Autodesk® FBX® SDK

SDK Object Model

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 2

- Create an importer and list the nodes of a FBX file

Autodesk FBX

demo

FBX SDK Webcast Agenda

- Day / Hour 1 – Welcome to FBX SDK
- Day / Hour 2 – Functionality basics
- **Day / Hour 3 – SDK Object Model**
- Day / Hour 4 – Geometry
- Day / Hour 5 – Animation

Autodesk FBX

SDK Object Model

- Collections
- Properties
- Connections
- Lights and Cameras
- Miscellaneous topics

Autodesk®
FBX®

Collections

Autodesk®
FBX®

Collections

- container classes are derived from the FbxCollection class
- Examples: FbxAnimLayer, FbxAnimStack, FbxDocument, and FbxScene

Autodesk FBX demo

Collections

- Add members - FbxCollection::AddMember()
- Remove members - FbxCollection::RemoveMember()
- Count members - FbxCollection::GetMemberCount()
- Get a member - FbxCollection::GetMember()
- Search its members - FbxCollection::FindMember()

Autodesk FBX demo

Properties

Autodesk FBX

FbxProperty

- Standalone class for properties
- provides methods for managing the connections between objects
- object such as FbxNode uses FBX properties (FbxProperty) rather than conventional C++ member variables
  - ensures that the data of a FbxObject is strongly typed
- When FbxObject created, static instance of FbxProperty are initialized

Autodesk FBX

demo

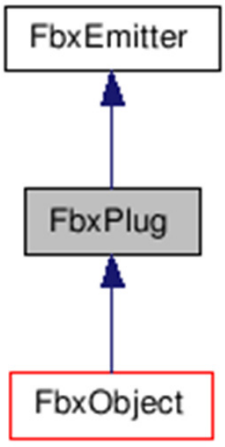
User-defined Data

- The FbxProperty class can contain user-defined data
- can be dynamically associated to a FbxObject at runtime

Autodesk FBX demo

FbxObject

- provides methods for managing the connections between objects
- provides a property (FbxProperty) mechanism to describe characteristics of objects in a scene



Autodesk FBX demo

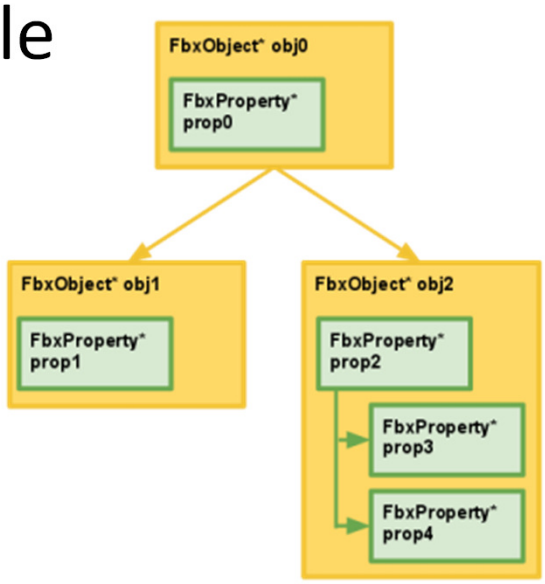
Connections

Connections

- internal data structure that manages the two-way relationship between FBX *objects* and/or FBX *properties*
- manipulated using the FbxObject and FbxProperty

Connection Example

- object-property
- object-object
- property-property



Autodesk FBX

object-property connections

- Properties are contained within objects as *sources*
- Calling FbxObject::GetSrcProperty() will return the object's *source* property at the given index
- Symmetrically, calling FbxProperty::GetDstObject() will return the property's *destination* object

Autodesk FBX

object-object connections

- Parent-child relationships among *objects* use connections (for example hierarchy of a scene)
- Typically, an object's children are *sources*, and are accessed using FbxObject::GetSrcObject()
- An object's parent is a *destination*, and is accessed using FbxObject::GetDstObject()

