
COLLADA
TM

祝１周年記念

SIGGRAPH 2004 で COLLADA を
発表してから１年経過

- 発表後
  - 4つの新しいワーキンググループがスタート
  - 2度のミーティング
  - 参加者の数が増加

COLLADA™

COLLADAプロジェクト発足のきっかけ

- コンテンツ制作の複雑化
  - ３Dデバイスで可能な表現の拡大化
  - コンテンツサイズの肥大化
  - 一方、費やせる制作期間は短縮されている
- コンテンツ制作フローのカスタマイズにはお金と時間が必要
- ツールやプラットフォーム、制作フローに依存しないファイルフォーマットが今後の制作過程において必要

COLLADA™

COLLADAプロジェクト発足のきっかけ

- アプリケーション間でデータを受け渡す標準交換フォーマットが存在しない
  - ソースデータはDCCの独自フォーマット内にある
  - こういったフォーマットを定義する活動が無い
  - コンテンツ制作者からの需要は高い
  - この分野がコンテンツ制作のネックになっている
- 制作チームごとに独自のフォーマットやエクスポータを作るのはとてももったいない
  - 1度標準的な良いものを作ってしまえば、ここに費やしている時間や努力そしてお金を大きく節約可能
- 共通な標準交換フォーマットがあれば、ツールを間でデータを共有して制作することも可能

COLLADA™

COLLADA とは?

- 配布フォーマットではありません
  - 3D アセットの交換フォーマット
  - アセットのバージョンの差異による情報の差分も全て保持
- 不可逆ファイルフォーマットではありません
  - ユーザが記述したそのままの情報を DCC ツール間で受け渡すことが可能
- シーングラフではありません
  - シーングラフ以上の表現を記述することが可能

COLLADA™

COLLADA とは?

- 独自フォーマットではありません
  - ドキュメントのしっかり整った、オープンな標準規格
  - UTF8、XML、URL、Base64、XML スキーマなどの標準ツールを使用して構築
- オープンプロジェクト
  - 多くの業界関係者を巻き込んで設計されたオープンソースプロジェクト
- 3D アセットに柔軟性を提供
  - メタ情報を持ったオープンな保存フォーマット
  - DCC ツールがアップグレードしても、それまでに作ったアセットを使い続けることが可能

COLLADA™

アプリとアプリをつなぎます

- アプリケーション間の交換フォーマットになるように設計
  - DCC ツールと ３次元処理を行うツールをつなげてコンテンツ制作フローの形成を手助け
  - アプリケーションが同じ言語で互いに話ができる手段を提供
  - 開発者とツールメーカーがこれまで作って来たものを失うことなく、それらをベースにして進化することを可能にする

COLLADA™

COLLADA ドキュメントの構造

- 各COLLADAドキュメントはXMLで表現され、以下のような構成になっています

ヘッダ

ライブラリ群

シーン群

```
<?xml version="1.0" encoding="UTF-8"?>
<COLLADA>
  <asset> ... </asset>

  <library type="GEOMETRY">
     ...
  </library>

  <scene id="DefaultScene">
    <node id="node" name="My Teapot">
      <instance url="#teapot" />
      <translate>0 0 0</translate>
      <rotate>0 0 1 0</rotate>
      <scale>1 1 1</scale>
    </node>
  </scene>

</COLLADA>
```

COLLADA ドキュメント

- COLLADA ドキュメントは以下である可能性があることから、「ファイル」ではなく「ドキュメント」と呼ばれます
  - ファイルシステム上の単なるファイル以上のもの
  - SQL データベースクエリーのフォーマット済みの結果
  - リモートサーバに送られた HTTP リクエストから生じるもの
  - DCC ツールからの一時的な出力フラグメント
  - 大規模なソースコントロールシステム内の特定の条件に当てはまる小さなサブセット

- 構造化されたデータをXMLで表現するため、3Dツールは全ての最新のデータベースツールを使用することが可能

ライブラリ

- 数多くあるオブジェクト型を総称してライブラリと呼びます

```
<animation>  <image>
<camera>     <light>
<code>       <material>
<controller> <program>
<geometry>   <texture>
```

- COLLADA は新しい型をライブラリに追加することで、拡張可能です

アセット

- すべてのCOLLADAドキュメントと内部オブジェクトは<asset>と表示され、メタ情報を使用して管理することが可能

```
<?xml version="1.0" encoding="UTF-8"?>
<COLLADA>
  <library type="GEOMETRY">
    <mesh id="MyTeapot">
      <asset>
        <author>Mark Barnes</author>
        <revision>1.04</revision>
        <authoring_tool>Maya Collada exporter v0.7.4</authoring_tool>
        <modified>2004-12-20T22:10:18Z</modified>
        <up_axis>Y_UP</up_axis>
      </asset>
      ...
    </mesh>
  </library>
</COLLADA>
```

モデルとジオメトリ

- ジオメトリは、インデックス配列で表現される頂点ごとに複数のデータストリームを持つことが可能

```
<vertices id="revolve-Vtx">
  <input semantic="POSITION" source="#revolve-Pos"/>
</vertices>

<polygons count="720" material="#Default">
  <input semantic="VERTEX" source="#revolve-Vtx" idx="0"/>
  <input semantic="NORMAL" source="#revolve-0-Normal" idx="1"/>
  <input semantic="TEXCOORD" source="#revolve-0-TexCoord0" idx="2"/>
  <input semantic="TEXCOORD" source="#revolve-0-TexCoord1" idx="3"/>
  <input semantic="TEXCOORD" source="#revolve-0-TexCoord2" idx="4"/>
  <input semantic="TANGENT" source="#revolve-0-Tangent" idx="5"/>
  <p>0 0 0 0 0 0 31 1 1 1 1 1 32 2 2 2 2 2 1 3 3 3 3 3</p>
  <p>1 4 4 4 4 4 32 5 5 5 5 5 33 6 6 6 6 6 2 7 7 7 7 7</p>
  <p>2 8 8 8 8 8 33 9 9 9 9 9 34 10 10 10 10 10 3 11 11 11 11 11</p>
  <p>3 12 12 12 12 12 34 13 13 13 13 13 35 14 14 14 14 14 4 15 15 15 15 15</p>
  ...
</polygons>
```

