject: [FIXED] GLDraw,GLCtrl would be nice to work properly
Posted by luoganda on Fri, 17 Jun 2022 10:54:20 GMT
View Forum Message <> Reply to Message

Since this are a part of nonBazaar core packages,
this would be nice to work properly.
I tested this on w7x64 with latest upp-included-clang(16299) and msvc17
producing 32bit debug executable with reference example - GLDrawDemo.
But for this - i used msvc17 since debugger gave me a little more info.
I have also tested this with producing 64bit code,
but it was even worse(only blank white screen).

Anyhow, i attached screenshots where bug happens,
if one is a little bit more familiar with underlying code,
one could probably resolve this quite easily.

Commented code also mostly all produces such bugs.
~~~~~

Bazaar(uppHub) thing:
i suggest - don't throw away Bazaar archive - it has a lot of usefull code.
I didn't for example found Serial package or non-bazaar FontSel(dialog)
in uppHub(maybe i overlooked something), which are mostly just copy/paste
packages - since all works out of the box.

For Serial bazaar package i just added this, to adjust a few things,
for linux, i am not sure if this is needed or how it's done.

```
 void SetWriteTimeout(dword ms){
   COMMTIMEOUTS ct;memset(&ct,0,sizeof(ct));
   ct.WriteTotalTimeoutConstant=ms;SetCommTimeouts(fd,&ct);
 }
 void SetTimeouts(dword msRead,dword msWrite){
   COMMTIMEOUTS ct;memset(&ct,0,sizeof(ct));
   ct.ReadIntervalTimeout=ct.ReadTotalTimeoutMultiplier=MAXDWORD;
   ct.ReadTotalTimeoutConstant=msRead;
   /*ct.WriteTotalTimeoutMultiplier=MAXDWORD;*/
   ct.WriteTotalTimeoutConstant=msWrite; SetCommTimeouts(fd,&ct);
 }
```

File Attachments

1) GLDrawDemo-sshots.7z, downloaded 330 times

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by Klugier on Sat, 18 Jun 2022 07:53:30 GMT
~~~~~

Hello,

Thank you for your feedback! In context of UppHub this is the step forward in comparison to bazzar and we shouldn't back to it anymore. The reason for that is that many packages are outdated and there are lack of maintainer. I know that currently we do not have Serial package on UppHub, but why do not create one? If you are interested you can create UppHub repository with Serial package from bazaar. This package will be maintain by you.

To create UppHub package, just create git repository and full-fill them accordingly. You could follow the repository structure of AutoScroller. If everything will be ready just let us know that you have created Serial package, so we could add it to UppHub registry.

BTW, You put two topics into one thread. Please raise separate topic for GLDraw issue. For me it works correctly on Windows10. Are you compiling GLDrawDemo on virtual machine or something, there might be a problem with a drivers. Also, you should migrate to more modern system. Windows7 hit end of the life, so there is no grantee that everything will be working as expected. Could you try on Windows10. Windows11 or GNU/Linux?

EDIT: OK, after testing we have a fatal error at the begging of the program when targeting 32 bit operating system on Windows. On 64 bit everything works correctly. If you compile with USEMALLOCK it should works fine.

Klugier

File Attachments

1) Screenshot 2022-06-18 100342.png, downloaded 1261 times

Help

GUI USEMALLOC | CLANG Debug

main.cpp | Gtypes.h | anne.svg | main.cpp | Util.cpp | HeapImp.h | Index.hpp

```
1  #include <CtrlLib/CtrlLib.h>
2  #include <GLDraw/GLDraw.h>
3  #include <GLCtrl/GLCtrl.h>
4
5  using namespace Upp;
6
7  struct OpenGLExample : GLCtrl {
8      Point point;
9
10     virtual void GLPaint() {
11         Size sz = GetSize();
12
13         GLDraw w;
14
15         w.Init(sz);
16
17         w.DrawRect(0, 0, sz.cx, sz.cy, SWhite);
18         w.DrawRect(10, 10, 30, 30, SRed);
19         w.DrawLine(45, 45, 80, 120, 4, Blue);
20         w.DrawLine(80, 90, 400, 0, PEN_DASHDOT);
21         w.DrawImage(200, 10, CtrlImg::reporticon());
22         w.DrawImage(280, 10, CtrlImg::reporticon(), Red());
23         const char *text = "This text is centered";
24         Size tsz = GetTextSize(text, Arial(25).Bold());
25         w.DrawText((sz.cx - tsz.cx) / 2, (sz.cy - tsz.cy) / 2, text, Arial(2
26         w.Clipoff(200, 50, 95, 100);
27         w.DrawText(0, 80, "CLIPPED", Roman(25));
28         w.End();
29
30         for(int a
31             w.Dra
32     }
33
34     virtual void
35         point = p
36         Refresh()
37     }
38 };
39
40 GUI_APP_MAIN
41 {
42     Ctrl::GlobalE
43     TopWindow win
44     OpenGLExample
45     gl.SetFrame(I
46     win.Add(gl.HSizePos(10, 10).VSizePos(10, 10));
47     win.Sizeable().Zoomable();
48     win.Open();
49     win.Run();
50 }
51
```

