

Doomsday Engine - Bug #1125

Handling missing OpenGL support during launch

2013-07-25 01:25 - galt_gendo

Status:	New	Start date:	2013-07-25
Priority:	Low	% Done:	0%
Assignee:			
Category:	User experience		
Target version:	Rendering		
Description			
<p>First of all, Allura as commit browser utterly sucks. This might be already fixed in the repo, but as it is, I can't tell without 'git clone'.</p> <p>I've just built 1.11.0, got hit by '-reconfig' problem, but following is independent (tested by moving .doomsday dir).</p> <p>(it may matter: Qt 4.8.5)</p> <p>All I'm getting from doomsday is a segfault right as the loading wheel is about to completely fill. The backtrace isn't very enlightening (there's a noticeable lack of debugging symbols in doomsday - I might fix that, but not sure it it helps):</p> <pre>#0 0x081e43e7 in FontLineWrapping::font() const () #1 0x081db320 in GLTextComposer::update() () #2 0x081e78ad in LabelWidget::update() () #3 0xb7e1505b in de::Widget::notifyTree(de::Widget::NotifyArgs const&) () from /usr/games/lib/libdeng2.so.2 #4 0xb7e10de3 in de::RootWidget::update() () from /usr/games/lib/libdeng2.so.2 #5 0x08203ecf in WindowSystem::timeChanged(de::Clock const&) () #6 0xb7e1ba57 in de::App::timeChanged(de::Clock const&) () from /usr/games/lib/libdeng2.so.2 #7 0xb7e22dc2 in de::Clock::setTime(de::Time const&) () from /usr/games/lib/libdeng2.so.2 #8 0xb7c75b51 in de::GuiApp::loopIteration() () from /usr/games/lib/libdeng_gui.so.1 #9 0xb7e34306 in de::Loop::nextLoopIteration() () from /usr/games/lib/libdeng2.so.2 #10 0xb7e3939b in de::Loop::qt_static_metacall(QObject*, QMetaObject::Call, int, void**) () from /usr/games/lib/libdeng2.so.2 #11 0xb6d8ef72 in QMetaObject::activate (sender=0x8706458, m=0xb6edf48c <QTimer::staticMetaObject>, local_signal_index=0, argv=0x0) at kernel/qobject.cpp:3547 #12 0xb6ddfc07 in QTimer::timeout (this=0x8706458) at .moc/release-shared/moc_qtimer.cpp:147 #13 0xb6d961be in QTimer::timerEvent (this=0x8706458, e=0xbfffea7c) at kernel/qtimer.cpp:280 #14 0xb6d8cdd4 in QObject::event (this=0x8706458, e=0xbfffea7c) at kernel/qobject.cpp:1156 #15 0xb7169934 in QApplicationPrivate::notify_helper (this=0x86be378, receiver=0x8706458, e=0xbfffea7c) at kernel/qapplication.cpp:4562 #16 0xb716e53e in QApplication::notify (this=0xbfffea18, receiver=0x8706458, e=0xbfffea7c) at kernel/qapplication.cpp:3944 #17 0xb7c75a13 in de::GuiApp::notify(QObject*, QEvent*) () from /usr/games/lib/libdeng_gui.so.1 #18 0xb6d76eda in QCoreApplication::notifyInternal (this=0xbfffea18, receiver=0x8706458, event=0xbfffea7c) at kernel/qcoreapplication.cpp:949 #19 0xb6daba7e in sendEvent (event=0xbfffea7c, receiver=<optimized out>) at kernel/qcoreapplication.h:231 #20 QTimerInfoList::activateTimers (this=0x86bfb34) at kernel/qeventdispatcher_unix.cpp:621 #21 0xb6da856a in timerSourceDispatch (source=0x86bfb00) at kernel/qeventdispatcher_glib.cpp:186 #22 0xb6b24be2 in g_main_dispatch (context=0x86bef58) at gmain.c:3054 #23 g_main_context_dispatch (context=0x86bef58) at gmain.c:3630</pre>			

```
#24 0xb6b24f68 in g_main_context_iterate (context=0x86bef58, block=1,
dispatch=1, self=<optimized out>) at gmain.c:3701
#25 0xb6b2504c in g_main_context_iteration (context=0x86bef58, may_block=1)
#26 0xb6da8d3c in QEventDispatcherGlib::processEvents (this=0x86bee80,
flags=...) at kernel/qeventdispatcher_glib.cpp:425
#27 0xb721afc5 in QGuiEventDispatcherGlib::processEvents (this=0x86bee80,
flags=...) at kernel/qguieventdispatcher_glib.cpp:204
#28 0xb6d75979 in QEventLoop::processEvents (this=0xbffed04, flags=...)
at kernel/qeventloop.cpp:149
#29 0xb6d75c82 in QEventLoop::exec (this=0xbffed04, flags=...)
at kernel/qeventloop.cpp:204
#30 0xb6d7ac6f in QApplication::exec ()
at kernel/qcoreapplication.cpp:1221
#31 0xb7167717 in QApplication::exec () at kernel/qapplication.cpp:3823
#32 0xb7c75ea5 in de::GuiApp::execLoop() ()
from /usr/games/lib/libdeng_gui.so.1
#33 0x080fd8af in main ()
```

On the other hand, there's little that can fail in `FontLineWrapping::font()`.

