



X3D C/C++/C# Language Bindings (Updates)

SIGGRAPH Web3D Standardization Meeting (Korea Chapter)
Los Angeles, USA

July 29, 2019

Myeong Won Lee

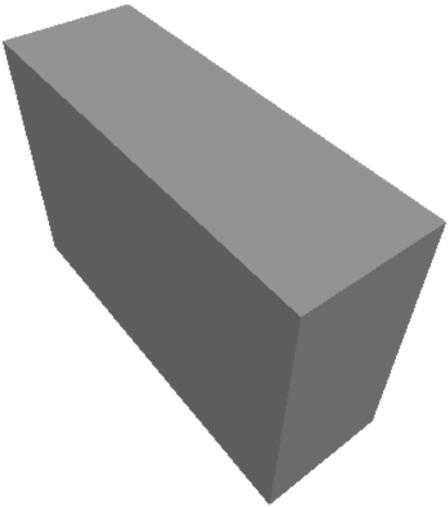
Status of Work

- ISO/IEC 19777-3 X3D C Language Binding
 - CD in progress
- ISO/IEC 19777-4 X3D C++ Language Binding
 - CD in progress
- ISO/IEC 19777-5 X3D C# Language Binding
 - CD in progress

C/C++/C# Language Binding Concepts

- What is C/C++/C# language binding?
 - X3D scene access interface using C, C++ and C# languages
 - Specify 19775-2 X3D Scene Access Interface using C, C++, C# languages
 - Development of C, C++/C# programs using X3D data types and functions
 - X3D scene read, update, store, and exchange in C, C++/C# applications
- Scope
 - Provides a browser implementation independent way of accessing a browser's capabilities via the languages
 - Provides a set of implementation independent base classes and interfaces that represent possible interactions with an X3D scene through an SAI
 - Provides C, C++ and C# API formats for X3D scene access

A Simple Example of X3D Scene Access API



getX3D
getScene
getBackground
getViewpoint
getShape
getBox
getApperance
getMaterial

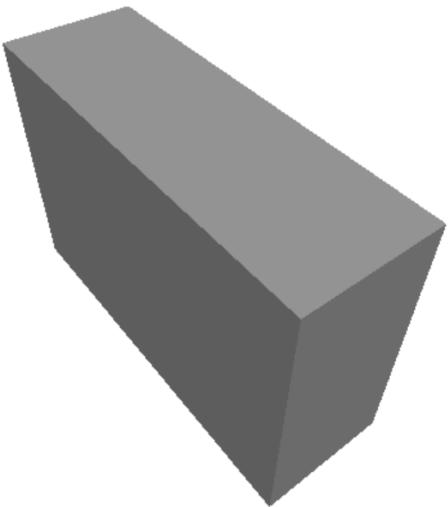
setX3D
setScene
setBackground
setViewpoint
setShape
setBox
setApperance
setMaterial

```
<X3D>
<Scene>
  <Background skyColor='1 1 1' />
  <Viewpoint description='Book View'
    orientation='-0.747 -0.624 -0.231 1.05' position='-
    1.81 3.12 2.59' />
  <Shape>
    <Box size='1 2 3' />
    <Appearance>
      <Material />
    </Appearance>
  </Shape>
</Scene>
</X3D>
```

X3D

X3D Scene Access Interface (SAI)

A Sample of X3D Scene Access API (C/C++)



getX3D
getScene
getBackground
getViewpoint
getShape
getBox
getApperance
getMaterial

setX3D
setScene
setBackground
setViewpoint
setShape
setBox
setApperance
setMaterial

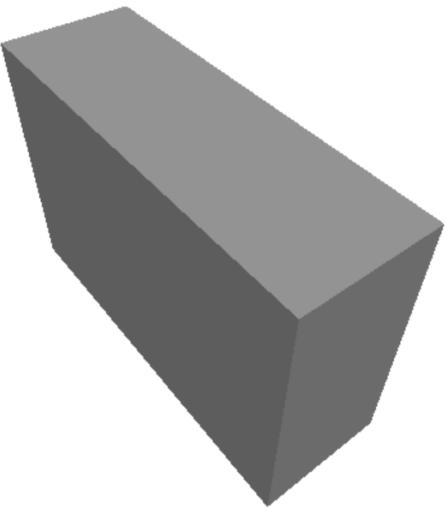
getX3D (&pX3D)
getScene(&pScene)
getBackground(&pBackground)
getViewpoint(&pViewpoint)
getShape(&pShape)
getBox(&pBox)
getApperance(&pAppearance)
getMaterial(&pMaterial)

setX3D (pX3D)
setScene(pScene)
setBackground(pBackground)
setViewpoint(pViewpoint)
setShape(pShape)
setBox(pBox)
setApperance(pAppearance)
setMaterial(pMaterial)

X3D C++ encoding

X3D Scene Access Interface (SAI)

A Sample of X3D Scene Access API (C#)



getX3D
getScene
getBackground
getViewpoint
getShape
getBox
getApperance
getMaterial

setX3D
setScene
setBackground
setViewpoint
setShape
setBox
setApperance
setMaterial

getX3D (pX3D)
getScene(pScene)
getBackground(pBackground)
getViewpoint(pViewpoint)
getShape(pShape)
getBox(pBox)
getApperance(pAppearance)
getMaterial(pMaterial)

setX3D (pX3D)
setScene(pScene)
setBackground(pBackground)
setViewpoint(pViewpoint)
setShape(pShape)
setBox(pBox)
setApperance(pAppearance)
setMaterial(pMaterial)

X3D C# encoding

X3D Scene Access Interface (SAI)

X3D C/C++ Binding Viewer Program Example

1. X3DBindingViewer

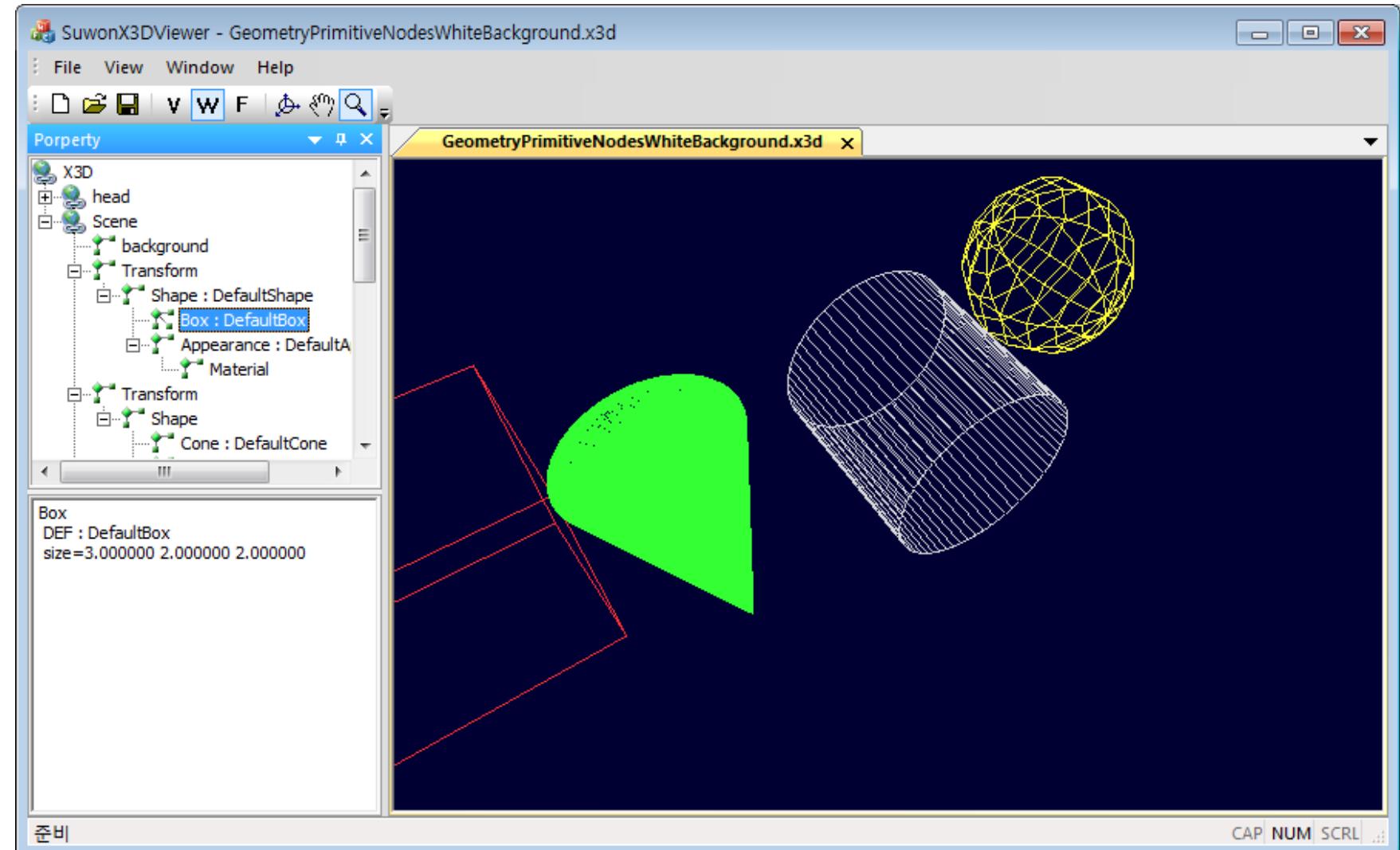
- 1) Load X3DLib.dll
- 2) Parse an X3D file with X3DLib
- 3) Read, update, draw, and store the X3D file using X3DLib classes