Main package configuration(s)

Flags

GUI USEMALLOC

Configurati

GUI applicati

Set | Flag

HEAPDB
HEAPLO
HEAPOVI
HEAPSTA
IVTEST
LINUXFB
MT
NODEPRI
NOGTK
NOI18N
NONAME
NOSIMD
OSX
PROFILEM
SDLFB
SKELETO
SO
STACKTR
STATIC_P
STD_NEW
TESTXPC
TEST_XM
TIMING
TURTLE
USEMALL
VIRTUALG
WIN10
WIN32
WINALT
WINFB
WINGL
X11

Other flags:

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by mirek on Sun, 19 Jun 2022 18:16:51 GMT
View Forum Message <> Reply to Message

luoganda wrote on Fri, 17 June 2022 12:54
Bazaar(uppHub) thing:
i suggest - don't throw away Bazaar archive - it has a lot of usefull code.
I didn't for example found Serial package or non-bazaar FontSel(dialog)
in uppHub(maybe i overlooked something), which are mostly just copy/paste
packages - since all works out of the box.

For Serial bazaar package i just added this, to adjust a few things,
for linux, i am not sure if this is needed or how it's done.

```
 void SetWriteTimeout(dword ms){
   COMMTIMEOUTS ct;memset(&ct,0,sizeof(ct));
   ct.WriteTotalTimeoutConstant=ms;SetCommTimeouts(fd,&ct);
 }
 void SetTimeouts(dword msRead,dword msWrite){
   COMMTIMEOUTS ct;memset(&ct,0,sizeof(ct));
   ct.ReadIntervalTimeout=ct.ReadTotalTimeoutMultiplier=MAXDWORD;
   ct.ReadTotalTimeoutConstant=msRead;
   /*ct.WriteTotalTimeoutMultiplier=MAXDWORD;*/
   ct.WriteTotalTimeoutConstant=msWrite; SetCommTimeouts(fd,&ct);
 }
```

Why do not you start UppHub package for serial?

Seriously, the problem with bazaar was that while it contained a lot of great code, 90% of it was unmaintained.

Mirek

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by luoganda on Mon, 20 Jun 2022 12:29:32 GMT
View Forum Message <> Reply to Message

I think that at least Serial and Sound(wrapped PortAudio soundsystem+plugin/wav)
should go into plugin or core packages, since they are fundamentals.
Serial code won't change in the future, and probably neither PortAudio,
so maintaining code is not an issue, copy/paste packages to core would be fine.

Of course, i have made copies of bazaar archive, just in case you'll be stubborn .
Personally, i prefer main upp-download to be as small as possible, without bazaar and git,
but bazaar and/or github as an option. Upphub may be a plus for some situations
and of course - it could be a handy option and probably "flaten out" newVers diferences

between bazaar and upp bugs.
If one is eager for upphub thing, then one could probably create converter
from bazaar to upphub, and mark/review packages as fullyFuncional or unmaintained, etc.
If this is started: from my experience - Serial and Sound are fully functional.
SoundExample code is also there and functional albeit for blocking mode. Streamed mode
is from my experience fully functional also.

The same goes for FontSel that was proposed soooo many times but not integrated into
core(uppsrc/CtrlLib),
all is there only this is lacking for dialogs(colorchooser,FileSel,etc) to be complete.
FontSel code was written by someone else - but i think i changed it a bit some time ago,
anyhow - it's attached(if integrated - only check/correct FontSel.h and FSel.cpp header paths -
2linesOfCode).

What about GLDraw and GLCtrl - core of this topic?
This are also fundamentals..
It's a simple correction for one that knowns how underlying code should work.

