

フィリップ・ロリン  
(Philippe Rollin)



**nVIDIA**®

# 最新クロスプラットフォーム シェーダの開発

# アジェンダ

- **FX Composer 2.0の概要**
- **クロスプラットフォームのシェーダオーサリング**
- **プロダクションパイプラインの統合**
- **結論**
- **Q&A**

# FX Composer 2.0

## 対象作業と対象ユーザ

### ● 対象作業

- シェーダオーサリングIDE
- デバッグおよびプロファイリング
- シーンの統合
- アセット管理

### ● 対象ユーザ

- グラフィックスプログラマ
- テクニカルディレクタ
- テクニカルアーティスト
- アーティスト



# ユーザからの要求事項

- 複雑なレンダリングを処理
- 高度なカスタマイズが可能なレイアウト管理
- 強力なユーザインタフェース
- シェーダパフォーマンスのプロファイリング
- プラグインベースのアーキテクチャ
- スクリプト記述が可能

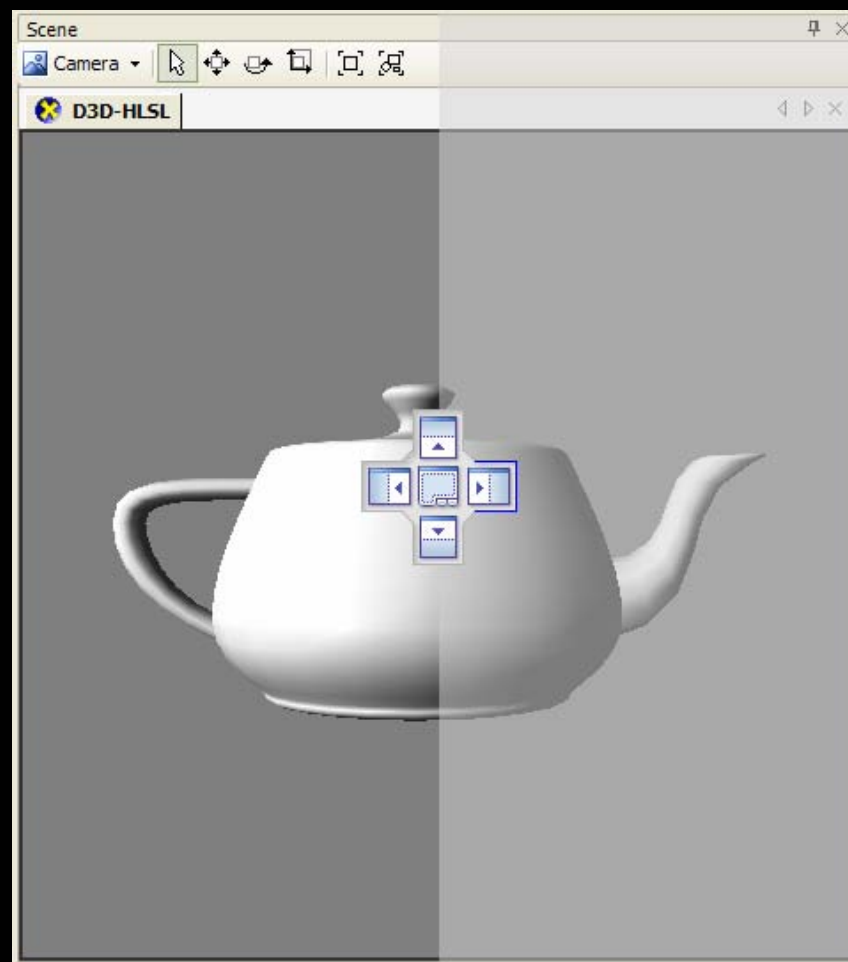
# 柔軟なレンダリング

- APIとシェーディング言語の多数の組み合わせ
- サーフェスシェーダおよびフルスクリーンシェーダ



# 向上したユーザインタフェース

- カスタマイズされたユーザレイアウト
  - パネルを表示または非表示
  - VC2005ドッキングスタイル
  - レイアウトの保存/読み込み



# 向上したユーザインタフェース



The screenshot displays a complex game engine interface with several overlapping windows and panels:

- Code Editor:** Shows a C++ shader function `float4 fresPS_t(vertexOutput IN) : COLOR` with lines 246-301. The code includes texture sampling and material property assignments.
- Scene View:** A 3D perspective view of a scene with a central cube and several spheres. A wireframe bounding box is visible around the central cube.
- Material Scene Bindings:** A dropdown menu showing `PointLight` selected, with a `<Unbound>` warning icon.
- Node Properties:** A table listing properties for the selected `PointLight` node:

