Pipeline Integration with FX Composer

Chris Maughan

NVIDIA Corporation

Agenda

- FX Composer 1.6
  - Recap
  - DXSAS
  - Plugins
  - Scripting
- Preview of FX Composer 2

FX Composer History

- **FX Composer 1.0 shipped last January**
  - **~100 .fx files, ~30 projects out of the box**
  - **Support for all ps/vs profiles in DX9b**
- **FX Composer 1.5 shipped November last year**
  - **Many new features, such as DXSAS .86, scripting, SDK, etc.**

FX Composer 1.6

- **FX Composer 1.6 update in February**
  - **Mainly a service release; very stable build**
  - **New .fxproj format, unpackaged**
    - **Moving towards a standard XML format (XNA, COLLADA)**
- **1.7 in the near future**
  - **Shaderperf updates**
  - **Installer improvements**

Introducing DXSAS .86

- Specification from Microsoft
  - Updated version in the current DX SDK
  - FX Composer currently supports .86
    - Waiting for scripting additions
- Defines a standard set of semantics and annotations
- Help menu brings up the current list of annotations/semantics
  - You can also use fxmapping.xml to map your own custom annotations/semantics to the spec

Example Semantics

Semantics

Semantic	Description	Data Type	Supported	FX Composer Only
diffuse#	Color value to be used as the diffuse color. The fourth channel represents diffuse alpha.	float4,float3,texture	yes	-
specular#	Color value to be used as the specular color. The fourth channel represents specular alpha.	float4,float3,texture	yes	-
	Color value to be used as			

```
float4 myColor : DIFFUSE;
float4x4 myMat : WORLDVIEW;
float elapsed : TIME;
```

Example Annotations

Annotations

Annotation	Description	Data Type	Supported	FX Composer Only
frustum	This is type is associated with a frustum	matrix	yes	-
uiname	This is a string that describes variable, i.e. a pretty name used for labeling an ui dialog	string	yes	-
uihelp	This is the string for helpful information that is displayed to a user in a tool.	string	-	-
uiwidget	Described the widget to be used to edit the value.	string	yes	-

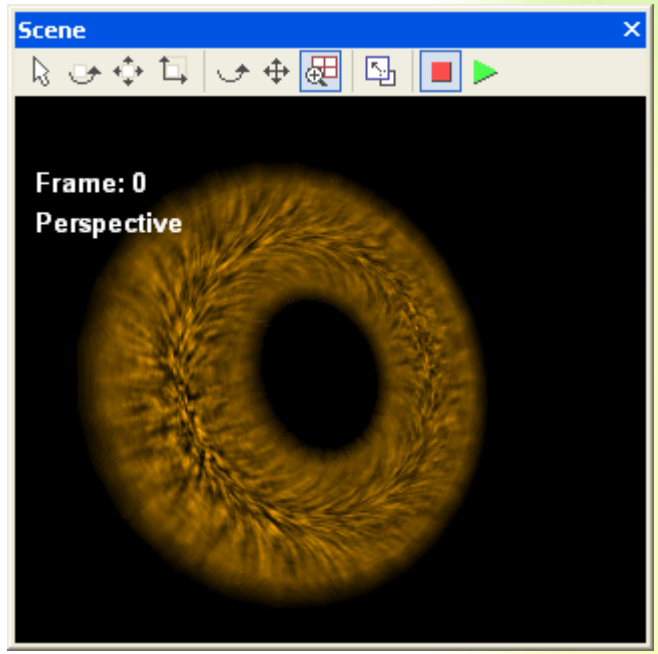
```
float4 myColor : DIFFUSE
<
        UIName=“Paint Tint”;
        UIWidget = “color”;
>;
```

ScriptExecute - .86 style

- Designed to help the effect interaction problem
- Adds powerful scripting features to effects
- A superset of the XML ‘scene commands’ that FX Composer 1.1 shipped with
  - More powerful/general
- All FX Composer effects support .86
  - Old scene command XML automatically interpreted as ScriptExecute.
  - FX Composer 2 will do SAS 1.x