シーンの記述

- **<scene>** で、同一空間におけるオブジェクトの階層構造やシーングラフを記述

```
<scene id="DefaultScene">
  <node id="Camera">
    <instance url="#camera"/>
    <lookat> ... </lookat>
  </node>
  <node id="Light" name="Light">
    <instance url="#Lt-Light"/>
    <translate>-5.0 10.0 4.0</translate>
    <rotate>0 0 1 0</rotate>
      <node id="Teapot" name="Teapot">
        <instance url="#teapot-lod-02"/>
        <rotate>0 0 1 0</rotate>
      </node>
  </node>
</scene>
```

アニメーションとスキニング

- COLLADA 1.3 での新機能はアニメーションとスキニング
  - Siggraph 2004 後に仕様のリリース

- アニメーションはカーブまたはキーフレームで制御可能
  - `<animation>` には値の`<source>` 配列を少なくても1 つ含む
  - `<sampler>`オブジェクトではこれらの`<source>`を取り込んで、インデックス化または補間して出力
  - 最終的に、`<channel>`を使って出力結果をそのシーンにおける値とします

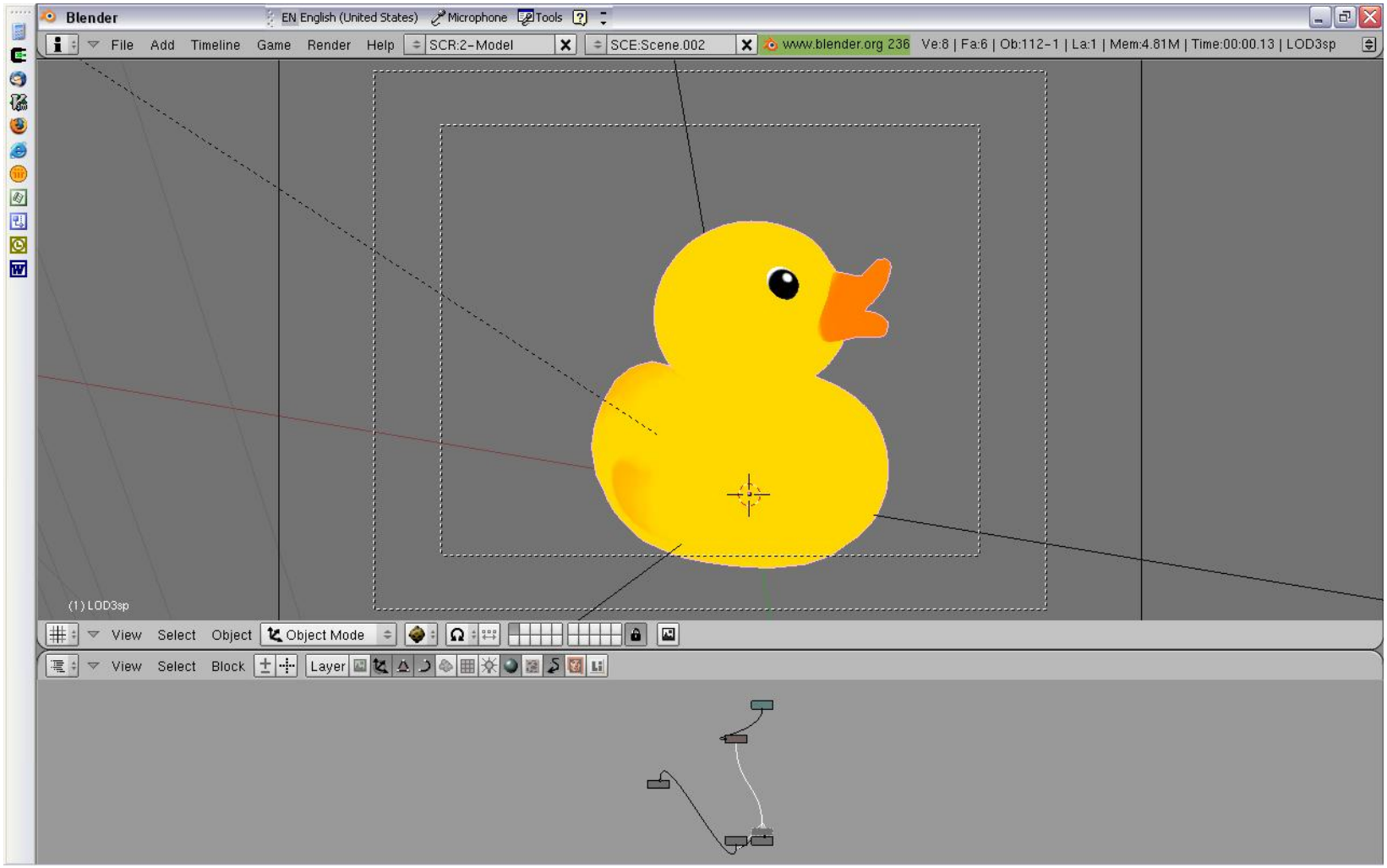
    例：「`elbow/rotation.ANGLE`」

- スキニングアニメーションのサポート
  - １つのスケルトンで多数のメッシュをアニメーション
  - メッシュは重みつき因数の配列と逆ボーン変換マトリクスによってデフォルトのポーズにバインドされる

