

---

Subject: Re: styling of widgets ( animation / look and feel)

Posted by [mirek](#) on Thu, 20 Apr 2023 19:51:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

dodobar wrote on Thu, 20 April 2023 19:17 Thanks, however the base classes I provided could still be relevant with a few modifications

To be more in line with what you're suggesting , possibly functions passed to it as the easing

Those are trivialities. Obviously you need some time based transition to animate, but that is not the problem.

Problem is that following code:

Quote:

```
Color start_color = SColorFace();
Color end_color = SColorHighlight();
Color current_color = animation.LerpColor(start_color, end_color);
```

cannot be code. It has to be parametrised so it can be replaced it with any (or no) animation

Think about Button. Its look (and feel) is parametrized as

```
struct Style : ChStyle<Style> {
    Value look[4];
    Color monocolour[4], textcolor[4];
    Point pressoffset;
    int focusmargin;
    int overpaint;
    Font font;
    Image ok, cancel, exit;
    bool transparent;
    bool focus_use_ok;
};
```

Now buton has 4 basic states (normal, hot (when mouse is over), pushed, disabled). Proper animation support most provide transition between any 2 of them (that is 12 transitions if I count right) while allowing to set such animation in Style somehow and also while not singificantly complicating Option::Draw. Ideally, animation code, if any, should be completely outside of Option code.

EDIT: Moreover, it needs correct transitions between partial states (e.g. user moves mouse over, so hot state ends before animation is complete).