File Attachments
1) FontSel.7z, downloaded 334 times

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by Klugier on Tue, 21 Jun 2022 23:39:55 GMT
View Forum Message <> Reply to Message

Hello Luoganda,

Serial is niche. It shouldn't be part of the Core package and any developer that is currently engage in Upp framework development do not have much experience with that. So putting it there doesn't make sens to me.

However, the different story is with UppHub. It fits there perfectly. It is great addition that expands capabilities of whole Upp framework.

As you see there is no maitaner and the person who wants to move that package there. Could you help with this? Do you want to be a person deserved to our community? If yes, please create GitHub repository in accordance to instruction in this documentation page and let us know when it will be ready. We will handle the package publishing process for ourselves.

Luoganda, I believe in you that you publish your first package on UppHub! Good luck!

Klugier

Subject: Re: GLDraw,GLCtrl would be nice to work properly

Posted by mirek on Sun, 03 Jul 2022 18:55:20 GMT
View Forum Message <> Reply to Message

luoganda wrote on Mon, 20 June 2022 14:29
What about GLDraw and GLCtrl - core of this topic?

Tested (with RX580), works. Unfortunately this can be GPU/GPU driver releated; I will try again once I have got Nvidia sometimes this or next month.

Meanwhile, why do not you try to investigate? Have tried to e.g. get backtrace for starters?

Mirek

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by mirek on Sun, 03 Jul 2022 19:00:13 GMT
View Forum Message <> Reply to Message

luoganda wrote on Mon, 20 June 2022 14:29I think that at least Serial and Sound(wrapped PortAudio soundsystem+plugin/wav)
should go into plugin or core packages, since they are fundamentals.
Serial code won't change in the future, and probably neither PortAudio,
so maintaining code is not an issue, copy/paste packages to core would be fine.

I am not fundamentally opposed to this.

Quote:
If one is eager for upphub thing, then one could probably create converter
from bazaar to upphub, and mark/review packages as fullyFuncional or unmaintained, etc.

This would be a mistake.

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by Klugier on Sun, 03 Jul 2022 20:51:35 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Did you tested with 32 bit clang on Windows (not 64 bit)? I reproduce it with default CLANG build method (not CLANGx64). The error is only with 32 bit, 64 works without crash. It reproduces for both Nvidia and AMD GPU, so I am sure that it is not driver related. I attached screenshot with

crash.

Here is backtrace:

Upp::Panic(msg=b2430c "Writes to ") at C:\Prototable\upp\uppsrc\Core\Util.cpp 119
Upp::BlkHeap<Upp::Heap::LargeHeapDetail,256>::DbgFreeCheck(p=7acfd58, size=2024) at C:\Prototable\upp\uppsrc\Core\HeapImp.h 101
Upp::BlkHeap<Upp::Heap::LargeHeapDetail,256>::CheckFree(h=7acfd40) at C:\Prototable\upp\uppsrc\Core\HeapImp.h 120
Upp::BlkHeap<Upp::Heap::LargeHeapDetail,256>::MakeAlloc(h=7acfd40, wcount=8) at C:\Prototable\upp\uppsrc\Core\HeapImp.h 211
Upp::Heap::TryLAlloc(i0=5, wcount=8) at C:\Prototable\upp\uppsrc\Core\lheap.cpp 55
Upp::Heap::LAlloc(size=) at C:\Prototable\upp\uppsrc\Core\lheap.cpp 111
Upp::Heap::AllocSz(sz=) at C:\Prototable\upp\uppsrc\Core\sheap.cpp 167
Upp::MemoryAllocSz_(sz=) at C:\Prototable\upp\uppsrc\Core\sheap.cpp 458
Upp::MemoryAllocSz(size=) at C:\Prototable\upp\uppsrc\Core\heapdbg.cpp 142
Upp::MemoryAlloc(size=2032) at C:\Prototable\upp\uppsrc\Core\heapdbg.cpp 155
Upp::Index<Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key>::ReallocHash(n=73) at C:\Prototable\upp\uppsrc\Core\Index.hpp 41
Upp::Index<Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key>::GrowAdd<const Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key &>(k=, sh=2394942300) at C:\Prototable\upp\uppsrc\Core\Index.hpp 78
Upp::Index<Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key>::AddS<const Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key &>(m=, k=, sh=2394942300) at C:\Prototable\upp\uppsrc\Core\Index.hpp 86
Upp::Index<Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key>::AddS<const Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key &>(k=, sh=2394942300) at C:\Prototable\upp\uppsrc\Core\Index.hpp 102
Upp::Index<Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key>::Put0<const Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key &>(k=, sh=2394942300) at C:\Prototable\upp\uppsrc\Core\Index.hpp 237
Upp::Index<Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Key>::Put(k=) at C:\Prototable\upp\uppsrc\Core\Index.h 92
Upp::LRUCache<Upp::ImageGLData,Upp::Tuple<unsigned long long,unsigned long long,unsigned long> >::Get(m=) at C:\Prototable\upp\uppsrc\Core\Other.hpp 187
Upp::GetTextureForImage(flags=3, img=, context=0) at C:\Prototable\upp\uppsrc\GLDraw\Texture.cpp 90
Upp::GetTextureForImage(img=, context=167162738539356648) at C:\Prototable\upp\uppsrc\GLDraw\GLDraw.h 38
Upp::GLDraw::PutImage(p=, img=, src=, color= #000000) at

