Subject: Display transparent paper
Posted by Silvan on Sat, 29 Jan 2022 19:38:25 GMT
View Forum Message <> Reply to Message

There is a simple way to draw text on a display with transparent background?
And to set font size?

Thanks

Subject: Re: Display transparent paper
Posted by Lance on Sat, 29 Jan 2022 23:09:37 GMT
View Forum Message <> Reply to Message

If I want to change the background to something I specify, I just

draw.DrawRect(arect, acolor);

without this, the text will be drawn on whatever the current background is. Is this the transparent background you meant? So do nothing, you get a transparent background.

As for font.

Draw::DrawText(int x, int y, const String& text, Font font = StdFont(),...)

Just specify the font you intended there.

HTH,
Lance

Subject: Re: Display transparent paper
Posted by Silvan on Sun, 30 Jan 2022 08:34:02 GMT
View Forum Message <> Reply to Message

Thak you, I specify better my question-
I want to use the alignment capacity of Display and I use this:

Display::Paint(Draww,Rectrc,AttrText(sometext).Center(),Red, White,0);

White is the paper color, but I want to write on the existing background (transparent).

I finally used something like this:

Size size = GetTextSize(sometext, font);
Draw::.DrawText((sz.cx-size.cx)/2, (sz.cy-size.cy)/2, sometext, font, Red);

I'm wandering if there exist I direct way to use Display.

Bye
Silvano

Subject: Re: Display transparent paper
Posted by Lance on Sun, 30 Jan 2022 17:14:52 GMT
View Forum Message <> Reply to Message

Silvan:

Dig a little into Upp source code, I find something like this

```
void Display::PaintBackground(Draw& w, const Rect& r, const Value& q,
                              Color ink, Color paper, dword style) const
{
 if(IsType<AttrText>(q)) {
  const AttrText& t = ValueTo<AttrText>(q);
  if(!IsNull(t.paper))
   paper = t.paper;
  if(!IsNull(t.normalpaper) && !(style & (CURSOR|SELECT|READONLY)))
   paper = t.normalpaper;
 }
 w.DrawRect(r, paper);
}

void Display::Paint(Draw& w, const Rect& r, const Value& q, Color ink, Color paper, dword style)
const
{
 PaintBackground(w, r, q, ink, paper, style);
 Single<StdDisplayClass>().Paint0(w, r, q, ink, paper, style);
}
```

Do you want to try set both paper for AttrText and the Paint parameter to Null and see if you get what you want. Obviously when you set the paper color of AttrText to Null, then the paper color of passed in parameter to funciton Display::Paint will be used. I am not quite sure if set it to Null will make a do nothing DrawRect. I think its the only reasonable result, but before actually test it, I cannot really tell for sure.

HTH,
Lance

Subject: Re: Display transparent paper

Posted by Lance on Sun, 30 Jan 2022 17:20:27 GMT
View Forum Message <> Reply to Message

Yes, it should work as you wanted.

Test code

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;

struct App:TopWindow
{
 void Paint(Draw& w)override
 {
  w.DrawRect(GetSize(), Cyan());
//  w.DrawRect(GetSize()/2, SColorPaper());
  w.DrawRect(GetSize()/2, Null); // Do nothing, transparant background achieved.
 }
};

GUI_APP_MAIN
{
 App().Run();
}
```

Subject: Re: Display transparent paper
Posted by Silvan on Sun, 30 Jan 2022 18:15:23 GMT
View Forum Message <> Reply to Message

Yes it works. Thank you

```
 Display::Paint(aDraw,aRect,AttrText(sometext).Center().SetFo nt(Arial(40)),Red,Null,0);
```

The answer is to set to Null the paper color.

Ps:
I have to declare the Display object, but I read somewhere that it is static.
How can I use it directly without declaraction?

Like this:

```
Single<StdDisplayClass>().Paint(w, r, q, ink, paper, style); // don't compile
```

Subject: Re: Display transparent paper
Posted by Lance on Sun, 30 Jan 2022 22:17:28 GMT
View Forum Message <> Reply to Message

Please take a look at <Reference/ArrayCtrl>

In particular:

```
ArrayCtrlExample::ArrayCtrlExample()
{
	...
 array.AddColumn("with display").SetDisplay(Single<MyDisplay>());
	...
}
```

Subject: Re: Display transparent paper
Posted by Lance on Sun, 30 Jan 2022 22:23:28 GMT
View Forum Message <> Reply to Message

I have a feeling my answer is not what you're looking. However you can investigate ArrayCtrl code that uses Display* display;
to standardize painting work. Adapt it to your particular situation.

Or you can upload a minimal test package, not necessarily compile-able. Explain in comments what you intend to achieve.

Subject: Re: Display transparent paper
Posted by Silvan on Mon, 31 Jan 2022 11:20:00 GMT
View Forum Message <> Reply to Message

Thank you.

I refer to the content of the last paragraph of that page:
https://www.ultimatepp.org/src$Draw$Display_en-us.html

It says:
"Standard Displays are implemented as "functional globals" - functions returning constant reference to single global Display instance."

So I suppose it is possible to use StdDisplay without declaration, like me and you did.

Subject: Re: Display transparent paper
Posted by Oblivion on Mon, 31 Jan 2022 16:47:43 GMT
View Forum Message <> Reply to Message

Hello Silvan,

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

struct MyApp : TopWindow {
 MyApp()
 {
  Sizeable().Zoomable().CenterScreen().SetRect(0, 0, 640, 480);
 }

 void Paint(Draw& w) override
 {
  Size sz = GetSize();
  w.DrawRect(sz, Red());
  StdCenterDisplay().Paint(w, sz, "Centered", White, Null, 0L);  // <-- There are ready-made
displays which you can use in-place (they are functional globals)
 }

};

GUI_APP_MAIN
{
 MyApp().Run();
}
```

You can also create your own Display objects. See uppsrc/Draw/Display.cpp for details.

Best regards,
Oblivion

Subject: Re: Display transparent paper
Posted by Silvan on Mon, 31 Jan 2022 17:46:59 GMT
View Forum Message <> Reply to Message

Thank you!
yes, simple and direct!

Best Regards
Silvano