Doomsday Engine - Feature #2164

Interlaced 3D mode

2016-07-11 09:05 - skyjake

Status:	Closed	**Start date:**	2016-07-11
Priority:	Normal	**% Done:**	100%
Assignee:	skyjake		
Category:	3rd party compatibility		
Target version:	2.0 – Home UI & Packages		

Description

Use a post-processing shader to interlace the left and right eye frames.

See discussion: http://dengine.net/forums/viewtopic.php?f=7&t=2290&p=15210#p15210

Related issues:

Related to Feature #1678: Optimize stereoscopic pixel formats **Rejected**

History

#1 - 2016-07-11 09:05 - skyjake

- Related to Feature #1678: Optimize stereoscopic pixel formats added

#2 - 2016-07-11 09:09 - skyjake

- Description updated

#3 - 2016-07-12 15:47 - cmbruns

skyjake wrote:

> Use a post-processing shader to interlace the left and right eye frames.
>
> See discussion: http://dengine.net/forums/viewtopic.php?f=7&t=2290&p=15210#p15210

I'm copying my forum post on this subject here to preserve it, because my historical forum posts occasionally get deleted.

I have implemented interleaved stereo modes before, so I'd like to make some comments in case anyone wants to begin implementing this right away. However I probably won't have time to implement this myself in the near future.

Passive row-interleaved 3D like this works with old Zalman monitors, certain pre-2016 Samsung 3D TVs, the LG D2743 monitor, and, notably, certain late model LG 4K passive 3D TVs.

Compared with active 3D methods like Nvidia 3D vision, this sort of passive 3D has certain advantages: it works on any OS (Windows, Mac, Linux, whatever), with any video card. Also the passive glasses are far more comfortable and stylish. There are also disadvantages, but I don't want to get into the whole passive 3D vs. active 3D holy war at the moment.

1) The "easy" way to implement row interleaved 3D, would be to render the full left and right eye views to separate offscreen buffers, then compose those two images using a shader that fetches from the left eye image for even numbered scan lines, and from the right eye image for odd numbered scan lines. One appealing aspect of this approach is that those same left/right offscreen buffers could be reused for almost any stereoscopic 3D format one can imagine.

2) The "old fashioned" way to implement this would be to use OpenGL stencil operations to mask out the odd or even scan lines during the left and right eye render passes. I don't recommend this approach. Theoretically it seems like this approach could have better rendering performance than option "1", but in practice it seems to actually be slower.

3) The "optimal" way to implement this would be to render a compressed aspect ratio image into each of the left and right offscreen buffers, so the image is vertically squished smaller by a factor of two. (One could even pack both eye views into a single full-frame offscreen buffer). This approach would yield the best performance, because no extra pixels are rendered; and it yields the best appearance, because the vertical pixel filtering is more correct in the final displayed image.

- It's important to provide a manual stereoscopic "swap eyes" option with these modes, because the conventions vary between different displays, and it's so easy to get the parity wrong.
- For windowed display modes, it's important to track the global window position. Because when we talk about, say, "odd numbered scan lines", we mean global screen relative row indices, not local window relative indices. In particular, we want the player to be able to drag the game window around the screen, without ever inadvertently inverting the sense of the stereoscopic depth.

- It's important that the apparent screen resolution exactly matches the physical monitor resolution. These details are on the player's responsibility, but should be clearly documented. So A) the operating system "screen resolution" must match the true physical resolution of the interlaced display. And further, B) on recent versions of Windows, the display scale must be set to "100%". So if you have your Windows desktop set to show "large icons", you might need to make them small while playing in these modes.

#4 - 2016-10-09 08:57 - skyjake

See 3d7fa3438a8. Some tweaks should still be applied after gl2-viewcompositor has been merged.

#5 - 2016-10-09 08:57 - skyjake

- Status changed from New to In Progress

- Assignee set to skyjake

- % Done changed from 0 to 50

#6 - 2016-10-09 08:58 - skyjake

- Tags set to VR, Renderer

- Target version set to 2.0 – Home UI & Packages

#7 - 2016-10-15 22:00 - skyjake

- Status changed from In Progress to Closed

- % Done changed from 50 to 100