C:\Prototable\upp\uppsrc\GLDraw\GLDrawS.cpp 400
Upp::SDraw::SysDrawImageOp(x=1674, y=593, img=, src=, color= #000000) at
C:\Prototable\upp\uppsrc\Draw\SDrawPut.cpp 23
Upp::SDraw::DrawTextOp(x=1674, y=605, angle=1200, text=251e5d8, font=<:12>, ink= #000000,
n=1, dx=0) at C:\Prototable\upp\uppsrc\Draw\SDrawText.cpp 81
Upp::Draw::DrawTextA(x=1694, y=640, angle=1200, text=7b09610, font=<:254>, ink= #000000,
n=23, dx=0) at C:\Prototable\upp\uppsrc\Draw\DrawText.cpp 49
Upp::Draw::DrawTextA(x=1694, y=640, angle=1200, text=, font=<null>, ink= #FE0000, dx=0) at
C:\Prototable\upp\uppsrc\Draw\DrawText.cpp 162
Upp::Draw::DrawTextA(x=1694, y=640, angle=1200, text=7b468d0 "x Text w", charset=0,
font=<:254>, ink= #FE0000, n=23, dx=0) at C:\Prototable\upp\uppsrc\Draw\DrawText.cpp 174
Upp::Draw::DrawTextA(x=1694, y=640, angle=1200, text=7b468d0 "x Text w", font=<:254>, ink=
#FE0000, n=23, dx=0) at C:\Prototable\upp\uppsrc\Draw\DrawText.cpp 189
Upp::Draw::DrawTextA(x=1694, y=640, angle=1200, text=, font=<:640>, ink= #FE0000, dx=0) at
C:\Prototable\upp\uppsrc\Draw\DrawText.cpp 203
OpenGLExample::GLPaint() at C:\Prototable\upp\reference\GLDrawDemo\main.cpp 31
Upp::GLCtrl::DoGLPaint() at C:\Prototable\upp\uppsrc\GLCtrl\GLCtrl.cpp 24
Upp::GLCtrl::GLPane::WindowProc::<lambda0>::operator()() at
C:\Prototable\upp\uppsrc\GLCtrl\Win32GLCtrl.cpp 135
Upp::Function<void ()>::Wrapper<`lambda at
C:\Prototable\upp\uppsrc\GLCtrl\Win32GLCtrl.cpp:135:21'>::Execute() at
C:\Prototable\upp\uppsrc\Core\Function.h 17
Upp::Function<void ()>::operator()() at C:\Prototable\upp\uppsrc\Core\Function.h 76
Upp::GLCtrl::GLPane::ExecuteGL(hDC=e7011033, paint=, swap_buffers=1) at
C:\Prototable\upp\uppsrc\GLCtrl\Win32GLCtrl.cpp 113
Upp::GLCtrl::GLPane::WindowProc(message=15, wParam=0, lParam=0) at
C:\Prototable\upp\uppsrc\GLCtrl\Win32GLCtrl.cpp 135
Upp::Ctrl::WndProc(hWnd=25064c, message=15, wParam=0, lParam=0) at
C:\Prototable\upp\uppsrc\CtrlCore\Win32Wnd.cpp 645
759a7c92
7598714a
75985d7f
74bcc889
759a7c92
7598714a
75986bda
7598fa9f
76f4719c
Upp::Ctrl::UpdateDHCtrls() at C:\Prototable\upp\uppsrc\CtrlCore\DHCtrl.cpp 126
Upp::Ctrl::UpdateDHCtrls() at C:\Prototable\upp\uppsrc\CtrlCore\DHCtrl.cpp 123
Upp::Ctrl::WindowProc(message=15, wParam=0, lParam=0) at
C:\Prototable\upp\uppsrc\CtrlCore\Win32Proc.cpp 261
Upp::TopWindow::WindowProc(message=15, wParam=0, lParam=0) at
C:\Prototable\upp\uppsrc\CtrlCore\TopWin32.cpp 70
Upp::Ctrl::WndProc(hWnd=16067a, message=15, wParam=0, lParam=0) at
C:\Prototable\upp\uppsrc\CtrlCore\Win32Wnd.cpp 645
759a7c92
7598714a