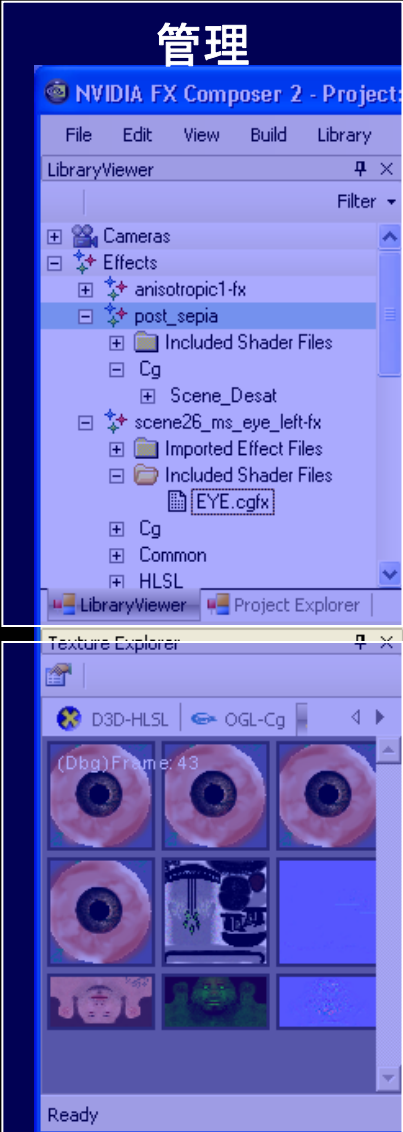
Property	Value
Name	<none>
Uri	Point_Light1 Point_Light0
Visible	
Frozen	
Object Scale	
Object Shear	
Object Rotation	
- Color Picker:** A circular color wheel with a gradient from blue to red, overlaid on a checkerboard pattern. Below it are sliders for Red (0.500), Green (0.375), Blue (0.234), and Alpha (1.000).
- Material Properties:** A list of material parameters:

Bright Surface Color	0.8 0.5 0.1
Bright Surface Color	0 0 0
Dark Surface Color	0.5 0.4 0.05
Gooch cool tone	0.05 0.05 0.6
- Slider:** A horizontal slider for `Minimum for Gloss Dr` with a value of `0.7` and a range from `0.200` to `1.000`.
- Navigation:** A central cross-shaped navigation pad with directional arrows.



# FX Composerの代表的なレイアウト

## 管理



**テクスチャ**

## コーディング



**情報、スクリプト記述、エラー**

## プロパティ



**プレビュー**

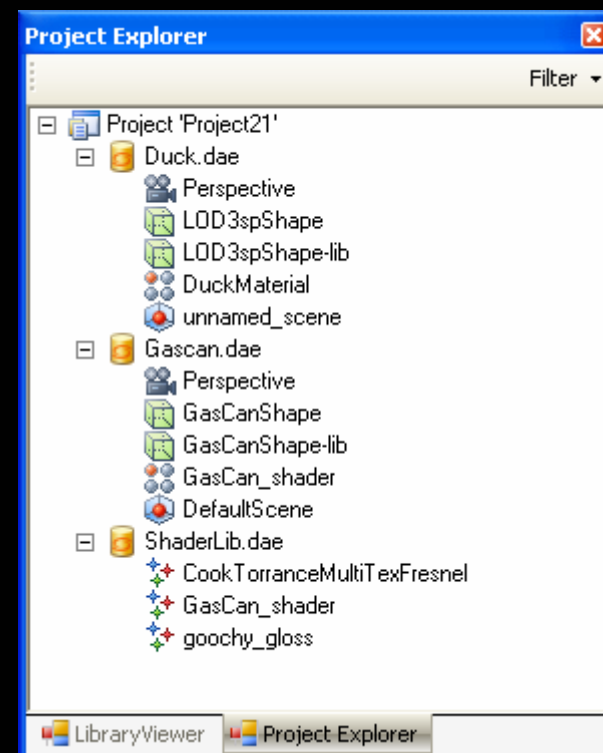


# デモ: シェーダオーサリング

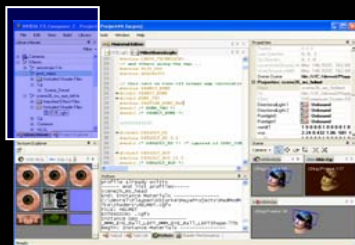
- プロジェクトの読み込み
- コンパイル
- エラーおよびタスク

# プロジェクトエクスプローラ

- 複数のドキュメントを管理
- ドキュメントは1つまたは複数のアセットを格納
- アセットは、エフェクト、マテリアル、メッシュ、およびその他のシーンエレメント
- アセットの整理
  - 移動、コピー、削除、名前の変更
  - ドラッグアンドドロップ
  - 1つまたは複数のドキュメント
  - エフェクトライブラリ



## ● COLLADA



- オープンアセットエクスチェンジフォーマット (.dae)