Autodesk FBX

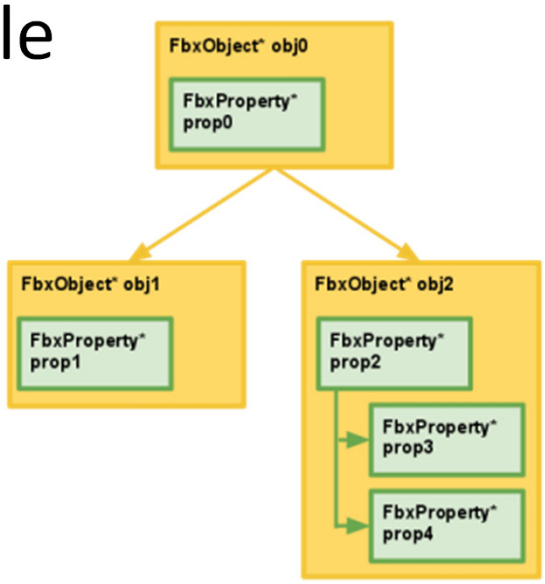
property-property connections

- Parent-child relationships among *properties* also use connections (for example: the property hierarchy of FbxIOSettings)
- Typically, a property's children are *sources*, and are accessed using FbxProperty::GetSrcProperty()
- A property's parent is referred to as a *destination*, and is accessed using FbxProperty::GetDstProperty()

Autodesk FBX

Connection Example

- object-property
- object-object
- property-property



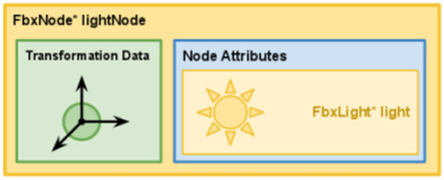
Autodesk FBX

demo

Lights and Cameras

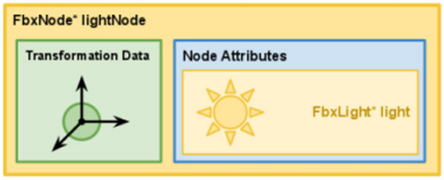
Autodesk FBX

FbxLight



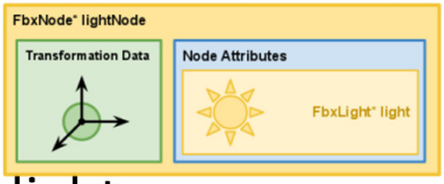
- Light Types:
  - ePoint
  - eDirectional
  - eSpot
  - eArea
  - eVolume

FbxLight



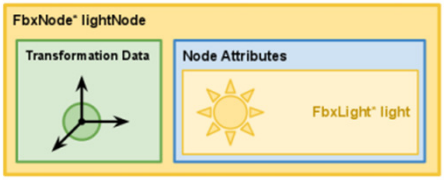

```
// Create a spotlight.
void CreateLight(FbxScene* pScene, char* pName)
{
    FbxLight* light = FbxLight::Create(pScene,pName);
    light->LightType.Set(FbxLight::eSPOT);
    light->CastLight.Set(true);

    FbxNode* lightNode = FbxNode::Create(pScene,pName);
    lightNode->SetNodeAttribute(light);

    FbxNode* rootNode = pScene->GetRootNode();
    RootNode->AddChild(lightNode);
}
```


FbxLight



- Pointing a spot/directional light
  - light's node must have its target set using FbxNode::SetTarget()
- Color
  - defined in its FbxLight::Color property
- Intensity
  - defined in its FbxLight::Intensity property