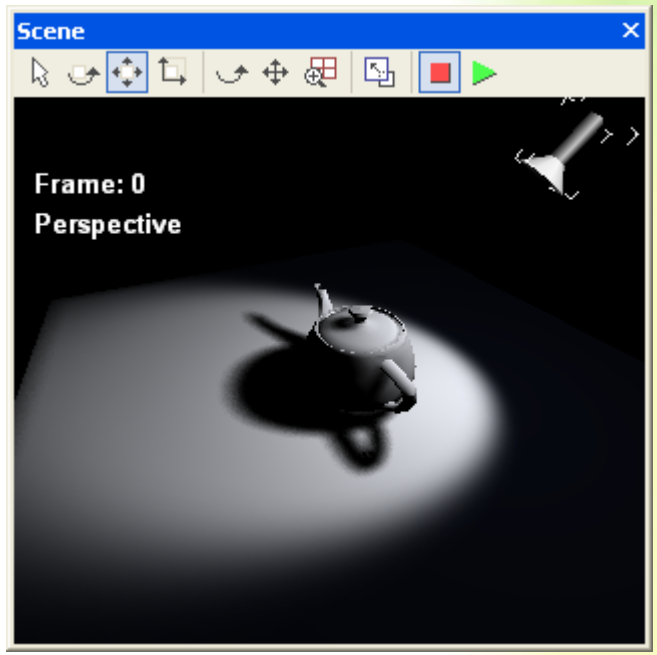
ScriptExecute: Fur Shells

- Script loops on a per-object basis
- Loop counter used to distance each fur shell
- Properties panel lets you tweak the appearance

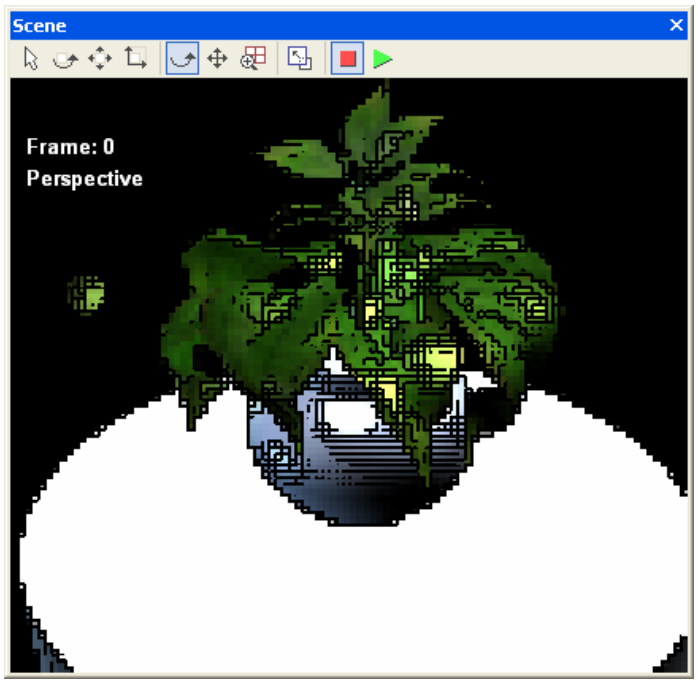


ScriptExecute: Renderport

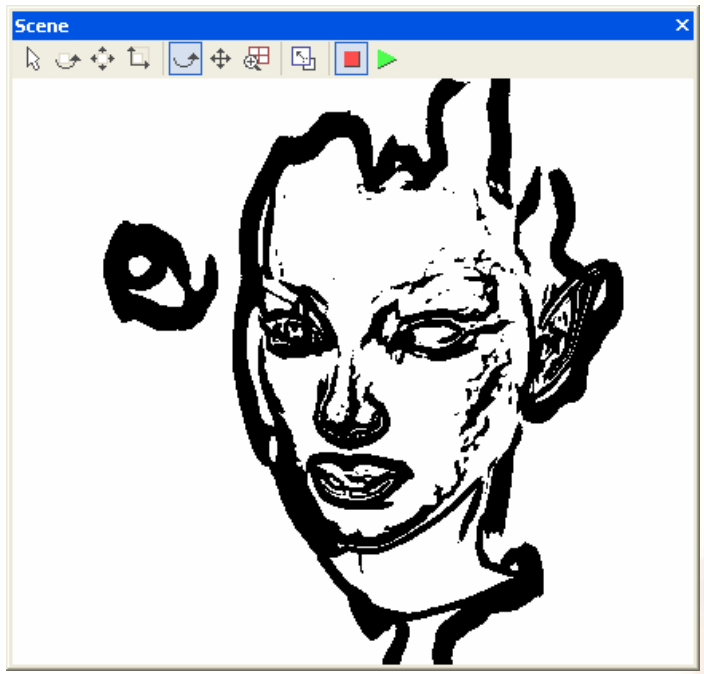
- Rendering from different POV
  - Switch camera from script
- Example scene, Soft Shadows
  - Depth map rendered from POV of light
- Current matrices are changed to use the values from the light



