
Subject: Re: Painter future

Posted by [Tom1](#) on Wed, 25 Feb 2009 13:09:09 GMT

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

Putting aside true subpixel precision and anti-aliasing, comparing just raw simple polygon rendering under Upp, I made a simple test app:

```
#include <GLCtrl/GLCtrl.h>
#include <CtrlLib/CtrlLib.h>
#include <Painter/Painter.h>
```

```
using namespace Upp;
```

```
struct PainterExample : Ctrl {
public:
    double delta;
    virtual void Paint(Draw &draw) {
        delta=0;
```

```
        ImageBuffer ib(draw.GetPagePixels());
        BufferPainter pntr(ib);
```

```
        dword begin=GetTickCount();
```

```
        int reps=0;
        for(reps=0;reps<5;reps++) for(int i=0;i<500;i++){
            //pntr.Move(i,0).Line(499,i).Line(499-i,499).Line(0,499-i).Close().Stroke(1,Red());
            pntr.Move(i,0).Line(499,i).Line(499-i,499).Line(0,499-i).Close().Fill(Red());
        }
        draw.DrawImage(0,0,ib);
        dword end=GetTickCount();
        delta=end-begin;
        delta/=reps;
    }
};
```

```
struct DrawExample : Ctrl {
public:
    double delta;
    virtual void Paint(Draw &draw) {
        delta=0;
```

```
        dword begin=GetTickCount();
        Rect rect(0,0,500,500);
```

```
        int reps=0;
        for(reps=0;reps<5;reps++) for(int i=0;i<500;i++){
```

```

    Point v[5]={Point(i,0),Point(499,i),Point(499-i,499),Point(0,499-i),Point(i,0)};
    //draw.DrawPolyline(v,5,0,Green());
    draw.DrawPolygon(v,5,Green());
}
dword end=GetTickCount();
delta=end-begin;
delta/=reps;
}
};

```

```

struct OpenGLExample : GLCtrl {
public:
    double delta;
    virtual void GLPaint() {
        delta=0;
        Size sz=GetSize();
        dword begin=GetTickCount();

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

        glViewport(0,0,sz.cx,sz.cy);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        glPushMatrix();
        glOrtho(0,sz.cx,sz.cy,0,-1,1);

        int reps=0;
        for(reps=0;reps<10;reps++) for(int i=0;i<500;i++){
            //glBegin(GL_LINE_STRIP); // Polyline
            glBegin(GL_POLYGON); // Polygon
            glColor3f(0.0f,0.0f,1.0f); // Blue
            glVertex2f((float)i,(float)0);
            glVertex2f((float)499,(float)i);
            glVertex2f((float)499-i,(float)499);
            glVertex2f((float)0,(float)499-i);
            glVertex2f((float)i,(float)0);
            glEnd();
            glFlush();
        }

        glPopMatrix();
        glFlush();

        dword end=GetTickCount();
        delta=end-begin;
        delta/=reps;
    }
};

```

```

}
};

class ExampleTopWindow: public TopWindow{
public:
    PainterExample pr;
    DrawExample dw;
    OpenGLExample gl;

    ExampleTopWindow(){
        pr.SetRect(0,0,500,500);
        dw.SetRect(500,0,500,500);
        gl.SetRect(1000,0,500,500);

        pr.BackPaint();
        dw.BackPaint();
        gl.BackPaint();

        pr.delta=0;
        dw.delta=0;
        gl.delta=0;

        Add(pr);
        Add(dw);
        Add(gl);
    }

    virtual void LeftDown(Point p,dword keyflags){
        Title(Format("Painter/Draw/OpenGL: %.3f/%.3f/%.3f ms",pr.delta,dw.delta,gl.delta));
        Refresh();
    }

};

GUI_APP_MAIN
{
    ExampleTopWindow win;
    win.Sizeable().Zoomable();
    win.Open();
    win.Run();
}

```

(Changing the limit for repeats on different platforms may be required for sufficient clock resolution. Also, this requires a screen width above 1500 pixels -- changing some values for smaller screens will be necessary.)

By clicking on the area not covered by the controls, you get the results of the previous round and simultaneously start the next one.

The results on my system (AMDx64/2.0GHz single core + NVidia 7600GS, running Vista Business x64) look as follows:

Painter polygons: 450 ms
Draw polygons (using GDI): 300 ms
OpenGL polygons (using GLCtrl): 51 ms

Painter polylines: 210 ms
Draw polylines(using GDI): 5 ms
OpenGL polylines (using GLCtrl): 32 ms

What are your results?

// Tom