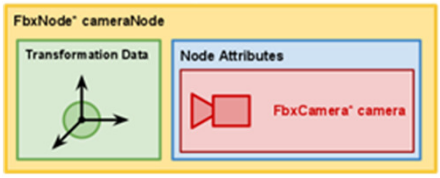
FbxLight



- Decay
  - defined in its FbxLight::DecayType property
- Shadows
  - enabled using the FbxLight::CastShadows boolean property
  - color of the light's shadow is defined in the FbxLight::ShadowColor property
  - A shadow texture may also be applied using FbxLight::SetShadowTexture().

demo

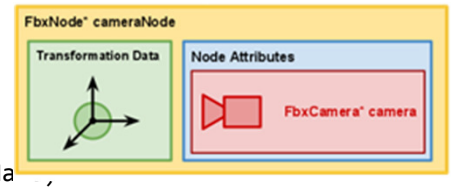
FbxCamera



- Types:
  - ePerspective
  - eOrthogonal

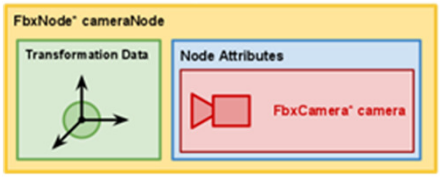
- FbxCameraStereo
  - Adds specialization for stereo cameras.

FbxCamera




```
void CreateCamera(FbxScene* pScene, char* pCameraNa
{
    FbxCamera* camera = FbxCamera::Create(pScene, pCameraName);

    FbxNode* cameraNode = FbxNode::Create(pScene, pCameraName);
    cameraNode->SetNodeAttribute(camera);

    FbxNode* rootNode = pScene->GetRootNode();
    RootNode->AddChild(cameraNode);

    // Once a camera has been created, it can be set as the scene's default camera.
    // A scene must have its default camera set explicitly,
    // even if there is only one camera in the scene.
    pScene->GetGlobalSettings().SetDefaultCamera((char *)camera->GetName());
}
```


FbxCamera



- Pointing the camera
  - camera's node must have its target set using FbxNode::SetTarget()
- See FbxCamera docs for all the properties that can be set
  - For example: Roll, Aspect, FOV, etc.

Autodesk FBX

demo

Miscellaneous

Autodesk FBX

Copying an FBX Object

- An FBX object may be copied by calling its Copy() member function
- Copies all of its associated FbxProperty instances, and their values
- Does NOT include any of its inter-object connections

Autodesk FBX

Copying an FBX Object

```
// Assume that pScene is a pointer to a valid scene object.
FbxMesh* pSourceMesh = FbxMesh::Create (pScene, "");
// ... Define control points, etc. for pSourceMesh.

// This mesh will be overwritten
FbxMesh* pTargetMesh = FbxMesh::Create (pScene, "");

// Copy the data from pSourceMesh into pTargetMesh. Note that
// the source object and the target object must be instances of
// the same class (FbxMesh in this case).
pTargetMesh->Copy(pSourceMesh);
```

Autodesk FBX

Error Handling

- Functions typically return false for failure
- `objectname->GetLastErrorString()` returns a string with an error message
- `objectname->GetLastErrorID()` returns an integer value.
  - use FBX enumerated values or defined values to handle the error appropriately

Autodesk FBX

Error Handling sample

- http://docs.autodesk.com/FBX/2013/ENU/FBX-SDK-Documentation/files/GUID-5509751F-8679-4D32-987D-8343FD8F6D1A.htm

Autodesk FBX

demo

Supported String Formats

- FBX SDK uses Unicode UTF-8 strings internally
- Converts its UTF-8 strings to the required string format when calling a function in an operating system's API
- FbxString defined in fbxstring.h has string

Customizing the SDK

- Custom User Data
  - FbxObject/FbxProperty have Get/SetUserDataPtr()
- Custom Properties
- Custom Classes
  - FbxManager::RegisterFbxClass()
- Custom User Data for Layer Elements
  - FbxLayerElementUserData
- Custom File Formats
  - Customizing File Formats with FBX SDK I/O Plug-ins

Exercise

- 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.

Autodesk®
FBX®