- クロノスグループが管理

- 多数のISVおよびIHVを含む
- 成熟したDCCプラグインで広範なサポート

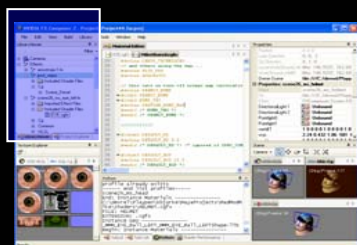
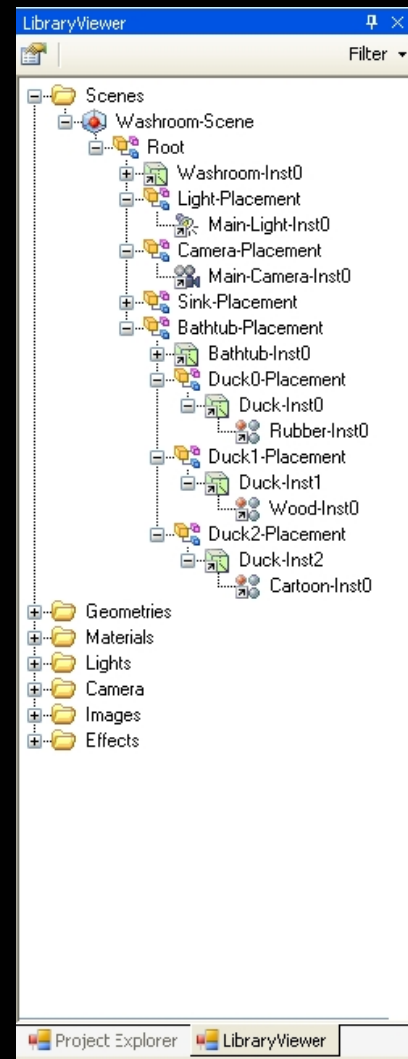


- FX Composer 2がサポート

- インポートおよびエクスポート
- エフェクトおよびマテリアルをサポート
- DCCアプリケーションとのアセット交換を円滑化
- その他のファイル形式にも対応

# ライブラリの表示

- 複数のドキュメントを整理
- アセットをタイプ別に並べ替え
- アセットの視覚化
  - シーン
  - エフェクト
- オーサリング










# エフェクトのオーサリング

## ● エフェクト構造の表示

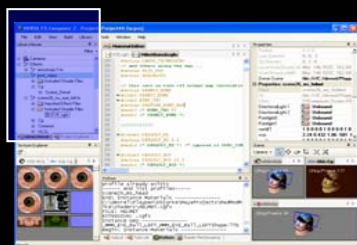
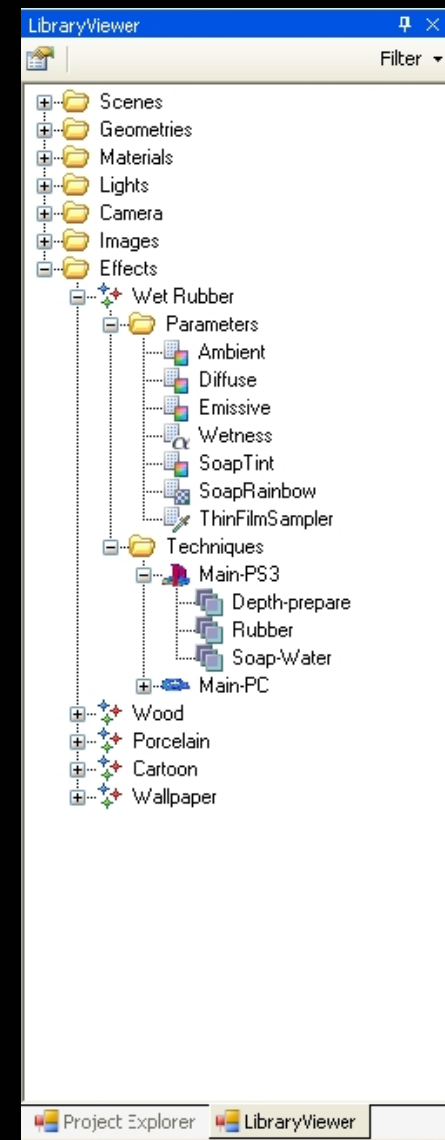
- テクニック
- パス
- パラメータ



- Vector 
- Matrix 
- Color 
- Surface 
- Sampler 
- Scalar 
- String 

## ● ツールバーとコンテキストメニューを使用したオーサリング

- 子の追加
- 子の削除
- 詳細オプション



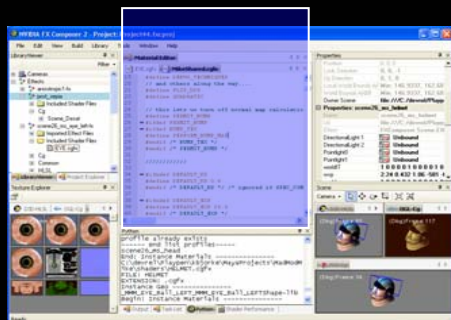
# CgFX & COLLADA FX Cg

## ● CgFX

- 手作業によるコーディング
- 少ないユーザインタフェースのアシスタント

## ● Cgオーサリング用COLLADA FX

- 最高のユーザフレンドリ体験
- ユーザインタフェースにより完全編集可能
- CgFXアセットの移動が可能
- 手作業によるコーディングを減少
  - ゼロXML
  - Cg、GLSL (GPUシェーダコードを記述するときのみ)



