Irrlicht Engine

Official forum of the Irrlicht Engine

Skip to content

Search… Search
Advanced search

- Board index ‹ Other Irrlicht Stuff ‹ Project Announcements
- Change font size
- Print view

- FAQ
- Register
- Login

Irrlicht i18n (Unicode, FreeType, etc.)

Post a reply

Search this topic… Search

140 posts • Page **1** of **10** • **1**, 2, 3, 4, 5 ... 10

Irrlicht i18n (Unicode, FreeType, etc.)

by **Nalin** » Fri Feb 12, 2010 1:52 pm

What is this?
These are a collection of classes I have written that provide Irrlicht the ability to handle/read/write/draw unicode text. Everything here is based off the abilities of the ustring class. The ustring class, CGUITTFont class, and the CStringTable class are all licensed under the zlib license, the same license that Irrlicht uses.

ustring
Download: irrUString.h
ustring is a UTF-16 string class. It can convert to/from UTF-8 and UTF-32. Using an iterator, one can gain access to each unicode character in the string. This is the base class used by everything else. It is heavily documented in doxygen format. It also supports C++0x move semantics for users of Visual C++ 2010 and GCC (when using -std=c++0x). This provides a performance improvement when dealing with a lot of temporary copies (ie, chaining multiple strings together with +).

To use everything else in this post, you must place this file in Irrlicht's include directory and add to irrlicht.h:
#include "irrUString.h"

XML reading/writing
Download: CXMLWriterUTF8.zip
Download: xml_io.patch
-- svn trunk

-- 1.7.2
These are tweaks to the XML reading/writing system. It enables Irrlicht to correctly read any type of UTF encoded XML file and write UTF-8 encoded files. As reading files is built into Irrlicht, a patch for CXMLReaderImpl.h is required. As Irrlicht uses CXMLWriter when serializing elements, a patch for CFileSystem.cpp is required to use the new CXMLWriterUTF8 class.

Add CXMLWriterUTF8.h and CXMLWriterUTF8.cpp to Irrlicht's source directory, add the files to the Irrlicht project, and apply the patch to gain access to this feature.

Multi-byte font rendering

Download: CGUITTFont.zip
-- svn trunk
-- 1.7.2
Requires: FreeType 2
To render multi-byte unicode, CGUITTFont was created. This requires FreeType 2 to work and a font that contains the glyphs you wish to render. The class will return a gui::IGUIFont that you can then bind to Irrlicht's GUI for rendering. The class has full doxygen documentation built-in.

Add CGUITTFont.h and CGUITTFont.cpp to your project to gain access to this ability. Remember, you must be using FreeType 2 in order to draw TrueType fonts.

Multi-language string tables

Download: CStringTable.zip
To help facilitate internationalization, the string table class allows you to change the language of your application just by loading in a specially designed XML file.

Add CStringTable.h and CStringTable.cpp to your project to gain access to this ability.

How does this all work?

Irrlicht makes heavy use of wchar_t, which is a data type that varies in size depending on your platform (16 bits on Windows, 32 bits on Linux, etc.) ustring has the ability to convert to/from a stringw holding its proper, platform specific unicode string (UTF-16 in Windows, UTF-32 in Linux, etc.) You read your data into a ustring and convert it to a stringw when you need to pass it through Irrlicht (like, for example, drawing it on the screen), converting it back to a ustring when it reaches its destination. CGUITTFont will convert the stringw back into a ustring before it draws the text. This allows us to have basic internationalization support without gutting Irrlicht.

How do I convert my ustring to other strings?

cpp Code: Select all

```cpp
ustring test("This is a test \xEF\xBF\xBD");
stringc test_char = test.toUTF8_s();
stringw test_wchar = test.toWCHAR_s();
io::path test_path = test.toPATH_s();
```

How do I use a ustring iterator?