2. X3DLib.dll

- 1) Parse an X3D file
- 2) Insert the parsed X3D into an internal class
- 3) Provide an interface to read X3D

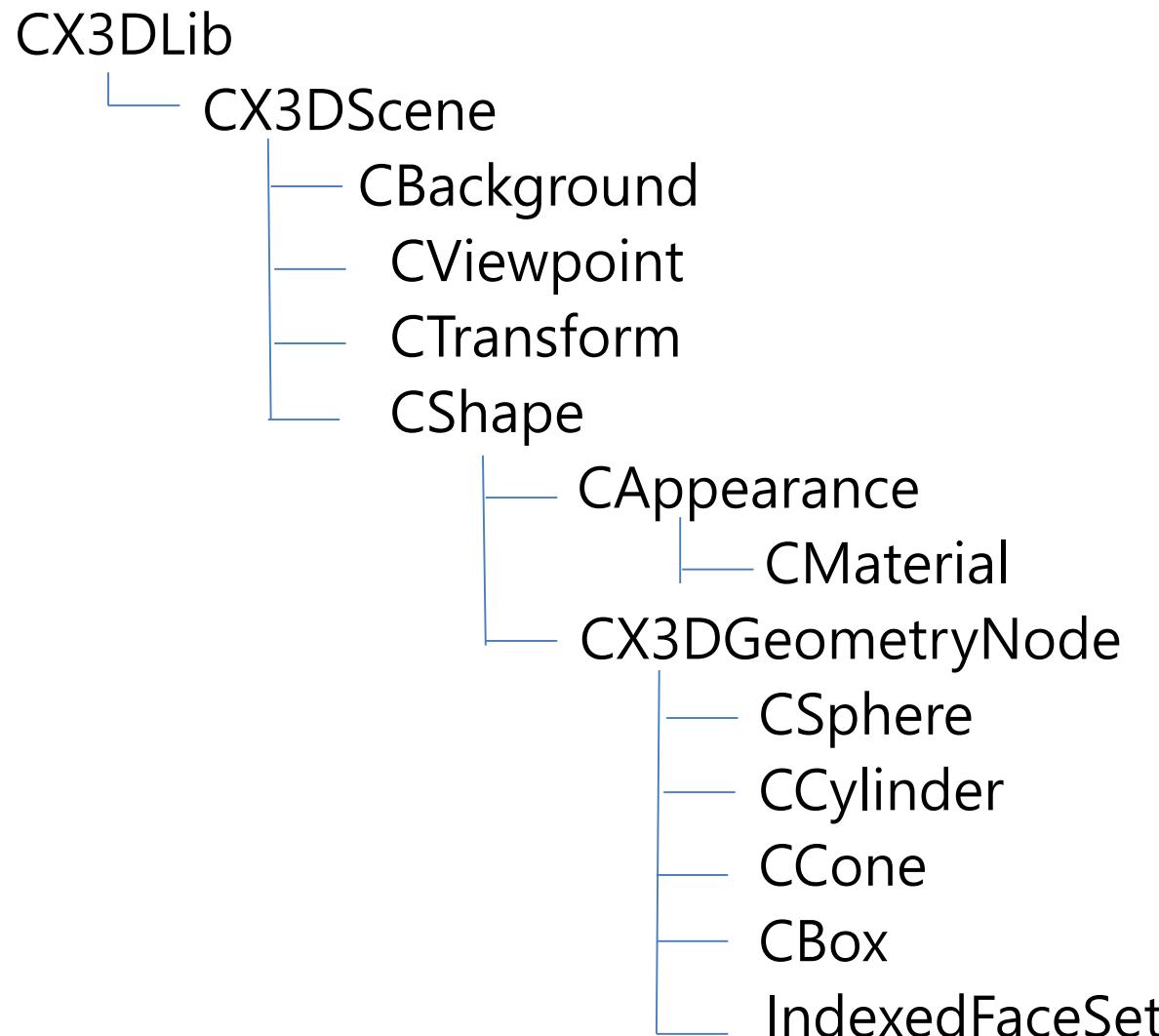
X3DBindingViewer (X3D C++ Binding Viewer)

X3D Tree View



Property View

X3D C++ Binding Class Structure (Partial)



Background

```
//C.3.6 Background
/** Background defines a concrete node interface that extends interface X3DBackgroundNode. */

class AFX_EXT_CLASS CBackground : public CX3DBackgroundNode
{
    DECLARE_DYNAMIC(CBackground);
public:
    CBackground();
    virtual ~CBackground();
//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return array of String results array [] from MFString inputOutput field named "backUrl" */
    CString* getBackUrl ();

    /** Return number of primitive values in "backUrl" array */
    int getNumBackUrl ();

    /** Assign String array [] to MFString inputOutput field named "backUrl" */
    void setBackUrl (CString* values, int size);

    X3DBackgroundNode : X3DBindableNode {
        SFBool [in]      set_bind
        MFFloat [in,out] groundAngle []      [0,π/2]
        MFCOLOR [in,out]groundColor []      [0,1]
        SFNode [in,out] metadata NULL      [X3DMetadataObject]
        MFFloat [in,out] skyAngle []      [0,π]
        MFCOLOR [in,out]skyColor 0 0 0 [0,1]
        SFFloat [in,out] transparency 0      [0,1]
        SFTime [out]     bindTime
        SFBool [out]     isBound
    }
}
```