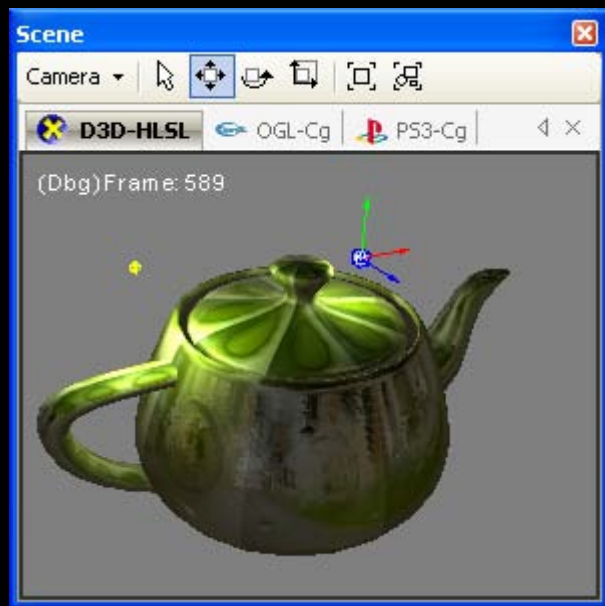
# デモ: マテリアルのオーサリング

- マテリアルの作成
- エフェクトの割り当て
- マテリアルパラメータの微調整

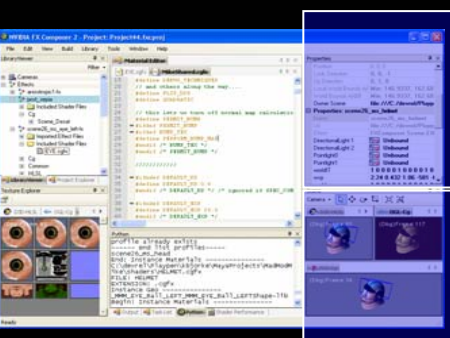


# デモ: シーンの統合

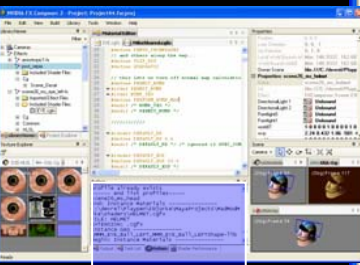
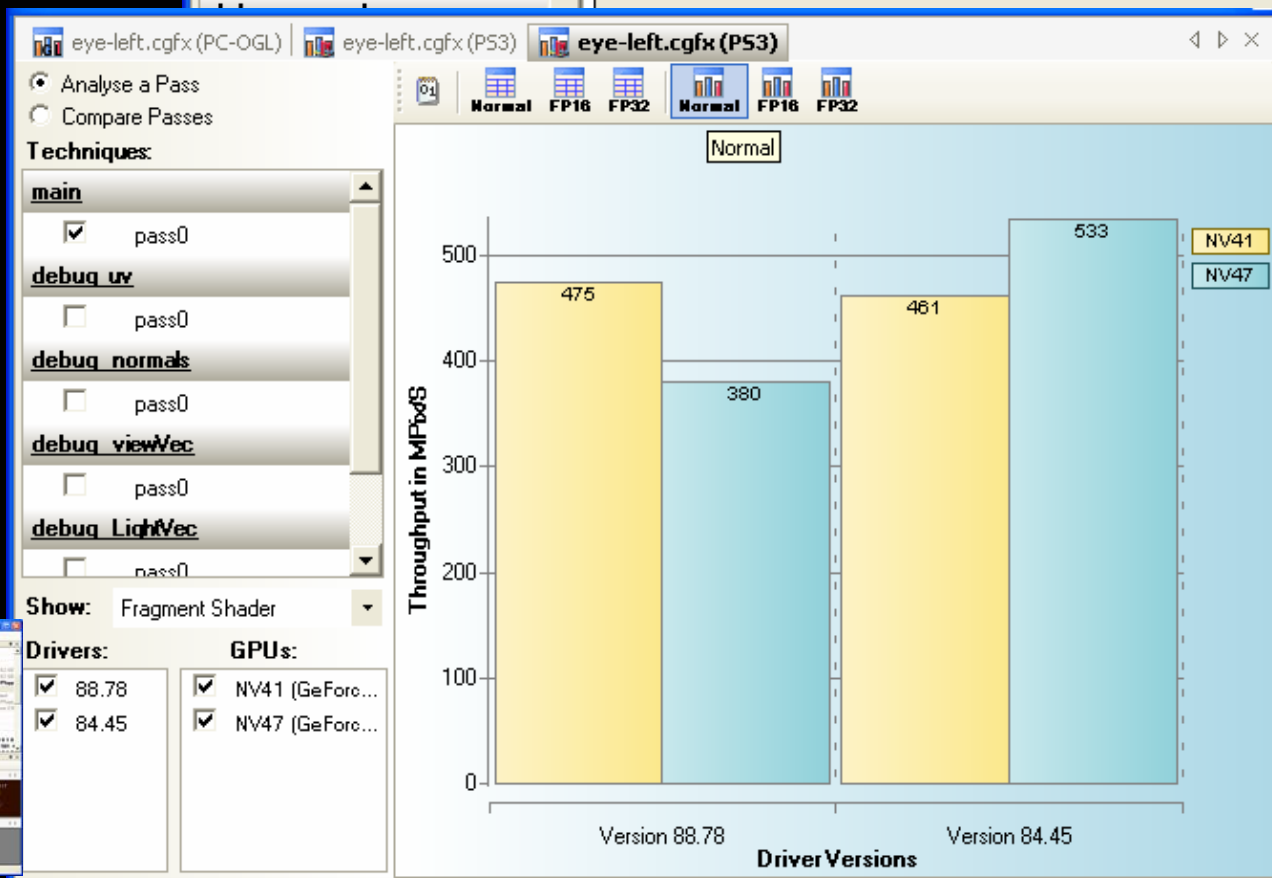
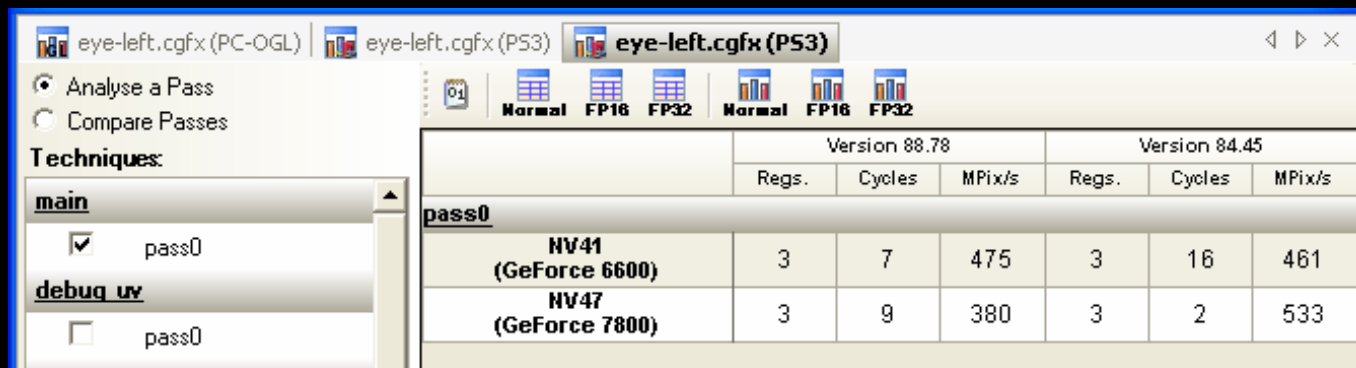
- ライトの作成
- シーンのトラバース
- ライトとマテリアルのバインド
- リアルタイム操作



