Subject: Re: Inverse palette conversion algorithm...
Posted by mr_ped on Fri, 07 Apr 2006 08:05:36 GMT

View Forum Message <> Reply to Message

I'm not sure what's this about and supposed to do,
I just catched up the "r = 255 * r / (RASTER_MAP_R - 1);" lines...

Why to do this with such expensive calculation? (1 multiply + 1 divide)

```
enum {
   ...
   RASTER_SHIFT2_R = ((4-RASTER_SHIFT_R)*2),
   RASTER_SHIFT2_G = ((4-RASTER_SHIFT_G)*2),
   RASTER_SHIFT2_B = ((4-RASTER_SHIFT_B)*2),
};

r = (r<<RASTER_SHIFT_R) + (r>>RASTER_SHIFT2_R);
g = (g<<RASTER_SHIFT_G) + (g>>RASTER_SHIFT2_G);
b = (b<<RASTER_SHIFT_B) + (b>>RASTER_SHIFT2_B);
```

... These are less expensive for older CPU (and at least of the same speed on modern ones), and provide a better distribution of converted values IMHO.

Than again I'm curious about the whole purpose of this piece of code.