Lightwave™

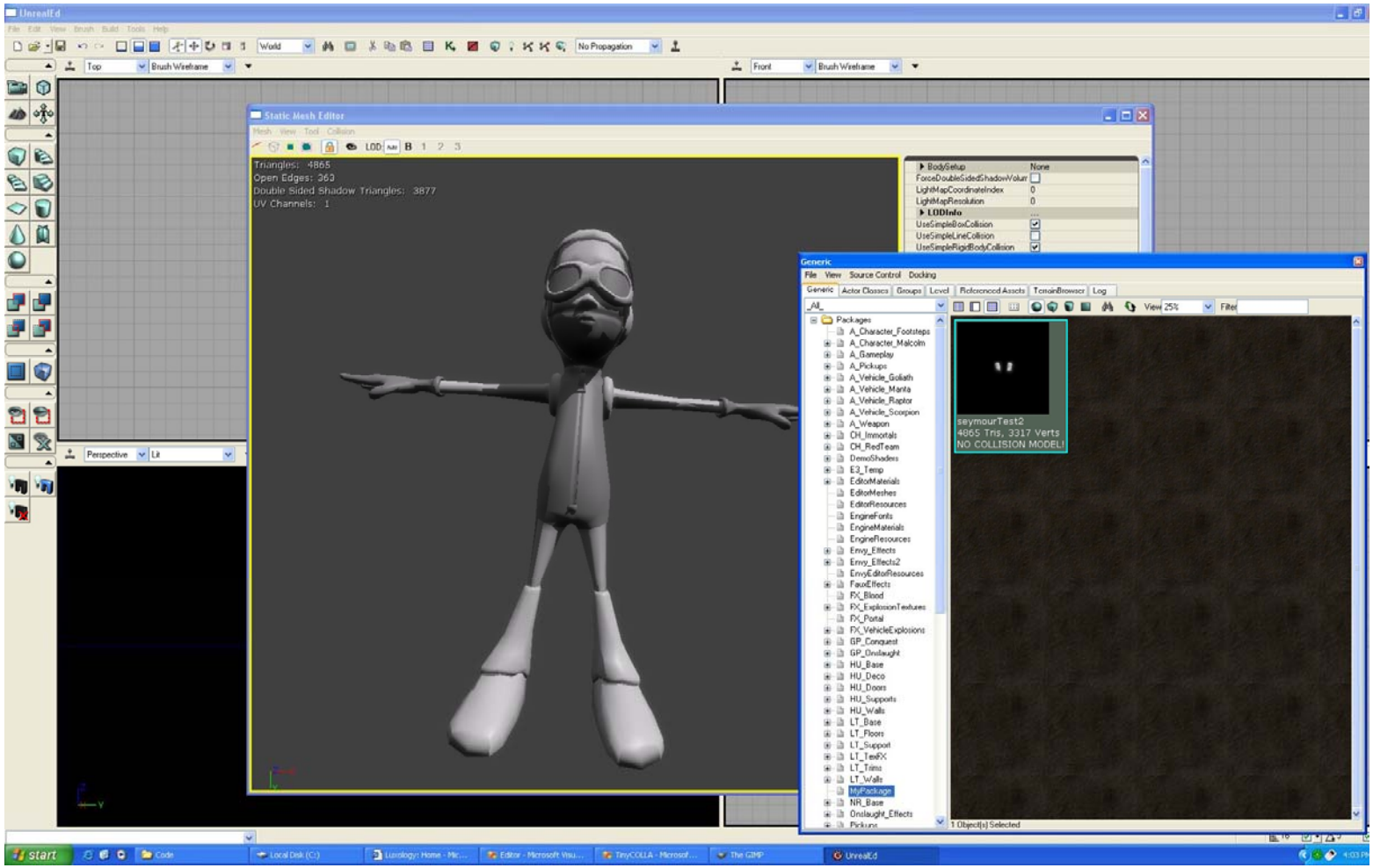
COLLADA™

Blender™



COLLADA™

Unreal Engine™



COLLADA™

COLLADA 1.4 での新機能

COLLADA FX

COLLADA フィジックス

テストと標準化

コンテンツ制作における
ワークフロー管理

COLLADA™

シェーダエフェクトとは?

- シェーダは GPU 上で動作し、視覚効果を生みだすプログラム

- シェーダエフェクトは加えて、シェーダが動作する環境を記述可能

  - レンダリングステート設定
    - （例：カリング、ブレンディング、デプステスト）
  - シェーダが使われるパイプラインステージ
  - 自動設定されるパラメータ
  - ユーザが微調整できるパラメータ
  - エフェクトを終了するのに必要なパスの数
  - パスで使用するテクスチャ
  - など

シェーダエフェクトの問題

- 各プラットフォームには、どれもマルチパスのシェーダエフェクトの表示用に独自のファイルフォーマットがあります。

Cg	*CgFX*エフェクトフォーマット
HLSL	*.FX*エフェクトフォーマット
GLSL	*現在のところ、エフェクトフォーマットなし*
OpenGL|ES	*標準エフェクトフォーマットなし*

- マルチプラットフォーム対応のゲームを設計する場合、管理がとても大変

COLLADA™

COLLADA FX

- COLLADA FX によって以下が可能になります
  - 複雑なマルチパスのシェーダエフェクトの交換
  - 複数エフェクトの記述をカプセル化
    - 例： LOD、昼間/夜間バージョン
  - 1 つのファイル内に複数のプラットフォームに対応したシェーダを記述
    - 例： 1 つのドキュメントでのCG 、HLSL、GLSL バージョンのエフェクト
  - 同一フォーマットでの固定パイプラインとプログラマブルパイプラインの両方のエフェクトをサポート

COLLADA™

COLLADA FX の動作

- 単一セットのレンダーステートや中間シェーダ言語を作成したりはしない
  - プラットフォーム固有のエレメントを用意することによって、そのプラットフォームでサポートされる全てのデータ型やレンダーステートにアクセス可能
  - ステートを操作するランタイム API の使用が前提
  - ランタイムでのスクリプト処理は必須ではありません

```
<effect id="MossyRockEffect">

  <technique sid="HLSL_medium_LOD">
    <profile_HLSL>
      ...
    </profile_HLSL>
  </technique>

  <technique sid="CG_with_lighting">
    <profile_CG>
      ...
    </profile_CG>
  </technique>

</effect>
```

テクニックとパス

- COLLADA FX ファイルはテクニックとパスで構成されます

```
<effect id="Level3-spaceship-damaged ">

  <technique sid="high-LOD-shadows">
    <profile_HLSL>
       <code> ... </code>

       <pass id="shadow-pass">
         ...
       </pass>

       <pass id="specular-reflection">
         ...
       </pass>

    </profile_HLSL>
  </technique>

</effect>
```