Properties	
Light1	Point_Light1
Light2	Point_Light2
Light1 Position	-1.85 3.39 -0.00721 0
Light1 Color	<input type="color"/> 1 1 1 1
lamp1 power	2
Light2 Position	0.677 3.15 0.161 0
Light2 Color	<input type="color"/> 1 1 1 1
lamp2 power	0.5
Ambient Light Color	<input type="color"/> 0.07 0.07 0.07 1
Surface Color	<input type="color"/> 1 1 1 1
diffuse	1
specular	1
specular power	12
Bumpiness	1
shine	0.2890625
colorTexture	{Hidden,Name,Uri,Image,W
normalTexture	{Hidden,Name,Uri,Image,W



# シェーダパフォーマンスのシミュレーション



# プロダクションパイプラインの統合

- シーンをバインドする演算式
- スクリプト記述による自動化
- プラグインベースのアーキテクチャ
- ソースコントロールの統合
- プロダクションパイプラインのサンプル

# シーンをバインドする演算式

セマンティック



```
float4 LightPos : Position
```

```
<
```

```
string Object = "PointLight";  
string Space = "Object";
```

← アノテーション

```
> = {-10.0f, 10.0f, -10.0f, 0.0f};
```

# カスタムセマンティックスおよびアノテーション



- シーンおよびシステムデータへのフックパラメータ
- xml設定ファイルによる式
- 演算子の豊富なライブラリ
  - ドットおよび外積、mux、demux、matrix ops ...
- カスタム演算子
  - 演算子を使用してXMLから構築
  - プラグイン経由

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

# スクリプト記述による自動化

- 自動割り当て
  - マテリアルをジオメトリに
  - シェーダパラメータをシーンオブジェクトに  
(直近のライト、カメラ、など)
  - モデルのアクセサリをアタッチポイントに
  
- 共通タスクツールバー (Maya/MELライク)

