Subject: DarkThemeCached
Posted by Didier on Thu, 20 May 2021 21:11:45 GMT
View Forum Message <> Reply to Message

Hello,

Looking at Pradip's code I saw this DarkThemeCached()
What is it supposed to do ?

Especially this line:
cache.ii = (cache.ii + 1) & (N - 1);

Subject: Re: DarkThemeCached
Posted by Oblivion on Thu, 20 May 2021 21:44:22 GMT
View Forum Message <> Reply to Message

Hello Didier,

That line is supposed to return the remainder of (cache.ii + 1) / N operation, namely (cache.ii + 1) % N (=8) Its function is to clamp the index between 0-7.
A known microptimization trick, if the ii is a positive integer and the N is a power of 2.

(Apparently, colors are cached in thread_local slots))

Best regards,
Oblivion

Subject: Re: DarkThemeCached
Posted by Didier on Thu, 20 May 2021 23:34:37 GMT
View Forum Message <> Reply to Message

Ok for the line, I did'n know this trick.

But since there are only 8 slots .... it seems to me there is a risk of overlap letting this beeing used freely ?
I still don't uderstand how this function can be used ?

I am really curius

Subject: Re: DarkThemeCached
Posted by Oblivion on Thu, 20 May 2021 23:52:36 GMT

View Forum Message <> Reply to Message

To my understanding,

It keeps a little map (icolors = input colors, ocolors = output colors, total: 8 colors);

When you call DarkThemeCached(c)

1) It looks up c in icolors.(keys)
2) If it finds it, it returns the corresponding alternate (dark) theme color, ocolor (values).
3) If it couldn't find it in the keys, then calls DarkTheme(c) to calculate the dark theme color (ocolor). (relatively expensive)
4) Then it puts the new dark theme color variant into the map value and returns it.

Now, probably the confusing part is the index: "cache" structure seems to keep track of the current position, so that it wouldn't overwrite the last cached and used color unless it is necessary. That line with modulo operation is apparently for this purpose.
It will start over from index position 0 only if it is iterated enough (>= 8), i, e. previous lookups failed to find their respective color 'c' value among the keys.

Keep in mind that this is what I see in this piece of code. Not the official explanation. :)

Best regards,
Oblivion

Subject: Re: DarkThemeCached
Posted by Didier on Fri, 21 May 2021 07:21:36 GMT
View Forum Message <> Reply to Message

Hello Oblivion,

Thanks for the explanations.

If you are using many colors, I don't think using this is a good idea : the cache will be overlapping all the time

Subject: Re: DarkThemeCached
Posted by mirek on Thu, 27 May 2021 07:21:01 GMT
View Forum Message <> Reply to Message

Didier wrote on Fri, 21 May 2021 09:21Hello Oblivion,

Thanks for the explanations.

If you are using many colors, I don't think using this is a good idea : the cache will be overlapping all the time

That is why this is separate function from DarkTheme(Color)... You are supposed to use it only where there is a chance of speedup. Actually, the only proper use in the whole uppsrc is in the RichText, where it is used to convert the text color. Usually you do not have that many text colors...

All that said, it is entirely possible that this could be improved somehow. But I just wanted tiny simple cache for RichText...

Mirek

Subject: Re: DarkThemeCached
Posted by Oblivion on Thu, 27 May 2021 07:59:41 GMT
View Forum Message <> Reply to Message

Well it is also a perfect fit for situations where the app/widget mainly uses the standard 8 ANSI colors and their brighter counterparts, such as TerminalCtrl. :)
For such cases this function seems like an an optimal solution.

Best regards,
Oblivion