Background

```
/** Assign single String value [] as the MFString array for inputOutput field named "backUrl" */
void setBackUrl (CString value);

/** Return array of String results array [] from MFString inputOutput field named "bottomUrl" */
CString* getBottomUrl ();

/** Return number of primitive values in "bottomUrl" array */
int getNumBottomUrl ();

/** Assign String array [] to MFString inputOutput field named "bottomUrl" */
void setBottomUrl (CString* values, int size);

/** Assign single String value [] as the MFString array for inputOutput field named "bottomUrl" */
void setBottomUrl (CString value);

/** Return array of String results array [] from MFString inputOutput field named "frontUrl" */
CString* getFrontUrl ();

/** Return number of primitive values in "frontUrl" array */
int getNumFrontUrl ();

/** Assign String array [] to MFString inputOutput field named "frontUrl" */
void setFrontUrl (CString* values, int size);
```

Background

```
/** Assign single String value [] as the MFString array for inputOutput field named "frontUrl" */
void setFrontUrl (CString value);

/** Return array of String results array [] from MFString inputOutput field named "leftUrl" */
CString* getLeftUrl ();

/** Return number of primitive values in "leftUrl" array */
int getNumLeftUrl ();

/** Assign String array [] to MFString inputOutput field named "leftUrl" */
void setLeftUrl (CString* values, int size);

/** Assign single String value [] as the MFString array for inputOutput field named "leftUrl" */
void setLeftUrl (CString value);

/** Return array of String results array [] from MFString inputOutput field named "rightUrl" */
CString* getRightUrl ();

/** Return number of primitive values in "rightUrl" array */
int getNumRightUrl ();

/** Assign String array [] to MFString inputOutput field named "rightUrl" */
void setRightUrl (CString* values, int size);
```

Background

```
/** Assign single String value [] as the MFString array for inputOutput field named "rightUrl" */
void setRightUrl (CString value);

/** Return array of String results array [] from MFString inputOutput field named "topUrl" */
CString* getTopUrl ();

/** Return number of primitive values in "topUrl" array */
int getNumTopUrl ();

/** Assign String array [] to MFString inputOutput field named "topUrl" */
void setTopUrl (CString* values, int size);

/** Assign single String value [] as the MFString array for inputOutput field named "topUrl" */
void setTopUrl (CString value);
}
```

Viewpoint

```
//C.3.242 Viewpoint
/** Viewpoint defines a concrete node interface that extends interface X3DViewpointNode. */
class AFX_EXT_CLASS CViewpoint : public CX3DViewpointNode
{
    DECLARE_DYNAMIC(CViewpoint);
public:
    CViewpoint();
    virtual ~CViewpoint();
//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return array of 3-tuple float results array in radians from SFVec3f inputOutput field named "centerOfRotation" */
    float* getCenterOfRotation ();

    /** Assign 3-tuple float array in radians to SFVec3f inputOutput field named "centerOfRotation" */
    void setCenterOfRotation (float* value);

    Viewpoint : X3DViewpointNode {
        SFBool      [in]      set_bind
        SFVec3f     [in,out]   centerOfRotation  0 0 0  (-∞,∞)
        SFString    [in,out]   description      ""
        SFFloat    [in,out]   fieldOfView     π/4  (0,π)
        SFBool      [in,out]   jump          TRUE
        SFNode     [in,out]   metadata       NULL   [X3DMetadataObject]
        SFRotation  [in,out]   orientation    0 0 1 0 [-1,1], (-∞,∞)
        SFVec3f     [in,out]   position       0 0 10 (-∞,∞)
        SFBool      [in,out]   retainUserOffsets FALSE
        SFTime     [out]      bindTime
        SFBool      [out]      isBound
    }
}
```

Viewpoint

```
/** Return float result [] from SFFloat inputOutput field named "fieldOfView" */
float getFieldOfView ();

/** Assign float value [] to SFFloat inputOutput field named "fieldOfView" */
void setFieldOfView (float value);

/** Return array of 3-tuple float results array [] from SFVec3f inputOutput field named "position" */
float* getPosition ();

/** Assign 3-tuple float array [] to SFVec3f inputOutput field named "position" */
void setPosition (float* value);
}
```

Transform

```
//C.3.232 Transform
/** Transform defines a concrete node interface that extends interface X3DGroupingNode. */

class AFX_EXT_CLASS CTransform : public CX3DGroupingNode
{
    DECLARE_DYNAMIC(CTransform);

public:
    CTransform();
    virtual ~CTransform();

//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return array of 3-tuple float results array [] from SFVec3f inputOutput field named "center" */
    float* getCenter();

    /** Assign 3-tuple float array [] to SFVec3f inputOutput field named "center" */
    void setCenter (float* value);

    Transform : X3DGroupingNode {
        MFNode      [in]      addChildren           [X3DChildNode]
        MFNode      [in]      removeChildren        [X3DChildNode]
        SFVec3f     [in,out]   center              0 0 0   (-∞,∞)
        MFNode      [in,out]   children             []      [X3DChildNode]
        SFNode      [in,out]   metadata             NULL   [X3DMetadataObject]
        SFRotation   [in,out]   rotation             0 0 1 0 [-1,1] or (-∞,∞)
        SFVec3f     [in,out]   scale                1 1 1   (-∞, ∞)
        SFRotation   [in,out]   scaleOrientation     0 0 1 0 [-1,1] or (-∞,∞)
        SFVec3f     [in,out]   translation          0 0 0   (-∞,∞)
        SFVec3f     []         bboxCenter          0 0 0   (-∞,∞)
        SFVec3f     []         bboxSize            -1 -1 -1 [0,∞) or -1 -1 -1
    }
}
```

Transform

```
/** Return array of 4-tuple float results array in radians from SFRotation inputOutput field named "rotation" */
float* getRotation ();

/** Assign 4-tuple float array in radians to SFRotation inputOutput field named "rotation" */
void setRotation (float* value);

/** Return array of 3-tuple float results array [] from SFVec3f inputOutput field named "scale" */
float* getScale ();

/** Assign 3-tuple float array [] to SFVec3f inputOutput field named "scale" */
void setScale (float* value);

/** Return array of 4-tuple float results array in radians from SFRotation inputOutput field named "scaleOrientation" */
float* getScaleOrientation ();

/** Assign 4-tuple float array in radians to SFRotation inputOutput field named "scaleOrientation" */
void setScaleOrientation (float* value);

/** Return array of 3-tuple float results array [] from SFVec3f inputOutput field named "translation" */
float* getTranslation ();

/** Assign 3-tuple float array [] to SFVec3f inputOutput field named "translation" */
void setTranslation (float* value);

}
```

Shape

```
//C.3.199 Shape
/** Shape defines a concrete node interface that extends interface X3DShapeNode. */
class AFX_EXT_CLASS CShape : public CX3DShapeNode
{
    DECLARE_DYNAMIC(CShape);
public:
    CShape();
    virtual ~CShape();

//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return X3DAppearanceNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput
    field named "appearance" */
    void getAppearance (CX3DNode result);

    /** Assign X3DAppearanceNode value (using a properly typed node) to SFNode inputOutput field named "appearance" */
    void setAppearance (X3DAppearanceNode node);

    Shape : X3DShapeNode {
        SFNode  [in,out] appearance NULL      [X3DAppearanceNode]
        SFNode  [in,out] geometry   NULL      [X3DGeometryNode]
        SFNode  [in,out] metadata    NULL      [X3DMetadataObject]
        SFVec3f []      bboxCenter  0 0 0     (-∞,∞)
        SFVec3f []      bboxSize    -1 -1 -1 [0,∞) or -1 -1 -1
    }
}
```

Shape

```
/** Assign X3DAppearanceNode value (using a properly typed protoInstance) */
void setAppearance (CX3DPrototypeInstance protoInstance);

/** Return X3DGeometryNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field
named "geometry" */
void getGeometry (CX3DNode result);

/** Assign X3DGeometryNode value (using a properly typed node) to SFNode inputOutput field named "geometry" */
void setGeometry (X3DGeometryNode node);

/** Assign X3DGeometryNode value (using a properly typed protoInstance) */
void setGeometry (CX3DPrototypeInstance protoInstance);
}
```