cpp Code: Select all

```cpp
ustring test("This is a test \xEF\xBF\xBD");
test += (uchar32_t)0x10400;
std::cout << "test.size(): " << test.size() << std::endl;
std::cout << "test.size_raw(): " << test.size_raw() << std::endl;
```

```
ustring::iterator end = test.end();
for (ustring::iterator i = test.begin(); i != end; ++i)
{
  uchar32_t character = *i;
  std::cout << "character: " << character << std::endl;
}

// Yes, we can directly output our ustring to std::cout.
std::cout << test << std::endl;

// The iterator is STL compatible, so we can use C++0x and the STL to increment each
character by 1.
std::for_each(test.begin(), test.end(), [](ustring::access& c) { ++c; });

// And we can output to std::wcout too.
std::wcout << test << std::wendl;
```

How do I use CGUITTFont?

cpp Code: Select all

```
// Get a 14 pixel high code2001 font.
gui::CGUITTFont* tt_font = gui::CGUITTFont::createTTFont(device->getGUIEnvironment(),
"code2001.ttf", 14);

// Set the font.
gui::IGUIEnvironment* env = device->getGUIEnvironment();
gui::IGUISkin* skin = env->getSkin();
skin->setFont(tt_font);

...

tt_font->drop();
```

How do string tables work?

A string table will hold all the translated strings for a given language. Your project should have a string table for each language you want to support.

cpp Code: Select all

```
// A string table can be loaded using the loadStringTable() function in the irr::st
namespace.
st::CStringTable* stringtable = st::loadStringTable("example.irrst", device-
>getFileSystem());

// Then we can set it to be the active string table.
st::setActiveStringTable(st);
```

cpp Code: Select all

```
// Alternatively, you can use the CStringTableManager class.
CStringTableManager mgr;

// The CStringTableManager class can load all .irrst files in a directory.
mgr.loadDirectory("translations", device->getFileSystem());

// We can then set the active table by the language of the string table.
mgr.setActiveStringTable("English");
```

cpp Code: Select all

```
// We can get a string directly from a string table.
st::CStringTable* english = st::loadStringTable("english.irrst", device->getFileSystem());

// Both getTranslation and _T will return the string associated with the given ID.
core::ustring username = english->getTranslation("Username");
core::ustring password = english->_T("Password");

// Or, we can set it as the active string table...
st::setActiveStringTable(english);

// ...and use the st::_T() function to grab a string from it.
core::ustring nickname = st::_T("Nickname");
```

cpp Code: Select all

```
// The same thing works with the CStringTableManager.
CStringTableManager mgr;
mgr.loadDirectory("translations");
mgr.setActiveStringTable("Español");
core::ustring nickname = st::_T("Nickname");
...
gui::IGUIStaticText* passtext = env->addStaticText(st::_T("Password").toWCHAR_s().c_str(),
core::recti(200, 180, 200 + 200, 180 + 20), false, false, window, GUI_LOGIN_PASSWORD,
false);
```

cpp Code: Select all

```
// We can also manipulate parameters in the strings.
// We will be using this string:
//     Hello, {{%1}}!  How are you?
// In our Spanish language file, it is translated as this:
//     ¡Hola, {{%1}}!  ¿Cómo estás?
// Assume that the currently active string table is the Spanish table.
const core::ustring nick_name("Bob");
...
// This will return:
//     ¡Hola, Bob!  ¿Cómo estás?
core::ustring greeting = st::_T("Hello, {{%1}}!  How are you?").param(nick_name);
...
// By default, parameters are in the format:
//   {{%1}}, {{%2}}, ...
// But that can be changed:
st::CStringTable* english = st::loadStringTable("english.irrst", device->getFileSystem());
```

```
// This sets the parameter format to the following:
//     [#1], [#2], ...
english->setParameterFormat("[#", "]");
```

The CStringTable.zip download contains a sample .irrst file that illustrates how to assemble the file. But, as a small example:

cpp Code: Select all

```
<?xml version="1.0"?>
<irr_stringtable>
        <language value="Español" />
        <entry>
                <id>Nickname</id>
                <translation>Apodo</translation>
        </entry>
        <entry id="Username"><![CDATA[<Nombre de usuario>]]></entry>
        <entry id="Password">Contraseña</entry>
        <s id="Language">Idioma</s>
</irr_stringtable>
```

What's next?
I'm not sure. Care to suggest something?