ScriptExecute: Shader Stacks



Tiles.fx + EdgeDetect.fx



Corona.fx + EdgeDetect.fx

ScriptExecute Summary

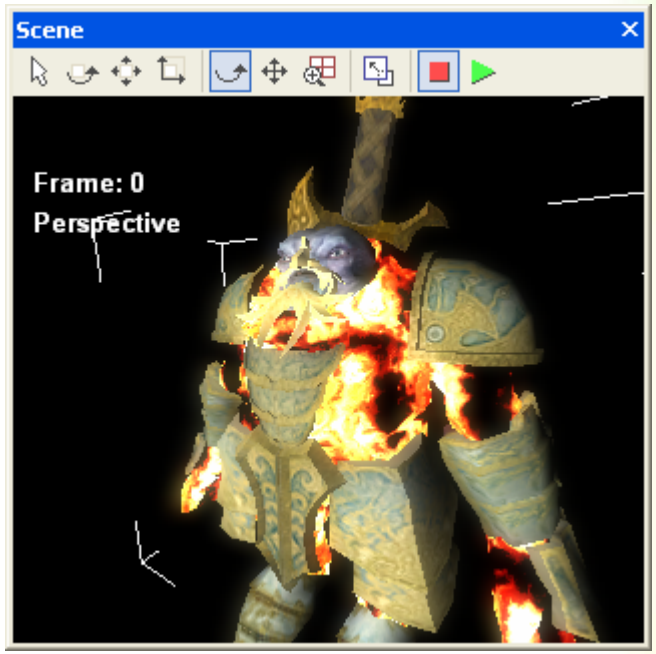
- Flexible way to solve interacting effect problems
- Not too hard to implement in the engine
  - Sample DXSAS code took about 2 weeks
- Really powerful
- A Standard – but look out for the 1.1 spec from Microsoft

DXSAS Implementations

- DCC Companies are working towards 1.x spec
- FX Composer will try to implement as full support as possible
- Microsoft working on full sample implementation for next SDK
- DXSAS .86 version implementation on latest NVSDK

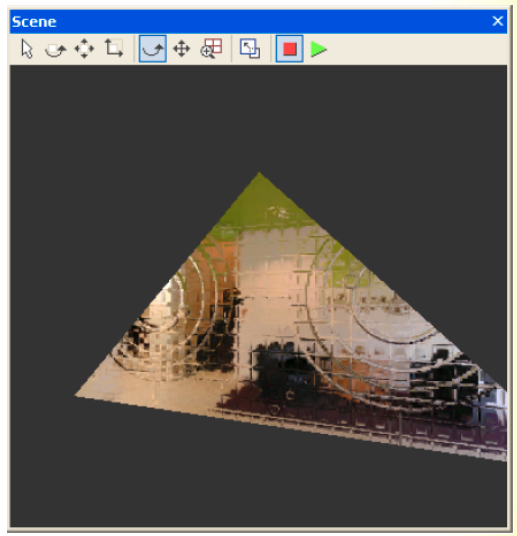
FX Composer SDK

- FX Composer 1.5 was the first version with an SDK
  - Can write plug-ins
  - Can write scripts in C# or VB.NET
- 1.6 adds samples & documentation
  - .x, .obj
  - Material exporter



Plugin Tutorial in 1.6

- **Bumpy shiny triangles!**
  - Cutting edge technology ☺
- **Already external plugins**
  - No problems reported
- **Fairly simple to build**
  - User guide has full documentation
- **VC 2003 does most of the work with a wizard**