Appearance

```
//C.3.2 Appearance
/** Appearance defines a concrete node interface that extends interface X3DAppearanceNode. */
class AFX_EXT_CLASS CAppearance : public CX3DAppearanceNode
{
    DECLARE_DYNAMIC(CAppearance);
public:
    CAppearance();
    virtual ~CAppearance();

//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return array of X3DShaderNode results array (using a properly typed node array or X3DPrototypeInstance array) from
MFNode inputOutput field named "shaders" */
    CX3DNode* getShaders ();

    /** Return number of nodes in "shaders" array */
    int getNumShaders ();

    /** Assign X3DShaderNode array (using a properly typed node array) to MFNode inputOutput field named "shaders" */
    void setShaders (CX3DShaderNode* nodes, int size);

    Appearance : X3DAppearanceNode {
        SFNode [in,out] fillProperties NULL [FillProperties]
        SFNode [in,out] lineProperties NULL [LineProperties]
        SFNode [in,out] material NULL [X3DMaterialNode]
        SFNode [in,out] metadata NULL [X3DMetadataObject]
        MFNode [in,out] shaders [] [X3DShaderNode]
        SFNode [in,out] texture NULL [X3DTextureNode]
        SFNode [in,out] textureTransform NULL [X3DTTextureTransformNode]
    }
}
```

Appearance

```
/** Assign single X3DShaderNode value (using a properly typed node) as the MFNode array for inputOutput field named "shaders" */
void setShaders (CX3DShaderNode node);

/** Assign X3DShaderNode array (using a properly typed protoInstance array) to MFNode inputOutput field named "shaders" */
void setShaders (CX3DPrototypeInstance node);

/** Assign X3DShaderNode array (using a properly typed node array) to MFNode inputOutput field named "shaders" */
void setShaders (CX3DNode* nodes, int size);

/** Return FillProperties result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named
"fillProperties" */
void getFillProperties (CX3DNode result);

/** Assign FillProperties value (using a properly typed node) to SFNode inputOutput field named "fillProperties" */
void setFillProperties (CFillProperties node);

/** Assign FillProperties value (using a properly typed protoInstance) */
void setFillProperties (CX3DPrototypeInstance protoInstance);

/** Return LineProperties result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named
"lineProperties" */
void getLineProperties (CX3DNode result);
```

Appearance

```
/** Assign LineProperties value (using a properly typed node) to SFNode inputOutput field named "lineProperties" */
void setLineProperties (CLineProperties node);

/** Assign LineProperties value (using a properly typed protoInstance) */
void setLineProperties (CX3DPrototypeInstance protoInstance);

/** Return X3DMaterialNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named
"material" */
void getMaterial (CX3DNode result);

/** Assign X3DMaterialNode value (using a properly typed node) to SFNode inputOutput field named "material" */
void setMaterial (CX3DMaterialNode node);

/** Assign X3DMaterialNode value (using a properly typed protoInstance) */
void setMaterial (CX3DPrototypeInstance protoInstance);

/** Return X3DTextureNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named
"texture" */
void getTexture (CX3DNode result);

/** Assign X3DTextureNode value (using a properly typed node) to SFNode inputOutput field named "texture" */
void setTexture (CX3DTextureNode node);
```

Appearance

```
/** Assign X3DTextureNode value (using a properly typed protoInstance) */
void setTexture (CX3DPrototypeInstance protoInstance);

/** Return X3DTextureTransformNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field
named "textureTransform" */
void getTextureTransform (CX3DNode result);

/** Assign X3DTextureTransformNode value (using a properly typed node) to SFNode inputOutput field named "textureTransform" */
void setTextureTransform (X3DTextureTransformNode node);

/** Assign X3DTextureTransformNode value (using a properly typed protoInstance) */
void setTextureTransform (CX3DPrototypeInstance protoInstance);
}
```

Material

```
//C.3.121 Material
/** Material defines a concrete node interface that extends interface X3DMaterialNode. */
class AFX_EXT_CLASS CMaterial : public CX3DMaterialNode
{
    DECLARE_DYNAMIC(CMaterial);

public:
    CMaterial();
    virtual ~CMaterial();

//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return float result [] from intensityType type inputOutput field named "ambientIntensity" */
    float getAmbientIntensity ();

    /** Assign float value [] to intensityType type inputOutput field named "ambientIntensity" */
    void setAmbientIntensity (float value);

    /** Return array of 3-tuple float results array using RGB values [0..1] from SFColor inputOutput field named "diffuseColor" */
    float* getDiffuseColor ();
    /** Assign 3-tuple float array using RGB values [0..1] to SFColor inputOutput field named "diffuseColor" */
    void setDiffuseColor (float* color);

    Material : X3DMaterialNode {
        SFFloat [in,out] ambientIntensity 0.2 [0,1]
        SFColor [in,out] diffuseColor 0.8 0.8 0.8 [0,1]
        SFColor [in,out] emissiveColor 0 0 0 [0,1]
        SFNode [in,out] metadata NULL [X3DMetadataObject]
        SFFloat [in,out] shininess 0.2 [0,1]
        SFColor [in,out] specularColor 0 0 0 [0,1]
        SFFloat [in,out] transparency 0 [0,1]
    }
}
```

Material

```
/** Return array of 3-tuple float results array using RGB values [0..1] from SFColor inputOutput field named "emissiveColor" */
float* getEmissiveColor ();

/** Assign 3-tuple float array using RGB values [0..1] to SFColor inputOutput field named "emissiveColor" */
void setEmissiveColor (float* color);

/** Return float result [] from intensityType type inputOutput field named "shininess" */
float getShininess ();

/** Assign float value [] to intensityType type inputOutput field named "shininess" */
void setShininess (float value);

/** Return array of 3-tuple float results array using RGB values [0..1] from SFColor inputOutput field named "specularColor" */
float* getSpecularColor ();

/** Assign 3-tuple float array using RGB values [0..1] to SFColor inputOutput field named "specularColor" */
void setSpecularColor (float* color);

/** Return float result [] from intensityType type inputOutput field named "transparency" */
float getTransparency ();

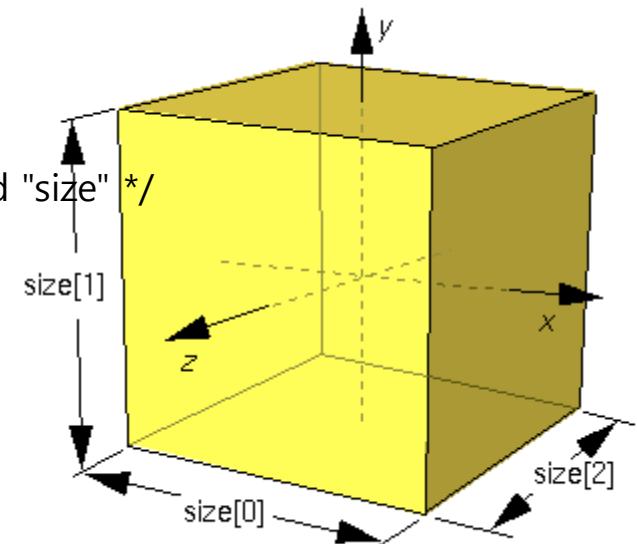
/** Assign float value [] to intensityType type inputOutput field named "transparency" */
void setTransparency (float value);

}
```

