State of the Art
Cross Platform Shader Development with FX Composer 2
Agenda

- FX Composer 2.0 Overview
- Cross-Platform Shader Authoring
- Production Pipeline Integration
- Conclusion
- Q&A
FX Composer 2.0

What and Who?

What is it for?
- Shader Authoring IDE
- Debugging and Profiling
- Scene Integration
- Asset Management

Who is it for?
- Graphics Programmers
- Technical Directors
- Technical Artists
- Artists
Your Requirements

- Handling of complex rendering
- Highly customizable layout management
- Powerful user interface
- Shader performance profiling
- Plug-in based architecture
- Scriptable
Flexible Render

- Many API & shading language combinations
  - OpenGL, Direct3D, PlayStation3™, OpenGL ES*
  - Cg, HLSL, GLSL*
- Surface and fullscreen shaders
Improved User Interface

- Customized user layouts
- Show or hide panels
- VC2005 docking style
- Save/Load layouts
Improved User Interface
Typical FX Composer Layout

Management

Coding

Properties

Textures

Info, Scripting, Errors

Preview
DEMO: Shader Authoring

- Loading Project
- Compiling
- Errors and Tasks
Project Explorer

- Manage multiple documents
- Documents contain multiple assets
- Assets are effects, materials, meshes, and other scene elements
- Organize your assets
  - Move, copy, delete, rename
  - Drag and drop
  - One or many documents
  - Effect Libraries
- COLLADA
Open asset exchange format (.dae)

Governed by the Khronos Group
- Includes numerous ISVs and IHVs
- Mature DCC plugins for extensive support

Supported by FX Composer 2
- Import & Export
- Supports effects and materials
- Facilitates asset exchange with DCC apps
- Other file formats supported
Library View

- Organize across documents
- Sort assets by type
- Visualize Assets
  - Scenes
  - Effect
- Authoring
Effect Authoring

- View effect structure
  - Techniques
  - Passes
  - Parameters

- Authoring using toolbars and context menus
  - Add children
  - Remove children
  - Advanced options
CgFX & COLLADA FX Cg

CgFX
- Hand coded
- Less user interface assistance

COLLADA FX for Cg authoring
- Most user friendly experience
- Fully editable via user interface
- Can migrate your CgFX assets
- Less hand coding
- Zero XML
- Cg, GLSL only when writing the GPU shader code
DEMO: Material Authoring

- Creating materials
- Assigning effects
- Tweaking material parameters
DEMO: Scene Integration

- Light creation
- Scene traversal
- Bind light to material
- Realtime manipulation
Shader Performance Simulation
Production Pipeline Integration

- Scene binding expressions
- Automation via scripting
- Plug-in based architecture
- Source control integration
- Production Pipeline Samples
Scene binding expressions

float4 LightPos : Position
<
string Object = "PointLight";
string Space = "Object";
> = {-10.0f, 10.0f, -10.0f, 0.0f};
Custom Semantics and Annotations

- Hook parameter to scene and system data
- Expressions via xml configuration file
- Extensive Library of Operators
  - dot & cross products, mux, demux, matrix ops, ...
- Custom operators
  - Built from XML using operators
  - Via plug-in

```xml
1 : <RemappedSemantic name="myWorldView">
2 :  <MatrixMultiply description="World * View">
3 :   <input type="internalsemantic" value="world"/>
4 :  </input>
5 :  <input type="internalsemantic" value="view"/>
6 : </MatrixMultiply>
7 : </RemappedSemantic>
```
Automation Via Scripting

- Automatic assignment of materials to geometry
- Shader parameters to scene objects (nearest lights, cameras, etc...)
- A model’s accessories to attachment points

- Common-tasks toolbar (ala Maya/MEL)
Python scripting

# Convert any Possible Profile to COLLADA FX
def ConvertToCOLLADA():
    effects = FXRuntime.Instance.Library.FindLibraryItems(FXEffect)
    for effect in effects:
        for profile in effect.Profiles:
            if profile.CanConvertToColladaFX() == True:
                profile.ConvertToColladaFX()

# Create an effect
def bindMMM():
    CmdGroupBegin.Do("script: assign cgfx files to MMM ")
    SelectRenderPort("OpenGL")
    ForceRedraw()
    CmdGroupEnd.Do()

    # get the cgfx files to assign to MMM
    files = FXEffectUtils.GetEffectFiles()
    for pathname in files:
        ...

DEMO: Scripting

Automatic effect loading & assignment
Custom Plug-ins

- Importers
- Exporters
- Semantic expression operators
- Rendering devices
- Procedural geometry generation (fins, hair, etc...)
- Custom authoring environment

...etc...
Custom Plug-ins: Sky is the limit
Source Control Integration

- Seamless integration into source control software
- Documents and assets reflect file-based state
Pipelines: FXC2 centric

FX Composer 2.0

COLLADA Scene with FX

Conditioners and Compilers

Game Binary
- MAC
- PC
- Console
Pipelines: DCC centric

FX Composer 2.0

Effect Library

DCC Application

Scene

Conditioner

Game Binary

Console
Pipelines: Engine

FX Composer 2.0

Game Importer
Rendering Game Adapter

Game Engine

CgFX
D3D FX
COLLADA FX

Game Binary

Game Binary
FX Composer 2.0 Alpha 5

- Alpha5 release ETA end of summer ’06
  - Document and asset management
  - COLLADA FX authoring
  - Shader parameter scene binding
  - Custom semantic and annotation support
  - Python scripting
  - Shader performance
  - Available to Sony PS3 developers and limited partners

- Beta release ETA end of fall ’06
  - Open to public
Conclusion

- Next-generation of shader IDE is on its way
- Production-ready with powerful features
- NVIDIA is closely working with Khronos and others to deliver a professional-grade authoring tool
Q&A

Send us emails for early alpha and beta releases

fxcomposer@nvidia.com

Thanks

Ignacio Castaño (icastano@nvidia.com)
Daniel Horowitz (dhorowitz@nvidia.com)
The Source for GPU Programming

developer.nvidia.com

- Latest News
- Developer Events Calendar
- Technical Documentation
- Conference Presentations
- GPU Programming Guide
- Powerful Tools, SDKs and more ...

Join our FREE registered developer program for early access to NVIDIA drivers, cutting edge tools, online support forums, and more.

developer.nvidia.com

©2004 NVIDIA Corporation. NVIDIA, and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation. Nalu is ©2004 NVIDIA Corporation. All rights reserved.