パス

- 各パスはレンダーステートを設定し、プラットフォームがサポートできる限りのプログラマブルパイプラインのステージを宣言
- ユニフォーム型の入力は、変数または定数として表される
- プログラマブルシェーダのないプロファイルはレンダーステートの設定のみが行える
- レンダーステートは定数によって、あるいはパラメータを読むことによって設定

```
<pass sid="pass0">

  <shader stage="VERTEXSHADER">
    <compilertarget>vs_2_x</compilertarget>
    <entry>vertex_function</entry>
    <bind symbol="damage">
      <param ref="damage_amount"/>
    </bind>
  </shader>

  <depthtestenable> TRUE </depthtestenable>
  <depthfunc> LEQUAL </depthfunc>
  <cullface> BACK </cullface>

  <shader stage="PIXELSHADER">
    <compilertarget>ps_2_x</compilertarget>
    <entry>pixel_function</entry>
  </shader>

</pass>
```

COLLADA™

パラメータ

```
<library type="EFFECT">
  <effect id="ThinFilm2">
    <technique>
    <profile_HLSL>

    <include url="#thinfilm_code"/>

    <setparam ref="LightPosP1">
      <annotate name="UIName">
         <string>Light Pos 1</string>
      </annotate>
      <annotate name="Object">
         <string>PointLight</string>
      </annotate>
      <annotate name="Space">
         <string>World</string>
      </annotate>
      <float4> 10.0 10.0 10.0 1.0 </float4>
    </setparam>

    ...
```

- ソースコード中で宣言されたパラメータには <setparam> を使用して値を割り当てることが可能
- パラメータには複数のコメントをつけることが可能。
  COLLADA FX はこれらのコメントを解釈しない
- パラメータは<newparam> を使うことで、ソースコード内で定義されたデータ型から生成可能

テクスチャへのレンダリング

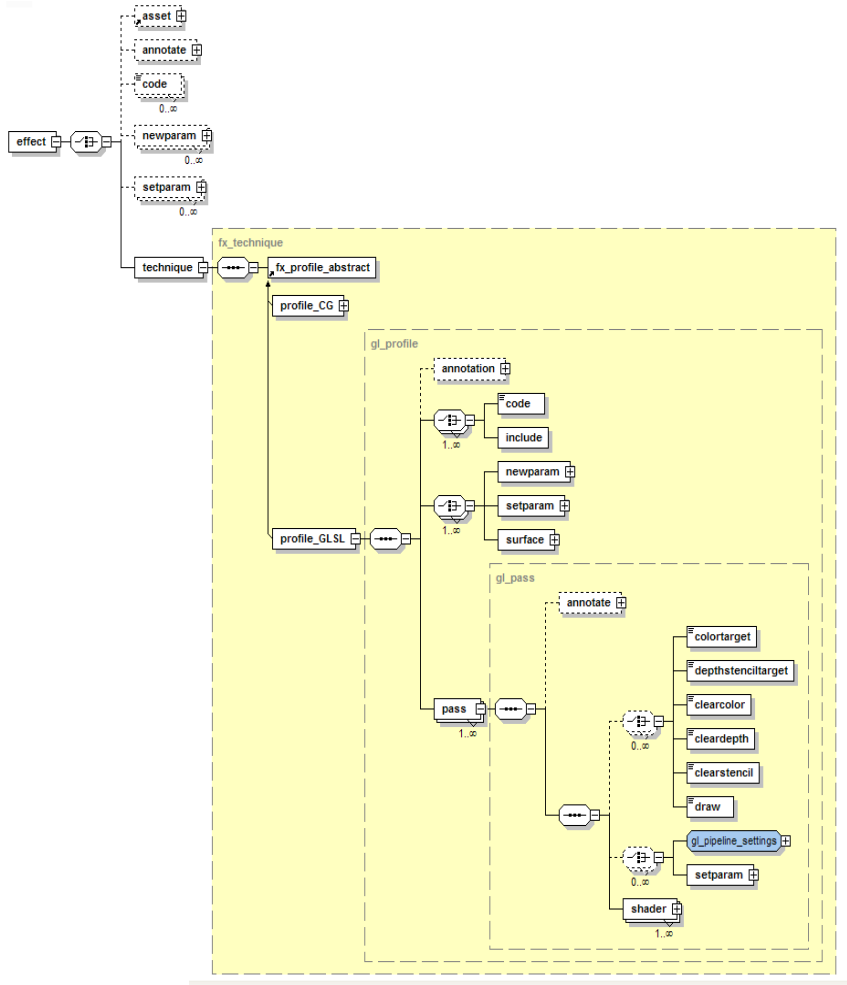
- IDが割り振られた<surface>オブジェクトを使うことで複数のレンダーターゲットが指定でき、ターゲットの指定はパス間で共有することが可能

```
<pass sid="pass0">

   <colortarget index="0"> color_surface </colortarget>
   <colortarget index="1"> normal_surface </colortarget>
   <depthstenciltarget> depth_surface </depthstenciltarget>

   <clearcolor index="0"> 0 0 0 </clearcolor>
   <clearcolor index="1"> 0 0 0 </clearcolor>
   <cleardepth> 1.0 </cleardepth>
   <clearstencil> 0 </clearstencil>

   <draw>SCENE</draw>

   ...
</pass>
```

COLLADA FX スキーマ

- スキーマは、エフェクトの宣言や、実装を明確にすることで、柔軟性と記述性に重きをおいて設計されています

- 各シェーダランタイムには、XML ドキュメントとランタイム APIが明確に1対1にマッピングがされています



物理シミュレーションファイルフォーマット

- 物理特性や物理シミュレーション用の標準フォーマットが存在しない
  - 各 API がそれぞれ独自のファイルフォーマットを持っている
  - 各 API も各 DCC プラットフォームに独自のエクスポータやプラグインを提供しなければならない

- 異なる物理エンジン間で同じ物理設定を交換する標準的な方法が何もない

- 多くの場合、直接プログラムを書く以外に物理シミュレーションを実装するのは不可能に近い

COLLADA フィジックス

- COLLADA フィジックスが提供するもの
  - 基本的な剛体とジョイントモデルに対して物理エンジンAPI非依存な切り口
    - 重心、内部速度、慣性モーメント、摩擦係数、減衰、重力係数、他
  - 剛体に対するバウンディングボールやバウンディングボックスを使用した衝突処理または凸包近似による衝突処理
  - 複合体に対する関節の可動域と自由度
  - 静的あるいは動的ジオメトリによるシーンの分割