# スクリプト記述による自動化



```
#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:
    ...
```



# デモ: スクリプト記述

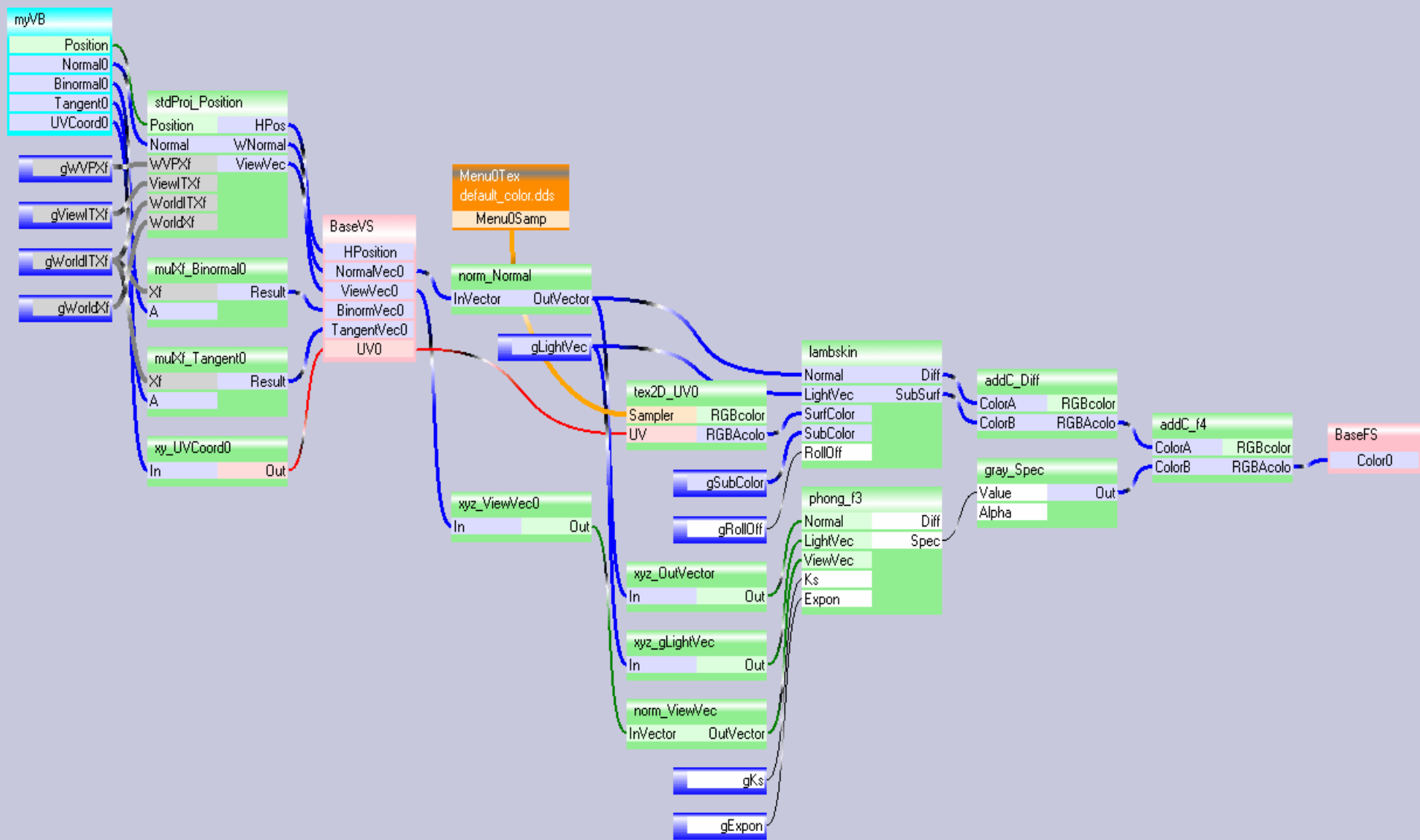
- 自動割り当て
- その他

# カスタムプラグイン

- インポータ
- エクスポータ
- セマンティック式の演算子
- レンダリングデバイス
- 手順によるジオメトリの生成(ヒレ、ヘアなど)
- カスタムオーサリング環境

...その他...

