Subject: PATCH: Image load progress update
Posted by aroman on Wed, 23 Aug 2006 09:06:09 GMT
View Forum Message <> Reply to Message

Loading large images can be slow (1+ sec), and in some applications it's nice to have a responsive UI during this time. I added a progress callback to the image loading code. The interface is the same except that loading functions now optionally take an additional progress callback.

This can be used very simply with:

```
Progress progress_window("Loading big image");
Image im = StreamRaster::LoadFileAny("big_image.jpg",progress_window);
```

Here is an example program that demonstrates both loading images and using the callback:

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

// NOTE: In order to load JPEG images, you need to include the "plugin\jpg" package! Once that
// is included, you are all set to go.

// Example 1: This just loads and displays the file "image_to_load.jpg" in the directory of
// the executable. It probably won't work if you run it without having that file, so I made
// the slightly fancier version below where you can pick a file.
//
// This version is here so that you can see the bare minimum code necessary to load and
// display an image.
class ReallySimpleImageViewer : public TopWindow {
public:
  ReallySimpleImageViewer() {
    // Set up our window to be resizeable
    Zoomable().Sizeable().Title("Simple Image Viewer");

    // Add a control that can display images.
    Label image_display;
    image_display.SetPaintRect(PaintRect(ImageDisplay()));
    image_display.AddFrame(ThinInsetFrame());
    Add(image_display.HSizePosZ(5,5).VSizePosZ(5,5));

    // While loading the image, we'll show a nice progress box so that the user
    // can cancel if they get bored. This is optional! If you leave out the
    // second parameter to "LoadFileAny", it will load the image until it finishes
    // or an error occurs.
    Progress progress_window("Loading image");
    Image im = StreamRaster::LoadFileAny("image_to_load.jpg",progress_window);

    // im will be either the image we tried to load or a null image if the load failed.
```

```
    // We can check for that with "IsNullInstance()"
    if (im.IsNullInstance())
      PromptOK("Could not load image");

    // Display the image. Won't display anything for a null image.
    image_display.SetImage(im);
  }
};

// Example 2: This allows the user to choose an image to load.
class SimpleImageViewer : public TopWindow {
public:
  typedef SimpleImageViewer CLASSNAME; // Necessary for THISBACK()
  MenuBar menu;
  StatusBar status;
  Label image_display;

  SimpleImageViewer() {
    Zoomable().Sizeable().Title("Simple Image Viewer");
    AddFrame(menu);
    menu.Set(THISBACK(MainMenu));
    AddFrame(status);

    image_display.SetPaintRect(PaintRect(ImageDisplay()));
    image_display.AddFrame(ThinInsetFrame());
    Add(image_display.HSizePosZ(5,5).VSizePosZ(5,5));
  }
  void MainMenu(Bar& bar) {
    bar.Add("File",THISBACK(FileMenu));
  }
  void FileMenu(Bar& bar) {
    bar.Add("Open image...",THISBACK(OpenImage));
    bar.Add("Exit",THISBACK(Close));
  }
  void OpenImage() {
    FileSel fs;
    fs.Type("Image files","*.jpg,*.png,*.bmp");
    fs.AllFilesType();
    fs.ActiveDir(GetFileDirectory(GetDataFile("x")));
    if (fs.ExecuteOpen("Select image to load")) {
      String filename = fs.GetFile(0);
      status = Format("Loading [%s]",filename);
      Image im = StreamRaster::LoadFileAny(filename,Progress("Loading image"));
      if (im.IsNullInstance()) {
        status = Format("Failed to load [%s]",filename);
      } else {
        status = Format("Loaded [%s]",filename);
      }
```

```
      image_display.SetImage(im);
    }
  }
};

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

Attached is the patch to add the necessary functionality.

File Attachments

1) upp_image_progress_feature.patch, downloaded 1876 times

Subject: Re: PATCH: Image load progress update
Posted by mirek on Wed, 23 Aug 2006 11:12:47 GMT
View Forum Message <> Reply to Message

Good idea. Added (but I think you have a small error there, I have changed it to:

```
 ImageBuffer b(cx, yy - y);
 RGBA* t = b;
 int y0 = y; // <<<!
 while(y < yy) {
  if(progress(y - y0, yy - y0)) // <<<!
   return Null;
  memcpy(t, ~GetLine(y) + x, cx * sizeof(RGBA));
  t += cx;
  y++;
 }
```

Mirek

Subject: Re: PATCH: Image load progress update
Posted by mirek on Wed, 23 Aug 2006 11:17:16 GMT
View Forum Message <> Reply to Message

P.S.: IsNullInstance is just "implementation" of "IsNull" (IsNull gets there via IsNull template). I have intended that you use

if(IsNull(image))

instead of

if(image.IsNullInstance())

(all in all, you cannot do

int x; if(x.IsNullInstance())

)

BTW, the "Instance" postfix is there because otherwise there is name clash problem in methods od classes that implement IsNull too...

(think

```
struct Foo {
  int x;
  bool IsNull() { return IsNull(x); }
};
```

-> IsNull(x) would be resolved as Foo::IsNull)

Mirek

Subject: Re: PATCH: Image load progress update
Posted by aroman on Thu, 24 Aug 2006 02:09:13 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 23 August 2006 04:12Good idea. Added (but I think you have a small error there, I have changed it to:

Good catch -- thanks, and thanks for the info about IsNull().

- Augusto