I have a keyboard that has non-ASCII characters. When I type into a text box in Irrlicht, it doesn't show the proper character! Why? (ALSO: Why can't a paste unicode text into Irrlicht?)
These patches don't solve this issue. If you want to be able to input unicode characters into Irrlicht, please take a look at MadHyde's patches:
http://irrlicht.sourceforge.net/phpBB2/viewtopic.php?p=220810#220810

Please note that as of the time of this post edit, Linux support in MadHyde's patches is incomplete.

Reported issues:
Incompatible with acki's irrExtensions.

Since new post notification e-mails don't work, and I don't always check this thread, I may not realize that you have posted something for a while. If you need help, you may want to consider sending me an e-mail:
john@suckerfreegames.removethis.com
Last edited by Nalin on Tue Jan 10, 2012 7:30 pm, edited 18 times in total.

Nalin

Posts: 194
Joined: Thu Mar 30, 2006 12:34 am
Location: Lacey, WA, USA

- Website
- YIM

Top

by **Dorth** » Fri Feb 12, 2010 4:32 pm

I dare say this is most likely the best current project open to this community, maybe only in competition with Spark, xEffects and CG integration.

Dorth

Posts: 931
Joined: Sat May 26, 2007 11:03 pm

Top

by **shadowslair** » Fri Feb 12, 2010 5:58 pm

Indeed it is useful. Thanks for sharing. I may try implementing it afer minimizing my unbelievably long todo list. 😄
"Although we walk on the ground and step in the mud... our dreams and endeavors reach the immense skies..."

User avatar
shadowslair

Posts: 758
Joined: Mon Mar 31, 2008 3:32 pm
Location: Bulgaria

Top

by **zet.dp.ua** » Fri Feb 12, 2010 7:41 pm

Really good work, very useful!

zet.dp.ua

Posts: 66
Joined: Sat Jul 07, 2007 8:10 am

Top

by **drr00t** » Sun Feb 14, 2010 12:49 pm

hi,

Very nice adding. I would like to try, but the links isn't working.

drr00t

Posts: 14
Joined: Wed Aug 05, 2009 2:11 pm
Location: Brasilia/DF - Brasil

Top

by **tinhtoitrangtay** » Mon Feb 15, 2010 1:16 pm

Thank you so muchbuti can't download it link dead please reupload

tinhtoitrangtay

Posts: 70
Joined: Tue Oct 28, 2008 12:59 pm

Top

---

by **Nalin** » Wed Feb 17, 2010 1:40 am

> *tinhtoitrangtay wrote:*Thank you so muchbuti can't download it link dead please reupload

Sorry about that. I just recently switched hosts and wouldn't you know, those files were the one thing I forgot to switch over.

Anyways, it is fixed now. You just have to wait for the DNS to update.

Nalin

Posts: 194
Joined: Thu Mar 30, 2006 12:34 am
Location: Lacey, WA, USA

- Website
- YIM

Top

---

by **Nalin** » Sun Feb 21, 2010 4:27 am

I've uploaded a new version of ustring and CGUITTFont. Download links are in the OP.

Changes in ustring:

- Renamed the class to ustring16, added custom allocators back in, and added a ustring typedef.
- Renamed all the byte order mark constants in the unicode namespace to better convey their functions.
- Added endianness conversion features.
- The class checks for byte order marks when you construct a new string.
- Ability to add byte order marks and specify endianness when using the string conversion functions.
- Various bugs fixed.

The big new features are the byte order marks and the endianness features. When you construct a new string, it will check to see if a byte order mark is present. If one is, it will correctly construct the string, taking into account the endianness specified by the mark. The byte order mark will NOT be saved into the string, however.

The string now saves its EUTF_ENCODE type. The encoding of the string can be retrieved via the getEncoding() member function. Currently, the only valid return values you should get are EUTFE_UTF16_LE and EUTFE_UTF16_BE.

I added a new enum, EUTF_ENDIAN. The getEndianness() member function will also return the endianness of the ustring.

I also updated all the unicode conversion routines (the toUTF8, toUTF16, toUTF32 member functions.) You can choose whether or not the function adds the appropriate byte order mark to the output string by passing true to the addBOM argument. Also, for the toUTF16 and toUTF32 functions, you can specify which endianness you wish the resulting string to have: EUTFEE_NATIVE, EUTFEE_LITTLE, or EUTFEE_BIG.