COLLADA フィジックスの動作

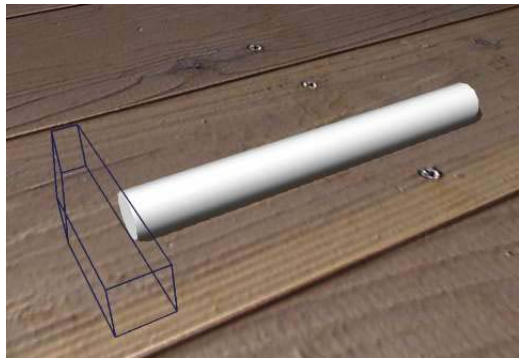
- シェーダにおけるmaterialと同様に<physics_material>を用いることでオブジェクトの物理属性を指定することが可能

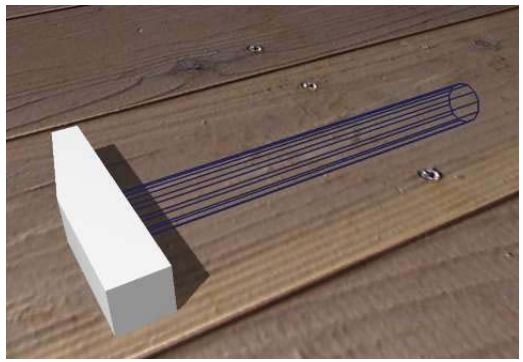
```
<material id="Wood">
   <effect ref="MyWoodShader"/>

   <physics_material id="WoodPhysMtl">
      <technique profile="COMMON">
         <static_friction>
            <float semantic="X"> 0.23 </float>
         </static_friction>
         <dynamic_friction>
            <float semantic="X"> 0.12 </float>
         </dynamic_friction>
         <density> <float> 0.05 </float> </density>
         <elasticity> <float> 0.05 </float> </elasticity>
      </technique>
   </physics_material>

</material>
```

COLLADA フィジックスの動作

- 剛体は分析的形状と物理的材質または凸包と物理的材質の組み合わせで作成

```
<library type="PHYSICS">

  <rigid_body id="HammerRigidBody">
    <mass> 1.25 </mass>
    <shape>
      <mass> 0.25 </mass>
      <geometry url="#hammerHandleForPhysics"/>
      <physics_material url="#WoodPhysMtl"/>
    </shape>

    <shape>
      <mass> 1.0 </mass>
      <translate> 0.0 8.0 0.0 </translate>
      <geometry url="#hammerHeadForPhysics"/>
      <physics_material url="#SteelPhysMtl"/>
    </shape>
  </rigid_body>

</library>
```

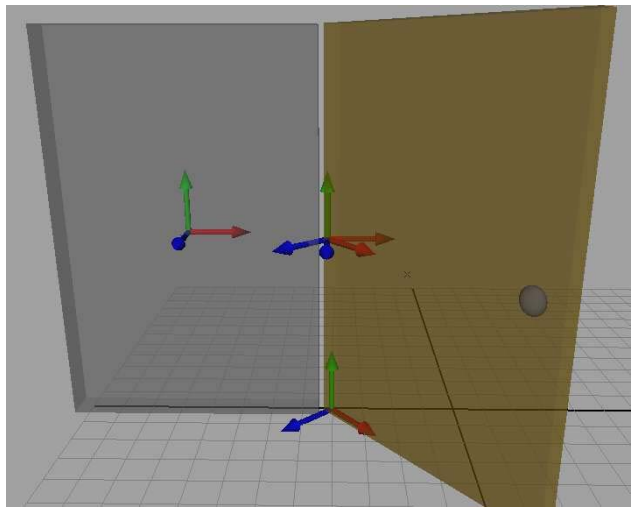




COLLADA フィジックスの動作

- 複合体では、接合部分に可動域の制約が適用されます

```
<library type="PHYSICS">

  <rigid_body id="doorRigidBody"/>
  <rigid_body id="wallRigidBody"/>

  <rigid_constraint id="rigidHingeConstraint">
    <attachment body="#wallRigidBody">
      <translate sid="translate">5 0 0</translate>
    </attachment>
    <attachment body="#doorRigidBody">
      <translate sid="translate">0 8 0</translate>
      <rotate sid="rotateX">0 1 0 -45.0</rotate>
    </attachment>
    <rot_limit_min>0 -90 0 </rot_limit_min>
    <rot_limit_max>0 +90 0</rot_limit_max>
  </rigid_constraint>

</library>
```



テストと標準化

- DCC プラグインを信頼性を高めるためには、様々なケースをテストし、それぞれが正しい挙動をすることを検証する必要がある
  - 例えば、COLLADA エクスポータはインポートとエクスポートの規則に従う必要がある
    - インポートされたものは、たとえツールで認識されていなくてもエクスポートできる必要がある
    - 未認識のデータは変換されたり廃棄されたりしてはいけない
- ユーザと開発者が協力して、信頼性のある高品質なエクスポータとインポータをすべての DCC アプリケーション向けに作成することが望まれている
- この目標に到達するため
  - 自由に利用可能なテストセットを作成
  - 主要なエクスポーターすべての適合テストの結果を発表

COLLADA™

COLLADA 適合テストセット

- 目下 500 を超える機能テストは今も増え続けています

Maya テスト：200	*XSI* テスト：150
3DS Max テスト：150	*XML 構文解析*テスト：30

- COLLADA 開発者は全員、COLLADA、ネイティブソースファイル、および参照イメージが自由に利用可能

- テスト結果が公表されるので比較することが可能
  - http://www.collada.org/ にアクセスし、現在のテスト結果をご覧ください
  - COLLADA では、ウェブベースのアプリケーションを通した自動テストシステムへのアクセスを間もなく可能

