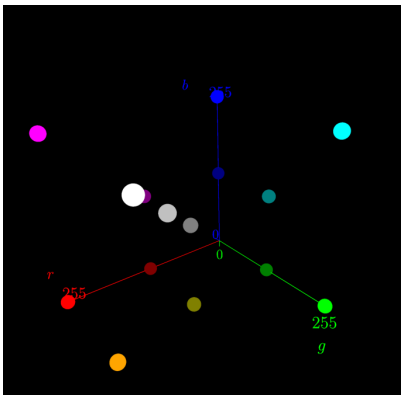
Mapping an RGB color to the nearest palette color in ColdFusion

Posted At : January 22, 2013 6:22 AM | Posted By : William Eatman
Related Categories: coldfusion



I ran into a need to map an RGB value to a palette color the last time I worked with the Google calendar API. I kept getting errors back when creating new calendars, and realized that I needed to pass a hex color that was in a defined palette. This left me with fewer options for the end-user, as I would now have to present them with a list of valid colors. I could avoid some type of palette chooser by mapping a given RGB value to the nearest Google color. In my situation an exact color match was not necessary.

Last week I posted an article with a function that would return the 17 CSS 2.1 colors. The code for the function was simple, but I needed it for this article. I actually thought about being cruel and writing the function to return the web safe color palette.

Mapping an RGB value to a palette color is a problem of finding the palette color that is the nearest in distance to a given RGB value. We can visualize this as shown on the left by plotting each palette color in 3d space using each color component as an axis.

We can get the Euclidean distance between two points by using the formula shown below that returns the distance between points P and Q.

$$ d(p,q) = \sqrt{(p_r - q_r)^2 + (p_g - q_g)^2 + (p_b - q_b)^2} $$

Here is the same equation in CF code.

```
<cfset d = SQR( ((p.r - q.r) ^ 2) + ((p.g - q.g) ^ 2) + ((p.b - q.b) ^ 2))
```

Now it's just a matter of iterating through the palette colors to find the shortest distance. Below is sample output from a program that generates a random RGB value and finds the nearest CSS 2.1 palette color.

Random Color - RGB(29,100,92)

Nearest Palette Color - RGB(0,128,128)

Live Demo

Full Source

```
<cfset randomRGB = {
  r = RandRange(0, 255)
 ,g = RandRange(0, 255)
 ,b = RandRange(0, 255)
} />

<cfoutput>
<strong>Random Color - RGB(#randomRGB.r#,#randomRGB.g#,#randomRGB.b#)</strong>
<div style="width: 150px; height: 150px; background: rgb(#randomRGB.r#,#randomRGB.g#,#randomRGB.b#)"> </div>
</cfoutput>

<cfset qCSS21Colors = getCSS21Colors()>

<!--- this should be the greatest distance possible --->
<cfset leastDistance = 444>

<cfloop query="qCSS21Colors">
 <cfset CSS21RGB = hexToRGB(qCSS21Colors.color)>

 <!--- get the Euclidean distance --->
     <cfset distance = SQR( ((CSS21RGB.r - randomRGB.r) ^ 2) + ((CSS21RGB.g - randomRGB.g) ^ 2) + ((CSS21RGB.b - randomRGB.b) ^ 2) )>

     <!--- if a closer palette color is found, save it --->
     <cfif distance LT leastDistance>
         <cfset leastDistance = distance>
         <cfset nearestColorRGB = CSS21RGB>
     </cfif>
</cfloop>
```

```
<br />
<cfoutput>
<strong>Nearest Palette Color - RGB(#nearestColorRGB.r#,#nearestColorRGB.g#,#nearestColorRGB.b#)</strong>
<div style="width: 150px; height: 150px; background: rgb(#nearestColorRGB.r#,#nearestColorRGB.g#,#nearestColorRGB.b#)"> </div>
</cfoutput>
```