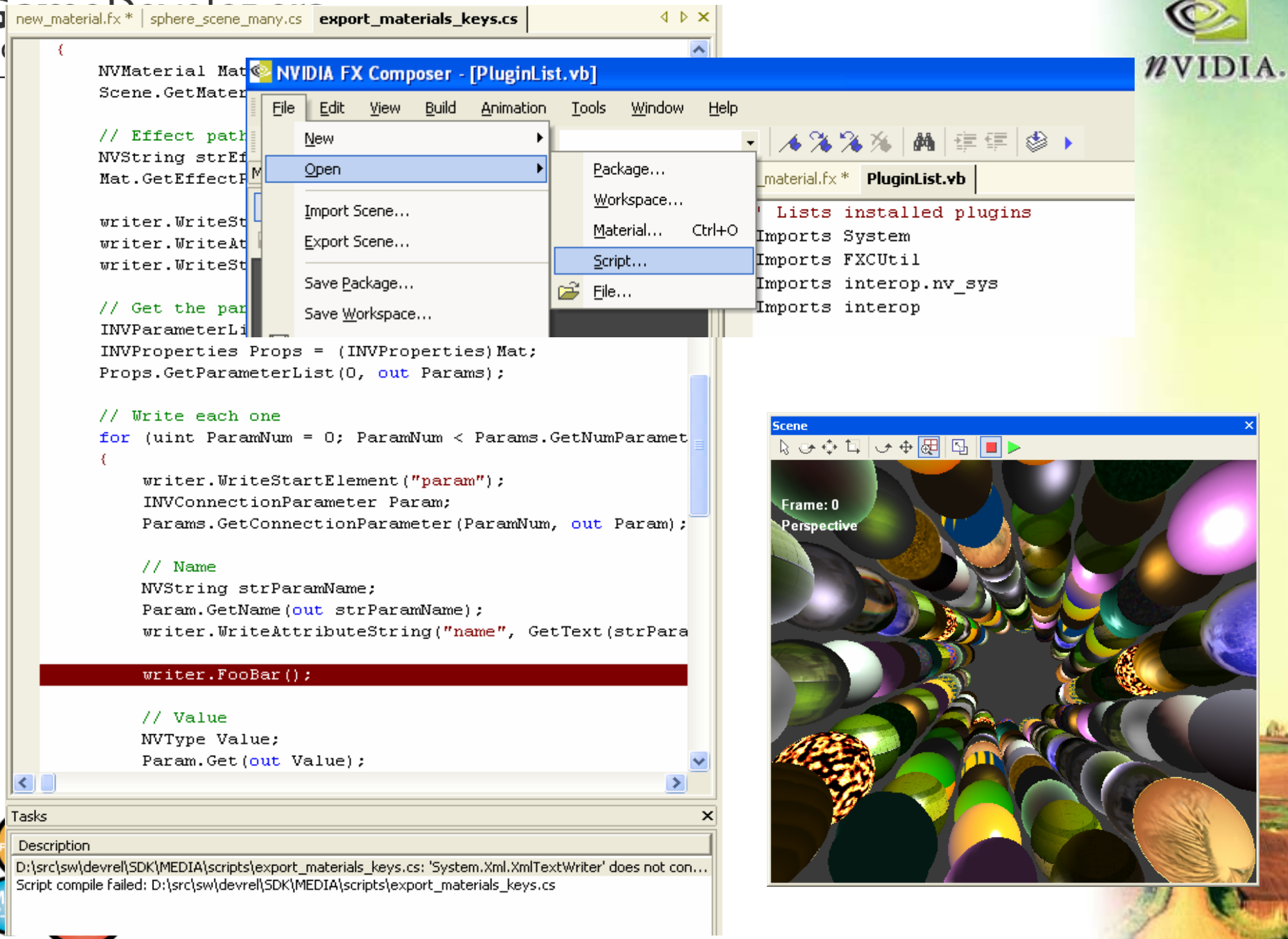


Scripting with C#/VB

- .NET scripting is a powerful new feature
  - C# and VB.NET
  - Editing & Compilation integrated into FX Composer
  - Errors displayed in task bar
    - Just like working with an effect
  - Full FX Composer engine is exposed to the script
- Disadvantages
  - No single stepping
  - No intellisense, not currently as integrated into the IDE as ‘real’ plugins