75986bda
7598fa9f
76f4719c
759855b0
Upp::Ctrl::sProcessMSG(msg=) at C:\Prototable\upp\uppsrc\CtrlCore\Win32Wnd.cpp 740
Upp::Ctrl::ProcessEvent(quit=251fb47) at C:\Prototable\upp\uppsrc\CtrlCore\Win32Wnd.cpp 762
Upp::Ctrl::ProcessEvents(quit=251fb47) at C:\Prototable\upp\uppsrc\CtrlCore\Win32Wnd.cpp 781
Upp::Ctrl::EventLoop(ctrl=251fcf8) at C:\Prototable\upp\uppsrc\CtrlCore\Win32Wnd.cpp 805
Upp::TopWindow::Run(appmodal=0) at C:\Prototable\upp\uppsrc\CtrlCore\TopWindow.cpp 323
GuiMainFn_() at C:\Prototable\upp\reference\GLDrawDemo\main.cpp 49
Upp::AppExecute__(app=621500) at C:\Prototable\upp\uppsrc\Core\App.cpp 450
WinMain(hInstance=620000, lpCmdLine=27e0721 "", nCmdShow=10) at
C:\Prototable\upp\reference\GLDrawDemo\main.cpp 40
main()
WinMainCRTStartup()
74d66739
76f38fef
76f38fbd

As I wrote previously, using malloc instead of our allocator solves the problem in this case.

Klugier

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by mirek on Sun, 03 Jul 2022 22:41:03 GMT
View Forum Message <> Reply to Message

Does not crash for me, but backtrace is helpful. You have not posted the screenshot, but it appears like "Writes to freed blocks detected" issue?

Obviously, this is is not detected with USEMALLOC for simple reason that normal allocator does not have any such detected.

Anyway, probably the problem is somewhere about image cache recycling/freeing (related to rendering the text). Can you try with

```
  w.DrawLine(80, 90, 400, 0, PEN_DASHDOT);
 // w.DrawImage(200, 10, CtrlImg::reporticon());
 // w.DrawImage(280, 10, CtrlImg::reporticon(), Red());
  const char *text = "This text is centered";
  Size tsz = GetTextSize(text, Arial(25).Bold());
 // w.DrawText((sz.cx - tsz.cx) / 2, (sz.cy - tsz.cy) / 2, text, Arial(27).Bold(), SBlue);
  w.Clipoff(200, 50, 95, 100);
 // w.DrawText(0, 80, "CLIPPED", Roman(25));
  w.End();
```

```
 // for(int angle = 0; angle < 360; angle += 30)
 //  w.DrawText(sz.cx - sz.cx / 3, sz.cy - sz.cy / 3, angle * 10, "x   Text with angle " +
AsString(angle));
```

just to test if we are onto something?

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by mirek on Sun, 03 Jul 2022 23:04:25 GMT
View Forum Message <> Reply to Message

Next thing to try is to comment GLDraw/Texture.cpp:43 SysImageRealized

---

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by Klugier on Mon, 04 Jul 2022 21:36:53 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Commenting SysImageRealized doesn't help. Also, commenting w.DrawImage(), too. What helped is commenting loop with angle:

```
  //for(int angle = 0; angle < 360; angle += 30)
  // w.DrawText(sz.cx - sz.cx / 3, sz.cy - sz.cy / 3, angle * 10, "x   Text with angle " +
AsString(angle));
```

