The OpenGL Shading Language on NVIDIA Hardware

Simon Green / Mark Kilgارد
Which Shading Language?

- **GLSL**
  - Recommended for cross-vendor development
  - OpenGL only

- **Cg**
  - Compatible with Microsoft’s HLSL
  - Can generate code for ARB_fragment / vertex program
  - New features: interfaces, subshaders
  - CgFX metafile format
  - Recommended for those who want cutting-edge features

- **HLSL**
  - Recommended for Direct3D applications
Availability

- Available for preview in 56.68 drivers (included in GDC SDK DVD)
- Linux GLSL support coming soon in Rel.60 drivers
- Supported on all GeForce / Quadro FX
  - Earlier generations support vertex shaders only
NVEmulate

- Simple tool to enable shader objects extension
- Available to registered developers
- Allows shader assembly to be dumped to files for debugging
NVIDIA GLSL Enhancements

• Supports HLSL-style types – float, half, fixed and equivalent vector, matrix types
  – half precision (fp16) is sufficient for most shading calculations (colors, unit vectors)
  – faster on GeForce FX series processors
  – no penalty on other hardware
  – makes porting shaders easier
Using Half Types Portably

- Can use preprocessor to make code portable:

  ```c
  #ifndef __GLSL.CG_DATA_TYPES
  # define half2 vec2
  # define half3 vec3
  # define half4 vec4
  #endif
  ```
Additional Features

- `#include` works (reserved word in spec but not defined)
- Support Cg standard library in GLSL
  - e.g. `refract()`
- Can use Cg code with OpenGL `ARB_shader_objects` extension using `EXT_Cg_shader`
Demos

- glsl_digital_clock
- glsl_dispersion
- glsl_physics
- glsl_skingning
- glsl_vnoise
Questions?

• Send bugs to:
  glsl-support@nvidia.com