



*graphic***REMEDY**

Maximize your
debugging & profiling skills

Avi Shapira
Graphic Remedy

OpenGL debugging & profiling model

Application is sending API calls



Something happens here!



Why is OGL debugging & profiling difficult?

The graphic system is viewed as a “black box”

- A render context is a huge “global” state machine
- You cannot watch state variable values
- You cannot put breakpoints on ES API functions
- You cannot view allocated graphic objects (textures, buffers, etc)
- Pipeline stages are highly parallel and programmable
- Caching is used in different stages
- Hardware uses a lot of algorithms
- IHVs keep implementation details secret...

gDEBugger can help you

Debugging with gDEBugger

- Transforms OpenGL debugging into a “white box” model
 - Lets you see what is happening within the graphic system implementation
 - Manipulate graphic pipeline stages to see the resulting effects
- Locate and fix problems faster!

Profiling with gDEBugger

- View the workload of each pipeline stage
- Disable pipeline stages to see the effect on render performance
- Locate performance bottlenecks quickly and easily!

Customer benefits

- **gDEBugger**
 - Shortens development time
 - Improves application quality
 - Optimizes application performance
 - Shortens render time and improves visual effects

- **gDEBugger ES**
 - Brings all of gDEBugger's debugging and profiling abilities to the OpenGL ES developer's world
 - Built-in OpenGL ES implementation when working on Windows PC



Demo

The screenshot displays the gDEDebugger - GRTeaPot interface with several panels:

- OpenGL Function Calls History:** Lists recent calls such as `glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)`, `glUseProgramObjectARB(3)`, `glUniform1fARB(0, 0.70)`, `glStringMarkerGREMEDY(Drawing scene objects)`, `glBindTexture(GL_TEXTURE_2D, 6)`, and `glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, ...)`.
- OpenGL State Variables:** Shows current state for `GL_VIEWPORT` (0, 0, 400, 400), `GL_PROJECTION_MATRIX` (2.00, 0.00, 0.00, 0.00)(0....), and `GL_MODELVIEW_MATRIX` (1.00, 0.00, 0.00, 0.00)(0....).
- Calls Stack:** Shows the current call stack starting with `tpDrawScene - grteapotapplication.cpp, line 1206`.
- Properties:** Displays details for the selected function: `tpDrawScene`, file path `c:\program files\graphic remedy\gdebugger\examples\teapot\src\grteapotapplication`, and line number `1206`.
- Performance Graph:** A line graph showing performance metrics over time.
- Counter Name Table:**

Counter Name	Value
Frames/sec: Context 1	64
CPU 0 Utilization	5
GPU0: % vertex_shader_busy	0
GPU0: % gpu_idle	92
GPU0: vertex_count	n
- Performance Dashboard:** A bar chart showing performance metrics for different components: Fra... (64), CPU... (5), GPU... (0), GPU... (92), and GPU... (100).
- Function Calls Statistics:**

OpenGL Function Name	%	# of Calls in Previ
glMaterialfv	9.30	4
glMatrixMode	9.30	4
glPopMatrix	6.98	3
glPushMatrix	6.98	3
glRotatef	6.98	3
glUseProgramObjectARB	6.98	3

Available performance counters

gDEBugger	ATI	NVIDIA	Win32
Frames / sec	% Hardware Busy	% GPU Idle	CPU utilization
# OGL function calls	% TCL Busy (Vertex Processor)	% Driver Waiting	CPU user mode utilization
# Texture objects	% VTX Fetch Busy	% Vertex Shader Busy	CPU privilege mode utilization
# Loaded texels	# Pixel Passed Z	% Pixel Shader Busy	Available physical memory
* All counters are "per render context" and available on all graphic hardware's	# Pixel Processed	% ROP Busy	Virtual memory usage
	# Pixel Blended	% Shader Waits for Texture	Virtual memory pages / sec
	# Post-cull Point Prims	% Shader Waits for ROP	Drivers virtual memory usage
	# Post-cull Line Prims	Video Memory Usage	
	# Post-cull Tri Prims	AGP / PCI-E Memory Usage	
	# Pre-cull Vertices	# Vertex Count	
	# Pre-cull Vertices	# Frame Batch	
	# Pre-cull Point Prims	# Frame Vertex	
	# Pre-cull Line Prims	# Frame Primitive	
	And more...	And more...	

Free gDEDebugger License for Academic users!

OpenGL ARB and Graphic Remedy Academic Program:

- Annual program for all OpenGL Academic users
- License of the full feature version for one year
- Includes all software updates
- A limited number of free licenses available for non-commercial developers who are not in academia

➤ Apply now

- Khronos booth
- Online: <http://academic.gremedy.com>



In the pipeline...

- Linux port – will be released in April 2007

The screenshot displays the gDEBugger application window titled "gDEBugger - tutorial2". The interface includes a menu bar (File, Edit, View, Debug, Breakpoints, Tools, Help) and a toolbar with various debugging icons. The main area is divided into several panels:

- OpenGL Function Calls History:** Lists function calls for "Context 1 - 82 OpenGL function calls", including `glColor3f(0.00, 0.00, 1.00)`, `glVertex3f(-10.00, -10.00, 10.00)`, `glEnd()`, and `glFlush()`.
- OpenGL State Variables:** Shows variables like `GL_VIEWPORT` (0, 0, 640, 480), `GL_PROJECTION_MATRIX`, and `GL_MODELVIEW_MATRIX`.
- Properties:** Displays details for the selected variable `GL_MODELVIEW_MATRIX`, including its name and value.
- Performance Graph:** A line graph showing performance metrics over time, with a table listing counter names and values: `Frames/sec: Context 1` (5148), `OGL vidmem MB` (4), `gpu_idle` (11), and `culled_primitive_count` (0).
- Performance Dashboard:** A bar chart showing performance metrics for different categories: `Frame...` (5545), `OGL v...` (256), `gpu_j...` (100), and `culle...` (100).
- Function Calls Statistics:** A table showing the percentage and number of calls for various OpenGL functions in the previous frame.

OpenGL Function Name	%	# of Calls in Previous Frame
<code>glColor3f</code>	42.86	36
<code>glVertex3f</code>	42.86	36
<code>glRotatf</code>	3.57	3
<code>glFlush</code>	2.38	2
<code>glBegin - GL_TRIANGLES</code>	1.19	1
<code>glClear</code>	1.19	1

- Windows Vista support
- Visual Studio 2005 Integration
- Buffer viewer - display pbuffers, FBOs, etc.
- And more...

Questions? More info...

- info (at) gremedy.com
- Khronos booth Wed-Fri
- Fully functional 30-day trial version available at: www.gremedy.com