# カスタムプラグイン: 無制限に拡張可能

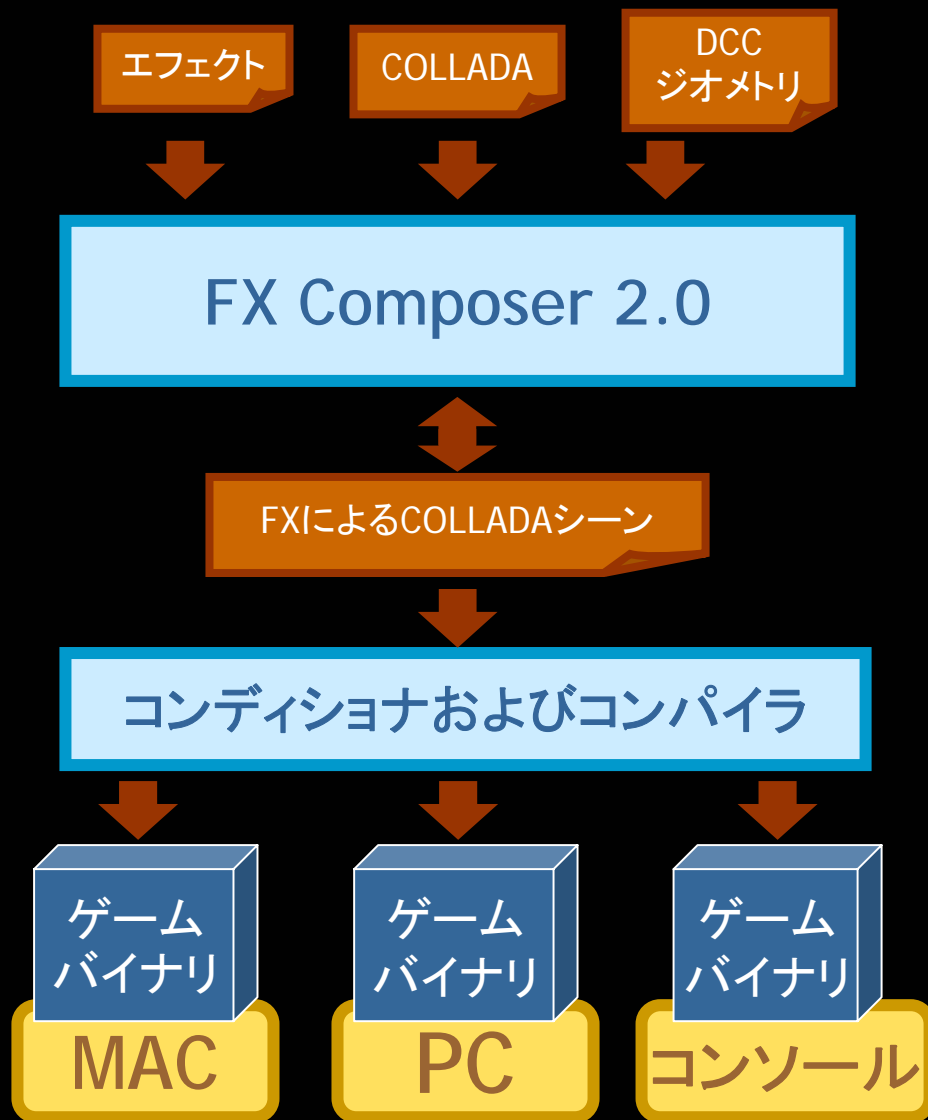


# ソースコントロールの統合

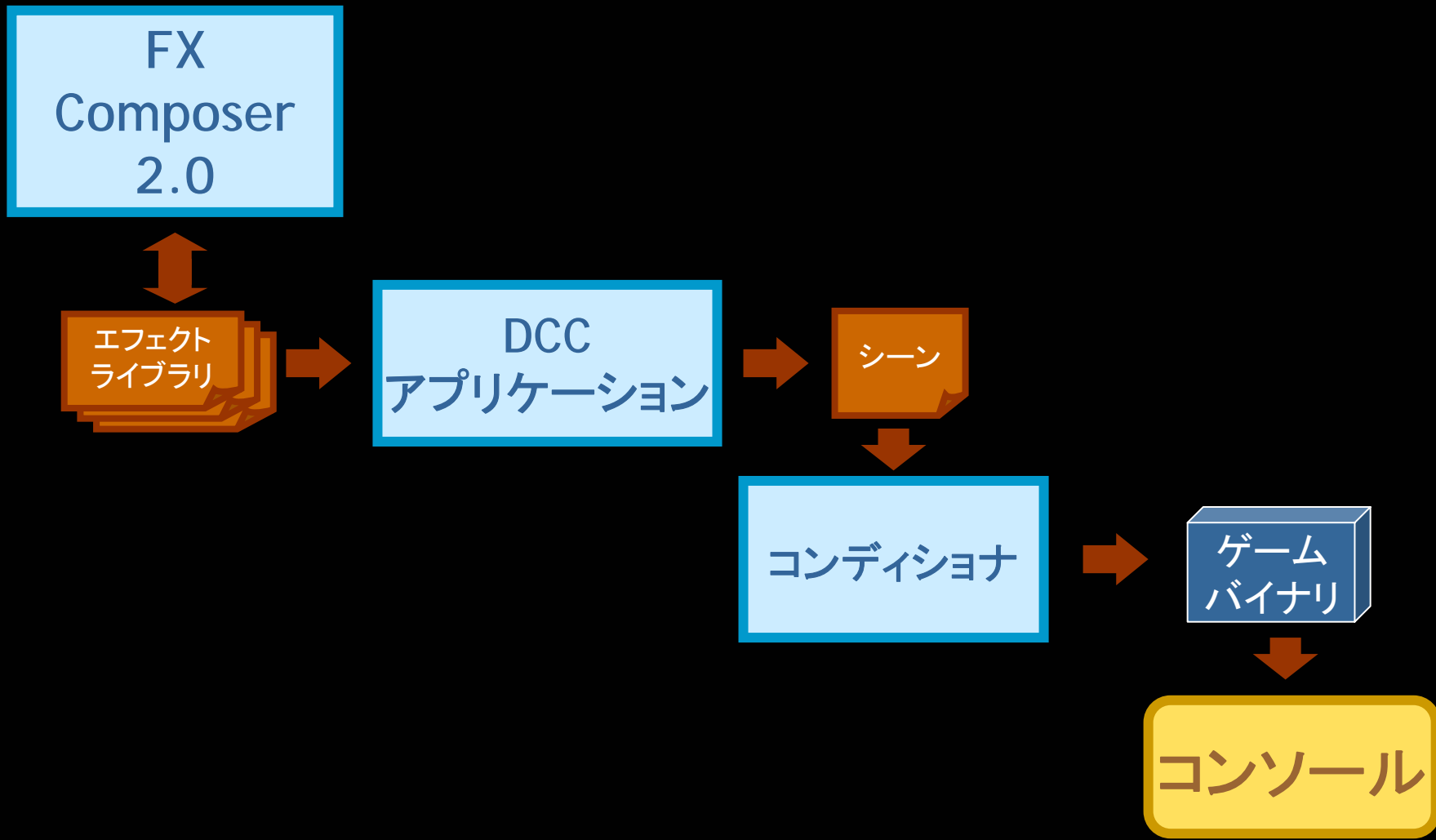
- ソースコントロールソフトウェアへのシームレスな統合
- ドキュメントおよびアセットはファイルベースの状態を反映



