User Guide

QuerySample

What Is the QuerySample?

The NVSDK QuerySample shows how to use the various query types exposed in DirectX9. DirectX9 supports the following queries (see also DirectX SDK documentation, 'Queries'):

- Vertex Cache Size (D3DQUERYTYPE_VCACHE)
- Resource Stats (D3DQUERYTYPE_RESOURCEMANAGER)
- Vertex Stats (D3DQUERYTYPE_VERTEXSTATS)
- Event (D3DQUERYTYPE_EVENT)
- Occlusion (D3DQUERYTYPE_OCCLUSION)
- Timstamp (D3DQUERYTYPE_TIMESTAMP)
- Timestamp Disjoint (D3DQUERYTYPE_TIMESTAMPDISJOINT)
- Timestamp Frequency (D3DQUERYTYPE_TIMESTAMPFREQ)
- Pipeline Timings (D3DQUERYTYPE_PIPELINETIMINGS)
- Interface Timings (D3DQUERYTYPE_INTERFACETIMINGS)
- Vertex Timings (D3DQUERYTYPE_VERTEXTIMINGS)
- Pixel Timings (D3DQUERYTYPE_PIXELTIMINGS)
- Cache Utilization (D3DQUERYTYPE_CACHEUTILIZATION)

This document discusses how to use the sample and then goes into detail what information the different query types provide.

Using this Sample

After starting the QuerySample, it displays on the left side of the screen a slowly rotating pinwheel. On the right side of the screen it shows the results of issuing the various query types. See Figure 1.

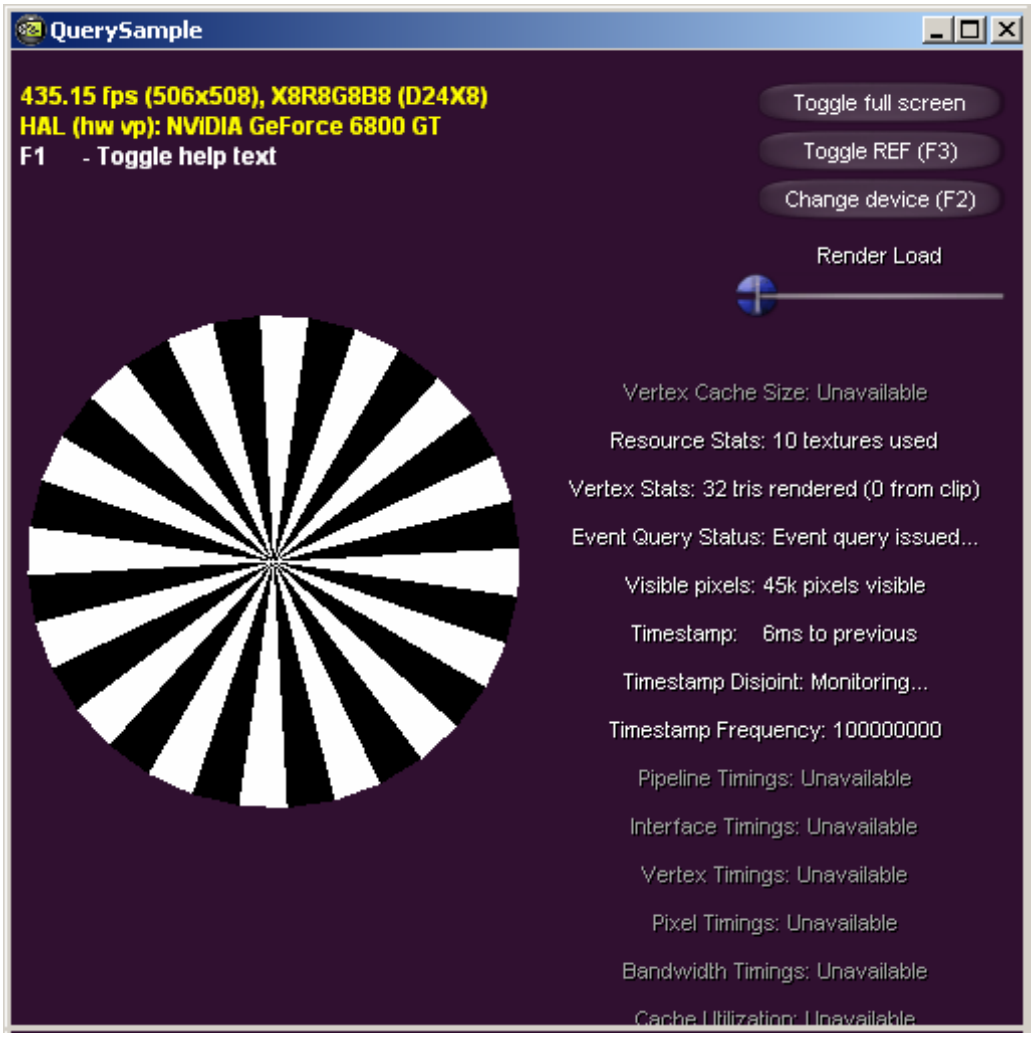

Figure 1. QuerySample on Start-up

The sample provides a standard set of UI elements for choosing rendering and display parameters. Table 1 summarizes the keyboard shortcuts to these elements.

