Quick Guide

This chapter aggregates recommendations concerning major points explained very shortly.

- Optimize Number of Nodes
- Avoid Buffer Clearing
- Disable Depth Test and Adjust Depth Range
- Vertex Buffer Optimization
- Startup Time

Optimize Number of Nodes

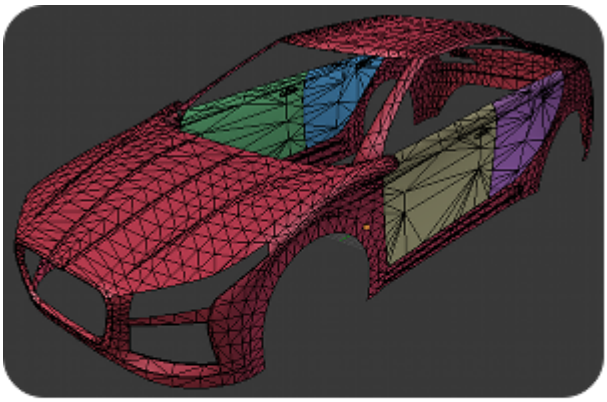
A reduced number of nodes lead to fewer node-related render state activations that have to be processed. Thus, whenever geometry can be safely combined to a single node like e.g. a mesh, then this way should be favoured. For instance a tire and a rim form a logical group sharing the same transformation. In that case the designer can unify separate geometries in a single mesh with the aid of a digital content creation (DCC) tool.
Consequently, the rim and tire do their texture lookup in a single texture, which further leverages performance gains by texture caching.

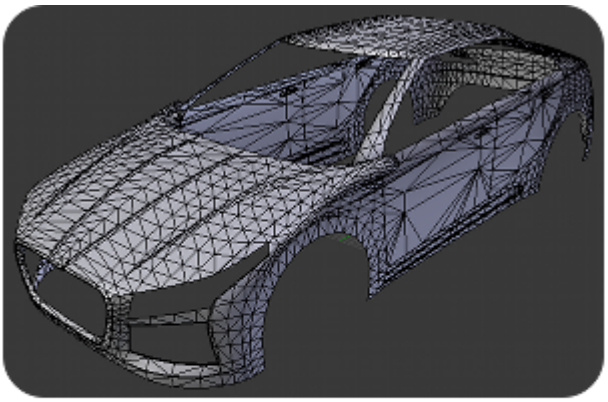


Example: Low-polygon wheel model consisting of tire, rim and brakes. The textures have been combined in a single texture atlas instead of using 3 separate textures.

See below crucial prerequisites for combining Nodes:

- Nodes share same transformation (position, rotation, scale).
- Nodes share the same Appearance (textures, shaders, material, render-mode, render-order)
- Nodes are within the same render-order bin.





Example: Instead of having 5 separate nodes for the chassis and doors, they can be combined into a single mesh if they don't need to move independently (i.e. the doors can't open) and have the same appearance (i.e. a car-paint).

Avoid Buffer Clearing

Clear only buffers that need to be cleared.

- Color buffer: The color buffer does not have to be cleared if the scene is known to cover all of it.
  For instance, this is the case if the background image or a sky map covers the whole surface.
- Stencil buffer: Only needs to be cleared if it is actually used.
- Depth buffer: If depth test is enabled.

Disable Depth Test and Adjust Depth Range

Disable Depth Test if Not Required

If it can be easily predicted that an object is not occluded by other objects then disable depth test (See Render-Mode).

Adjust Depth Range

As the float resolution in depth buffer is highly limited, the depth range should be as small as possible to reduce depth-fighting artifacts. Further, geometry beyond the near and far plane can be discarded in the clipping stage

Vertex Buffer Optimization

Vertex Buffer Editor

Loading of vertex geometry has impact on the bandwidth and on the performance of the caching mechanism for vertices. Scene Composer offers a Vertex Buffer Editor that can be used to remove the unused/unneeded vertex buffer attributes so the performance increase.

Read more...

Shader Consistency Checker

The information contained in the vertex buffer and the shader attributes should match otherwise conversion is needed and that costs performance. Scene Composer validates the shader parameters, the attributes and the uniforms defined in shaders and gives warnings that can indicate performance issues.

Read more...

Node Attachment Optimization

The more node attachments are shared the more performance increases because of reusing the data instead of loading it. SceneComposer provides a wizard to automatically detect node attachments which could be replaced by shared node attachments templates.

Read more...

Startup Time

Loading Data

Asset Tool can be used to partition an asset for a startup scene in a faster flash memory.

An another option would be to load a small startup scene and use asynchronous asset loading to load next scene in background.

Loading a lot of data at startup is generally a bad idea. Distribute loading tasks over time.

Also avoid to copy the whole asset from NAND flash to RAM. A better solution would be to implement a paging and caching mechanism using a custom Asset Repository.

Persist Pre-compiled Shaders

Use this feature by enabling CANDERA_SHADER_PROGRAM_PERSIST_INTERFACE_ENABLED flag in the CMake project.

Scope Masks

Upload data to VRAM using Scope Masks as presented in the documentation and in the example.

Startup Animation

Here are some hints for the startup animation:

- Avoid large images
- Use image compression
- Restrict the frame rate (e.g. 30 fps)
- Realize a flipbook startup animation with asynchronous image fetching