
Subject: Re: GLCtrl in GTK

Posted by [mirek](#) on Thu, 06 Dec 2018 07:10:09 GMT

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

Thank you. Now let us try adding some LOGs to GLCtrl/GTKGLCtrl.cpp.

Please change it to:

```
#include "GLCtrl.h"
```

```
#ifdef GUI_GTK
```

```
#include <GL/glx.h>
```

```
#include <GL/gl.h>
```

```
#include <gdk/gdkx.h>
```

```
namespace Upp {
```

```
static XVisualInfo *s_XVisualInfo;
```

```
static Colormap    s_Colormap;
```

```
static GLXContext  s_GLXContext;
```

```
static ::Display   *s_Display;
```

```
EXITBLOCK {
```

```
    if(s_GLXContext)
```

```
        glXDestroyContext(s_Display, s_GLXContext);
```

```
}
```

```
void GLCtrl::Create()
```

```
{
```

```
    DLOG("Create");
```

```
    Ctrl *top = GetTopCtrl();
```

```
    if(!top)
```

```
        return;
```

```
    GdkWindow *gdk = top->gdk();
```

```
    if(!gdk)
```

```
        return;
```

```
    DDUMP(gdk);
```

```
    Window w = gdk_x11_drawable_get_xid((GdkDrawable *)gdk);
```

```
    DDUMP(w);
```

```
    ONCELOCK {
```

```
        s_Display = gdk_x11_drawable_get_xdisplay((GdkDrawable *)gdk);
```

```

int samples = numberOfSamples;
GLXFBConfig *fbc;
do {
    Vector<int> attr;
    attr << GLX_RGBA << GLX_DEPTH_SIZE << depthSize
        << GLX_STENCIL_SIZE << stencilSize;
    if(doubleBuffering)
        attr << GLX_DOUBLEBUFFER;
    if(samples > 1)
        attr << GLX_SAMPLE_BUFFERS_ARB << 1 << GLX_SAMPLES_ARB << samples;
    attr << 0;

    samples >>= 1;
    int fbcount;
    fbc = glXChooseFBConfig(s_Display, DefaultScreen(s_Display), attr, &fbcount);
}
while(!fbc && samples > 0);
DDUMP(fbc);
if(!fbc)
    return;
s_XVisualInfo = glXGetVisualFromFBConfig(s_Display, fbc[0]);
s_Colormap = XCreateColormap(s_Display, RootWindow(s_Display, s_XVisualInfo->screen),
s_XVisualInfo->visual, AllocNone);
s_GLXContext = glXCreateContext(s_Display, s_XVisualInfo, NULL, GL_TRUE);

DDUMP(s_XVisualInfo);
DDUMP(s_Colormap);
DDUMP(s_GLXContext);
}

if(!s_GLXContext)
    return;

XSetWindowAttributes swa;
swa.colormap = s_Colormap;
swa.border_pixel = 0;
swa.event_mask = 0;

win = XCreateWindow(s_Display, w, 0, 0, 1, 1, 0,
                    s_XVisualInfo->depth, InputOutput, s_XVisualInfo->visual,
                    CWBorderPixel|CWColormap|CWEventMask, &swa);
visible = false;
position = Null;

DDUMP(win);
}

void GLCtrl::Sync()

```

```

{
    if(win) {
        Rect r = GetScreenView() - GetTopCtrl()->GetScreenRect().TopLeft();
        bool b = IsVisible() && r.GetWidth() > 0 && r.GetHeight() > 0;
        if(b != visible) {
            visible = b;
            position = Null;
            if(b)
                XMapWindow(s_Display, win);
            else
                XUnmapWindow(s_Display, win);
        }
        if(r != position && visible) {
            position = r;
            XMoveResizeWindow(s_Display, win, r.left, r.top, r.Width(), r.Height());
        }
        DDUMP(visible);
        DDUMP(r);
    }
}

```

```

void GLCtrl::State(int reason)
{
    DLOG("State");
    switch(reason) {
        case CLOSE:
            DLOG("Destroy");
            XDestroyWindow(s_Display, win);
            break;
        case OPEN:
            DLOG("Create");
            Create();
        default:
            DLOG("Sync");
            Sync();
            break;
    }
}

```

```

void GLCtrl::Paint(Draw& w)
{
    Size sz = GetSize();
    DLOG("Paint");
    DDUMP(s_GLXContext);
    DDUMP(sz);
    DDUMP(s_Display);

    if(!s_GLXContext || sz.cx == 0 || sz.cy == 0)

```

```
return;

glXMakeCurrent(s_Display, win, s_GLXContext);

ONCELOCK {
    glewInit();
}

DLOG("DoGLPaint");
DoGLPaint();

if(doubleBuffering)
    glXSwapBuffers(s_Display, win);
else
    glFlush();

glXMakeCurrent(s_Display, None, NULL);
}

}

#endif
```

run reference/OpenGL (without changing anything else) and post a .log file here (Alt+L).

Thanks. We need to figure this out...

Mirek