The "Render Load" slider in the upper right corner increases the render load of the scene. At the left most settings the polygons in the pinwheel are rendered once. At the right most setting the same draw call is repeated 20,001 times. Since the z-test is set to 'less' all pixels in the pinwheel are drawn exactly once on-screen, even with the slider at the right most setting.

Right-click dragging the mouse rotates the pinwheel; the mouse wheel zooms the camera in and out.

Table 1. Keyboard Commands

Key	Description
F1	Toggle help text
F2	Change Device
F3	Toggle Ref
H	Toggle UI
Alt + Enter	Toggle Full screen mode
Esc	Exit

Known Bugs

None.

What this Sample Shows

This section describes in detail what the various queries measure and what is necessary to access a query's data. The sample comes with full source code and hence serves as a working example as how to use and access each of the query's data.

Vertex Cache Size

The vertex cache query returns hints as to how to best optimize a mesh's vertex layout for the GPU's PostTnL cache. It returns the optimization method to use (longest strips versus vertex cache usage), and the size of the vertex cache.

All current NVIDIA GPUs prefer the vertex cache usage and have a PostTnL cache that is a strict FIFO. The cache size is 16 vertices for GeForce 1, GeForce 2, and GeForce 4 MX architectures and 24 vertices in all GeForce 3, GeForce 4, GeForce FX, and GeForce 6 architectures. For more information, see also the NVIDIA NVTriStrip library: http://developer.nvidia.com/object/nvtristrip_library.html.

Current NVIDIA drivers (71.xx and earlier) do not support the vertex cache size query.

Resource Stats

The resource stats query provides detailed information for all Direct3D resources currently allocated by your application. It provides statistics for all allocated 2D, volume, and cubemap textures, vertex and index buffers, as well as surfaces.

For each of these resource types it shows whether that resource is thrashing, how many total bytes are allocated, etc. See D3DRESOURCESTATS Structure in the DirectX SDK manual for details.

The resource stats query is only supported when running with the DirectX Debug Runtime. The query sample provides source code that shows how to query and access the resulting data.

Vertex Stats

The vertex stats query shows how many triangles are being rendered and how many of these rendered triangles are generated due to clipping. Moving the render load slider therefore changes the displayed vertex stats results.

Like the resource stats query above, the vertex stats query is only available when running with the DirectX Debug Runtime.

Event

An event query inserts a token into the driver's command stream. The driver signals the query as completed as soon as the GPU consumes it. Event queries are supported across all NVIDIA architectures since the release of DirectX 9.

Event queries are useful to determine (and hence potentially restrict) how much rendering data the driver is currently buffering.

Occlusion

An occlusion query returns how many pixels pass the z-test during the time the query is outstanding. Changing the render load slider in the QuerySample therefore does not change the number of visible pixels, as only the first draw call's pixels pass the z-test (see above). Rotating the pinwheel, however, does change the number of pixels reported visible, since fewer (or more) pixels are drawn.

All GeForce 3, GeForce 4, GeForce FX, and GeForce 6 architectures support occlusion queries.

Occlusion queries determine whether an object is visible on-screen, and hence are useful in culling algorithms. Note that an occlusion query typically has high latency, i.e., it takes a non-trivial amount of time (measured on the order of frames) between issuing an occlusion query and receiving its result.

Timestamp

A timestamp query acts very much like an event query (see above). Unlike an event query, however, it also records at what time the GPU consumes the timestamp token.

Timestamp tokens are therefore useful in determining relative timings of how long it took the GPU to consume a sequence of tokens without requiring the CPU to constantly poll query states.

Note, however, that the timestamp tokens operate using their own timer that does not directly relate to wall-time. Attempting to relate timestamp queries to wall-time typically results in time-drifting – the QuerySample source code has commented out code that shows that effect. See also the source code of the QuerySample for how to decode the timestamp data.

NVIDIA drivers 71.xx and later support timestamp queries across on GeForce 6 and earlier architectures.

Timestamp Disjoint

The timestamp disjoint query lets developers monitor whether the timestamp query's (see above) counter frequency has changed. Counter frequencies may change, for example, when running on a laptop that switches into battery savings mode.

NVIDIA drivers 71.xx and later support timestamp disjoint queries across GeForce 6 and earlier architectures.

Timestamp Frequency

The timestamp frequency query provides the frequency at which the timestamp query's (see above) counter operates. The QuerySample source code shows how to use the result of the timestamp frequency query to convert the timings provided by a timestamp query into milliseconds.

NVIDIA drivers 71.xx and later support timestamp disjoint queries across GeForce 6 and earlier architectures.

Pipeline Timings

The pipeline timings query provides GPU utilization and timing data. Current NVIDIA drivers (71.xx and earlier) do not support this type of query.

Interface Timings

The interface timings query provides GPU utilization and timing data. Current NVIDIA drivers (71.xx and earlier) do not support this type of query.

Vertex Timings

The vertex timings query provides GPU utilization and timing data. Current NVIDIA drivers (71.xx and earlier) do not support this type of query.

Pixel Timings

The pixel timings query provides GPU utilization and timing data. Current NVIDIA drivers (71.xx and earlier) do not support this type of query.

Cache Utilization

The cache utilization query provides GPU vertex and texture cache hit rates. Current NVIDIA drivers (71.xx and earlier) do not support this type of query.

Notice

Trademarks

Copyright

NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
www.nvidia.com