

Webcast 音声のご利用方法

電話でご利用できます

電話

オーディオ会議(有料) : 03-3298-4797

オーディオ会議(無料) : 0034-800-900364 (注)

参加コード : 119949

(注) オーディオシステムは、AT&Tを使用しており、無料ダイヤルは、NTT以外のキャリア、IP電話、携帯電話などでは使用できません。

電話会議ログイン後は* (米印) 6を押して電話をミュートにしてください。

Autodesk® FBX® SDK

ジオメトリ



はじめる前に

LiveMeeting と音声会議 の使用方法

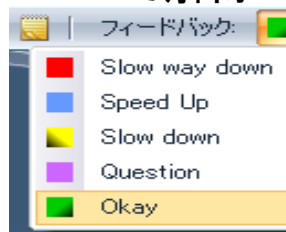
フルスクリーン モード



フルスクリーンに拡張
ESC で解除



フィードバック



リアルタイム Q&A



電話をミュートする *6



本日のプレゼンテーション

録画がポストされます

ADN Extranet

<http://adn.autodesk.com/>



エクスサイズ – ユニット 3

- カメラとライトを保持するシーンを作成
- カスタムプロパティーを追加
- シーンの接続を出力
- ASCII FBX ファイルをしてシーンをエクスポート

FBX SDK Webcast Agenda

- Day / Hour 1 – ようこそFBX SDKへ
- Day / Hour 2 – FBX の基礎 (インポート/エクスポート/シーン)
- Day / Hour 3 – SDKオブジェクトモデル
- **Day / Hour 4 – ジオメトリ**
- Day / Hour 5 – マテリアルとアニメーション



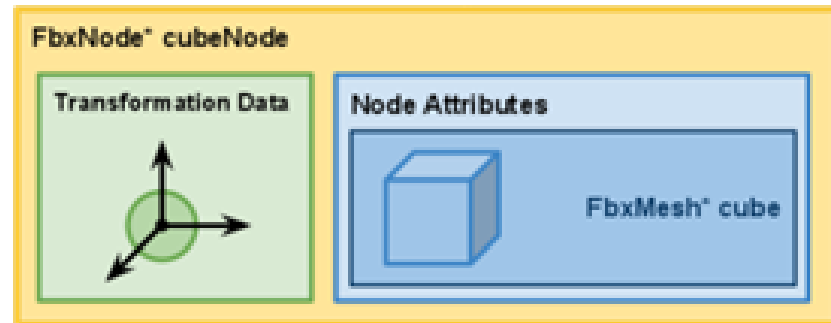
ジオメトリ

- FBXノード
 - トランスフォームデータ
 - ノードの属性
- ジオメトリ構造
- メッシュ
- マテリアル
- テクスチャ

FBX ノード

FbxNode – ユニット 2より

1. トランスフォームを提供
2. 「ノードの属性」への接続を提供
 - 「ノードの属性」は複数ノードで共有可能



1. トランスフォームデータ

トランスフォームデータ



- トランスフォームの提供
 - デフォルトの移動, 回転, スケールベクターはプロパティー
 - LclTranslation
 - LclRotation
 - LclScaling

トランスフォームの提供

//ノードの規定TRSプロパティを取得

```
FbxDouble3 lTranslation = lNode->LclTranslation.Get();  
FbxDouble3 lRotation    = lNode->LclRotation.Get();  
FbxDouble3 lScaling     = lNode->LclScaling.Get();
```



トランスフォームデータ



- トランスフォームを提供
 - グローバルとローカルトランスフォームマトリックス
 - EvaluateGlobalTransform
 - EvaluateLocalTransform



グローバルとローカルトランスフォームマトリックス

// ノードのデフォルトグローバルトランスフォームマトリックスを取得

```
FbxXMatrix& IGlobalTransform = INode->EvaluateGlobalTransform();
```

// ノードのデフォルトローカルトランスフォームマトリックスを取得

```
FbxXMatrix& ILocalTransform = INode->EvaluateLocalTransform();
```

グローバルとローカルトランスフォームマトリックス

- [FbxScene](#)のアニメーションエバリュエーターも使用可能
 - [FbxAnimEvaluator::GetNodeGlobalTransform\(\)](#)
 - [FbxAnimEvaluator::GetNodeLocalTransform\(\)](#)



