
Subject: ColorScheme for individual cells

Posted by [piotr5](#) on Sat, 05 May 2012 11:57:02 GMT

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

I had to modify ArrayCtrl.h and ArrayCtrl.cpp, to achieve what I want:

in ArrayCtrl.cpp I changed the lines marked with "+":

```
const Display& ArrayCtrl::GetCellInfo(int i, int j, bool f0,
                                     Value& v, Color& fg, Color& bg, dword& st)
{
...
}
if(hasfocus)
    st |= Display::FOCUS;
+ bg = BGColorInfo(i,j);
if(nobg)
    bg = Null;
+ fg = FGColorInfo(i,j);
if((st & Display::SELECT) ||
    !multiselect && (st & Display::CURSOR) && !nocursor ||
    drop) {

...

ArrayCtrl::ArrayCtrl() {
+ ColorScheme=NULL;
```

in ArrayCtrl.h I added to the class ArrayCtrl:

```
public:
struct BGFGFunc {
    virtual Color& BG (const ArrayCtrl* a, int i, int j) const=0;
    virtual Color& FG (const ArrayCtrl* a, int i, int j) const=0;
};
```

```
BGFGFunc *ColorScheme;
```

```
protected:
const Color& BGColorInfo(int i, int j) const
    {return ColorScheme ? ColorScheme->BG(this,i,j) :
      (i & 1 ? evenpaper : oddpaper);}
const Color& FGColorInfo(int i, int j) const
    {return ColorScheme ? ColorScheme->FG(this,i,j) :
      (i & 1 ? evenink : oddink);}
```

has anybody suggestions how to do that any better?

as for application, I'd like to add duplicate entries into the array, and when they get selected all duplicates get selected too. but of course the user would like to see up front if a duplicate exists, so I colour them differently...

I did think about achieving the same with some richer text-display, but replacing the display for each cell seems a bit of overkill for this simple goal. any other ideas?

Subject: Re: ColorScheme for individual cells

Posted by [nixnixnix](#) on Mon, 07 May 2012 02:29:27 GMT

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

Hi,

It is not overkill to assign a Display object to every individual cell. I use ArrayCtrl routinely with over a dozen columns and around 200,000 rows and I assign a custom Display object to almost every individual cell with no noticeable overhead. I have a couple of dozen of these ArrayCtrl objects in my app and it runs in a small amount of memory and fast.

My advice is to leave the UPP code alone. Whatever you can't achieve with Display can almost certainly be done by also using Convert.

All the best,

Nick

Subject: Re: ColorScheme for individual cells

Posted by [piotr5](#) on Thu, 17 May 2012 11:09:50 GMT

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

You're right, it isn't overkill in terms of ressources. what I meant however is it's overkill in terms of maintenance. one would need to copy the upp code for text-display into an own function and alter those lines. if upp-code changes you'd also need to change that code to keep appearance constant over all array-controls. in short, bad code-reuse.

imho arrayctrl needs some changes anyway. for example there's no callback for after editing a cell, only one for the case of successful editing. all the deprecated cursor-callbacks should be re-considered by adding an actual cursor in addition to the selection. (graphically a border around the current cell and/or a seperator-line between rows during drag'n'drop.)

the idea behind a color-scheme was to do such things like changing colour every 5 lines or removing the column-lines and display the control as a checker-board. and as I said, I'd like to add

redundant rows and grey them out so the user knows they aren't important. but the point is I'd need individual Display for rows and not for columns. that might be an alternative way of solving my problem except that it would be more rarely needed than for example changing colour every 5 rows.

I were asking for opinions on my implementation. for example maybe it would be better to use an actual colouring-scheme class passed as a template-parameter to arrayctrl. allows for inlining the actual colour-decision. but somehow this trick is rarely used in upp ctrl-derived classes...

Subject: Re: ColorScheme for individual cells
Posted by [nixnixnix](#) on Thu, 17 May 2012 16:10:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

I don't see why you would need to change UPP code but perhaps I don't fully appreciate what you're trying to do. I override the Display class and add it to the ArrayCtrl using ArrayCtrl::SetDisplay().

I'll have to step aside and let someone with more UPP chops take it from here as I clearly don't get the issue. Seems to me like you can easily do everything you need to with Display.

Best regards,

Nick
