Subject: High Performance Drawing
Posted by cbpporter on Sun, 16 Dec 2007 11:48:52 GMT

View Forum Message <> Reply to Message

Hi!

I need high performance flicker free graphics. The high performance part is relativelly covered, but not the flicker free one.

Code that draws stuff when the mouse is moved is pretty flickery, even when drawn to an Image first (x and y are updated by mouse move. The issue is most apparent when using larger windows and resolutions:

```
void aggtest::Paint(Draw& draw)
{
 if (GetSize().cx == 0 || GetSize().cy == 0)
  return;
 ImageDraw agdraw(GetSize());
 agdraw.DrawRect(GetSize(), White);
 agdraw.DrawLine(x,y,150,150,50,Blue);
 Image img(agdraw);
 draw.DrawImage(0,0,GetSize().cx,GetSize().cy,img);
}
```

I don't think that this is a buffer issue, it is rather and issue with screen refresh synchronization. Does anybody know a way to reduce this flicker under Windows/Linux. I don't want to use OpenGL or DirectX to wait for hardware sync with the monitor.

Also, creating a buffer at every paint seems rather costly for large resolutions. Isn't there a way to keep a buffer pre-allocated and then obtain Image objects out of it with non-destructive copy?

And the documentation about ImageDraw is inaccurate:
Quote:
Image img(100, 100);

ImageDraw idraw(img);

myobject.Paint(idraw);

idraw.Close();

// now the Image has been modified by the myobject's drawing

operations and can be e.g. saved to disk

ImageDraw does not have a ImageDraw(const Image&) constructor or a Close() method.

Furthermore, I couldn't find a way to create an ImageDraw object starting from an existing Image since the above methods are not present.