

Color representation in VB

RGB colors Color is represented as a long integer, a combination of red, green and blue components. Each of these primary colors has an allowable numeric range of 0 to 255 (2^8) If all 3 values are 255, the resultant color is white.

RGB syntax: RGB(red,green,blue)

This function computes the color value with the formula:

Color= blue*&HFF00& + green*&HFF& + red

You can decompose a color from its long integer representation:

r = color AND &HFF&
g = (Color AND &HFF00 \ &H100
b = (Color AND &HHFF0000 \ &H1000

Each color occupies exactly 2 hexadecimal digits.
Remember the predefined colors in VB:

VbBlack is &H0
VbRed is &HFF
VbGreen is &HFF00
VbYellow is &HFFFF
VbBlue is &HFF0000
VbMagenta is HFF00FF
VbCyan is &HFFFF00
VbWhite is &HFFFFFF

If you use the "Point" to get a pixel color and the pixel doesn't exist, a -1 is returned (&HFFFFFFF)

VB also defines 16 colors in the standard palette by a number code, used in DOS QuickBasic. Convert to the Color function using

Color = QBColor(code)

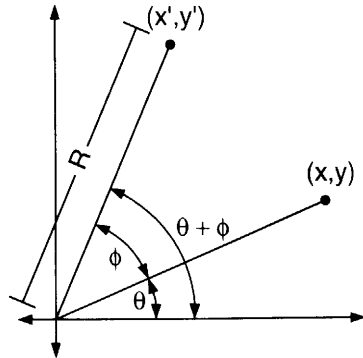
where 0 = black, 1 = blue, 2 = green, 3 = cyan, 4 = red, 5 = magenta, 6 = yellow, 7 = white, 8 = gray, 9 = light blue, 10 = light green, 11 = light cyan, 12 = light red, 13 light magenta, 14 = light yellow, and 15 = bright white.

Example of RGB use: Line(0,0)-(500,500), RGB(255,127,127),BF

Contrast: For a color image, look at each color's pixel brightness, red, green, blue from 0 to 255. A Grayscale image has equal pixel values for these three colors. Contrast enhancement is accomplished by 'stretching' the histograms.

Image Transformations

Rotating an image: map pixels from a source image to a destination image. Rotate the point (x,y) around the origin through the angle ϕ . The distance R from (cx,cy) to (x,y) is



$R = \sqrt{dx^2 + dy^2}$. Then the angle $\theta = \arctan(y/x)$. The new point's position (x', y') can then be solved as

$$\begin{aligned} X' &= R * \cos(\theta + \phi) \\ Y' &= R * \sin(\theta + \phi) \end{aligned}$$

Note that the size of a rotated image is usually taller and wider than the original. If the image's original width and height are W and H , then the rotated image area is

$$\begin{aligned} \text{Width} &= H * \sin(\theta) + W * \cos(\phi) \\ \text{Height} &= H * \cos(\theta) + W * \sin(\phi) \end{aligned}$$

When rotating generated curves (like an ellipse) by θ radians, transform the x and y coordinates of each point

$$\begin{aligned} \text{New } x(t) &= x(t) * \cos(\theta) \\ \text{New } y(t) &= y(t) * \sin(\theta) \end{aligned}$$

Two-dimensional transformations:

Translation: add an appropriate offset to the x and y coordinates of the points that make up an object. For example, to move an object 10 units (twips, millimeters, whatever) in the positive x direction and 3 units in the negative y direction, add 10 to the x coordinate and subtract 3 from the y coordinate:

$$\begin{aligned} \text{New}_x &= \text{old}_x + x_offset \\ \text{New}_y &= \text{old}_y + y_offset \end{aligned}$$

Scaling: to scale or stretch an object in the x direction, multiply the x coordinate of each of the object's points by some number. For example, to make an object twice as wide, multiply it by two.

$$\begin{aligned} \text{New}_x &= \text{old}_x * x_scale \\ \text{New}_y &= \text{old}_y * y_scale \end{aligned}$$

Rotation: rotate a point around the origin by an angle θ is

$$\begin{aligned} \text{New}_x &= \text{old}_x * \cos(\theta) - \text{old}_y * \sin(\theta) \\ \text{New}_y &= \text{old}_y * \sin(\theta) + \text{old}_x * \cos(\theta) \end{aligned}$$

VB coordinate systems

Xcoordinate: distance from left edge of drawing area to a point

Ycoordinate: distance from top edge of drawing area to a point

Standard coordinate systems: Set drawing objects ScaleMode property to:

VbTwips 1440 twips/inch, or 20 twips/point

VbPoints 72 printer's points per inch

VbPixels smallest displayable dot on the screen

VbCharacters characters measure 12 points (240 twips) vertically, 6points (120 twips)

horizontally (note in this coordinate system, horiz and vert spacing isn't the same)

Vb Inches, vbMillimeters, vbCentimeters

Regardless of coord system chosen, all values are converted to pixels.