History

#1 - 2013-07-15 16:16 - galt_gendo

- did you rerun qmake so that the latest defaultstyle.pack is included in the doomsday.pk3?
+ did you do a full rebuild? ("make clean" first)

The build was done straight from the tarball, so unless it was packaged wrong, it was clean.

- did you ensure there are no system-wide copies of the libdeng* or other such libraries, which might have been mixed up into the build?

Err...though I don't think so, if that was the case, it would still be a build system bug, as it should ensure the new libs are picked first.

- do you have the "Liberation Sans" font?

Of course **not**. Why should I have any **particular** font installed, if QtGui is using fontconfig and should be able to pick a reasonable fallback ?

PS.: Allura is formatting this post stupidly - it might be hard to read.

PPS.: correction, it's just the preview, that's broken.

#2 - 2013-07-17 05:32 - skyjake

Let's first rule out possible build/data problems:

- did you rerun qmake so that the latest defaultstyle.pack is included in the doomsday.pk3?
- did you do a full rebuild? ("make clean" first)
- did you ensure there are no system-wide copies of the libdeng* or other such libraries, which might have been mixed up into the build?
- do you have the "Liberation Sans" font?

PS. Use <http://github.com/skyjake/Doomsday-Engine/>

#3 - 2013-07-25 10:56 - skyjake

Err...though I don't think so, if that was the case, it would still be a build system bug, as it should ensure the new libs are picked first.

Indeed: [\[#1120\]](#)

So you are saying everything was linked against the correct libraries?

Would you mind doing a debug build (CONFIG+=debug) and checking if gdb reveals any extra information about the crash, if it still occurs?

QtGui is using fontconfig and should be able to pick a reasonable fallback

You are right, a fallback is picked and this does not matter...

#4 - 2013-07-25 14:58 - galt_gendo

So you are saying everything was linked against the correct libraries?

No, I didn't said that. Regardless, even rebuilding after 1.11.0 is already installed doesn't make a change, so it would be an independent problem.

Would you mind doing a debug build (CONFIG+=debug)...

A minor note here: unless I'm missing something (I don't really use qmake), this seems to require editing config.pri, cause config_user.pri comes too late to take effect.

If I only make sure '-g' is passed, the only real new info is
'0x081e7306 in FontLineWrapping::font (this=0x0)', which was sort of already obvious.
If, on the other hand, I do enough edits to recreate that debug entry in config_user.pri, the built executable is pretty much useless for me.
On this hardware, I need to run with '-noglcheck' - it still worked in 1.10.
On non-debug build, among the console output is:
ClientWindow > Bank > Bank::Job: Failed to load "generic.textured.color" from source:
[AllocError] (GLShader::alloc) Failed to create shader
ClientWindow: Error when initializing widget "
[LoadError] (Bank::data) Failed to load "generic.textured.color"
^ > Bank > Bank::Job: Failed to load "generic.textured.color" from source:
[AllocError] (GLShader::alloc) Failed to create shader
ClientWindow: Error when initializing widget "
[LoadError] (Bank::data) Failed to load "generic.textured.color"
^ > Bank > Bank::Job: Failed to load "generic.textured.color" from source:
[AllocError] (GLShader::alloc) Failed to create shader
ClientWindow: Error when initializing widget "
[LoadError] (Bank::data) Failed to load "generic.textured.color"

but the program continues, with debug build, that's an assert right there, long before anywhere close to last crash.

#5 - 2013-07-25 16:50 - skyjake

That certainly sounds like your OpenGL version is older than the required 2.0. Since you are running with -noglcheck, it looks like there isn't much to do here apart from replacing the crash with a graceful fatal error.

We'll be keeping 1.10.x around for some time still, where the OpenGL requirement is 1.4. The 1.10.4 release will be available this week or early next week.

CONFIG+=debug

This can be done on the command line when running qmake, as in *qmake -r ../doomsday/doomsday.pro CONFIG+=debug*

#6 - 2013-07-25 20:05 - galt_gendo

Well, I must mention, that back in 1.10, I was already running with '-noglcheck', but nothing seemed **noticeably** broken back then.

I've tried a more hamfisted approach to the sources, but the furthestmost I got was a graceful shutdown at '(Bank::data) Failed to load "fx.blur.horizontal"', so I declare 1.11 a bust on this machine.

Even getting that far required adding quite a few '! = 0' checks in that ui code.

#7 - 2013-07-26 10:32 - danij

- **Priority:** 1 --> 5

#8 - 2013-10-19 12:13 - skyjake

- *Tags set to Graphics, OpenGL*

- *Subject changed from doomsday segfaults before it really starts to Handling missing OpenGL support during launch*

- *Category set to User experience*

- *Priority changed from Normal to Low*

- *Target version deleted (1.11)*

Changed priority to Low because the OpenGL version check must be manually disabled for this situation to occur.

#9 - 2017-04-03 14:41 - skyjake

- Target version set to Rendering