Box

```
//C.3.16 Box
/** Box defines a concrete node interface that extends interface X3DGeometryNode. */
class AFX_EXT_CLASS CBox : public CX3DGeometryNode
{
    DECLARE_DYNAMIC(CBox);
public:
    CBox();
    virtual ~CBox();
//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();
    /** Return array of 3-tuple float results array [] from SFVec3f initializeOnly field named "size" */
    float* getSize ();
    /** Assign 3-tuple float array [] to SFVec3f initializeOnly field named "size" */
    void setSize (float* value);
    /** Return boolean result from SFBool initializeOnly field named "solid" */
    bool getSolid ();
    /** Assign boolean value to SFBool initializeOnly field named "solid" */
    void setSolid (bool value);
}
```

```
Box : X3DGeometryNode {
    SFNode [in,out] metadata NULL [X3DMetadataObject]
    SFVec3f [] size 2 2 2 (0,∞)
    SFBool [] solid TRUE
}
```



Cone

```
//C.3.40 Cone
/** Cone defines a concrete node interface that extends interface X3DGeometryNode. */

class AFX_EXT_CLASS CCone : public CX3DGeometryNode
{
    DECLARE_DYNAMIC(CCone);
public:
    CCone();
    virtual ~CCone();
//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return float result [] from type initializeOnly field named "bottomRadius" */
    float getBottomRadius ();

    /** Assign float value [] to type initializeOnly field named "bottomRadius" */
    void setBottomRadius (float value);

    Cone : X3DGeometryNode {
        SFNode [in,out] metadata NULL [X3DMetadataObject]
        SFBool [] bottom TRUE
        SFFloat [] bottomRadius 1 (0,∞)
        SFFloat [] height 2 (0,∞)
        SFBool [] side TRUE
        SFBool [] solid TRUE
    }
}
```

Cone

```
/** Return float result [] from type initializeOnly field named "height" */
float getHeight ();

/** Assign float value [] to type initializeOnly field named "height" */
void setHeight (float value);

/** Return boolean result from SFBool initializeOnly field named "side" */
bool getSide ();

/** Assign boolean value to SFBool initializeOnly field named "side" */
void setSide (bool value);

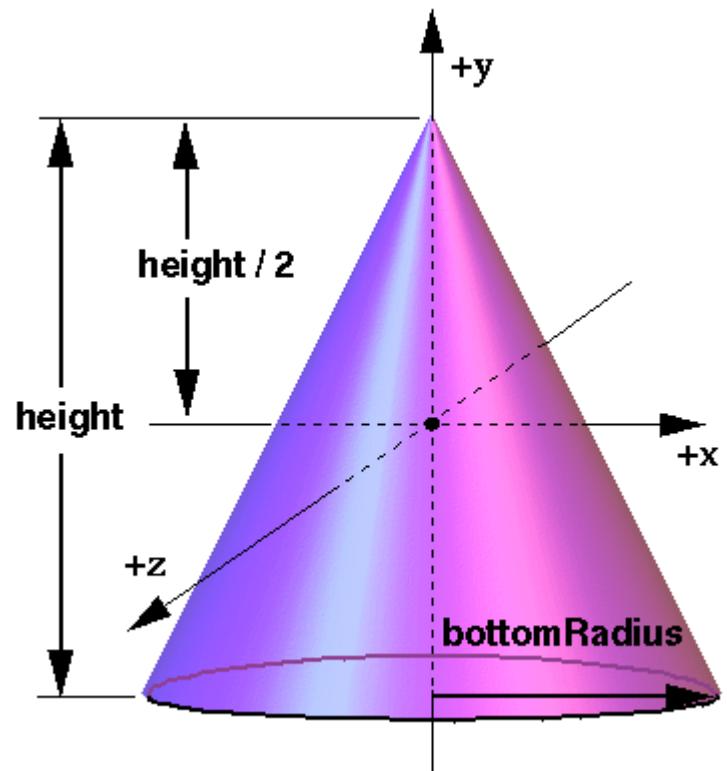
/** Return boolean result from SFBool initializeOnly field named "bottom" */
bool getBottom ();

/** Assign boolean value to SFBool initializeOnly field named "bottom" */
void setBottom (bool value);

/** Return boolean result from SFBool initializeOnly field named "solid" */
bool getSolid ();

/** Assign boolean value to SFBool initializeOnly field named "solid" */
void setSolid (bool value);

}
```



Cylinder

```
//C.3.52 Cylinder
/** Cylinder defines a concrete node interface that extends interface X3DGeometryNode. */

class AFX_EXT_CLASS CCylinder : public CX3DGeometryNode
{
    DECLARE_DYNAMIC(CCylinder);
public:
    CCylinder();
    virtual ~CCylinder();
//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return boolean result from SFBool initializeOnly field named "bottom" */
    bool getBottom ();

    /** Assign boolean value to SFBool initializeOnly field named "bottom" */
    void setBottom (bool value);

    /** Return float result [] from type initializeOnly field named "height" */
    float getHeight ();
}

Cylinder : X3DGeometryNode {
    SFNode  [in,out] metadata NULL [X3DMetadataObject]
    SFBool   []      bottom   TRUE
    SFFloat  []      height   2     (0,∞)
    SFFloat  []      radius   1     (0,∞)
    SFBool   []      side    TRUE
    SFBool   []      solid   TRUE
    SFBool   []      top     TRUE
}
```

Cylinder

```
/** Assign float value [] to type initializeOnly field named "height" */
void setHeight (float value);

/** Return float result [] from type initializeOnly field named "radius" */
float getRadius ();

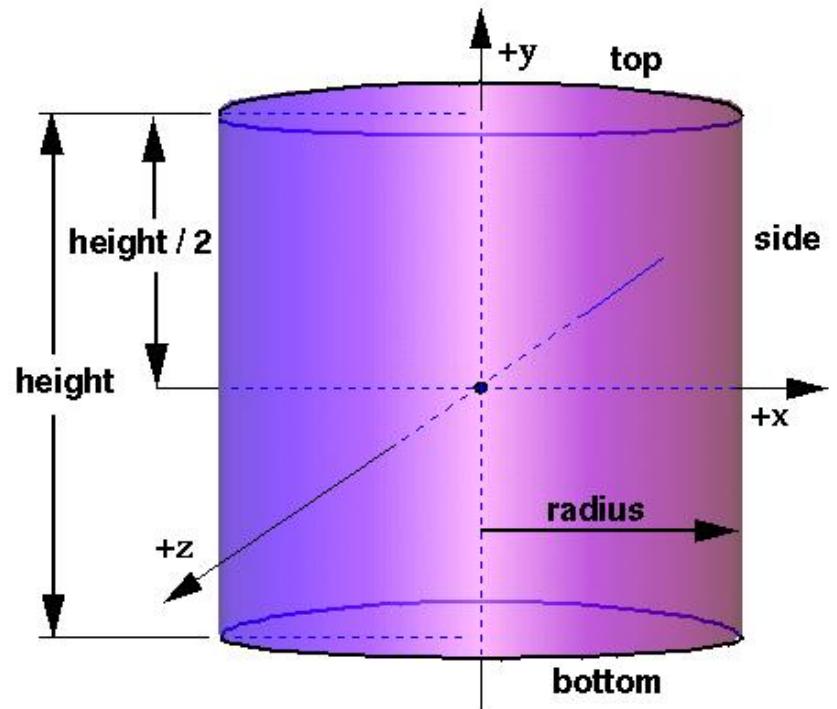
/** Assign float value [] to type initializeOnly field named "radius" */
void setRadius (float value);

/** Return boolean result from SFBool initializeOnly field named "side" */
bool getSide ();

/** Assign boolean value to SFBool initializeOnly field named "side" */
void setSide (bool value);

/** Return boolean result from SFBool initializeOnly field named "top" */
bool getTop ();
/** Assign boolean value to SFBool initializeOnly field named "top" */
void setTop (bool value);

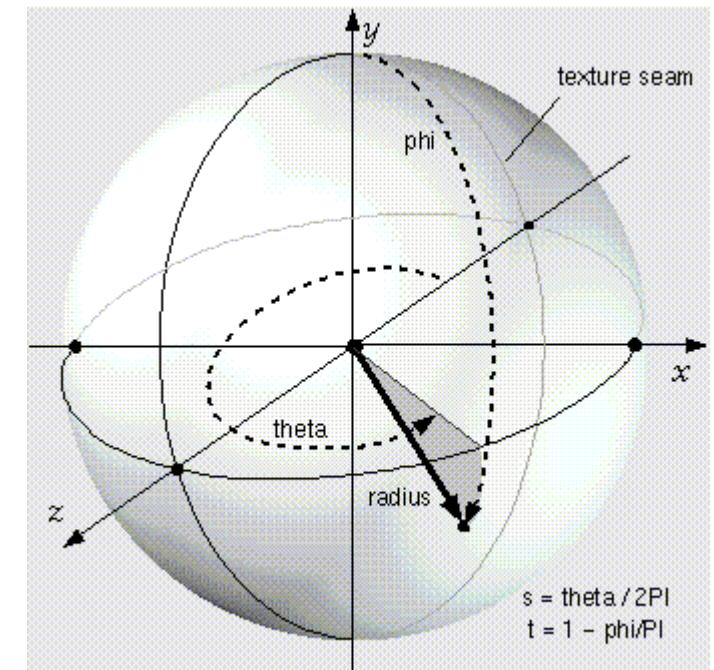
/** Return boolean result from SFBool initializeOnly field named "solid" */
bool getSolid ();
/** Assign boolean value to SFBool initializeOnly field named "solid" */
void setSolid (bool value);
```