グローバルとローカルトランスフォームマトリックス

// シーンのアニメーションエバリュエーターを取得

```
FbxAnimEvaluator* IEvaluator = IScene->getEvaluator();
```

// ノードのデフォルトグローバルトランスフォームマトリックスを取得

```
FbxXMatrix& IGlobalTransform = IEvaluator->GetNodeGlobalTransform(INode);
```

// ノードのデフォルトローカルトランスフォームマトリックスを取得

```
FbxXMatrix& ILocalTransform = IEvaluator->GetNodeLocalTransform(INode);
```



グローバルとローカルトランスフォームマトリックス

- アニメーションシーケンス間でFbxTimeを特定し使用
- 注意: ドキュメントではFbxXMatrixですが, 実際は FbxAMatrix

グローバルとローカルトランスフォームマトリックス

```
FbxTime lTime;
```

```
// 2秒で時間を設定
```

```
lTime.SetSecondDouble((float) 2);
```

```
// 2秒でのノードのグローバルトランスフォームマトリックスを取得
```

```
FbxXMatrix& lGlobalTransform = lNode->EvaluateGlobalTransform(lTime);
```



トランスフォームデータ

- シーンの軸システムとシステムユニットは[FbxGlobalSettings](#)オブジェクトで定義
- ノードのトランスフォームデータは親ノードに関連する移動、回転、スケールベクターで表現



ジオメトリートランスフォームプロパティー

- [FbxNodeAttribute](#)は[FbxNode](#)のローカルフレームの参照からのオフセット
 - [FbxNode::GeometricTranslation](#)
 - [FbxNode::GeometricRotation](#)
 - [FbxNode::GeometricScaling](#)
- 3ds Maxのオブジェクト-オフセットトランスフォームと同じ



トランスフォームコンポーネントのベイク

- 一般的に全てのpre/postトランスフォームコンポーネントは標準トランスフォームに全て統合される
- [FbxNode::ConvertPivotAnimationRecursive\(\)](#)



トランスフォームマトリックスの計算

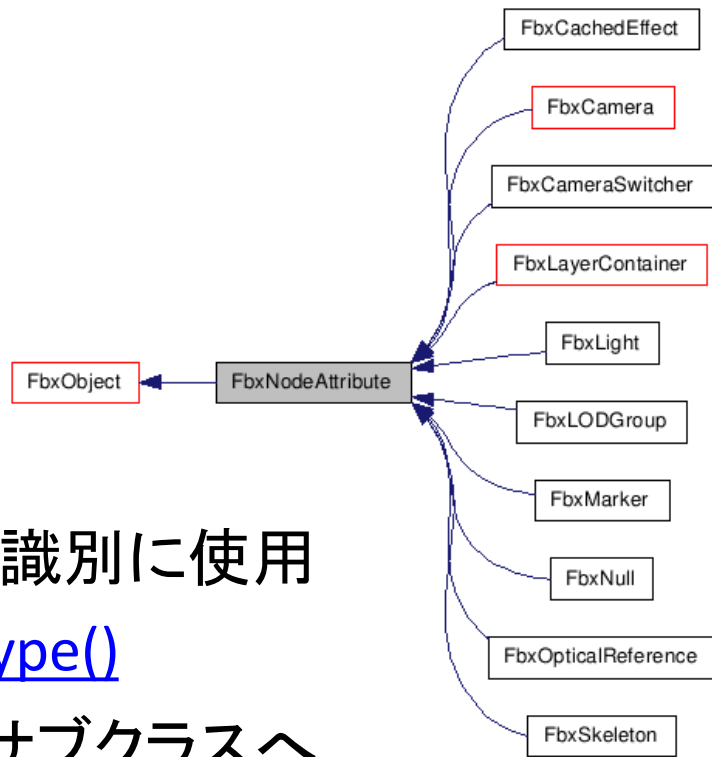
- FBX SDK と Maya は同じ
- 3ds Max は異なる
 - FBX インポーターとエクスポーターはトランスフォームマトリックスを3ds Maxの為にコンバート
- 詳細はこちら:
<http://docs.autodesk.com/FBX/2013/ENU/FBX-SDK-Documentation/files/GUID-10CDD63C-79C1-4F2D-BB28-AD2BE65A02ED.htm>



