

## Ryzom - Bug # 482

<b>Status:</b>	Closed	<b>Priority:</b>	Normal
<b>Author:</b>	kervala	<b>Category:</b>	NeL: General
<b>Created:</b>	04/12/2009	<b>Assignee:</b>	kervala
<b>Updated:</b>	09/30/2010	<b>Due date:</b>	
<b>Subject:</b>	Some shape rotations are bad with NL_NO_ASM		
<b>Description</b>	When we compile NeL with NL_NO_ASM, some shapes doesn't rotate as intended (its scale is also modified).		

### History

#### #1 - 05/11/2009 05:50 pm - kervala

Bug occurs in misc/fast\_floor.h OptFastFloor function. The C++ equivalent code is wrong in some rare cases.

#### #2 - 06/16/2009 08:06 pm - sfb

- File floortestcase.7z added

- Category set to Misc

Attached is a test case to prove the problem and below is the output.

This strikes another important bit of information. OptFastFloor doesn't exist on Linux (or other platforms) so these objects with bad rotations **will always** rotate incorrectly on non-Windows platforms unless a non-assembly, cross-platform variation of OptFastFloor is developed and all #ifdef NL\_OS\_WINDOWS references surrounding OptFastFloor calls are removed.

```
C:\Projects\floortestcase\build\bin\Release>floorTestCase.exe
INF 6e50 main.cpp 37 main floorTestCase.exe : starting comparison loop.
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.0 OptFstFlr: 0 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.1 OptFstFlr: 0 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.2 OptFstFlr: 0 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.3 OptFstFlr: 0 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.4 OptFstFlr: 0 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.5 OptFstFlr: 0 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.6 OptFstFlr: 1 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.7 OptFstFlr: 1 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.8 OptFstFlr: 1 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 0.9 OptFstFlr: 1 Std: 0.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.0 OptFstFlr: 1 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.1 OptFstFlr: 1 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.2 OptFstFlr: 1 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.3 OptFstFlr: 1 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.4 OptFstFlr: 1 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.5 OptFstFlr: 2 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.6 OptFstFlr: 2 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.7 OptFstFlr: 2 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.8 OptFstFlr: 2 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 1.9 OptFstFlr: 2 Std: 1.0
INF 6e50 main.cpp 46 main floorTestCase.exe : Orig: 2.0 OptFstFlr: 2 Std: 2.0
```

**#3 - 06/16/2009 08:36 pm - kerval**

Thanks a lot for this test-case :)

**#4 - 06/27/2009 05:47 pm - Spex**

Looking at the output this "floor" is more like a "round" towards even numbers; replacing it with a real floor() isn't obviously not working then.

Unfortunately, C++ doesn't define (yet?) any standard way how to handle rounding. The fast "floor" routines are based on manipulating the FP environment, which can be done using C99 functions as well.

[http://msdn.microsoft.com/en-us/library/aa289157\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/aa289157(VS.71).aspx) -- Microsoft Visual C++ Floating-Point Optimization

<http://www.kernel.org/doc/man-pages/online/pages/man3/fenv.3.html> -- fenv based functions according to C99; no idea how much overhead the actual rounding functions have (aka if they are fast or bring problems like FP state switches on each call, which is why these OptFastXXX functions exist in the first place)

**#5 - 07/26/2009 10:58 pm - kerval**

- File *fast\_floor.diff* added

I succeeded to fix it using SSE intrinsics under Windows, I post it as a patch for the moment since it could exist a better way to handle it.

As these functions are inline and they are available on all CPU with amd64 instructions (all CPU with AMD64 instructions have both SSE and SSE2), I suppose we don't need to add a check for the presence of SSE instruction.

**#6 - 07/26/2009 11:05 pm - kerval**

For a list of all supported intrinsic under VC++ 2008 :

<http://blogs.msdn.com/vcblog/archive/2007/10/18/new-intrinsic-support-in-visual-studio-2008.aspx>

**#7 - 07/27/2009 03:31 pm - kerval**

- Status changed from *New* to *Validated*

**#8 - 07/27/2009 03:32 pm - kerval**

- Status changed from *Validated* to *Assigned*

- Assignee set to *kerval*

**#9 - 07/31/2009 01:47 pm - kerval**

- Status changed from *Assigned* to *Resolved*

- % Done changed from *0* to *100*

Applied in changeset r1681.

**#10 - 09/03/2009 03:18 pm - sfb**

- Status changed from *Resolved* to *Closed*

Thanks kerval, this looks fantastic! I'm closing this issue.

**#11 - 09/29/2010 09:30 pm - kerval**

- Project changed from NeL to Ryzom
- Category deleted (Misc)
- Target version deleted (Version 0.7.0)

**#12 - 09/30/2010 02:41 pm - kerval**

- Category set to NeL: General
- Target version set to Version 0.7.0

**Files**

---

floortestcase.7z	7.6 kB	06/16/2009	sfb
fast_floor.diff	1.7 kB	07/26/2009	kerval