Sphere

```
//C.3.205 Sphere
/** Sphere defines a concrete node interface that extends interface X3DGeometryNode. */
class AFX_EXT_CLASS CSphere : public CX3DGeometryNode
{
    DECLARE_DYNAMIC(CSphere);
public:
    CSphere();
    virtual ~CSphere();
//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();
    /** Return float result [] from type initializeOnly field named "radius" */
    float getRadius ();
    /** Assign float value [] to type initializeOnly field named "radius" */
    void setRadius (float value);
    /** Return boolean result from SFBool initializeOnly field named "solid" */
    bool getSolid ();
    /** Assign boolean value to SFBool initializeOnly field named "solid" */
    void setSolid (bool value);
}
```

```
Sphere : X3DGeometryNode {
    SFNode  [in,out] metadata NULL [X3DMetadataObject]
    SFFloat []      radius   1     (0,∞)
    SFBool   []      solid    TRUE
```



IndexedFaceSet

```
//C.3.98 IndexedFaceSet
/** IndexedFaceSet defines a concrete node interface that extends interface X3DComposedGeometryNode. */

class AFX_EXT_CLASS CIndexedFaceSet : public CX3DComposedGeometryNode
{
    DECLARE_DYNAMIC(CIndexedFaceSet);

public:
    CIndexedFaceSet();
    virtual ~CIndexedFaceSet();

//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Assign MFInt32 value using RGB values [0..1] to
        MFInt32 inputOnly field named "set_colorIndex" */
    void setColorIndex (int32_t* colors, int size);

    IndexedFaceSet : X3DComposedGeometryNode {
        MFInt32 [in]      set_colorIndex
        MFInt32 [in]      set_coordIndex
        MFInt32 [in]      set_normalIndex
        MFInt32 [in]      set_texCoordIndex
        MFNode  [in,out]  attrib          [] [X3DVertexAttributeNode]
        SFNode   [in,out]  color           NULL [X3DColorNode]
        SFNode   [in,out]  coord           NULL [X3DCoordinateNode]
        SFNode   [in,out]  fogCoord       NULL [FogCoordinate]
        SFNode   [in,out]  metadata        NULL [X3DMetadataObject]
        SFNode   [in,out]  normal          NULL [X3DNormalNode]
        SFNode   [in,out]  texCoord        NULL [X3DTextureCoordinateNode]
        SFBool   []         ccw             TRUE
        MFInt32 []         colorIndex      [] [0,∞) or -1
        SFBool   []         colorPerVertex TRUE
        SFBool   []         convex          TRUE
        MFInt32 []         coordIndex     [] [0,∞) or -1
        SFFloat  []         creaseAngle    0 [0,∞)
        MFInt32 []         normalIndex    [] [0,∞) or -1
        SFBool   []         normalPerVertex TRUE
        SFBool   []         solid           TRUE
        MFInt32 []         texCoordIndex  [] [-1,∞)
```

IndexedFaceSet

```
/** Assign single SFInt32 value using RGB values [0..1] as the MFInt32 array for inputOnly field named "set_colorIndex" */
void setColorIndex (int32_t color);

/** Assign MFInt32 value [] to MFInt32 inputOnly field named "set_coordIndex" */
void setCoordIndex (int32_t* values, int size);

/** Assign single SFInt32 value [] as the MFInt32 array for inputOnly field named "set_coordIndex" */
void setCoordIndex (int32_t value);

/** Assign MFInt32 value [] to MFInt32 inputOnly field named "set_normalIndex" */
void setNormalIndex (int32_t* values, int size);

/** Assign single SFInt32 value [] as the MFInt32 array for inputOnly field named "set_normalIndex" */
void setNormalIndex (int32_t value);

/** Assign MFInt32 value [] to MFInt32 inputOnly field named "set_texCoordIndex" */
void setTexCoordIndex (int32_t* values, int size);

/** Assign single SFInt32 value [] as the MFInt32 array for inputOnly field named "set_texCoordIndex" */
void setTexCoordIndex (int32_t value);
```

IndexedFaceSet

```
/** Return boolean result from SFBool initializeOnly field named "convex" */
bool getConvex ();

/** Assign boolean value to SFBool initializeOnly field named "convex" */
void setConvex (bool value);

/** Return float result in radians from type initializeOnly field named "creaseAngle" */
float getCreaseAngle ();

/** Assign float value in radians to type initializeOnly field named "creaseAngle" */
void setCreaseAngle (float angle);

/** Return MFInt32 result using RGB values [0..1] from MFInt32 initializeOnly field named "colorIndex" */
int32_t* getColorIndex ();

/** Return number of primitive values in "colorIndex" array */
int getNumColorIndex ();

/** Return MFInt32 result [] from MFInt32 initializeOnly field named "coordIndex" */
int32_t* getCoordIndex ();
```