These new changes make it very easy to read/write proper Unicode formatted files complete with appropriate byte order marks. The class will take care of all the necessary endian conversions for you.

Code: Select all

```
// Passing true tells the class to include the byte order mark.
core::array<uchar8_t> utf8 = mystring.toUTF8(true);

io::IWriteFile* f = device->getFileSystem()->createAndWriteFile("mysettings.txt");
f->write(utf8.pointer(), utf8.size());
f->drop();
```

Changes in CGUITTFont:

- If the engine tries to draw a character that doesn't exist in the font, it will first try to draw the Unicode replacement character before drawing a blank space.

The CGUITTFont change requires the updated version of ustring.

Nalin

Posts: 194
Joined: Thu Mar 30, 2006 12:34 am
Location: Lacey, WA, USA

- Website
- YIM

Top

---

by **tinhtoitrangtay** » Mon Mar 01, 2010 11:27 am

WOW great new version
I have a problem when i merge to Irrlicht 1.7.1

```
------ Build started: Project: Irrlicht, Configuration: Release Win32 ------
Compiling...
CGUITTFont.cpp
Irrlicht.cpp
Linking...
Creating library ..\..\lib\Win32-visualstudio\Irrlicht.lib and object ..\..\lib\Win32-visualstudio\Irrlicht.exp
Generating code
Finished generating code
Embedding manifest...
Build log was saved at "file://c:\HOCTAPLT\Irr\source\Irrlicht\obj\Release\BuildLog.htm"
Irrlicht - 0 error(s), 0 warning(s)
------ Build started: Project: 05.UserInterface_vc8, Configuration: Release Win32 ------
Compiling...
```

```
main.cpp
Linking...
main.obj : error LNK2019: unresolved external symbol "public: static class
irr::gui::CGUITTFont * __cdecl irr::gui::CGUITTFont::createTTFont(class
irr::gui::IGUIEnvironment *,class irr::core::string<char,class irr::core::irrAllocator<char> >
const &,unsigned int,bool)" (?
createTTFont@CGUITTFont@gui@irr@@SAPAV123@PAVIGUIEnvironment@23@ABV?
$string@DV?$irrAllocator@D@core@irr@@@core@3@I_N@Z) referenced in function
_main
..\..\bin\Win32-VisualStudio\05.UserInterface.exe : fatal error LNK1120: 1 unresolved externals
Build log was saved at
"file://c:\HOCTAPLT\Irr\examples\05.UserInterface\Release\BuildLog.htm"
05.UserInterface_vc8 - 2 error(s), 0 warning(s)
========== Build: 1 succeeded, 1 failed, 0 up-to-date, 0 skipped ==========
```

i have change

```
static CGUITTFont* createTTFont(IGUIEnvironment *env, const io::path& filename, const u32
size, const bool size_is_pixels = false);
```

to

```
virtual CGUITTFont* createTTFont(IGUIEnvironment *env, const io::path& filename, const
u32 size, const bool size_is_pixels = false);
```

```
------ Build started: Project: 05.UserInterface_vc8, Configuration: Release Win32 ------
Compiling...
main.cpp
.\main.cpp(192) : error C2352: 'irr::gui::CGUITTFont::createTTFont' : illegal call of non-static
member function
..\..\include\CGUITTFont.h(90) : see declaration of 'irr::gui::CGUITTFont::createTTFont'
Build log was saved at
"file://c:\HOCTAPLT\Irr\examples\05.UserInterface\Release\BuildLog.htm"
05.UserInterface_vc8 - 1 error(s), 0 warning(s)
========== Build: 0 succeeded, 1 failed, 1 up-to-date, 0 skipped ==========
```

