
Subject: Painter bug?

Posted by [Mindtraveller](#) on Fri, 11 Mar 2011 11:51:22 GMT

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

```
painter
.Move(-.5*width2, h1)
.Line(-.5*width1, 0)
.Line( .5*width1, 0)
.Line( .5*width2, h1)
.Line(-.5*width2, h1)
.Fill(-.25*width2, h1, COLOR_HILIGHT, -.15*width2, .4*h1, .3*width2, COLOR_COMMON)
;
painter
.Rectangle(-.5*width2,h1,width2,length2)
.ColorStop(0.10, COLOR_COMMON)
.ColorStop(0.25, COLOR_HILIGHT)
.ColorStop(0.55, COLOR_COMMON)
.Fill(-.5*width2, 0., COLOR_COMMON, .5*width2, 0., COLOR_COMMON)
;
```

gives an empty line between 1st and 2nd figures:

U++: svn latest

OS: Windows 7 build 7600

File Attachments

1) [silo.png](#), downloaded 762 times

Subject: Re: Painter bug?

Posted by [dolik.rce](#) on Fri, 11 Mar 2011 21:55:59 GMT

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

Hi Pavel

It looks almost as if the h1 gets truncated or rounded to int somewhere in the process, because the line is visible only when h1 has some non-zero decimal part and the width of the gap is biggest for xy.5.

One other possible cause could lie in the subpixel accuracy rendering and related anti-aliasing. Unfortunately, from the quick look I had at it I couldn't figure where exactly the problem is...

Also I don't have much idea about Painter internals, so I don't know exactly where to look. Perhaps Mirek will know better...

Best regards,
Honza

Subject: Re: Painter bug?
Posted by [mirek](#) on Sat, 12 Mar 2011 07:45:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Could you please provide values for variables? Or more complete example?

Somehow I am unable to reproduce the problem...

Mirek

Subject: Re: Painter bug?
Posted by [dolik.rce](#) on Sat, 12 Mar 2011 08:36:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Sat, 12 March 2011 08:45 Could you please provide values for variables? Or more complete example?

Somehow I am unable to reproduce the problem...

Mirek

I got the gap for values `double width2=200;`
`double width1=100;`
`double h1=50.5;`

`double length2=100;` Actually only thing necessary to reproduce is `h1`, the rest doesn't influence it at all. If you set `h1=50`, no space is visible, for `h1=50.5` it is the widest and if you increase it further, it gradually disappears for `h1=51...`

Honza

Subject: Re: Painter bug?
Posted by [dolik.rce](#) on Sat, 12 Mar 2011 09:00:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

PS: It happens for antialiased and subpixel settings. With NOAA no gap is visible.

Subject: Re: Painter bug?
Posted by [mirek](#) on Sat, 12 Mar 2011 18:17:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am afraid this is unavoidable problem caused by antialiasing and subpixel accuracy... (and this would appear in most similiar systems. My guess is if you would put something like this to .pdf or .svg, you would get artifact as well).

Think:

At the beggining there is a white color of background - 255,255,255. Now the y is "half" of pixel, so we have to mix the color of polygon with this white with "alpha" 0.5. So if the color is black, there will now be 127, 127, 127 in canvas.

Ok, then paint another polygon, again whe have "half" of pixel, say the color is black, so we again have to apply it with "alpha" 0.5. Result: 64, 64, 64... (but should have been 0,0,0)

So it is a 'feature', not a bug. You have to account for it.

More on this e.g. here (AGG related):

<http://thread.gmane.org/gmane.comp.graphics.agg/2359/focus=2367>

Mirek

Subject: Re: Painter bug?

Posted by [Mindtraveller](#) on Sat, 12 Mar 2011 21:14:05 GMT

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

Thank you for the replies. The problem is clear now.

Currently I see the only solution, and I'm not really shure you want it.

Anyway. We could use temporary surface which stores color + alpha for a sequence of operations. On writing a pixel, it's color is calculated in usual way, but alpha is added to the current alpha value. E.g. we will have alpha = 256 in previous Mirek's example. Rendering this temporary surface to actual image will give solid black color which is absolutely right result.

This could look like this:

painter

```
.BeginInit()  
/*draw polygon 1*/  
/*fill polygon 1*/  
/*draw polygon 2*/  
/*fill polygon 2*/  
/*draw polygon 3*/  
/*fill polygon 3*/  
.EndInit()
```

;

UPDATE: OK, here is quick and dirty "back-end" solution. Let's imagine we draw polygons in

scaled coordinates:

```
painter.Scale(scale);
```

To avoid polygon stitching you may simply move polygon adjacent vertices with $0.5/scale$ towards neighbouring polygon, i.e.

```
painter.Move(x[i]+.5/scale,y[i]);
```

It moves polygon vertex to the next physical pixel if it's position is actually between pixels.

This technique seems like eliminating visual artifacts with U++ rendering of adjacent polygons.

Subject: Re: Painter bug?

Posted by [mirek](#) on Sat, 12 Mar 2011 22:28:05 GMT

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

Mindtraveller wrote on Sat, 12 March 2011 16:14

This technique seems like eliminating visual artifacts with U++ rendering of adjacent polygons.

Yep. I guess this is also what AGG author suggests by "dilating"...

BTW, the first time I have met this issue is a couple of years back when there was not even Painter in U++.

The problem was with "Microsoft XPS Document Writer" (pdf like format) output from U++, see here:

http://www.ultimatepp.org/forum/index.php?t=msg&goto=19408&&srch=Tom1+print#msg_19408

it is exactly the same issue, this time in Windows

Mirek

Subject: Re: Painter bug?

Posted by [tojocky](#) on Sun, 13 Mar 2011 10:55:44 GMT

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

Mindtraveller wrote on Sat, 12 March 2011 23:14 Thank you for the replies. The problem is clear now.

Currently I see the only solution, and I'm not really shure you want it.

Anyway. We could use temporary surface which stores color + alpha for a sequence of operations. On writing a pixel, it's color is calculated in usual way, but alpha is added to the current alpha value. E.g. we will have alpha = 256 in previous Mirek's example. Rendering this temporary surface to actual image will give solid black color which is absolutely right result.

This could look like this:

```
painter
```

```
.BeginComplex()
```

```
/*draw polygon 1*/
```

```
./*fill polygon 1*/  
./*draw polygon 2*/  
./*fill polygon 2*/  
./*draw polygon 3*/  
./*fill polygon 3*/  
.EndComplex()  
;
```

Pavel, What about just round your values and draw with integers coordinates for do not use sub-pixel?

Subject: Re: Painter bug?

Posted by [Mindtraveller](#) on Sun, 13 Mar 2011 17:59:19 GMT

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

The problem is scale factor here. I've mentioned above that with dynamic scale factor vertex can't be guaranteed to be aligned to physical pixels.
