Subject: DrawImage scaling
Posted by Coder on Sun, 31 Dec 2006 02:27:40 GMT
View Forum Message <> Reply to Message

Does Draw::DrawImage support point sampled (non-filtered) scaling now or in the future maybe?

Subject: Re: DrawImage scaling
Posted by mirek on Sun, 31 Dec 2006 08:32:25 GMT
View Forum Message <> Reply to Message

No. Support was considered at one point, but issue later remained silent, so it looks like it is not really needed much.

You can always use your routine. If printing is concern, you can register your own DrawData routine and get the same level of support as it was supported internally.

Mirek

Subject: Re: DrawImage scaling
Posted by Coder on Tue, 02 Jan 2007 01:43:39 GMT
View Forum Message <> Reply to Message

The reason I brought it up, is that most image editors (Gimp, Photoshop, Paintshop Pro, etc) and most image viewers, use nonfiltered zooming, to allow users to see the pixels as they are. In any case, thanks for responding!

Subject: Re: DrawImage scaling
Posted by nixnixnix on Mon, 07 May 2007 19:34:51 GMT
View Forum Message <> Reply to Message

I need this too. Is the only alternative that one needs to draw a rectangle to represent each pixel?

Cheers,

Nick

Subject: Re: DrawImage scaling
Posted by mirek on Tue, 08 May 2007 17:21:47 GMT
View Forum Message <> Reply to Message

nixnixnix wrote on Mon, 07 May 2007 15:34I need this too. Is the only alternative that one needs to draw a rectangle to represent each pixel?

Cheers,

Nick

If "zoom" is all you need, it is trivial to implement:

(IconDes/ImageOp.cpp 170):

```
Image Magnify(const Image& img, int nx, int ny)
{
 if(nx == 1 && ny == 1)
  return img;
 if(nx == 0 || ny == 0)
  return Image();
 Size sz = img.GetSize();
 bool xdown = nx < 0;
 nx = abs(nx);
 int ncx = xdown ? sz.cx / nx : sz.cx * nx;
 ImageBuffer b(ncx, sz.cy * ny);
 const RGBA *s = ~img;
 const RGBA *e = s + img.GetLength();
 RGBA *t = ~b;
 while(s < e) {
  RGBA *q = t;
  const RGBA *le = s + sz.cx;
  while(s < le) {
   Fill(q, *s, nx);
   q += nx;
   s++;
  }
  for(int n = ny - 1; n--;) {
   memcpy(q, t, ncx * sizeof(RGBA));
   q += ncx;
  }
  t = q;
 }
 return b;
}
```

BTW, if you are about to do any image processing, IconDes is quite good studying material...

Mirek