How to paint a translucent (not transparent) rectangle

> ***QUESTION***
>
> I want to have a background image and then draw a translucent, red rectangle over it. The effect would then be like looking through a piece of red glass. How to achieve that?

There are a number of different techniques, which vary in the overall effect. A simple algorithm, that doesn't model specular reflection or refraction, is demonstrated by this code:

```
procedure DrawTransparentRectangle(Canvas: TCanvas; Rect: TRect;
  Color: TColor; Transparency: Integer);
var
  X: Integer;
  Y: Integer;
  C: TColor;
  R, G, B: Integer;
  RR, RG, RB: Integer;
begin
  RR := GetRValue(Color);
  RG := GetGValue(Color);
  RB := GetBValue(Color);
  for Y := Rect.Top to Rect.Bottom - 1 do
  for X := Rect.Left to Rect.Right - 1 do
    begin
      C := Canvas.Pixels[X, Y];
      R := Round(
        0.01 * (Transparency * GetRValue(C) + (100 - Transparency) * RR)
      );
      G := Round(
        0.01 * (Transparency * GetGValue(C) + (100 - Transparency) * RG)
      );
      B := Round(
        0.01 * (Transparency * GetBValue(C) + (100 - Transparency) * RB)
      );
      Canvas.Pixels[X, Y] := RGB(R, G, B);
    end;
end;
```

This routine is meant to illustrate the principle; in reality, you'd use something other than the (very slow) *Pixels[]* property to access the individual pixels of the canvas. For example, if you were dealing with bitmaps, you could use the *Scanline* property.

The *Transparency* parameter ranges from 0 (completely opaque) to 100 (completely transparent). With this simple algorithm, transparency values greater than 50 work best. Note that this algorithm is non-physical. The results are not what you'd get with a real piece of colored glass.

Original resource:	The Delphi Pool
Author:	Steve Schafer
Added:	2013-01-27
Last updated:	2013-01-27