2. ノードの属性

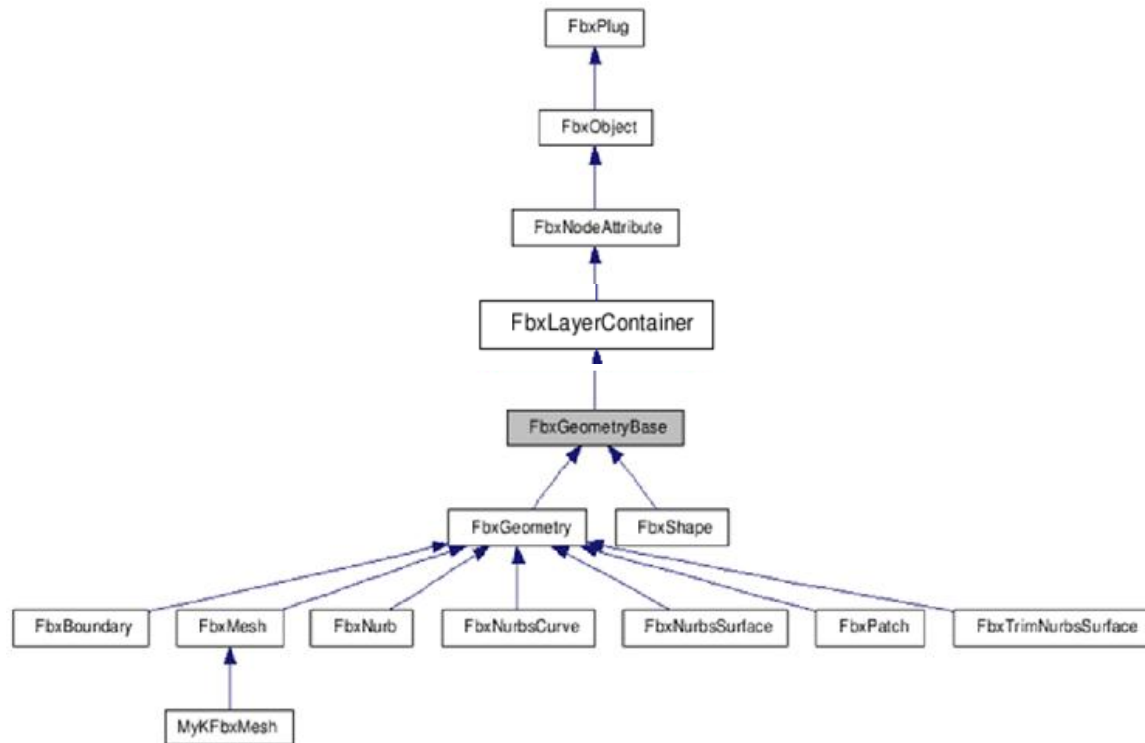
Fbx ノードの属性

- ライト
- カメラ
- ジオメトリ
- FbxNodeAttribute::EAttributeTypeは識別に使用
 - [FbxNodeAttribute::GetAttributeType\(\)](#)
 - ノード属性オブジェクトを適切なサブクラスへのキャストに使用