# FXC2を中心としたパイプライン



# DCCを中心としたパイプライン



# パイプライン: ハンドヘルド

ホストPC

ターゲットハンドヘルド

FX Composer 2.0 ES

リモート  
レンダリング

DCCツール  
XSI、Max Maya

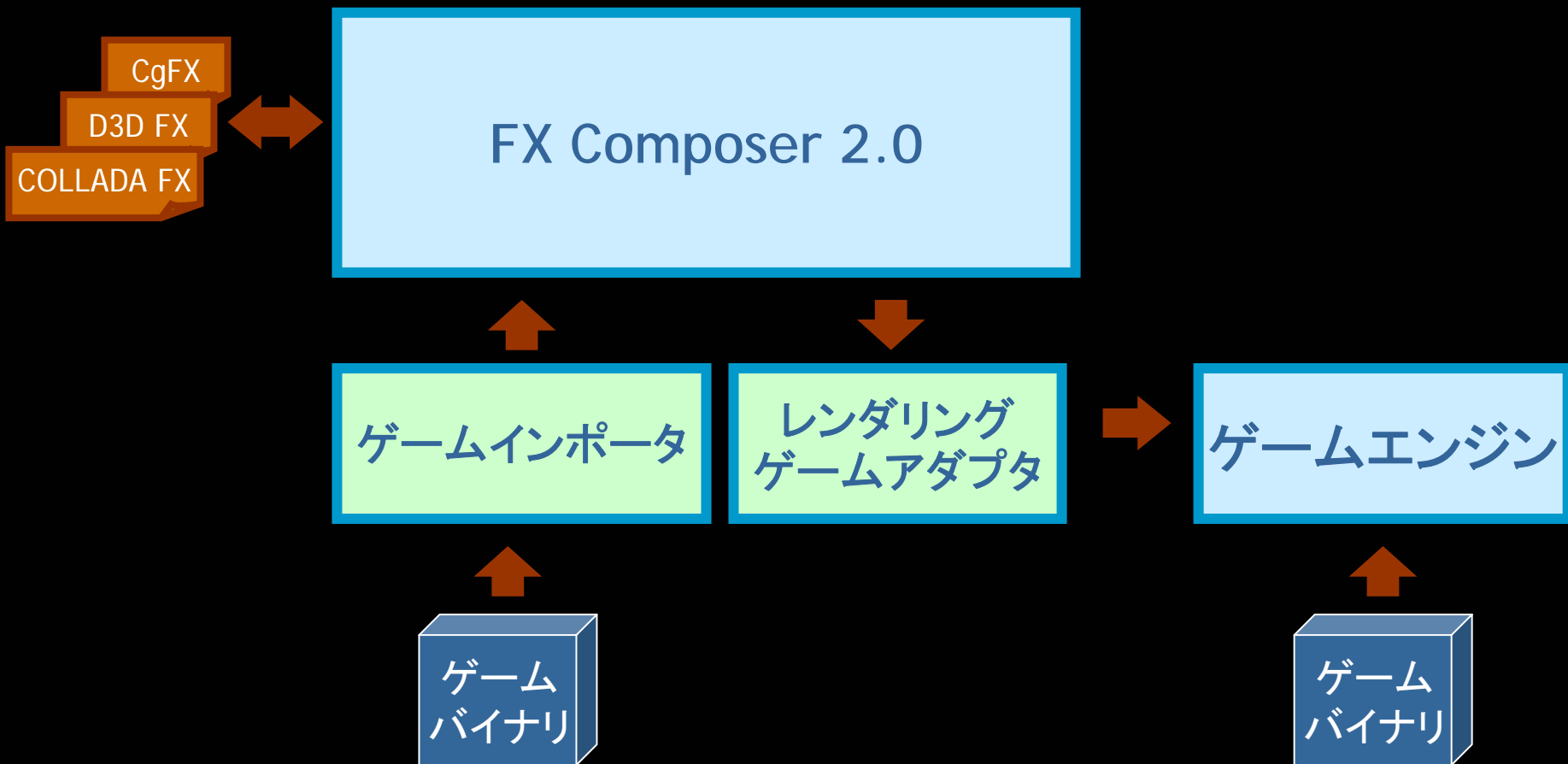
COLLADA

データコンディショナおよび  
コンパイラ

ゲーム  
エンジン



# パイプライン: エンジン



# FX Composer 2.0アルファ5

- アルファ5、2006年の夏の終り頃リリース予定
  - ドキュメントおよびアセット管理
  - COLLADA FXオーサリング
  - シェーダパラメータシーンバインディング
  - カスタムセマンティックおよびアノテーションのサポート
  - Pythonスクリプト
  - シェーダパフォーマンス
  - Sony PS3の開発者および一部のパートナーが入手可能
  
- ベータ版、2006年の秋の終り頃リリース予定
  - 一般に公開

# 結論



- 次世代シェーダIDE開発中
- 強力な機能を搭載して量産に対応
- NVIDIAは、Khronosおよび他の企業と協力して、プロフェッショナルグレードのオーサリングツールを実現

# Q&A



- アルファ版およびベータ版の早期リリースについては、  
電子メールでお問い合わせください。

[fxcomposer@nvidia.com](mailto:fxcomposer@nvidia.com)

- お疲れ様でした  
フィリップ・ロリン (Philippe Rollin、[prollin@nvidia.com](mailto:prollin@nvidia.com))

# The Source for GPU Programming

[developer.nvidia.com](http://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.

**NVIDIA**

[developer.nvidia.com](http://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.