**COLOR in WINBGIM**

**Colors for Windows BGI**

The [winbgim](http://www.cs.colorado.edu/~main/bgi/doc/bgi.html) package supports two types of colors that may be used with any of the functions that expect colors as arguments:

1. The sixteen ordinary BGI colors. These are the integers 0 through 15 or you may use the symbolic names:

BLACK BLUE GREEN CYAN

RED MAGENTA BROWN LIGHTGRAY

DARKGRAY LIGHTBLUE LIGHTGREEN LIGHTCYAN

LIGHTRED LIGHTMAGENTA YELLOW WHITE

1. A color may be specified from red, green and blue components using a new function called COLOR(r,g,b). Each of the r,g,b arguments must be a number in the range 0 to 255. For example, COLOR(255,100,0) is a mostly red color with some green and no blue. If you create one of these colors, it may be used as an argument to any of the BGI functions that expect a color. These colors may also be returned from BGI functions such as getbkcolor and the new function getdisplaycolor (which tells you what actual color will be displayed on the current monitor).

A function, converttorgb, and several other functions (RED\_VALUE, GREEN\_VALUE, BLUE\_VALUE, IS\_BGI\_COLOR, and IS\_RGB\_COLOR) are explained in the examples below.

**RGB Examples:**

setcolor(BLUE); // Change drawing color to BLUE.

setcolor(COLOR(255,100,0)); // Change drawing color to reddish-green.

setpalette(4, BLUE); // Change palette entry 4 to BLUE.

setpalette(4, COLOR(9,9,9));// Change palette entry 4 to nearly black.

int current = getcolor( ); // Set current to current drawing color.

if (IS\_BGI\_COLOR(current)) // Check whether it is a BGI color.

cout << "Current BGI drawing color is: " << current << endl;

if (IS\_RGB\_COLOR(current)) // Check whether it is an RGB color.

cout << "Current RGB drawing color has these components:\n"

<< "Red: " << RED\_VALUE(current) << '\n'

<< "Green: " << GREEN\_VALUE(current) << '\n'

<< "Blue: " << BLUE\_VALUE(current) << '\n';

cout << "The usual Windows RGB color int value is:\n"

<< converttorgb(current) << endl;