ジオメトリー構造

ジオメトリークラス構造



FbxLayerContainer

Colors, per polygon

レイヤー 0

Normals, per polygon-vertex
UVs, per vertex
Texture indices, per vertex

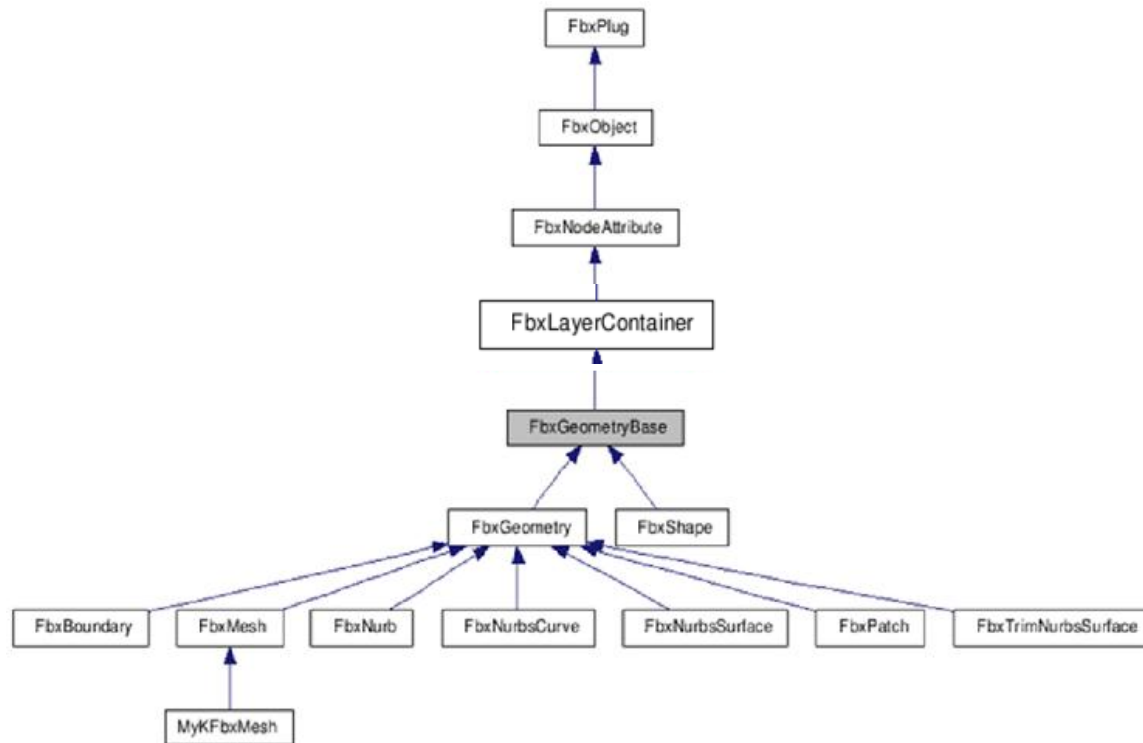
レイヤー 1

Normals, per polygon
UVs, per vertex
Texture indices, per vertex

レイヤー 2

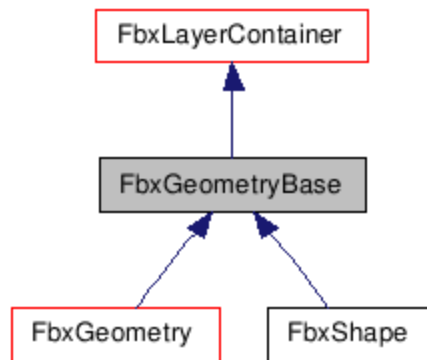


ジオメトリークラス構造

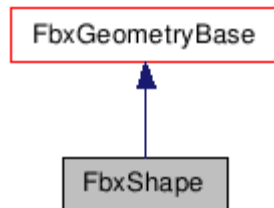


FbxGeometryBase

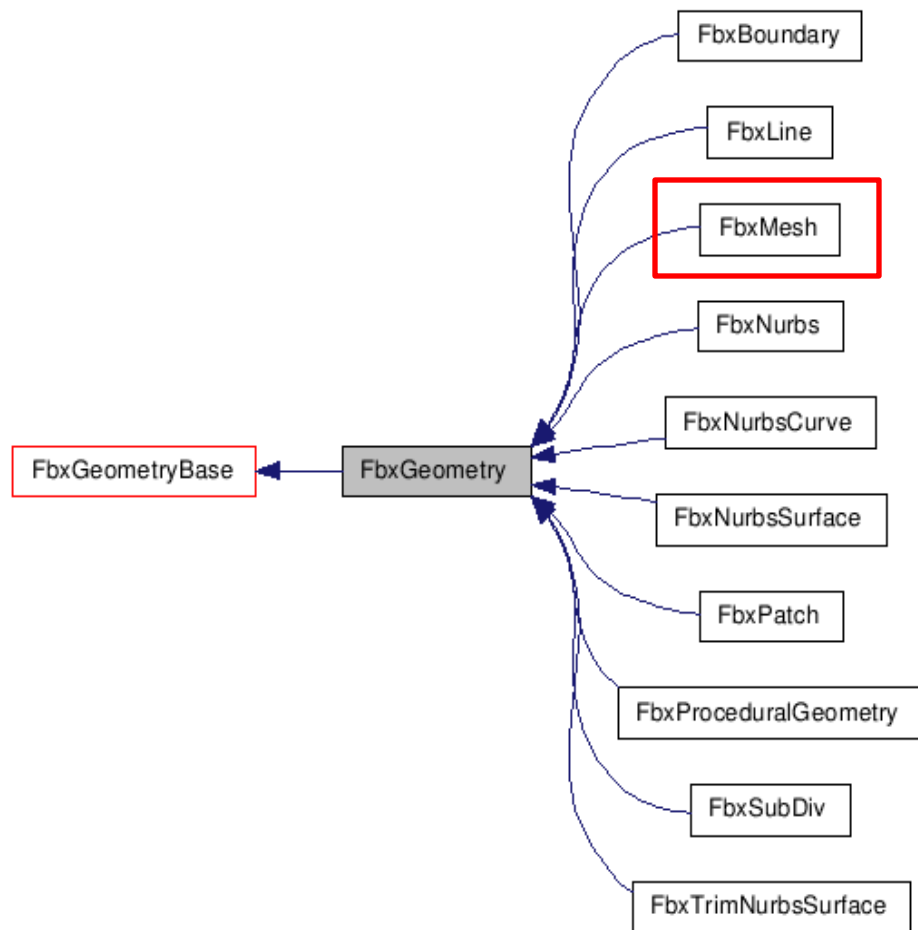
- FbxGeometry
 - デフォメーションをサポートするジオメトリオブジェクトの基本クラス
- FbxShape
 - 制御ポイントセットのデフォメーションを記述、MAYAのクラスターデフォーマーと同じ



FbxShape



FbxGeometry



メッシュ

FbxMesh

- 頂点又は“制御ポイント”のセット
- メッシュの法線、テクスチャやマテリアルを定義する為のレイヤーグループ

FbxMesh

```
void CreateMesh(FbxScene* pScene)
{
    FbxNode* meshNode = FbxNode::Create(pScene, "meshNode");

    FbxMesh* mesh = FbxMesh::Create(pScene, "mesh");