You can fix this problem

tinhtoitrangtay

Posts: 70
Joined: Tue Oct 28, 2008 12:59 pm

Top

by **tinhtoitrangtay** » Wed Mar 03, 2010 9:04 am

Hi Nalin
I have change code fix problem and i compiled success but i cannot run because it is crash.
From code

```
// Get a 14 pixel high code2001 font.
// If the third parameter is true, the size is in pixels. If false, it is in font points.
gui::CGUITTFont* tt_font = gui::CGUITTFont::createTTFont(device->getGUIEnvironment(),
"tahoma.ttf", 14, true);

// Set the font.
gui::IGUIEnvironment* env = device->getGUIEnvironment();
gui::IGUISkin* skin = env->getSkin();
skin->setFont(tt_font);

...

tt_font->drop();
```

change code

```
// Get a 14 pixel high code2001 font.
// If the third parameter is true, the size is in pixels. If false, it is in font points.
gui::CGUITTFont* tt_font = tt_font->createTTFont(device->getGUIEnvironment(),
"tahoma.ttf", 14, true);

// Set the font.
gui::IGUIEnvironment* env = device->getGUIEnvironment();
gui::IGUISkin* skin = env->getSkin();
skin->setFont(tt_font);

...

tt_font->drop();
```

This is fix problem but it is crash when i run app
code examples

```
/** Example 005 User Interface

This tutorial shows how to use the built in User Interface of
the Irrlicht Engine. It will give a brief overview and show
how to create and use windows, buttons, scroll bars, static
texts, and list boxes.

As always, we include the header files, and use the irrlicht
namespaces. We also store a pointer to the Irrlicht device,
a counter variable for changing the creation position of a window,
and a pointer to a listbox.
*/
#include <irrlicht.h>
#include "driverChoice.h"

using namespace irr;

using namespace core;
using namespace scene;
using namespace video;
using namespace io;
using namespace gui;

#ifdef _IRR_WINDOWS_
#pragma comment(lib, "Irrlicht.lib")
#endif

// Declare a structure to hold some context for the event receiver so that it
// has it available inside its OnEvent() method.
struct SAppContext
{
IrrlichtDevice *device;
s32 counter;
IGUIListBox* listbox;
};

// Define some values that we'll use to identify individual GUI controls.
enum
{
GUI_ID_QUIT_BUTTON = 101,
GUI_ID_NEW_WINDOW_BUTTON,
GUI_ID_FILE_OPEN_BUTTON,
GUI_ID_TRANSPARENCY_SCROLL_BAR
};

/*
The Event Receiver is not only capable of getting keyboard and
mouse input events, but also events of the graphical user interface
(gui). There are events for almost everything: Button click,
Listbox selection change, events that say that a element was hovered
and so on. To be able to react to some of these events, we create
an event receiver.
We only react to gui events, and if it's such an event, we get the
id of the caller (the gui element which caused the event) and get
the pointer to the gui environment.
*/
class MyEventReceiver : public IEventReceiver
{
public:
MyEventReceiver(SAppContext & context) : Context(context) { }

virtual bool OnEvent(const SEvent& event)
{
if (event.EventType == EET_GUI_EVENT)
{
s32 id = event.GUIEvent.Caller->getID();
IGUIEnvironment* env = Context.device->getGUIEnvironment();

switch(event.GUIEvent.EventType)
{

/*
If a scrollbar changed its scroll position, and it is
'our' scrollbar (the one with id GUI_ID_TRANSPARENCY_SCROLL_BAR), then we change
the transparency of all gui elements. This is a very
easy task: There is a skin object, in which all color
settings are stored. We simply go through all colors
```

```
stored in the skin and change their alpha value.
*/
case EGET_SCROLL_BAR_CHANGED:
if (id == GUI_ID_TRANSPARENCY_SCROLL_BAR)
{
s32 pos = ((IGUIScrollBar*)event.GUIEvent.Caller)->getPos();

for (u32 i=0; i<EGDC_COUNT ; ++i)
{
SColor col = env->getSkin()->getColor((EGUI_DEFAULT_COLOR)i);
col.setAlpha(pos);
env->getSkin()->setColor((EGUI_DEFAULT_COLOR)i, col);
}

}
break;

/*
If a button was clicked, it could be one of 'our'
three buttons. If it is the first, we shut down the engine.
If it is the second, we create a little window with some
text on it. We also add a string to the list box to log
what happened. And if it is the third button, we create
a file open dialog, and add also this as string to the list box.
That's all for the event receiver.
*/
case EGET_BUTTON_CLICKED:
switch(id)
{
case GUI_ID_QUIT_BUTTON:
Context.device->closeDevice();
return true;

case GUI_ID_NEW_WINDOW_BUTTON:
{
Context.listbox->addItem(L"Window created");
Context.counter += 30;
if (Context.counter > 200)