適合テスト結果

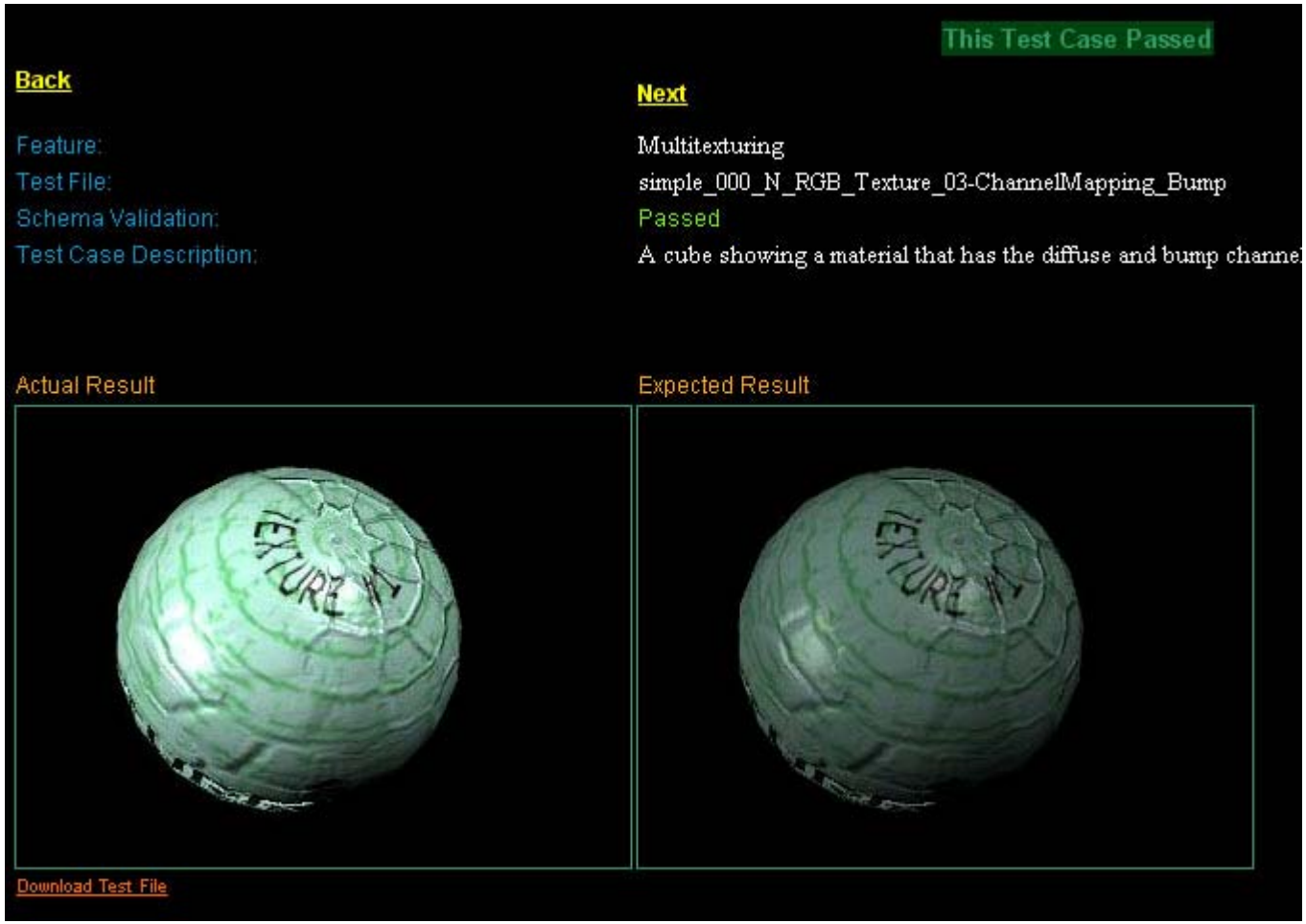
Collada Conformance Test Results

Download All Test Files

Feature	Priority		P/F			Comments	Test File	Description
EX: Export IM: Import Sch: Schema								
1) Export Options	**Ex**	**Im**	**Ex**	**Im**	**Sch**			
Export As Triangle List	Must	Must	P	P	P		simple_000_TriangleList_00	Simple, triangulate
Export As Poly List	Should	Should	P	P	P		simple_001_PolyList_00	5 sided polygon. N
Export with Bakend Matrices	Should	May	P	P	P		simple_002_BakeMatrices_00	A colletion of group
Export with Baked Lighting	May	NA	?	?	P	Test case didn't get run properly	simple_009_PerVertexLighting_00	A white sphere on
Export which allows the user to choose a sampling rate	May	May	?	?	P	Result Uncertain	simple_003_VariableSamplingRate_00	Sphere animated
Sampled Animation	Must	NA	?	?	P		simple_001_SampledAnimation_00	A sphere with an e
2) Transforms								
Translation	Must	Must	P	P	P		simple_000_Translation_00	A collection of obje
Scale	Must	Must	P	P	P		simple_001_Scaling_00	A collection of sph
Rotation	Must	Must	P	P	P	Rotation order problem has been fixed	simple_002_Rotation_00	The same object c
Parenting	Must	Must	P	P	P		simple_003_Parenting_00	Two groups of obj
Instancing	Must	Must	P	P	P		simple_004_Instancing_00	One object, instan
Pivot Point Translation	Must	Must	?	?	P	Object lacks a texture, but otherwise it's OK	simple_006_Pivot_00	A cube with pivot p
Pivot Point Translation	Must	Must	P	P	P		simple_006_Pivot_01	A shape with pivot
Pivot Point Translation	Must	Must	P	P	P		simple_006_Pivot_02	A collection of sha
3) Materials								
Multitexturing	Should	Should	P	P	P		simple_000_N_RGB_Texture_02-ChannelMapping_Ambient	A cube with a mate
Multitexturing	Should	Should	P	P	P		simple_000_N_RGB_Texture_03-ChannelMapping_Bump	A cube showing a
Multitexturing	Should	Should	P	P	P		simple_000_N_RGB_Texture_04-ChannelMapping_Incandescence	A cube showing a
Multitexturing	Should	Should	P	P	P		simple_000_N_RGB_Texture_05-ChannelMapping_Specularity	A cube showing a
Material Parameter, Transparency	Must	Must	P	P	P	The lighting on reimport appears strange	simple_003_TransparencyAsMaterialParam_00	A cube with a mate
Material Parameters, Common Profile	Must	Must	P	P	P		simple_006_Common_Profile_Material_00	A collection of sph
Per Polygon Material	Must	Must	P	P	P		simple_007_Per_Polygon_Materal_00	A grid of 9 polygon
Material Parameter, Color	Must	Must	P	P	P		simple_008_MaterialAttribute-Color_00	An array of sphere
Texture Format, BMP	Must	Must	P	P	P		simple_010_TextureFormats-BMP_00	A single polygon p
Texture Format, JPG	Must	Must	P	P	P		simple_010_TextureFormats-JPG_01	A single polygon p
Texture Format, PNG	Must	Must	P	P	P		simple_010_TextureFormats-PNG_02	A single polygon p
Texture Format, TGA	Must	Must	P	P	P		simple_010_TextureFormats-TGA_03	A single polygon p
4) VertexAttributes								
Texture Coordinates	Must	Must	F	F	P	The 2nd and 3rd UV sets are lost	simple_000_N_TexCoord_sets_00	A cube with a mate
Per Vertex Normals	Must	Must	P	P	P		simple_001_Normals_00	A sphere with verte
Per Vertex Colors	Must	Must	?	?	?	Crashed On Export, no DAE file made	simple_002_VertexColors_00	A cube with differe
Texture Coordinates	Must	Must	P	P	P		simple_005_Bake_Procedural_Texcoords_00	A textured cube. Th

