Autodesk® FBX® SDK

Geometry

Video

ftp://sparks:Sparks2012@87.106.97.50/
or directly:
IP address: 87.106.97.50

- User name: *sparks*
- Password: *Sparks2012*

- *FBX_SDK_webcast* folder contains the video and handouts from yesterday

Exercise – Unit 3

- Create a scene that contains one camera and one light.
- Add custom properties.
- Output the connections in the scene.
- Export the scene to an ASCII FBX file.

FBX SDK Webcast Agenda

- Day / Hour 1 – Welcome to FBX SDK
- Day / Hour 2 – FBX Basics (Import/Export/Scenes)
- Day / Hour 3 – SDK Object Model
- **Day / Hour 4 – Geometry**
- Day / Hour 5 – Animation

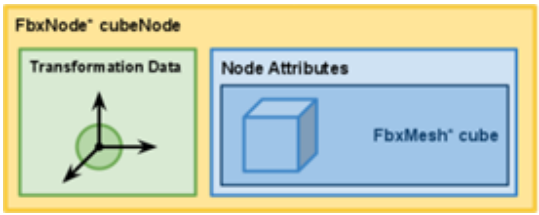
Geometry

- FBX Nodes
  - Transformation Data
  - Node Attributes
- Geometry Structure
- Meshes
- Materials
- Textures

FBX Nodes

FbxNode – Reminder from Unit 2

1. Provides transformation
2. Provides connections to node attribute
   - node attribute can be shared among multiple nodes



1. Transformation Data

Transformation Data

- Provides transformation
  - Default Translation, Rotation, and Scaling Vectors are properties
    - LclTranslation
    - LclRotation
    - LclScaling

Autodesk FBX demo 9

Provides transformation

```
//Get the node's default TRS properties
FbxDouble3 lTranslation = lNode->
   LclTranslation.Get();
FbxDouble3 lRotation    = lNode->
   LclRotation.Get();
FbxDouble3 lScaling     = lNode->
   LclScaling.Get();
```

Autodesk FBX demo 10

Transformation Data

- Provides transformation
  - Global and Local Transformation Matrices
    - EvaluateGlobalTransform
    - EvaluateLocalTransform

Global and Local Trans. Matrices

```
// Get the node's default global transformation matrix.
FbxXMatrix& lGlobalTransform = lNode->EvaluateGlobalTransform();

// Get the node's default local transformation matrix.
FbxXMatrix& lLocalTransform = lNode->EvaluateLocalTransform();
```

Global and Local Trans. Matrices

- Can also use FbxScene's animation evaluator
  - FbxAnimEvaluator::GetNodeGlobalTransform()
  - FbxAnimEvaluator::GetNodeLocalTransform()

Global and Local Trans. Matrices

```
// Get the scene's animation evaluator.
FbxAnimEvaluator* lEvaluator = lScene->getEvaluator();

// Get the node's default global transformation matrix.
FbxXMatrix& lGlobalTransform = lEvaluator->GetNodeGlobalTransform(lNode);

// Get the node's default local transformation matrix.
FbxXMatrix& lLocalTransform = lEvaluator->GetNodeLocalTransform(lNode);
```

Global and Local Trans. Matrices

- Can also use them with a specific FbxTime during animation sequences
- Note: Documentation refers to FbxXMatrix, but it should be FbxAMatrix

Global and Local Trans. Matrices

```
FbxTime lTime;

// Set the time at two seconds.
lTime.SetSecondDouble((float) 2);

// Get the node's global transformation matrix at 2 seconds.
FbxXMatrix& lGlobalTransform = lNode->EvaluateGlobalTransform(lTime);
```

Transformation Data

- The scene's axis system and system units are defined in its FbxGlobalSettings object
- The transformation data of a node includes its translation, rotation and scaling vectors with respect to its parent.