So, following code works fine on 32-bit windows:

```
 virtual void GLPaint() {
  Size sz = GetSize();

  GLDraw w;

  w.Init(sz);

  w.DrawRect(0, 0, sz.cx, sz.cy, SWhite);
  w.DrawRect(10, 10, 30, 30, SRed);
  w.DrawLine(45, 45, 80, 120, 4, Blue);
  w.DrawLine(80, 90, 400, 0, PEN_DASHDOT);

  w.DrawImage(200, 10, CtrlImg::reporticon());
  w.DrawImage(280, 10, CtrlImg::reporticon(), Red());
```

```
  const char *text = "This text is centered";
  Size tsz = GetTextSize(text, Arial(25).Bold());

  w.DrawText((sz.cx - tsz.cx) / 2, (sz.cy - tsz.cy) / 2, text, Arial(27).Bold(), SBlue);
  w.Clipoff(200, 50, 95, 100);
  w.DrawText(0, 80, "CLIPPED", Roman(25));

  //for(int angle = 0; angle < 360; angle += 30)
  // w.DrawText(sz.cx - sz.cx / 3, sz.cy - sz.cy / 3, angle * 10, "x   Text with angle " +
AsString(angle));

  w.End();
 }
```

OK, after digging more AsString(angle) is causing crash. It works good with following line of code:

```
w.DrawText(sz.cx - sz.cx / 3, sz.cy - sz.cy / 3, angle * 10, "x   Text with angle " /*+
AsString(angle)*/);
```

Replacing AsString with IntStr cause the same error. I think the problem must be with utoa32 (Format.cpp) function and it's memory management on 32 bit architecture.

Klugier

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by mirek on Mon, 04 Jul 2022 23:05:27 GMT
View Forum Message <> Reply to Message

Klugier wrote on Mon, 04 July 2022 23:36
```
w.DrawText(sz.cx - sz.cx / 3, sz.cy - sz.cy / 3, angle * 10, "x   Text with angle " /*+
AsString(angle)*/);
```

Replacing AsString with IntStr cause the same error. I think the problem must be with utoa32 (Format.cpp) function and it's memory management on 32 bit architecture.

Klugier

Well, accidentally, AsString in this case does not allocate any memory....

What is more likely that glyphs for digits are maybe causing the problem. Maybe you can replace AsString with "123" ?

Mirek

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by Klugier on Tue, 05 Jul 2022 08:44:14 GMT
View Forum Message <> Reply to Message

Hello Mirek,

You are right inserting numbers in the string cause crash. Below line crashes:

w.DrawText(sz.cx - sz.cx / 3, sz.cy - sz.cy / 3, angle * 10, "x Text with angle 123");

So, it is not a problem with number to string conversion. Thanks for explanation about memory allocation

Klugier

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by mirek on Tue, 05 Jul 2022 09:13:20 GMT
View Forum Message <> Reply to Message

Klugier wrote on Tue, 05 July 2022 10:44Hello Mirek,

You are right inserting numbers in the string cause crash. Below line crashes:

w.DrawText(sz.cx - sz.cx / 3, sz.cy - sz.cy / 3, angle * 10, "x Text with angle 123");

So, it is not a problem with number to string conversion. Thanks for explanation about memory allocation

Klugier

Keep experimenting. What about "xxx" instead of "123" ? (These are glyphs that already should be cached).

Mirek

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by Klugier on Tue, 05 Jul 2022 09:28:35 GMT
View Forum Message <> Reply to Message

Hello Mirek,

xxx works correctly without crash.

Klugier

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by luoganda on Tue, 05 Jul 2022 10:43:39 GMT
View Forum Message <> Reply to Message

The card is a little older
intel Q965/Q963 Express Chipset Family.
So yes, Mirek - it might be this.
Although i think that with gl2.0 should also work fine for basic line etc drawing,
if there are limits what glVer upp needs - then it might be this.
On w7x64 i have also directx11.

On windows - there are only drivers till 2.0 for opengl(rev 8.15.10.1930),
on linux - i think there are for 2.1 too.

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by luoganda on Mon, 25 Dec 2023 10:24:10 GMT
View Forum Message <> Reply to Message

Checked,
minimum OpenGL version on any platform is fully supported v2.1.

Meaning:
with older graphics card mentioned in previous post,
in this case - GLDraw/GLCtrl works ok on linux with latest drivers but not on win7(drivers not supporting full 2.1 capabilities).

Subject: Re: GLDraw,GLCtrl would be nice to work properly
Posted by Klugier on Mon, 25 Dec 2023 16:06:07 GMT
View Forum Message <> Reply to Message

Hello luoganda,

Any chances to update your Windows machine at least to Windows 10. Our policy right now is not to invest in operating systems that are not supported by their maintainers. Windows 7 is not supported since January 14, 2020. You can read more about this topic here.

Klugier