Script examples…

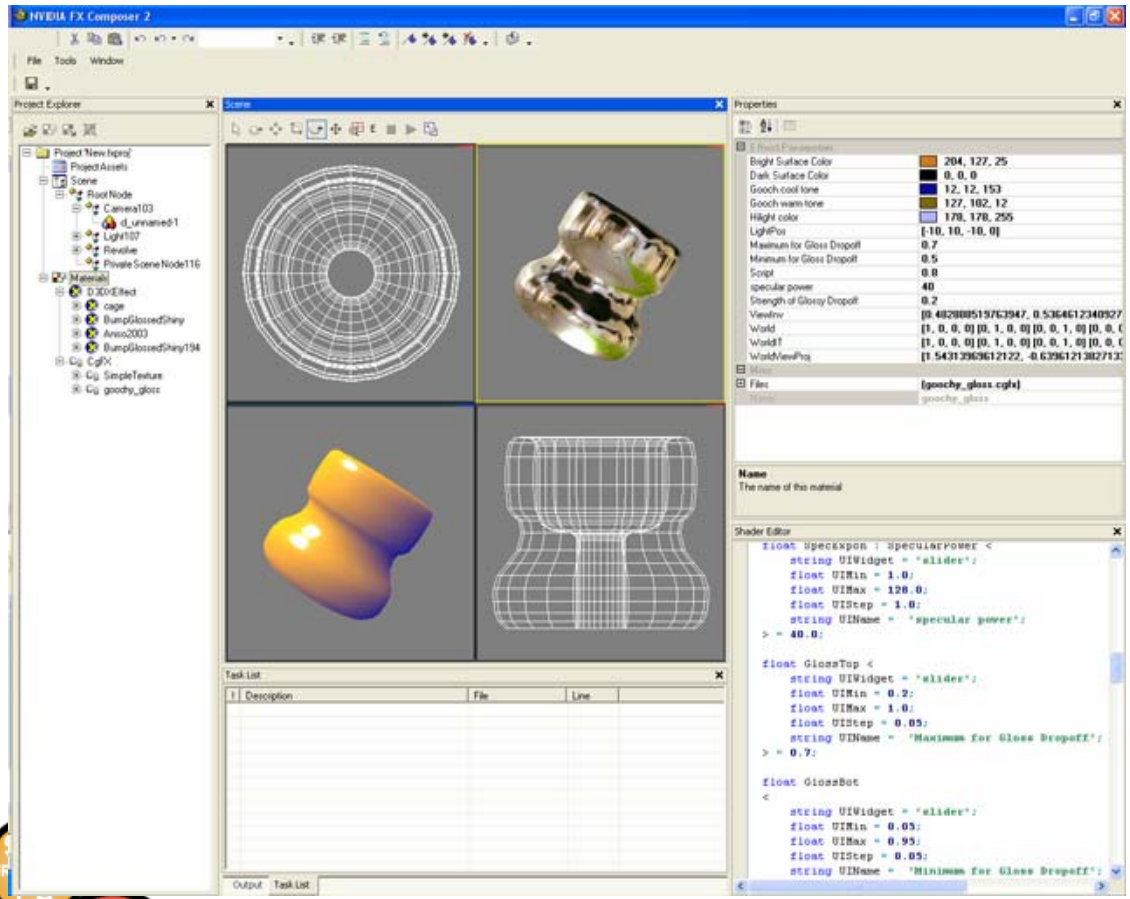
- **Examples**
  - **Import/export of scene & material data**
  - **Custom built scenes**
  - **Material parameter setting/restoring**
  - **Generation of effect files, based on data**
  - **Communication between FX Composer & your engine**
  - **Regression testing, batch processing of materials/effects**
    - **We have scripts to build screenshots of effects & projects**
    - **Other samples to copy**


new_material.fx *
sphere_scene_many.cs
export_materials_keys.cs
{
NVMaterial Mat
Scene.GetMater
// Effect path
NVString strEf
Mat.GetEffectP
writer.WriteSt
writer.WriteAt
writer.WriteSt
// Get the par
INVParameterLi
INVProperties Props = (INVProperties)Mat;
Props.GetParameterList(0, out Params);
// Write each one
for (uint ParamNum = 0; ParamNum < Params.GetNumParamet
{
writer.WriteStartElement("param");
INVConnectionParameter Param;
Params.GetConnectionParameter(ParamNum, out Param);
// Name
NVString strParamName;
Param.GetName(out strParamName);
writer.WriteAttributeString("name", GetText(strPara
writer.FooBar();
// Value
NVType Value;
Param.Get(out Value);
NVIDIA FX Composer - [PluginList.vb]
File
Edit
View
Build
Animation
Tools
Window
Help
New
Open
Import Scene...
Export Scene...
Save Package...
Save Workspace...
Package...
Workspace...
Material... Ctrl+O
Script...
File...
material.fx *
PluginList.vb
' Lists installed plugins
Imports System
Imports FXCUtil
Imports interop.nv_sys
Imports interop
Scene
Frame: 0
Perspective
Tasks
Description
D:\src\sw\devrel\SDK\MEDIA\scripts\export_materials_keys.cs: 'System.Xml.XmlTextWriter' does not con...
Script compile failed: D:\src\sw\devrel\SDK\MEDIA\scripts\export_materials_keys.cs

Soul of FX Composer 2.0

- **Major update**
  - User interface improvements
  - New engine
  - New plugin architecture
  - Device Independence
  - Platform Independence
  - Collada project files
- **Fair warning...**
  - This is still pre-alpha
  - Shipping later this year