Geometric Transformation Properties

- how a FbxNodeAttribute is offset from the FbxNode's local frame of reference
  - FbxNode::GeometricTranslation
  - FbxNode::GeometricRotation
  - FbxNode::GeometricScaling
- These are similar to 3ds Max's object-offset transformation

Baking Transform Components

- Basically combine all pre- and post transform components into the standard transforms
- FbxNode::ConvertPivotAnimationRecursive()

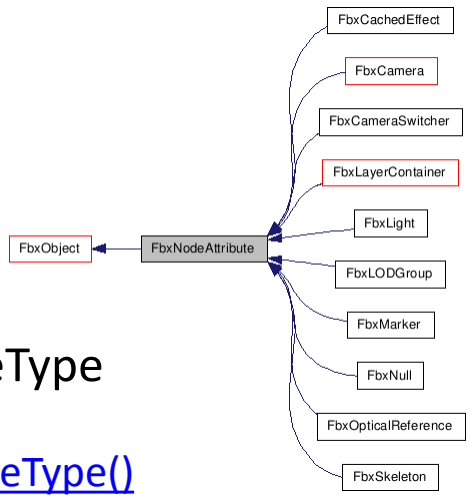
Computing Transformation Matrices

- FBX SDK and Maya are the same
- 3ds Max is different
  - the FBX importers and exporters for 3ds Max automatically convert transformation matrices
- Details are here: http://docs.autodesk.com/FBX/2013/ENU/FBX-SDK-Documentation/files/GUID-10CDD63C-79C1-4F2D-BB28-AD2BE65A02ED.htm

2. Node Attributes

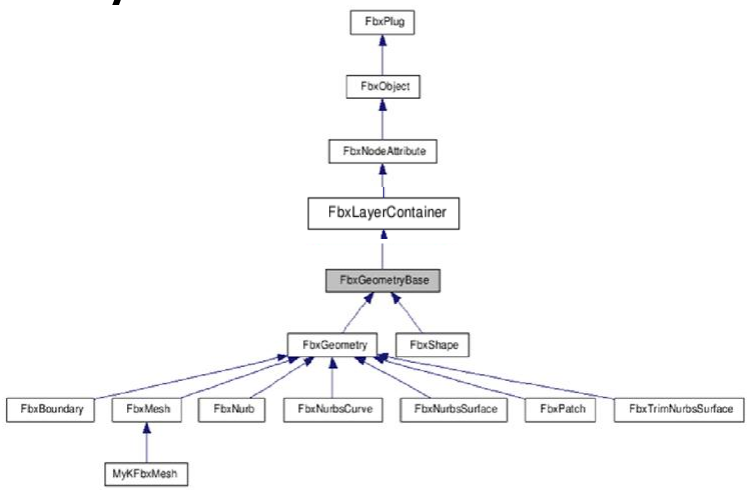
Fbx Node Attributes

- Lights
- Cameras
- Geometry
- FbxNodeAttribute::EAttributeType used for identification
  - FbxNodeAttribute::GetAttributeType()
  - useful for downcasting the node attribute object to its appropriate subclass



