

User Interface - Bug #5582

webgl renderer is significantly slower than canvas2d (performance regression)

08/04/2021 07:23 AM - Greg Shah

Status: New	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	
billable: No	case_num:
vendor_id: GCD	
Description	
Related issues:	
Related to User Interface - Support #4955: web client/javascript drawing - fa...	Closed

History

#1 - 08/04/2021 07:23 AM - Greg Shah

- Related to Support #4955: web client/javascript drawing - fast vs crisp added

#2 - 08/04/2021 07:33 AM - Greg Shah

At one time, the webgl renderer was faster than canvas2d for some browser versions and at least equivalent to canvas2d in the other commonly used browser versions. Then something changed such that webgl is almost always significantly slower no matter which browser version is used. This seems like a regression, but it could also be somehow related to browser rendering engine changes.

From #5384-6:

Sergey/Hynek: Please note that in our recent testing, both Eugenie and Eric have found that the Canvas2D drawing is much faster than the WebGL drawing. In addition, the graphics caching seems to hurt performance as well.

I'm speculating, but it could also mean the browser rendering engines went through optimizations in the image bitmap manipulation use cases, which we found to be sluggish. One indication is the graphics caching slowing down Canvas 2D. Graphics caching was introduced to overcome just those slow image bitmap operations in Canvas 2D, but introduces overhead by itself. As the bitmap operations get faster (speculation) the overhead of graphics caching starts to show up.

Hynek did some profiling (see #5384-62) and found:

I am looking for the poor performance results of WebGL renderer. The biggest issue, which introduces 300-400 ms of additional load time for the <customer_application_screen> on my system, is the combo box selection artifact workaround ([#4955-490](#)) in revision 11973.