Emulator - Emulator Issues #9387

Metroid Prime 2: Echoes (Refraction Effects issues)

03/01/2016 02:34 AM - DamekCritou

Status:	New	**% Done:**	0%
Priority:	Normal		
Assignee:			
Category:			
Target version:			
Operating system:	N/A	**Relates to performance:**	No
Issue type:	Bug	**Easy:**	No
Milestone:		**Relates to maintainability:**	No
Regression:	No	**Regression start:**	
Relates to usability:	No	**Fixed in:**	

Description

Game Name?

Metroid Prime 2: Echoes

Game ID? (right click the game in the game list, properties, info tab)

G2ME01

MD5 Hash? (right click the game in the game list, properties, info tab, MD5 Hash: Compute)

ce781ad1452311ca86667cf8dbd7d112

What's the problem? Describe what went wrong.

Any refraction effect, which includes water drops on the visor, the water splash on the visor, portal transparency, and the like, will severely slow down the game.

What steps will reproduce the problem?
Can either find and stare at a portal or jump into a pool of water deep enough to trigger refraction.

The portals start to get a little bit better if you stare at them long enough. But certain areas of the game become increasingly difficult to travel through with the massive slow down involved with these effects.

Which versions of Dolphin did you test on? Does using an older version of Dolphin solve your issue? If yes, which versions of Dolphin used to work?

Tested on: 4.0-9050
Worked before 4.0 versions.
Stated in another thread that the *OpenGL* refraction problem was reversed after it was implemented. But I am running this on the Dx12 experimental backend. I checked, and it does this on Dx11 now too.

What are your PC specifications? (CPU, GPU, Operating System, more)
Intel i7 4790k @ 4.6 GHz
Nvidia GeForce GTX 780
16 GB RAM
Windows 10
Direct X 12 backend
Dual core speedup enabled
Running on forced 16:9 at 2x Native with 2x MSAA and 2x anisotropic filtering with: Scaled EFB Copy, Force Texture Filtering, Ignore Format Changes, Disabled XFB, and Disable Bounding Box checked.

Is there any other relevant information? (e.g. logs, screenshots,

configuration files)

All right, so I know after reading some of the other submitted issues for the same problem that you guys don't accept performance problems as a real bug. But I don't believe this to be a strict performance issue, as the claim on your wikipedia states this particular problem does not happen with DirectX. I have here a video showcasing that this is simply not true anymore.

https://www.youtube.com/watch?v=LDCcpDSZ2lM

(there's a portal on the ledge I keep looking at)

Included is the optional rendering stats from Dolphin advanced settings.

I'm not entirely sure how refraction works through Dolphin, but it does seem odd that when the effect starts to generate, the draw calls and texture creations shoot way up above normal. Especially when the portal refraction doesn't change, which might explain why it (eventually) gets better for portals, but never for water refraction. I only bring this up because Direct X is supposed to be unaffected by this. This implies some coding change that has broken texture generation when refraction is being processed.

History

#1 - 03/01/2016 02:38 AM - JMC4789

Texture pooling was supposed to mostly solve this; it makes it a lot faster for me in all of those cases, faster than it used to be in older builds.

If you really want to make an issue about this, then don't be lazy and bisect. You say it works in X build, then find the build that broke it. If it's a change that was necessary due to accuracy, then the issue probably get rejected (as the performance drop was due to previously wrong emulation) or we move to another way of solving it.

Particularly, GPU texture decoding takes the load off the CPU for this particular bug. But, to be convinced that this feature needs implementing would require some very hard data.

#2 - 03/01/2016 12:18 PM - Jebeld17@gmail.com

As you said, this same issue was happening in OpenGL in the past, but also looking back, there was a similar issue that happened with Metroid Prime 1 for Prime Trilogy (Wii) in Dolphin, fixed by v4.0-5124, commit 5357b9c95ff2458e3b8c3112e303924a0d460ce6. (Source [[http://bit.ly/1T5Rwog]])

Have you tried running Prime 2: Echoes in OpenGL, by chance? Since both Prime 1 and Prime 2 use basically the same rendering engine developed by Retro Studios, I can't see too much why Echoes *wouldn't* work under OGL, given the specs on your machine.

JMC4789 wrote:

> Texture pooling was supposed to mostly solve this; it makes it a lot faster for me in all of those cases, faster than it used to be in older builds.
>
> If you really want to make an issue about this, then don't be lazy and bisect. You say it works in X build, then find the build that broke it. If it's a change that was necessary due to accuracy, then the issue probably get rejected (as the performance drop was due to previously wrong emulation) or we move to another way of solving it.
>
> Particularly, GPU texture decoding takes the load off the CPU for this particular bug. But, to be convinced that this feature needs implementing would require some very hard data.

#3 - 03/26/2016 03:36 AM - JMC4789

The issue is that EFB2RAM forces a GPU sync, meaning it's still slow even with Texture Pooling. Metroid Prime 1 doesn't need a GPU sync because everything works in EFB2Tex, but unfortunately Metroid Prime 2/3 need EFB2RAM for the scan visors.

#4 - 04/12/2018 06:10 AM - VoidMaw

I came up with a work-around for this issue. Set the EFB2Tex toggle to the same button you use for the scan visor and just tap it again before you put away the scan visor.