FX Composer 2 - Screenshot



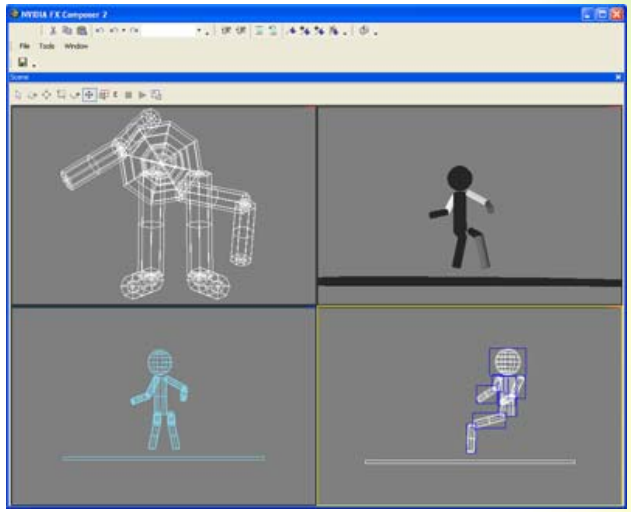
- Screenshot shows 4 Viewports, 3 are DirectX, one is OpenGL
- Also visible are the properties panel, the editor & the project browser

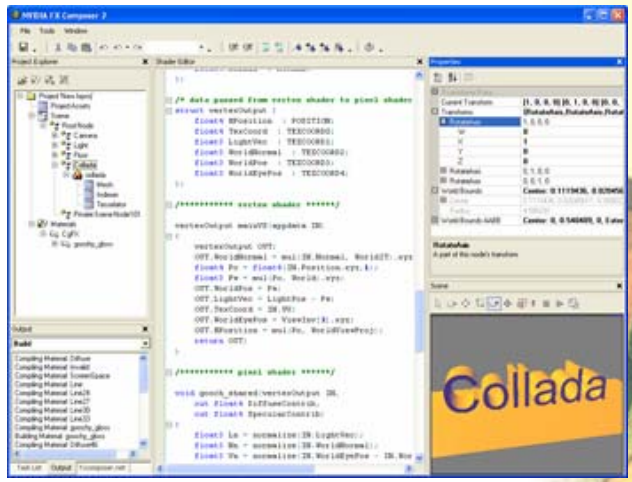
Why such a major update?

- **Improved user interface**
  - Better float dock support, plugin integration, layout management.
- **Updated engine**
  - More generalized & efficient pipeline
  - C#/.NET core
- **Multiple device and shader format support**
  - DirectX still the ‘first class’ citizen
  - Also GLSL, CgFX,…?
- **Collada project file support**
  - XNA? When it’s available…

User Interface

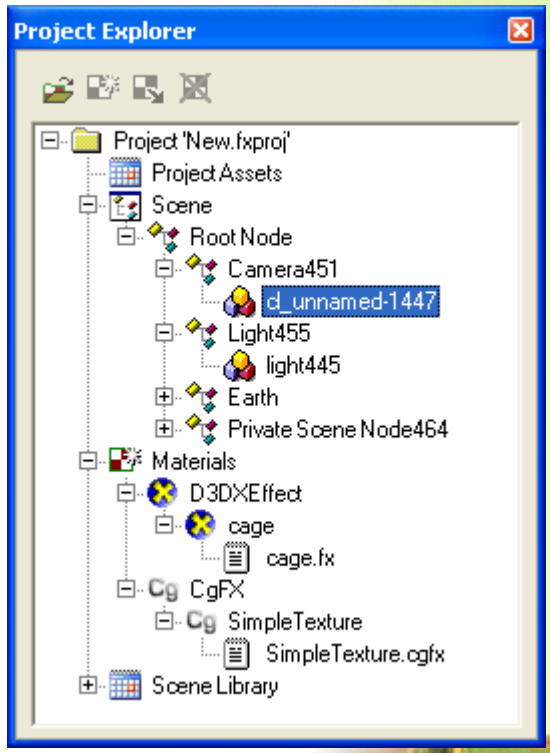
- **Supports more advanced window management**
  - Custom user layouts
  - ‘Artist’ mode, ‘Programmer’ mode, etc…
  - Doesn’t need the editor to be visible
- **Multiple viewports**
  - Each one with different device
  - Each one with different camera
    - Cameras can be shared
  - Easy viewing of same scene with different material types





