



Status Update

OpenGL ARB Meeting

June 18-19, 2002

Technical Progress

- First draft of OpenGL Shading Language Specification was released to ARB participants last week
- First draft of GL2 extension specs to support the OpenGL Shading Language were released to ARB participants last week
- Most of the functionality in these documents has been implemented by 3Dlabs

OpenGL 2.0 Functionality Demo

- Can create/delete shader objects and program objects
- Can compile shader objects
- Can link program objects
- Can use program object as part of current state
- Can set user-defined vertex attributes
- Can set uniform values
- Demo shows that shader source code files can be modified, saved, re-read by the application, and then compiled/linked/used by OpenGL

White Papers "GL2" Extensions

- First draft complete, as promised at the last ARB meeting
- Allows implementation, usage, and adoption of OpenGL 2.0 features before OpenGL 2.0 spec is finalized
- **GL2_shader_objects**
 - Supports generic operations on objects
 - Introduces shader objects and program objects
 - Defines interaction with shaders through uniform variables
 - Defines loading, compiling, linking, and using shaders
- **GL2_vertex_shaders**
 - Introduces vertex shader objects
 - Defines loading of vertex attributes
 - Defines interaction with the OpenGL rendering pipeline
- **GL2_fragment_shaders**
 - Introduces fragment shader objects
 - Defines interaction with the OpenGL rendering pipeline

OpenGL Shading Language

- First draft of spec complete
- Some issues have deliberately been left listed as “open”
- Need review and comment to help polish it
- We have released source code for a lexical analyzer and parser (this is now slightly out-of-date, will be updated)
- People have been downloading and experimenting with it

Next Steps (Technical)

- **Finalize OpenGL Shading Language Specification**
 - 3Dlabs will continue editing the spec
 - 3Dlabs will update the open source lex/parser to conform to the latest spec
 - This can happen relatively quickly
- **Finalize GL2 extension specifications**
 - 3Dlabs will continue editing the specs
 - 3Dlabs will consider making available the source for the device-independent parts of the GL2_shader_objects extension
 - This can happen relatively quickly
- **Implementations need to progress**
 - Someone should implement a code generator for ARB_vertex_program
 - Someone should implement a code generator for ATI_fragment_program
 - Someone could implement a code generator for DX8 vertex/pixel shaders
 - (And consider making these open source)
- **Consider additional OpenGL 2.0 functionality**
 - GL2_async_core extension spec is ready for review

Next Steps (Technical)

- **Develop a specification for an intermediate representation (IR) of the OpenGL Shading Language Specification**
 - Several reviewers have expressed strong views on the need for a hardware-independent binary representation of the language
 - The current implementation has an intermediate representation that is little more than a binary representation of the source
 - Something a little lower level (but still device-independent) is needed (e.g., a byte code representation)

Next Steps (Technical)

- Benefits of defining an IR include:
 - Intermediate representations are a proven part of optimizing compiler technology
 - Allows for some optimizations to be applied in a device-independent way
 - Compact representation
 - Can support a class of high-level languages
 - Much easier to construct shaders dynamically
 - Faster, simpler compiling and linking
 - Still hardware independent
 - Could allow us to move lex/parse of string-based HLL to GLU
 - Provides somewhat better protection of shader IP

Next Steps (ARB Process)

- Need to have a GL2 working group
- Review and comment on current specs
- Identify and resolve issues
- Set a timeline for spec completion

Next Steps (ARB Process)

- Need to agree this is the ARB's direction, no longer a 3DLabs proposal
- Will give OpenGL credibility and sustain momentum
- Agreement that this is the ARB's direction doesn't mean that the specs can't be changed (still need ARB to approve final specs)
- If this is not the direction for OpenGL, what is?

OpenGL 2.0 Proposed Timeline

OpenGL 1.4 Spec OpenGL Shading Language Spec OpenGL Foundation Extension Specs Initial Extension Implementations	OpenGL2 Foundation Extension Implementations	OpenGL2 Specification Work	OpenGL2 Spec Initial OpenGL2 Implementations
Siggraph 2002	2H02	1H03	Siggraph 2003