Context.counter = 0;

IGUIWindow* window = env->addWindow(
rect<s32>(100 + Context.counter, 100 + Context.counter, 300 + Context.counter, 200 +
Context.counter),
false, // modal?
L"Test window");

env->addStaticText(L"Please close me",
rect<s32>(35,35,140,50),
true, // border?
false, // wordwrap?
window);
}
return true;

case GUI_ID_FILE_OPEN_BUTTON:
Context.listbox->addItem(L"File open");
env->addFileOpenDialog(L"Please choose a file.");
return true;
default:
return false;
}
break;

default:
break;
}
}

return false;
}

private:
SAppContext & Context;
};


gui::IGUIFontVector* tt_font;

/*
Ok, now for the more interesting part. First, create the Irrlicht device. As in
some examples before, we ask the user which driver he wants to use for this
example:
*/
int main()
{
// ask user for driver
video::E_DRIVER_TYPE driverType=driverChoiceConsole();
if (driverType==video::EDT_COUNT)
return 1;

// create device and exit if creation failed

IrrlichtDevice * device = createDevice(driverType, core::dimension2d<u32>(640, 480));

if (device == 0)
return 1; // could not create selected driver.

/* The creation was successful, now we set the event receiver and
store pointers to the driver and to the gui environment. */

device->setWindowCaption(L"Irrlicht Engine - User Interface Demo");
device->setResizable(true);

video::IVideoDriver* driver = device->getVideoDriver();
//IGUIEnvironment* env = device->getGUIEnvironment();

/*
To make the font a little bit nicer, we load an external font
and set it as the new default font in the skin.
To keep the standard font for tool tip text, we set it to
the built-in font.
*/


// Set the font.
// Get a 14 pixel high code2001 font.
// If the third parameter is true, the size is in pixels. If false, it is in font points.
gui::IGUIFontVector* tt_font = tt_font->createTTFont(device->getGUIEnvironment(),
"tahoma.ttf", 14, true);

// Set the font.
gui::IGUIEnvironment* env = device->getGUIEnvironment();
gui::IGUISkin* skin = env->getSkin();
skin->setFont(tt_font);

//IGUISkin* skin = env->getSkin();
//IGUIFont* font = env->getFont("../../media/fonthaettenschweiler.bmp");
//if (font)
// skin->setFont(font);

//skin->setFont(env->getBuiltInFont(), EGDF_TOOLTIP);

/*
We add three buttons. The first one closes the engine. The second
creates a window and the third opens a file open dialog. The third
parameter is the id of the button, with which we can easily identify
the button in the event receiver.
*/

env->addButton(rect<s32>(10,240,110,240 + 32), 0, GUI_ID_QUIT_BUTTON,
L"Quit", L"Exits Program");
env->addButton(rect<s32>(10,280,110,280 + 32), 0, GUI_ID_NEW_WINDOW_BUTTON,
L"New Window", L"Launches a new Window");
env->addButton(rect<s32>(10,320,110,320 + 32), 0, GUI_ID_FILE_OPEN_BUTTON,
L"File Open", L"Opens a file");

/*
Now, we add a static text and a scrollbar, which modifies the
transparency of all gui elements. We set the maximum value of
the scrollbar to 255, because that's the maximal value for
a color value.
Then we create an other static text and a list box.
*/

env->addStaticText(L"Transparent Control:", rect<s32>(150,20,350,40), true);
IGUIScrollBar* scrollbar = env->addScrollBar(true,
rect<s32>(150, 45, 350, 60), 0, GUI_ID_TRANSPARENCY_SCROLL_BAR);
scrollbar->setMax(255);

// set scrollbar position to alpha value of an arbitrary element
scrollbar->setPos(env->getSkin()->getColor(EGDC_WINDOW).getAlpha());

env->addStaticText(L"Logging ListBox:", rect<s32>(50,110,250,130), true);
IGUIListBox * listbox = env->addListBox(rect<s32>(50, 140, 250, 210));
env->addEditBox(L"Editable Text", rect<s32>(350, 80, 550, 100));

// Store the appropriate data in a context structure.
SAppContext context;
context.device = device;
context.counter = 0;
context.listbox = listbox;

// Then create the event receiver, giving it that context structure.
```

```
MyEventReceiver receiver(context);

// And tell the device to use our custom event receiver.
device->setEventReceiver(&receiver);


/*
And at last, we create a nice Irrlicht Engine logo in the top left corner.
*/
env->addImage(driver->getTexture("../../media/irrlichtlogo2.png"),
position2d<int>(10,10));


/*
That's all, we only have to draw everything.
*/

while(device->run() && driver)
if (device->isWindowActive())
{
driver->beginScene(true, true, SColor(0,200,200,200));

env->drawAll();

driver->endScene();
}

tt_font->drop();

device->drop();

return 0;
}

/*
**/
```