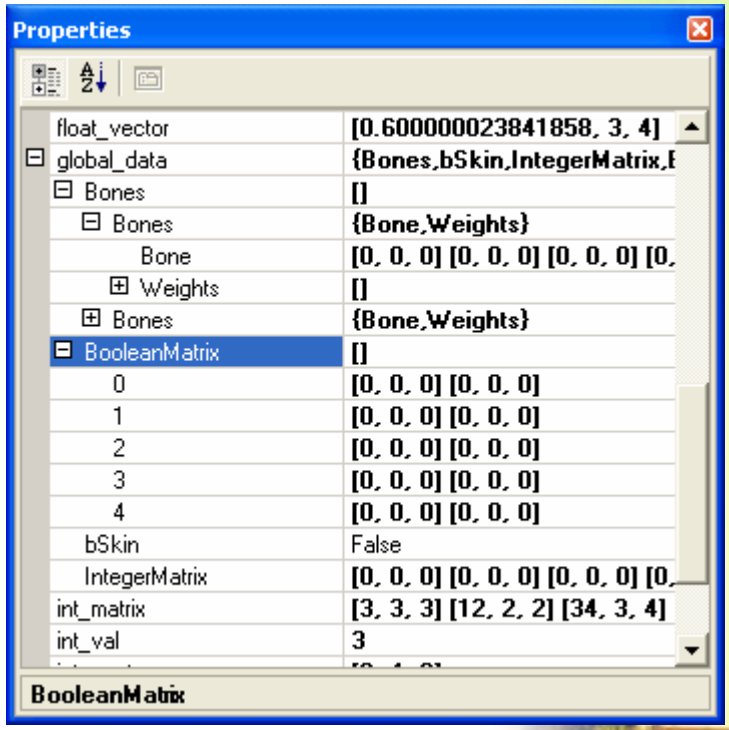
User Interface – 2

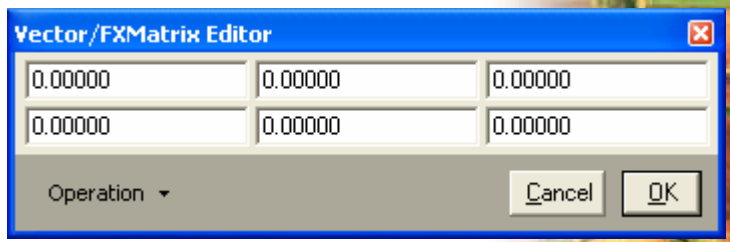
- **Project Explorer**
  - **Integration of material browser, texture browser and scene window**
  - **Control over building effects that aren't based on a metafile**
  - **Multiple material types**



User Interface – 3

- Improved property editor
  - Full support for all types
  - Structures
  - Arrays
- New widgets for editing matrices, colors, etc.



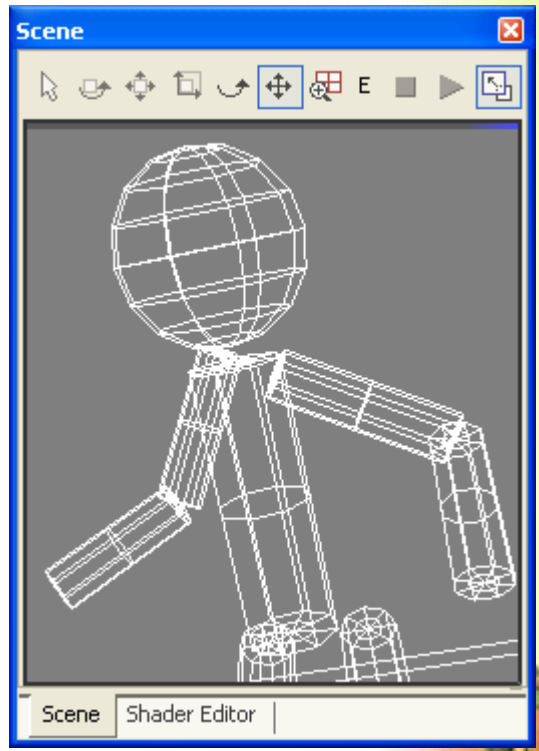


User Interface – 4

- Many usability features based on feedback
- Full drag-drop for materials, projects, tree nodes, etc.
- Better management of media files and paths
- Better management of project paths, etc.
- Still work in progress

New Engine

- **Supports n-sided polygons & multiple index sets**
  - **Needed to ensure DCC<->FXC interoperability**
  - **Tessellation to triangles before rendering**
  - **Pipeline stages can use polygon data if preferred**
- **More efficient scripting renderer**
  - **Enables DXSAS, Collada materials, etc. to run with the same core engine**