IndexedFaceSet

```
/** Return number of primitive values in "coordIndex" array */
int getNumCoordIndex ();

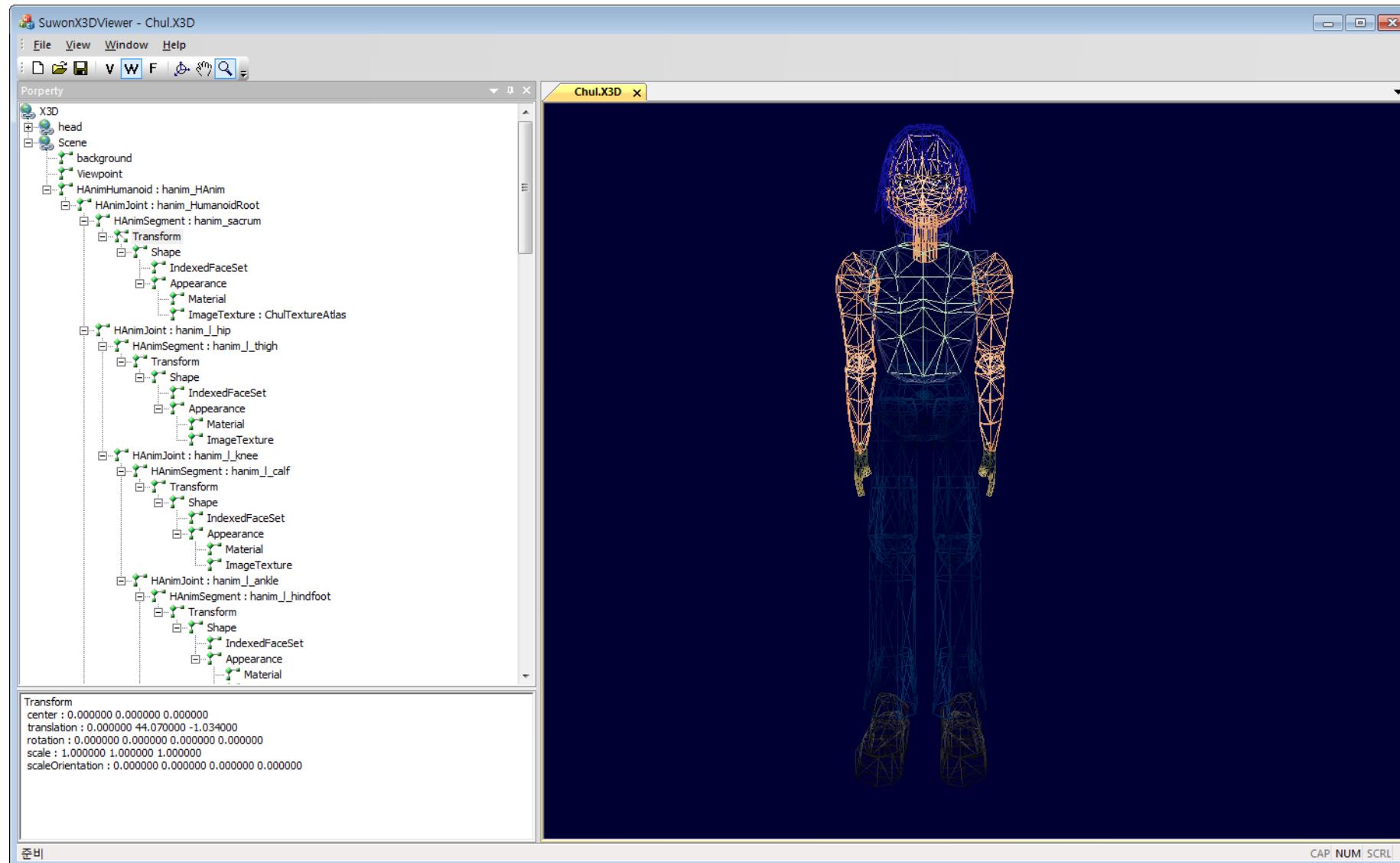
/** Return MFInt32 result [] from MFInt32 initializeOnly field named "normalIndex" */
int32_t* getNormalIndex ();

/** Return number of primitive values in "normalIndex" array */
int getNumNormalIndex ();

/** Return MFInt32 result [] from MFInt32 initializeOnly field named "texCoordIndex" */
int32_t* getTexCoordIndex ();

/** Return number of primitive values in "texCoordIndex" array */
int getNumTexCoordIndex ();
}
```

IndexedFaceSet Sample



X3DComposedGeometryNode (IndexedFaceSet Public Node)

```
//B.2.9 X3DComposedGeometryNode
/** X3DComposedGeometryNode defines an abstract node interface that extends interfaces X3DNode.
 * Composed geometry nodes produce renderable geometry, can contain Color Coordinate Normal TextureCoordinate, and are contained by a Shape node. */

class AFX_EXT_CLASS CX3DComposedGeometryNode : public CX3DGeometryNode
{
    DECLARE_DYNAMIC(CX3DComposedGeometryNode);

public:
    CX3DComposedGeometryNode();
    virtual ~CX3DComposedGeometryNode();

//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return bool result from SFBool initializeOnly field named "ccw" */
    bool getCcw();
```

X3DComposedGeometryNode (IndexedFaceSet Public Node)

```
/** Assign bool value to SFBool initializeOnly field named "ccw" */
void setCcw (bool value);

/** Return bool result from SFBool initializeOnly field named "colorPerVertex" */
bool getColorPerVertex ();

/** Assign bool value to SFBool initializeOnly field named "colorPerVertex" */
void setColorPerVertex (bool color);

/** Return bool result from SFBool initializeOnly field named "normalPerVertex" */
bool getNormalPerVertex ();

/** Assign bool value to SFBool initializeOnly field named "normalPerVertex" */
void setNormalPerVertex (bool value);

/** Return bool result from SFBool initializeOnly field named "solid" */
bool getSolid ();

/** Assign bool value to SFBool initializeOnly field named "solid" */
void setSolid (bool value);
```

X3DComposedGeometryNode (IndexedFaceSet Public Node)

```
/** Return array of X3DVertexAttributeNode results array (using a properly typed node array or X3DPrototypeInstance array) from
MFNode inputOutput field named "attrib" */
CX3DNode* getAttrib ();

/** Return number of nodes in "attrib" array */
int getNumAttrib ();

/** Assign X3DVertexAttributeNode array (using a properly typed node array) to MFNode inputOutput field named "attrib" */
void setAttrib (X3DVertexAttributeNode* nodes);

/** Assign single X3DVertexAttributeNode value (using a properly typed node) as the MFNode array for inputOutput field named
"attrib" */
void setAttrib (X3DVertexAttributeNode node);

/** Assign X3DVertexAttributeNode array (using a properly typed protoInstance array) to MFNode inputOutput field named "attrib" */
void setAttrib (X3DPrototypeInstance node);

/** Assign X3DVertexAttributeNode array (using a properly typed node array) to MFNode inputOutput field named "attrib" */
void setAttrib (CX3DNode* nodes);

/** Return X3DColorNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named
"color" */
void getColor (CX3DNode result);
```

X3DComposedGeometryNode (IndexedFaceSet Public Node)

```
/** Assign X3DColorNode value (using a properly typed node) to SFNode inputOutput field named "color" */
void setColor (CX3DCOLORNODE color);

/** Assign X3DCOLORNODE value (using a properly typed protoInstance) */
void setColor (CX3DPrototypeInstance protoInstance);

/** Return X3DCoordinateNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named
"coord" */
//CX3DNode* getCoord ();
void getCoord (CX3DNode result);

/** Assign X3DCoordinateNode value (using a properly typed node) to SFNode inputOutput field named "coord" */
void setCoord (CX3DCoordinateNode node);

/** Assign X3DCoordinateNode value (using a properly typed protoInstance) */
void setCoord (CX3DPrototypeInstance protoInstance);

/** Return FogCoordinate result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named
"fogCoord" */
void getFogCoord (CX3DNode result);

/** Assign FogCoordinate value (using a properly typed node) to SFNode inputOutput field named "fogCoord" */
void setFogCoord (CFogCoordinate node);
```

