Subject: Showing bitmaps or Icons
Posted by gprentice on Sat, 26 Nov 2005 07:55:20 GMT
View Forum Message <> Reply to Message

Where would I look for info on how to load and show a .bmp bitmap file or an icon (given an icon handle HICON on Win32) and then get mouse events (mouse move, mouse click) for this image?

Second question : if there's a short answer, any hints on how to do this?

e.g. suppose I want a customised checkbox - I create two bitmaps (checked and unchecked) that I want to switch when the image is clicked.

Last question : I recall mention of a grid component created by Daniel. Is this part of UPP? If not, it's ok. Just wondering.

Graeme

Subject: Re: Showing bitmaps or Icons
Posted by mirek on Sat, 26 Nov 2005 09:33:42 GMT
View Forum Message <> Reply to Message

Quote:
Where would I look for info on how to load and show a .bmp bitmap file or an icon (given an icon handle HICON on Win32) and then get mouse events (mouse move, mouse click) for this image?

Load it in to the Image, then create your own Ctrl, paint it in the Paint an react to the MouseMove/LeftDown etc...

To load it into the Image, use BmpEncoder. Note: this part a little bit complex, as Tom designed the interface (sorry for this little retaliation Tom :):

Image img = BmpEncode::NewBmp()->LoadImage(....

Quote:
suppose I want a customised checkbox - I create two bitmaps (checked and unchecked) that I want to switch when the image is clicked.

In this case using .bmp is quite ineffective, for U++. Just design your images in the ImageDes (in .iml file) and then use them (in U++, Image is simple value object - use it as it was String.... .iml will provide you a set of "function-constants" that will return your images designed in .iml).

See reference/iml

Quote:I recall mention of a grid component created by Daniel. Is this part of UPP? If not, it's ok. Just wondering.

Not yet. I am lagging  (I have to add his DatePicker first

---

Subject: Re: Showing bitmaps or Icons
Posted by gprentice on Sat, 26 Nov 2005 11:56:06 GMT
View Forum Message <> Reply to Message

Quote:In this case using .bmp is quite ineffective, for U++. Just design your images in the ImageDes (in .iml file) and then use them (in U++, Image is simple value object - use it as it was String.... .iml will provide you a set of "function-constants" that will return your images designed in .iml).

See reference/iml

Interesting, thanks.  I tried the iml example and notice some "flickering effects" in the image when the window is resized.  Even with a non centered image whose position isn't changing when the window is resized, I get a little bit of flickering.  Is this expected?  Any ideas for how to avoid it?  If I drag the window round the screen I don't get any flickering in the image.

I remember flickering was an issue with wxWidgets as discussed here -
http://wiki.wxwidgets.org//wiki.pl?action=browse&id=Flic ker-Free_Drawing&revision=18

I vaguely recall one of the solutions was to avoid erasing.

Graeme

---

Subject: Re: Showing bitmaps or Icons
Posted by mirek on Sat, 26 Nov 2005 12:24:00 GMT
View Forum Message <> Reply to Message

Quote:
Interesting, thanks. I tried the iml example and notice some "flickering effects" in the image when the window is resized. Even with a non centered image whose position isn't changing when the window is resized, I get a little bit of flickering. Is this expected? Any ideas for how to avoid it? If I drag the window round the screen I don't get any flickering in the image.

Well, simple solution is Ctrl::BackPaint. That will make Ctrl painting double-buffered.

However, this is not on by default because of performance issues.

This is the purpose of the "new Draw" research - if the "new Draw" proves to be doable/viable, "native" double-buffering is one of advantages.

---

Subject: Re: Showing bitmaps or Icons
Posted by fudadmin on Mon, 20 Mar 2006 06:45:40 GMT
View Forum Message <> Reply to Message

Why this doesn't work?

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

GUI_APP_MAIN
{    TopWindow w;
     Label l;
     Image img;

     w.Add(l);
 l.SetPos(l.PosLeft(0, 100), l.PosTop(0, 30));

 img = BmpEncoder::NewBmp()->LoadImage("testImg.bmp");
 //img.Exclamation();    //at least this should work?
 l.SetImage(img);

 w.Run();
}
```

And how to check if image was loaded?

---

Subject: Re: Showing bitmaps or Icons
Posted by mirek on Mon, 20 Mar 2006 08:12:38 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Mon, 20 March 2006 01:45Why this doesn't work?

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

GUI_APP_MAIN
{    TopWindow w;
     Label l;
```

```
    Image img;

    w.Add(l);
 l.SetPos(l.PosLeft(0, 100), l.PosTop(0, 30));

 img = BmpEncoder::NewBmp()->LoadImage("testImg.bmp");
 //img.Exclamation();    //at least this should work?
 l.SetImage(img);

 w.Run();
}
```

And how to check if image was loaded?

