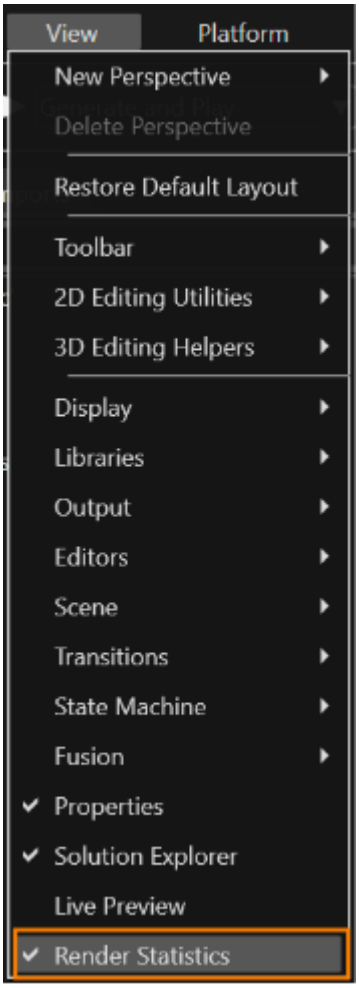
Display Renderer Statistics

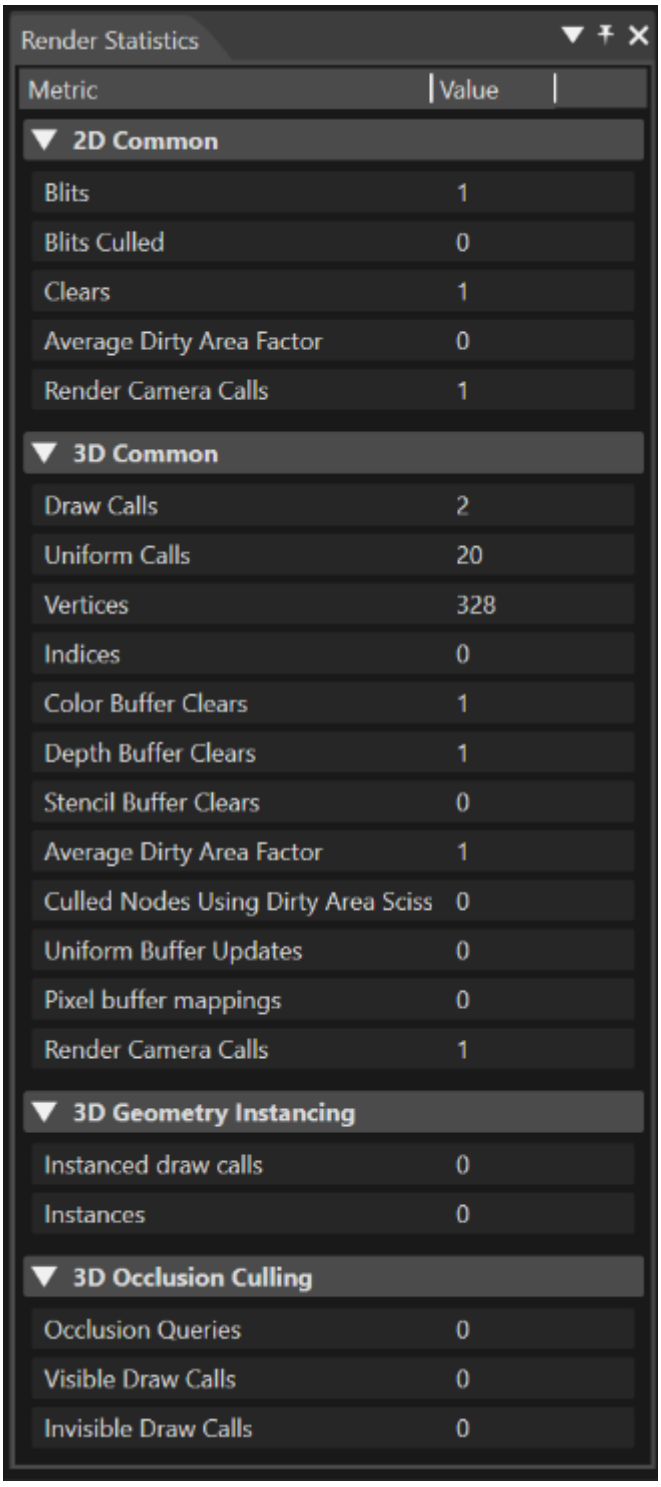
Add display

Render Statistics inform the user about current performance indicators during rendering. Therefore they are of major help when trying to identify performance bottlenecks. This information is presented to the user in a dedicated panel which can be made visible through the "Render Statistics" menu item from the "View" menu.