COLLADA™

機能テスト


This Test Case Passed
Back
Next
Feature:
Multitexturing
Test File:
simple_000_N_RGB_Texture_03-ChannelMapping_Bump
Schema Validation:
Passed
Test Case Description:
A cube showing a material that has the diffuse and bump channe
Actual Result
Expected Result
Download Test File

COLLADA と Khronos グループ

- Khronos グループは活動的な標準化ワーキンググループとして COLLADA を認定
  - Khronos は3D の世界を代表する業界コンソーシアムで、 OpenGL ES、OpenML、などの標準APIの策定を行う
  - COLLADA は Khronos に移り、協力者、パートナー、および参加者という更に強いコンソーシアムの下で開発は継続
  - ソニー・コンピュータエンタテインメントは、プロモータとして Khronos に加入し、より活発に標準規格の開発に参加

COLLADA™

Khronos グループ

COLLADA™

コンテンツ制作におけるワークフロー

- 3D アセットは、ゲームデータとして使用されるようになる前に、多くの処理が施されるケースが多い

  - 三角形化
  - ストリップ化
  - グローバルイルミネーションと PRT
  - 重心
  - BSP ツリー生成
  - 凸包
  - 慣性モーメント
  - LOD 生成
  - ジオメトリ圧縮
  - ゲームバイナリへのエクスポート

- DCC ツールのエクスポータにこれらの処理が組み込まれていない場合、ツールを購入するか独自のツールを作る必要があります
- 簡単なコマンドラインツールで 3D アセットを利用することができない

COLLADA™

アセット管理ツール

- COLLADA の目標は、 交換フォーマットとなるようなフレームワークを提供すること
  - ユーザが標準のツールを使用して独自のコンテンツ制作フローを再構成できるようにすること

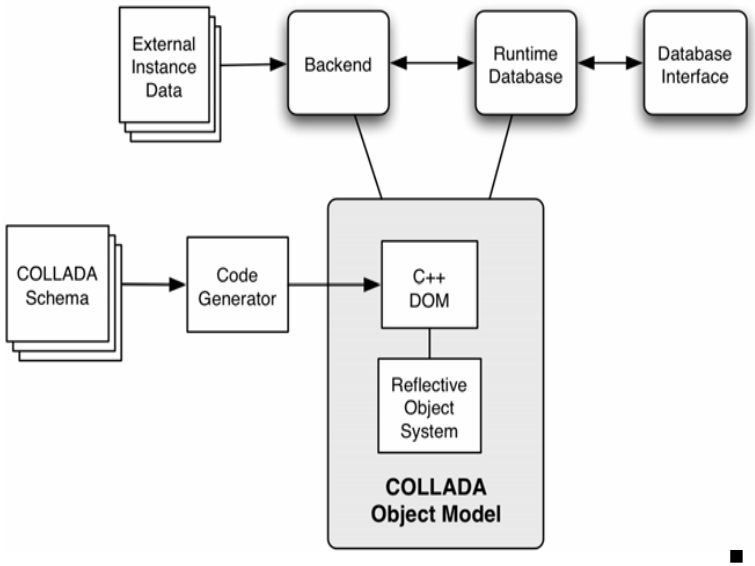
- ツールセットは以下のもので構成される

1. COLLADA DOM
2. アセットタグを使った複数のアセット表現
3. コンディショナー

COLLADA DOM

- COLLADA DOM の導入
  - COLLADA ドキュメントの読み出し、書き込み、および検証に適用
  - C++ で実装され、COLLADA フォーマットの XML スキーマから自動生成
  - コールバックで実装され高速かつ効率的
  - あらゆるプロジェクトにおいて、１から作り直す手間を省く

COLLADA DOM の動作



- リフレクティブなオブジェクトモデルがベース
  - COLLADAのXMLスキーマはオブジェクト型の内部データベースに処理される
  - このデータベースにより、クラスが型、関係、および構造を評価
  - ロード時にCOLLADA XMLドキュメントがスキャンされ、内部にツリー構造のオブジェクトデータベースが構築される
    - ツリーが構築されると、コールバックが引き起こされるので、これを使って独自の表現を内部に構築できます
- 読み出し後、このデータベースは、クエリー、変更、およびエクスポート可能になる

COLLADA DOM の使い方

1. まず、新しいデータベースに XML ドキュメントを開く

```
DAE *input = new DAE;
errval = input->load(“input_url_name.xml”);
```

2. ロード後、中にストリング「mossyRock」が入っている<image>エレメントの数を問い合わせることが可能

```
imageCount = input->getDatabase()->getElementCount
                                    ("mossyRock", "image", NULL);
```

COLLADA DOM の使い方

3. 次に、データベース内のすべてのイメージエレメントに対して
   ループ処理を行う

```
for (unsigned int i=0; i<imageCount; i++)
{
   error = input->getDatabase()->getElement
    ((daeElement**)&mossyRockImage, i, "mossyRock","image",NULL);

   // if successful, process the mossyRockImage
}
```

4. 処理後、変更されたデータベースを新しいドキュメントに書き出
   すことが可能

```
error = input->save(output_url.c_str());
```