    meshNode->SetNodeAttribute(mesh);

    FbxNode *rootNode = pScene->GetRootNode();
    lRootNode->AddChild(meshNode);
}
```

FbxMesh – ジオメトリー定義

- 頂点は制御ポイント
 - FbxMeshは制御ポイントの配列を使用
- 法線
 - [FbxLayerElementNormal](#)のインスタンスで定義



インスタンス

FbxGeometry – インスタンス

- FbxNodeの複数インスタンスに一つのインスタンスをバインド可能
- メモリーの節約に

インスタンス例

// ノード属性として与えられたメッシュでcubeのインスタンスを作成し、シーンに追加

```
FbxNode* CreateCubeInstance(FbxScene* pScene, const char* pName, FbxMesh*
pFirstCube)
{
    // FbxNodeの作成
    FbxNode* lNode = FbxNode::Create(pScene, pName);

    // ノード属性を設定
    lNode->SetNodeAttribute(pFirstCube);

    // cubeをスケール
    lNode->LclScaling.Set(FbxVector4(0.3, 0.3, 0.3));

    // シーンにノードを追加
    pScene->GetRootNode()->AddChild(lNode);

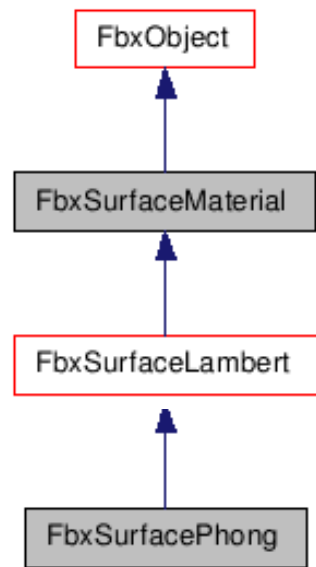
    // FbxNodeを返す
    return lNode;
}
```



