

Front Buffer Rendering

v. 0.4

Android's standard OpenGL ES implementation triple buffers window surfaces, trading increased latency for increased smoothness. This is a debatable tradeoff for most applications, but is clearly bad for VR. An Android application that heavily loads the GPU and never forces a synchronization may have over 50 milliseconds of latency from the time `eglSwapBuffers()` is called to the time pixels start changing on the screen, even running at 60 FPS.

Android should probably offer a strictly double buffered mode, and possibly a swap-tear option, but it is a sad truth that if you present a buffer to the system, there is a good chance it won't do what you want with it immediately. The best case is to have no chain of buffers at all -- a single buffer that you can render to while it is being scanned to the screen. To avoid tear lines, it is up to the application to draw only in areas of the window that aren't currently being scanned out.

The mobile displays are internally scanned in portrait mode from top to bottom when the home button is on the bottom, or left to right when the device is in the headset. VrLib receives timestamped events at display vsync, and uses them to determine where the video raster is scanning at a given time. The current code waits until the right eye is being displayed to warp the left eye, then waits until the left eye is being displayed to warp the right eye. This gives a latency of only 8 milliseconds before the first pixel is changed on the screen. It takes 8 milliseconds to display scan half of the screen, so the latency will vary by that much across each eye.

Last Update: Nov 10, 2014

OCULUS VR is a registered trademark of Oculus VR, LLC. (C) Oculus VR, LLC. All rights reserved. All other trademarks are the property of their respective owners.