COLLADA™

アセットタグによる管理

- アセットタグは、アセットの履歴、意味、および用途の記述に使用されるメタデータ

```
<asset>
  <author>Joe Schmoe</author>
  <keywords>Blink animation level_04</keywords>
  <unit name="kilometer" meter="0.001"/>
  <up_axis>Y_UP</up_axis>
  <revision>1.04</revision>
  <authoring_tool>Maya Collada exporter v0.7.4</authoring_tool>
  <modified>2004-12-20T22:10:18Z</modified>
</asset>
```

  - アセットタグは COLLADA ドキュメントのほとんどのオブジェクトにつけることが可能
  - アセットは複数の表現方法を持つことが可能
  - タグを使うことで、誰がいつアセットを作ったのかをデータベース全般で追跡することが可能

3DアセットのMakefileを目指して

- アセットタグにより、コンテンツ制作フローをより効率化することが可能
  - 親子関係を使用して、編集の影響を受けたアセットのみを再構築したり再エクスポートしたりすることが可能
  - アセットタグを使用すると、コマンドライン、パラメータ、ツール、および設定を記憶することができ、自動生成スクリプトを作成することが可能
  - アセットタグにより、より細かくバージョン管理システムと協調動作させることも可能
    - アセットを何千ものファイルに分割して管理する必要がない
  - アセットマネージメントはまだ研究段階なので、COLLADA のワーキンググループに加わって検討を手伝ってください!

COLLADA™

コンディショナー

- コンディショナーは以下の COLLADA DOM の機能を使用して書かれるプログラム

1. COLLADA ドキュメントのロード
2. アセットのサブセットの処理
3. COLLADA ドキュメントへの出力

- コンディショナーは 3D コンテンツ制作フローの 1 過程であり、以下を行う
  - N面のポリゴンのデータを最適化したトライアングルストリップに変換
  - グローバルイルミネーションモデルの前処理を行い、新しいバージョンのモデルとして結果の書き戻す
  - 複数の LOD、可視性構造、あるいは衝突処理を自動生成し、結果を直接データベースに埋め込む

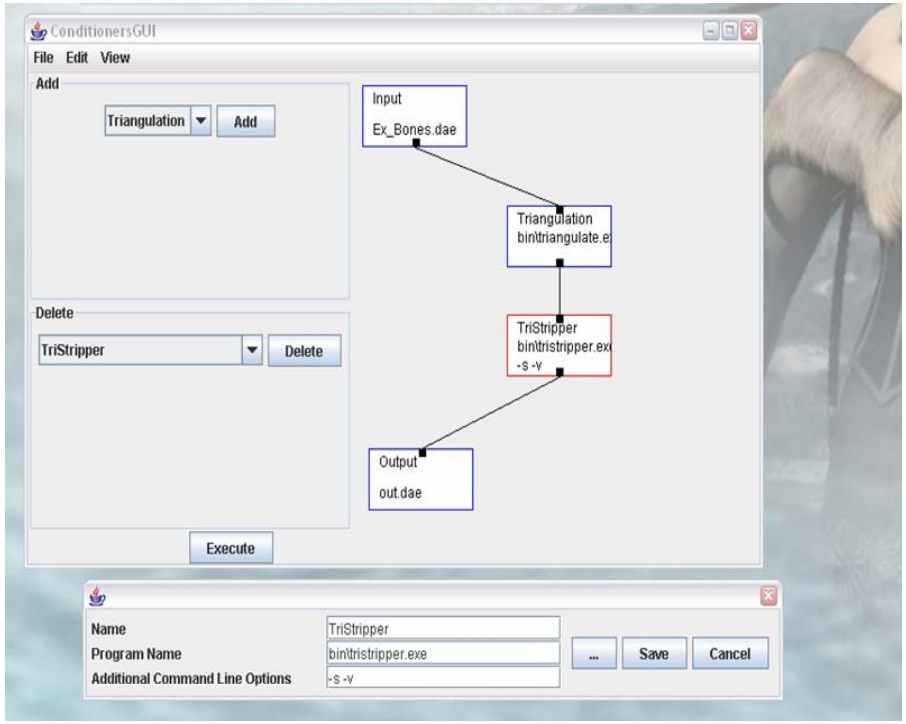
コンディショナー

- COLLADA DOM のサンプルには以下の目的で有用なコンディショナーをいくつか用意してある
  - ファイル名とアセット ID のパッチあて
  - N 面のポリゴンモデルの三角形化
  - 一定のタイムステップでのアニメーションカーブの再サンプリング
  - スキニングしたメッシュの非インデックス化
    - (glDrawElements にインデックスを用意)
  - キャッシュオプティマイザ
    - (三角形を並べ換えて描画を高速化)

- 付加的なコンディショナー案
  - トライアングルストリップ化
  - テクスチャ処理されたモデルへの Tangent と Binormal の生成
  - 連続 LOD 生成
  - その他

コンテンツ制作フローエディタ

- コンディショナーの結合はシェーダ編集のように単純である必要がある
- モジュールのコンポーネントからデータパイプラインを設計すると、実験と最適化が可能
- 再構成可能な制作フローと自動生成されたコンディショナーのビジュアル編集ツールに取り組み中



Javaベースのフロージェネレータ

COLLADA™

2005 年開発ロードマップ

5月6日	COLLADA v1.3.0 スキーマリリース
6月	COLLADA DOM v1.0 ベータテスト
7月29日	COLLADA DOM v1.0 ベータ 8 リリース
8月3日	COLLADA 技術講演（於：SIGGRAPH）
8月19日	COLLADA v1.3.1 スキーマパッチリリース
8月19日	COLLADA DOM v1.0 リリース
8月29日	COLLADA プレゼンテーション（於：CEDEC）
9月1日	COLLADA 業界セミナー（於：Eurographics）
9月 /10月	COLLADA 1.4.0 スキーマリリース

まとめ

COLLADA™ は、対話型エンタテインメント業界のオープンスタンダード

- 大学や研究者の皆様、COLLADA を研究してください!
  - 共同開発の、オープンソースプロジェクトで、MIT ライセンスを使用
  - ツールを作成し、規格を拡張してください
- ドキュメントを読んで、以下に公開しているウェブサイトにアクセスしてください

http://www.collada.org/

- Khronos に加入し、ワーキンググループに参加し、規格の策定にご協力をお願いします

COLLADA™