Designing for Performance

v. 0.4

1. Frame Rate

Thanks to the asynchronous [TimeWarp](), looking around in the Gear VR will always be smooth and judder-free at 60 FPS, regardless of how fast or slow the application is rendering. This does not mean that performance is no longer a concern, but it gives a lot more margin in normal operation, and improves the experience for applications that do not hold perfectly at 60 FPS.

If an application does not consistently run at 60 FPS, then animating objects move choppier, rapid head turns pull some black in at the edges, player movement doesn't feel as smooth, and gamepad turning looks especially bad. However, the asynchronous TimeWarp does not require emptying the GPU pipeline and makes it easier to hold 60 FPS than without.

Drawing anything that is stuck to the view will look bad if the frame rate is not held at 60 FPS, because it will only move on eye frame updates, instead of on every video frame. Don't make heads up displays. If something needs to stay in front of the player, like a floating GUI panel, leave it stationary most of the time, and have it quickly rush back to center when necessary, instead of dragging it continuously with the head orientation.

2. Performance Advice for Early Titles

Be conservative on performance. Even though two threads are dedicated to the VR application, a lot happens on Android systems that we can't control, and performance has more of a statistical character than we would like. Some background tasks even use the GPU occasionally. Pushing right up to the limit will undoubtedly cause more frame drops, and make the experience less pleasant.

You aren't going to be able to pull off graphics effects under these performance constraints that people haven't seen years ago on other platforms, so don't try to compete there. The magic of a VR experience comes from interesting things happening in well-composed scenes, and the graphics should largely try not to call attention to themselves.

Even if you consistently hold 60 FPS, more aggressive drawing consumes more battery power, and subtle improvements in visual quality generally aren't worth taking 20 minutes off the battery life for a title.

Keep rendering straightforward. Draw everything to one view, in a single pass for each mesh. Tricks with resetting the depth buffer and multiple camera layers are bad for VR, regardless of their performance issues. If the geometry doesn't work correctly - all rendered into a single view (FPS hands, et cetera) - then it will cause perception issues in VR, and you should fix the design.

You can't handle a lot of blending for performance reasons. If you have limited navigation capabilities in the title and can guarantee that the effects will never cover the entire screen, then you will be ok.

Don't use alpha tested / pixel discard transparency -- the aliasing will be awful, and performance can still be problematic. Coverage from alpha can help, but designing a title that doesn't require a lot of cut out geometry is even better.

Most VR scenes should be built to work with 16 bit depth buffer resolution and 2x MSAA. If your world is mostly pre-lit to compressed textures, there will be little difference between 16 and 32 bit color buffers.

Favor modest "scenes" instead of "open worlds". There are both theoretical and pragmatic reasons why you should, at least in the near term. The first generation of titles should be all about the low hanging fruit, not the challenges.

The best-looking scenes will be uniquely textured models. You can load quite a lot of textures -- 128 Megs of textures is okay. With global illumination baked into the textures, or data actually sampled from the real world, you can make reasonably photo realistic scenes that still run 60 FPS stereo. The contrast with much lower fidelity dynamic elements may be jarring, so there are important stylistic decisions to be made.

Panoramic photos make excellent and efficient backdrops for scenes. If you aren't too picky about global illumination, allowing them to be swapped out is often nice. Full image-based lighting models aren't performance-practical for entire scenes, but are probably okay for characters that can't cover the screen.

Last Update: Nov 10, 2014