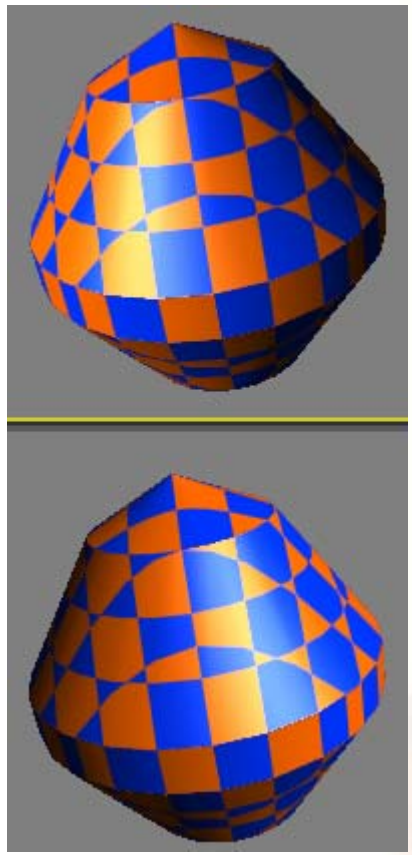


New Plugin Architecture

- Plugins now much more powerful
  - Can extend the user interface
  - Can add menu items, etc.
- Legacy plugins still supported
  - Wrapper plugin enables them to work
- Boundary between scripting and plugins more blurred
  - The same API, language
  - Can work either way

Device Independance

- Core application doesn't know or care about rendering
- All materials & devices are pluggable
  - CgFX & HLSL out of the box
  - GLSL to follow
  - SDK enables extension
    - User can implement their own device and material support
    - Works alongside existing materials in multiple viewports



Platform Independence

- FX Composer 2 is 100% .NET
- The Mono and DotGnu projects enable .NET on other platforms
  - www.mono-project.com
  - www.dotgnu.org
- Future version of FX Composer on MAC & Linux
- Watch this space!

Collada Project Files

- FX Composer 1.6 used its own XML project format
  - Proprietary, nobody liked it…
- FX Composer 2.0 can have different project formats
  - Currently Collada
  - Legacy importer/convertor provided if required for old projects

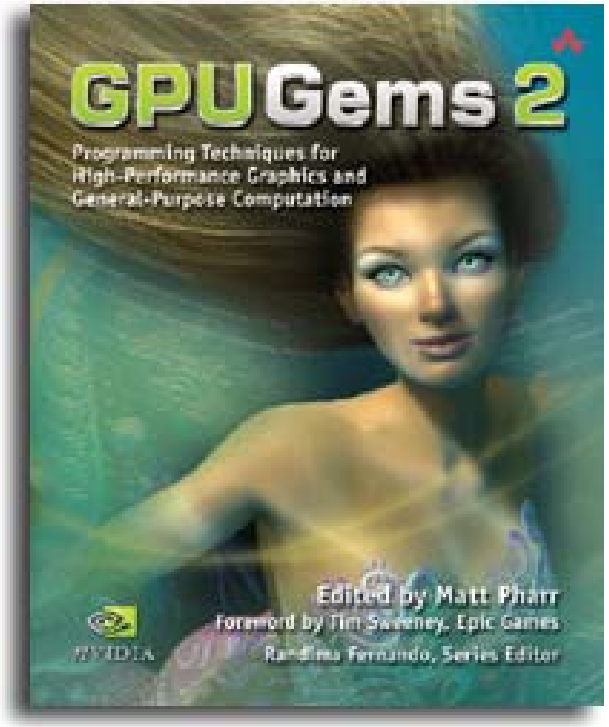
Questions?

- Suggestions, bug reports, early access
  - fxcomposer@nvidia.com
  - http://developer.nvidia.com/fxcomposer
- Me
  - cmaughan@nvidia.com

Thanks for listening...

GPU Gems 2

Programming Techniques for High-Performance Graphi and General-Purpose Computation



- **880 full-color pages, 330 figures, hard cover**
- **$59.99**
- **Experts from universities and industry**

"The topics covered in *GPU Gems 2* are critical to the next generation of game engines."

— Gary McTaggart, Software Engineer at Valve, Creators of Half-Life and Counter-Strike

"*GPU Gems 2* isn't meant to simply adorn your bookshelf—it's required reading for anyone trying to keep pace with the rapid evolution of programmable graphics. If you're serious about graphics, this book will take you to the edge of what the GPU can do."

—Rémi Arnaud, Graphics Architect at Sony Computer Entertainment