"Render Statistics" Menu

The panel will be displayed by default in the top right region of the SceneComposer user interface. Also this panel can be moved in any other area.



"Render Statistics" panel

The Renderer Statistics structure keeps track of per pass data that is relevant to the user. This includes statistics about the metrics of:

- Draw Calls
- Uniform Calls
- Vertices
- Indices
- ColorBufferClears
- DepthBufferClears

- StencilBufferClears
- AverageDirtyAreaFactor
- CulledNodesUsingDirtyAreaScissor
- Uniform Buffer Updates
- Pixcel buffer mappings
- Render Camera Calls
- Occlusion Queries (available on OpenGL ES 3.0 API only)
- Visible Draw Calls (available on OpenGL ES 3.0 API only)
- Invisible Draw Calls (available on OpenGL ES 3.0 API only)

Some specific pieces of information are the following:

2D:

- Uploaded BitmapImages
- Total Uploaded BitmapImages
- Render Camera2D Calls

3D:

- Uploaded Textures (incl. breakdown to BitmapTextureImages, CubeMapTextureImages, DirectTextureImages)
- Total Uploaded Textures (incl. breakdown to BitmapTextureImages, CubeMapTextureImages, DirectTextureImages)
- Uploaded Shader Programs
- Total Uploaded Shader Programs
- Uploaded VertexBuffers
- Total Uploaded VertexBuffers
- Render Camera Calls

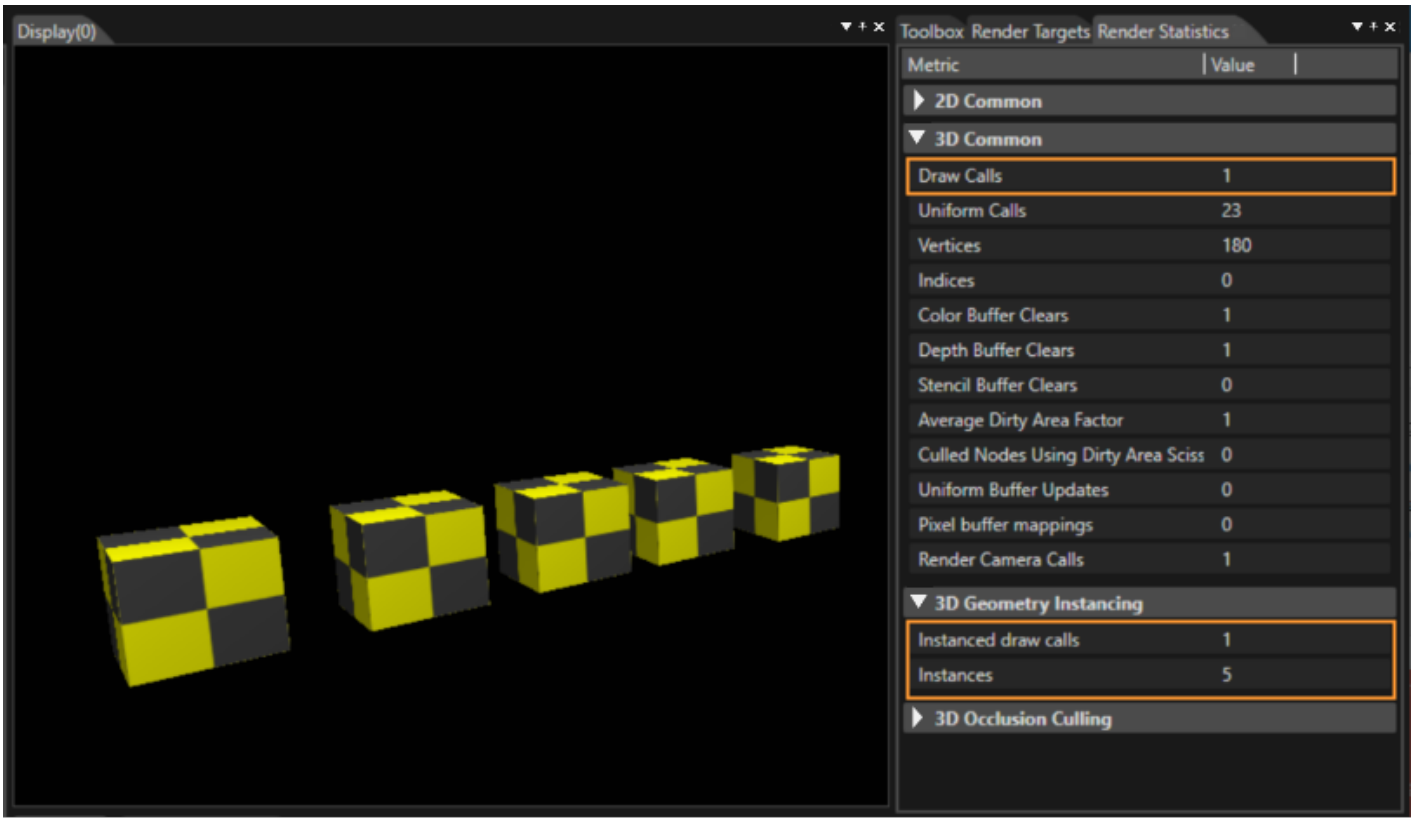
The numbers shown in the statistic accumulate a typical render pass conducted by RenderAllCameras. The user can disable or enable cameras and render targets within the RenderTarget panel to sum up and view statistics for only those selected items. To gather statistics, three basic conditions have to be fulfilled. First, the display has to be set active. Second, the "Enable Rendering" property of any camera has to be checked. Third, the camera(s) should have the rendering enabled.