マテリアル

マテリアル

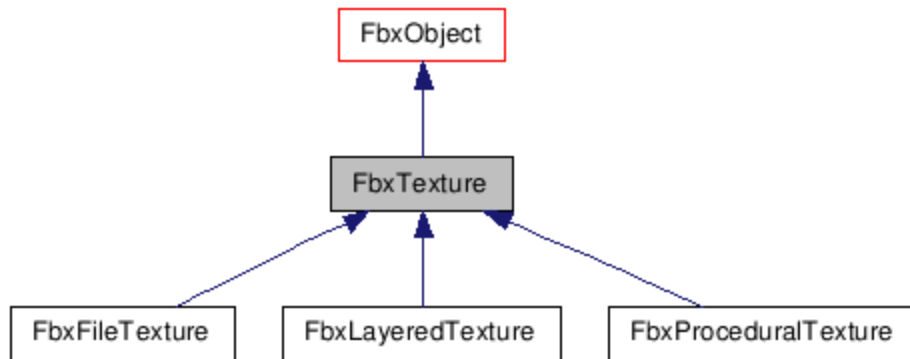
- [FbxNode::AddMaterial\(\)](#)を介して[FbxNode](#)のインスタンスを結びつける
- FbxSurfaceMaterial
 - FbxSurfaceLambert
 - FbxSurfacePhong
- FbxLayerElementMaterial
 - ジオメトリーへのマテリアルをマップするレイヤーエレメント
 - 注意, ドキュメントには古いクラス名 FbxGeometryElementMaterial を使用している箇所有り



テクスチャー

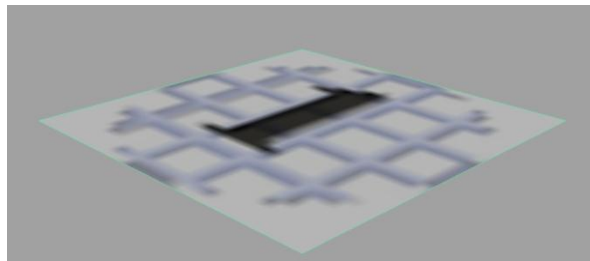
テクスチャ

- [FbxTexture](#)が基本クラス
- [FbxFileTexture](#)がファイルからのロードでテクスチャを表示

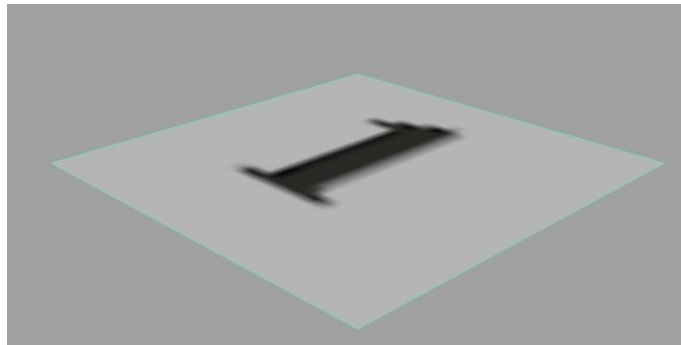


レイヤーテクスチャ

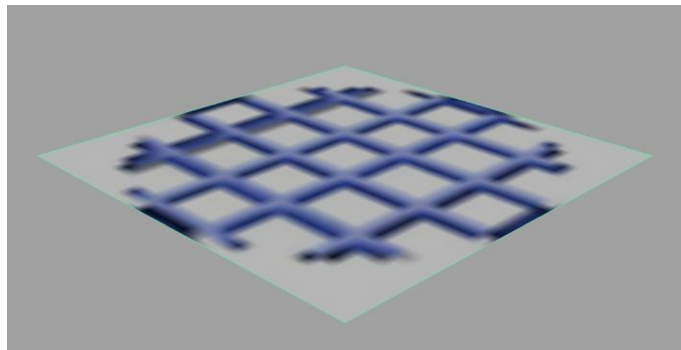
- ブレンドによる複数のテクスチャのコンビネーション



Result



Layer 1



Layer 0



エクササイズ

- cubeを作成
- cubeの各面にマテリアルを作成
- オプション: sphereを作成
- オプション: sphereにテクスチャーを追加
- ASCII FBXフォーマットにてシーンをエクスポート (既に体験済み)
- テキストエディターにてファイルのコンテンツを確認



お疲れ様でした

Autodesk®

ビデオ

<ftp://sparks:Sparks2012@87.106.99.61/>

又は直接:

IP address: 87.106.99.61

- User name: *sparks*
- Password: *Sparks2012*
- *FBX SDK webcast_Japanese* フォルダーにビデオとサンプルを置きます

Q&A

akira.kudo@autodesk.com