Are you sure that path is correct? (Well, there is no path, so it likely is not).

If it was not loaded, result will be empty (e.g. GetSize() == Size(0, 0))

Mirek

---

Subject: Re: Showing bitmaps or Icons
Posted by fudadmin on Mon, 20 Mar 2006 13:27:13 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 20 March 2006 08:12fudadmin wrote on Mon, 20 March 2006 01:45Why this doesn't work?

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

GUI_APP_MAIN
{    TopWindow w;
     Label l;
     Image img;

     w.Add(l);
 l.SetPos(l.PosLeft(0, 100), l.PosTop(0, 30));

 img = BmpEncoder::NewBmp()->LoadImage("testImg.bmp");
 //img.Exclamation();    //at least this should work?
 l.SetImage(img);

 w.Run();
}
```

And how to check if image was loaded?

Are you sure that path is correct? (Well, there is no path, so it likely is not).

If it was not loaded, result will be empty (e.g. GetSize() == Size(0, 0))

Mirek

Are you sure that you know Ultimate++?...

The image loading works this way:

```
GUI_APP_MAIN
{
 TopWindow w;
 Label l1,l2;
 Image img1,img2;

 w.Add(l1);
 l1.SetPos(l1.PosLeft(10, 100), l1.PosTop(10, 30));

 img1= PngEncoder::New()->LoadImageFile("testImg.png");
 l1.SetImage(img1);


 w.Add(l2);
 img2 = BmpEncoder::NewBmp()->LoadImageFile("testImg.bmp");
 l2.SetImage(img2);
 l2.SetPos(l2.PosLeft(150, 100), l2.PosTop(10, 30));

 w.Run();
}
```

---

Subject: Re: Showing bitmaps or Icons
Posted by fudadmin on Wed, 29 Mar 2006 01:26:52 GMT
View Forum Message <> Reply to Message

gprentice wrote on Sat, 26 November 2005 07:55
e.g. suppose I want a customised checkbox - I create two bitmaps (checked and unchecked) that I want to switch when the image is clicked.

Graeme

BTW, you can use:

```
ButtonOption opt;
...
 opt.SetImage( BmpEncoder::NewBmp()->LoadImageFile("OffIMG.bmp"),
 PngEncoder::New()->LoadImageFile("OnIMG.png") );
//you can choose image formats...
//don't forget to adjust sizes accordingly...
```

---

Subject: Re: Showing bitmaps or Icons
Posted by gprentice on Sat, 15 Apr 2006 11:20:06 GMT
View Forum Message <> Reply to Message

Is the code below valid - it seems to work?
gp1 is the name of a bitmap in an .iml file, just like Smiley() in the reference/iml example. What does img1 = gp1() do - does img1 store a copy of the bitmap?

```
GUI_APP_MAIN
{
 App w;
 Label l1;
 Image img1;

 w.Add(l1);
 l1.SetPos(l1.PosLeft(10, 100), l1.PosTop(10, 30));
 img1= gp1();
 l1.SetImage(img1);
 w.Run();
}
```

In the iml example there is
w.DrawImage((sz.cx - isz.cx) / 2, (sz.cy - isz.cy) / 2, Smiley());

What does "Smiley()" do here - does it get the "address" of the Smiley bitmap that has been embedded in the app due to the iml file being part of the project?

Graeme

BTW - in the reference/events example, the Log("Paint") in the Paint function causes continuous events ... I can't remember if this was originally a commented line and I uncommented it myself or not.

Subject: Re: Showing bitmaps or Icons
Posted by mirek on Mon, 17 Apr 2006 16:52:50 GMT
View Forum Message <> Reply to Message

Quote:
gp1 is the name of a bitmap in an .iml file, just like Smiley() in the reference/iml example.  What does img1 = gp1() do - does img1 store a copy of the bitmap?

Think about Image as of any other "value" type like "int", "Date" or "String". Copies are full and cheap (internally there is reference counting mechanism, but that is nothing you should bother you).

Now "Smiley()" is the Image you have designed and stored in .iml - it is Image constant. In fact, in ideal world, its definition would be (in header)

extern Image Smiley;

However, there are two problems with this:

- first, certain platforms (namely Win32) do not support global variables in .dll (not that we are using .dlls all that often, but it is better to be ready).

- second, global variables in C++ has unpleasant problem of "initialization" order. In practice, in constructor of global variable you are never sure which other global variables are already constructed (unless they are in the same .cpp file).

Therefore, instead of global variable, we are (quite often) using  "functional constants". E.g.

const Image& Smiley();

This solves both problems.

Mirek