> This selection is already available, see the panels RenderTargets/RenderTargetView or RenderTargets/CameraGroupView. The render statistics dialog may display values which are different than zero on some platforms, even if there is no camera configured to render. This may happen, for example, on iMX6 platform, because of the software layers rendering simulation.

To see how the Geometry Instancing feature works, the user has the opportunity to observe its functionality through Render Statistics. The next steps will be followed:

(To perform these steps, you need a system that supports OpenGL ES 3.0.)

- Open/create an OpenGL ES 3.0 Solution like 3D Basic Solution;
- Import an .fbx object;
- Create a 3D scene containing the object and a camera (do not activate occlusion culling render strategy);
- Create a display, a render target and set the camera to render on the display;
- Add shader with instancing support (ex: RefTransLight1Inst_RefColorTex) to the appearance;
- Add UniformSetter to the appearance;
- Set either "Lights Activation" to false or "Light Coordinate Space" to World on the UniformSetter;
- Save its appearance collection as a template and enable Instance Sharing;
- Drag the newly created template back on the mesh;
- Clone the object a few times;
- At this point each Cube must have a shared Appearance with a shader that supports instancing and a uniform setter;
- Enable the Render Statistics Panel;
- If Geometry Instancing IS NOT working then Draw Calls will be "N", Instanced Draw Calls will be "0" and Instances will be "0" (where "N" is the number of identical meshes);
- If Geometry Instancing IS working then Draw Calls will be "1", Instanced Draw Calls will be "1", Instances will be "N" (where "N" is the number of identical meshes).

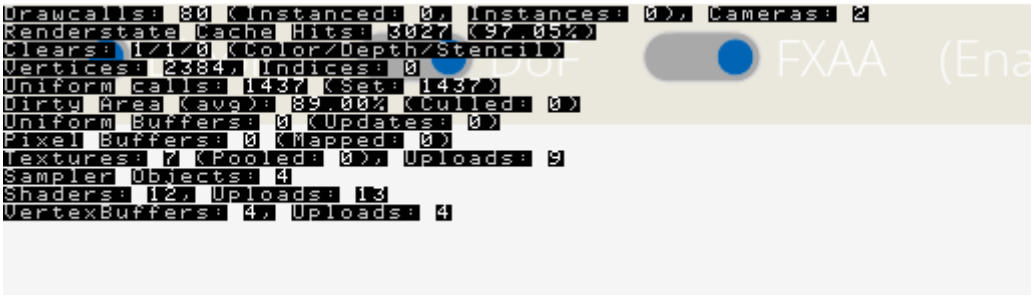


Geometry Instancing through "Render Statistics" panel

If CANDERA_RENDER_STATE_CACHING_ENABLED and CANDERA_DETAIL_STATISTICS_ENABLED are set to true, additional information about Render State Cache Effectivity is available.

- Render State Cache Hits informs about the number of OpenGL state changes that were saved because the value was already cached.
- Render State Cache Misses informs about the number of OpenGL state changes that had to be executed because the new value was not in the cache.
- With these values the Render State Cache Hit Rate can be calculated, where 100% would mean no state changes.

These values are visible either in the Render Statistics Overlay of the Player, the Render Statistics Panel in SceneComposer (only for applicable properties) or directly accessible from the API.



Render Statistic Overlay

Revision #6
Created 2023-02-13 05:40:26 UTC by Kanai Tomoaki
Updated 2025-01-20 23:32:06 UTC by Elisabeth Zauner