

Color-aware ClearType requires access to fixed background pixels, which is a problem if you don't know what the background pixels are, or if they aren't fixed

 devblogs.microsoft.com/oldnewthing/20150129-00

January 29, 2015



Raymond Chen

ClearType is a technology that blends text pixels with background pixels with the goal of reducing visual artifacts during text rendering.¹ If you go for the full-meal version of ClearType, it uses knowledge about the physical properties of LCD screens in order to carry out its work. Some time ago, I noted one catch with this model, which is that overdraw gives the ClearType algorithm the wrong background pixels. Another gotcha is that if you don't know what the background pixels are at all, then you can't use ClearType. For example, you might be rendering a semi-transparent bitmap that will be drawn on top of other content. If you try to use ClearType to render text onto this bitmap, the ClearType engine will see transparent pixels as the background, and blend accordingly. (My guess is that it will treat them as black pixels.) But when you actually draw this bitmap to the screen, those transparent pixels will allow the pixels below the bitmap to shine through, and those underlying pixels are not transparent. The result is color fringes because the ClearType engine was given incorrect background pixels. Another assumption that the ClearType engine makes is that the bitmap will be drawn on exact pixel boundaries without any stretching or shrinking or rotation. Since ClearType is doing math based on the physical LCD, you break the ClearType model if you scale the bitmap, rotate it, or render it on a fractional-pixel boundary because each carefully-crafted ClearType pixel does not end up as a single pixel on the LCD panel. (The mapping of original pixels to transformed pixels is controlled by something called an *interpolation mode*. For example, Direct2D offers a variety of interpolation modes.) If you break this rule and use ClearType anyway, you will once again get the dreaded color fringes. Nearly all of these factors come into play on the Windows 8 Start screen. The word *Start* on that page is rendered onto a transparent bitmap because it needs to overlay on top of your Start background, and most of the Start backgrounds are not fixed; they scroll slowly as you pan through your tiles. The tiles themselves translate when you pan the Start screen, and they scale when you go into Customize mode or when you press them, and they rotate when you tap on them. This means that any text drawn onto a tile cannot use the color-aware version of ClearType. These concerns apply more generally to any bitmap that will be scrolled. A bitmap that scrolls with subpixel positioning cannot use the

color-aware version of ClearType because ClearType assumes integer-pixel positioning. (This is why Internet Explorer doesn't use color-aware ClearType. Scrolling is performed with Direct Manipulation, and Direct Manipulation does subpixel scrolling.) Now, of course, you could work around this problem. You could design your interface so that it doesn't require transparent bitmaps, say by using fixed backgrounds. And you could design your interface so it doesn't use subpixel positioning, scaling, or rotation. Or you could simply stop pre-rendering text and instead rerender them on-the-fly each time something changes. The first two workarounds impair your design. The second impairs your performance, since moving a bitmap is no longer a simple update of a sprite's coordinates and transformation matrix; you have to go through a full text rendering pass, including a pixel read-back in order to figure out what the current background pixels are. (And pixel read-back from video memory is not cheap since it forces the composition tree to flatten.) Both trade-offs are pretty expensive, and the cheaper alternative is usually simply to stop using color-aware ClearType.

¹ Yes, there are people who don't like ClearType. The point of today's article isn't about whether ClearType is good or bad; it's about ClearType's limitations. I tried to remain neutral on the subject by saying that improved text rendering is the *goal* of ClearType, making no statement about whether it *achieves* that goal.

Raymond Chen

Follow

