Ilya Seletsky

Validation Framework

My real time graphics engine will most likely need to be tested with different scenes on different hardware. I might have a really detailed scene using very complex shaders and other materials that are targeted towards high end hardware. I might also have a really detailed scene but with less complex shaders and effects targeted for lower end systems. Then I might have much smaller scenes with less objects.

I need to basically see if the work I'm doing with Occlusion Culling improves the framerate rather than adding even more overhead. I predict that on large detailed scenes I will definitely see an improvement. I will avoid sending a large amount of objects to the card to draw, as well as avoiding running complex effects for those objects.

On smaller scenes there will most likely be a cutoff point where the work I'm doing trying to cull objects out will take more time that it would to just render those objects even if they aren't visible.

One thing that currently happens in my technique is objects pop in a frame late. I have ideas for some improvements that may either increase this or reduce this. If the pop in time is increased I need to objectively decide if this is acceptable compared to the performance increase. If the pop in time is reduced I need to determine if the short hitch in performance loss doesn't negatively affect the gameplay. It'll most likely have some incorrect results and draw more objects than necessary for the first initial frames but then normalize and speed up. This would happen after any significant viewport changes such as turning around very quickly or teleporting.