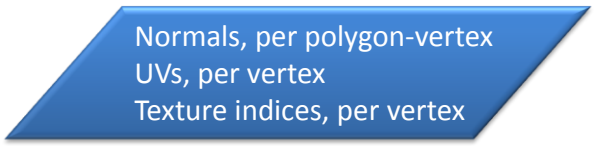
Geometry Structure

Geometry Class Structure

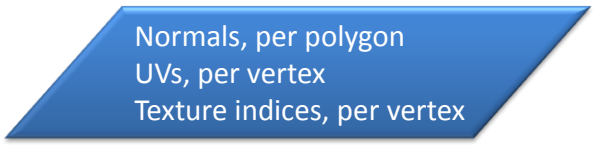


FbxLayerContainer

Layer 0

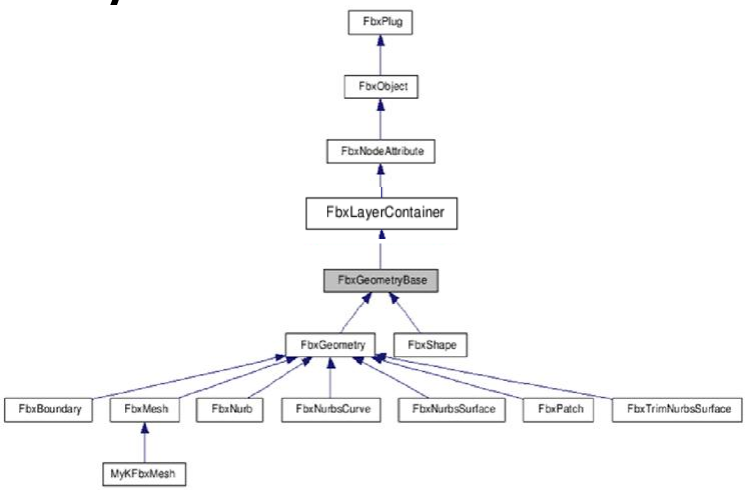


Layer 1



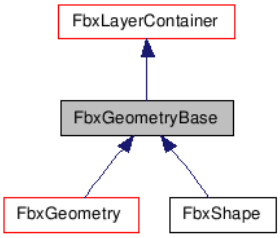
Layer 2

Geometry Class Structure

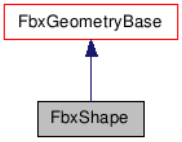


FbxGeometryBase

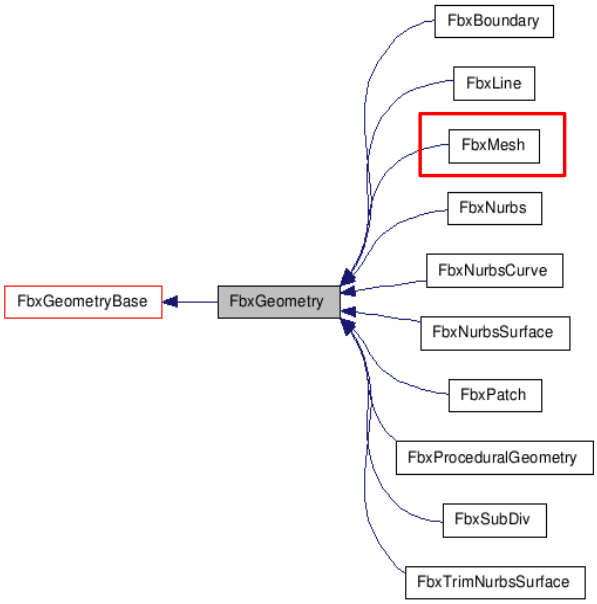
- FbxGeometry
  - base class for geometric objects that support control point deformations
- FbxShape
  - describes the deformation on a set of control points, that is similar to the cluster deformer in Maya.



FbxShape



FbxGeometry



Meshes

FbxMesh

- set of vertices, or "control points“
- a group of layers to define the mesh's normals, textures, and materials

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 – Geometry Definition

- Vertices are known as control points
  - FbxMesh uses an array of control points
- Normals
  - defined in an instance of FbxLayerElementNormal.

Instancing

FbxGeometry - Instancing

- Able to bind a single instance to multiple instances of FbxNode
- Saves memory by having only one definition

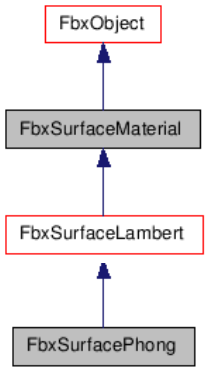
Instancing Example

```
// Create a cube instance with the given mesh as node attribute, and add it to
the scene.
FbxNode* CreateCubeInstance(FbxScene* pScene, const char* pName, FbxMesh*
pFirstCube)
{
    // create a FbxNode
    FbxNode* lNode = FbxNode::Create(pScene,pName);

    // set the node attribute
    lNode->SetNodeAttribute(pFirstCube);

    // rescale the cube
    lNode->LclScaling.Set(FbxVector4(0.3, 0.3, 0.3));

    // Add node to the scene
    pScene->GetRootNode()->AddChild(lNode);

    // return the FbxNode
    return lNode;
}
```

Materials

Materials

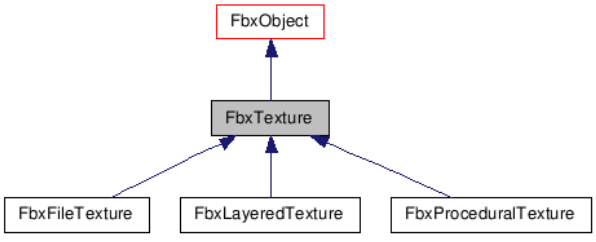
- bound to instances of FbxNode via FbxNode::AddMaterial()
- FbxSurfaceMaterial
  - FbxSurfaceLambert
    - FbxSurfacePhong
- FbxLayerElementMaterial
  - Layer element for mapping materials to a geometry
  - Note, the docs sometime refer to this under it's old name: FbxGeometryElementMaterial



Textures

Textures

- FbxTexture is the base class
- FbxFileTexture represents any texture loaded from a file



Layered Textures

- a combination of multiple textures blended sequentially

Exercise

- Create a cube
- Create materials for each side of the cube
- OPTIONAL: Create a sphere
- OPTIONAL: add a texture to the sphere
- Export the scene to ASCII FBX format (nothing new to do here)
- Open FBX file in ASCII text editor and review files contents