You can fix this problem. Thank Nalin so much

tinhtoitrangtay

Posts: 70
Joined: Tue Oct 28, 2008 12:59 pm

Top

---

by **Nalin** » Thu Mar 04, 2010 8:24 am

Your issue is that you removed the "static" keyword from the createTTFont() function. That "static" keyword is very important. If you remove it, it changes the whole meaning of the function and the way the class operates. Basically, it will completely break it. Instead of changing it, we should address your first issue: the linker error.

Try adding **#include <CGUITTFont.h>** to the top of your source file. It doesn't seem that you added it, and it is very important that you do.

EDIT: Also, are you adding CGUITTFont.h/cpp to your project, or to the Irrlicht library directly? You should be adding it to your project, not to the Irrlicht library.

Nalin

Posts: 194
Joined: Thu Mar 30, 2006 12:34 am
Location: Lacey, WA, USA

- Website
- YIM

Top

---

by **BlindSide** » Thu Mar 04, 2010 8:44 pm

Hi there, this looks very useful, thanks for the contribution!
ShadowMapping for Irrlicht!: Get it here
Need help? Come on the IRC!: #irrlicht on irc://irc.freenode.net

BlindSide
Admin

Posts: 2821
Joined: Thu Dec 08, 2005 9:09 am
Location: NZ!

Top

---

by **stefbuet** » Thu Mar 04, 2010 9:22 pm

Hi,
thanks for sharing your usefull classes 😉

However I got a little compiler error:
line 1912: irrUString.h incomplete universal character name \u16

The concerned line is:

Code: Select all

```
ustring& trim(const ustring& whitespace = " \u16\n\r") { ...
```

User avatar
stefbuet
Competition winner

Posts: 495
Joined: Sun Dec 09, 2007 4:13 pm
Location: france

Top

---

by **Nalin** » Fri Mar 05, 2010 12:28 am

> *stefbuet wrote:*Hi,
> thanks for sharing your usefull classes 😉

> However I got a little compiler error:
> line 1912: irrUString.h incomplete universal character name \u16
>
> The concerned line is:
>
> Code: Select all
> ```
> ustring& trim(const ustring& whitespace = " \u16\n\r") { ...
> ```

You have an older version of irrUString.h. That was fixed in one of the recent versions. Just download it again from the OP.

Nalin

Posts: 194
Joined: Thu Mar 30, 2006 12:34 am
Location: Lacey, WA, USA

- Website
- YIM

Top

---

by **stefbuet** » Sat Mar 06, 2010 8:03 pm

Yes, I used directly the guifont zip file which doesn't contain the newest version. I got now the other good file.

So, firstly I just put irrUString in my Irrlicht 1.7.1 include directory and added the line #include "irrUString.h" after all includes in the irrlicht.h file.

The compiler doesn't like it.
Did you use irrlicht 1.7.1? Which version of gcc is working with it? I'm using DevC++ which doesn't use the lastest gcc compiler.

Here is the compiler output:
http://dl.free.fr/iG99KnTro

Thanks.

User avatar
stefbuet

Competition winner

Posts: 495
Joined: Sun Dec 09, 2007 4:13 pm
Location: france

Top

Next Display posts from previous: All posts ▼ Sort by Post time ▼ Ascending ▼ Go

---

Post a reply
140 posts • Page **1** of **10** • **1**, 2, 3, 4, 5 ... 10

Return to Project Announcements

Jump to: Project Announcements ▼ Go

Who is online

Users browsing this forum: No registered users and 1 guest

- Board index
- The team • Delete all board cookies • All times are UTC