X3DComposedGeometryNode (IndexedFaceSet Public Node)

```
/** Assign FogCoordinate value (using a properly typed protoInstance) */
void setFogCoord (CX3DPrototypeInstance protoInstance);

/** Return X3DNormalNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named "normal" */
//CX3DNode* getNormal ();
void getNormal (CX3DNode result);

/** Assign X3DNormalNode value (using a properly typed node) to SFNode inputOutput field named "normal" */
void setNormal (X3DNormalNode node);

/** Assign X3DNormalNode value (using a properly typed protoInstance) */
void setNormal (CX3DPrototypeInstance protoInstance);

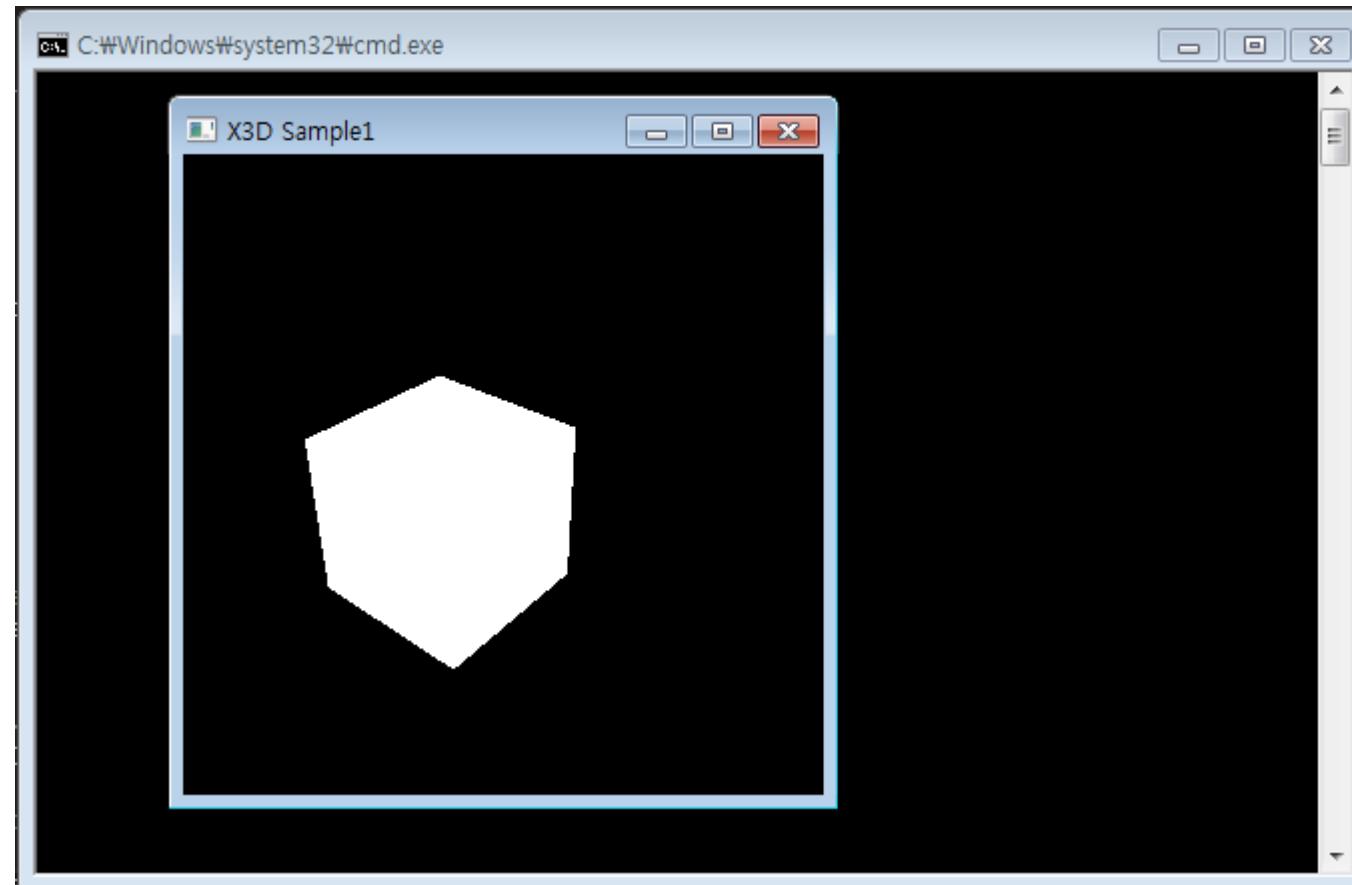
/** Return X3DTextureCoordinateNode result (using a properly typed node or X3DPrototypeInstance) from SFNode inputOutput field named "texCoord" */
void getTexCoord (CX3DNode result);

/** Assign X3DTextureCoordinateNode value (using a properly typed node) to SFNode inputOutput field named "texCoord" */
void setTexCoord (X3DTextureCoordinateNode node);

/** Assign X3DTextureCoordinateNode value (using a properly typed protoInstance) */
void setTexCoord (CX3DPrototypeInstance protoInstance);
```

X3D C++ Binding Example

- ISO/IEC WD 19777-3 X3D language binding: C Annex C
- ISO/IEC WD 19777-4 X3D language binding: C++ Annex C



X3D C++ Binding Sample Source (Win32)

```
#include <glut.h>
#include "..\X3DLib\X3DLib.h"           → GLUT Library include
                                         → X3D C++ Library include

#ifndef _DEBUG
#define new DEBUG_NEW
#endif

CWinApp theApp;
using namespace std;
CX3DScene* m_pScene = NULL;

void changeSize(int w, int h) {

    if(h == 0)
        h = 1;

    float ratio = 1.0* w / h;

    // Reset the coordinate system before modifying
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h);
}
```

X3D C++ Binding Sample Source (Win32)

```
• // Set the correct perspective.  
•     gluPerspective(45, ratio, 1, 1000);  
•     glMatrixMode(GL_MODELVIEW);  
•     glLoadIdentity();  
•     gluLookAt(5.0, 5.0, 5.0,  
•               0.0, 0.0, -1.0,  
•               0.0f, 1.0f, 0.0f);  
• }  
  
• void DrawNode(CX3DNode * pNode)  
• {  
•     if(pNode==NULL)  
•         return;  
  
•     ::glPushMatrix();  
•     if (pNode->isType(NODE_SHAPE))  
•         ((CX3DShapeNode*)pNode)->geometry->Draw();  
  
•     ::glPopMatrix();  
• }  
  
• void renderScene(void)  
• {  
•     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
•     glPushMatrix();
```

X3D C++ Binding Sample Source (Win32)

```
int nCount = m_pScene->m_Objects.GetCount();
for(int i=0; i<nCount; i++)
{
    CX3DNode* pChild =(CX3DNode*)(m_pScene->m_Objects).GetAt(i);
    if (pChild)
        DrawNode(pChild);
}

glPopMatrix();
glutSwapBuffers();
}

void initialize(void)
{
    m_pScene = new CX3DScene();

    CX3DShapeNode* shape = new CX3DShapeNode;
    CBox* box = new CBox;
    shape->setGeometry((CX3DGeometryNode*)box);

    m_pScene->AddNode(shape, NULL);
}
```

X3D C++ Binding Sample Source (Win32)

```
int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;

    HMODULE hModule = ::GetModuleHandle(NULL);

    if (hModule != NULL)
    {
        if (!AfxWinInit(hModule, NULL, ::GetCommandLine(), 0))
        {
            _tprintf(_T("심각한 오류: MFC를 초기화하지 못했습니다.\n"));
            nRetCode = 1;
        }
        else
        {
            glutInit(&argc, argv);
            glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
            glutInitWindowPosition(100,100);
            glutInitWindowSize(320,320);
            glutCreateWindow("X3D Sample1");

            initialize();

            glutDisplayFunc(renderScene);
            glutIdleFunc(renderScene);
            glutReshapeFunc(changeSize);
        }
    }
}
```

X3D C++ Binding Sample Source (Win32)

```
        glutMainLoop();
    }
else
{
    _tprintf(_T("심각한 오류: GetModuleHandle 실패\n"));
    nRetCode = 1;
}

return nRetCode;
}
```

X3D C++ Binding Example Source (Win32)

```
#include <glut.h>
#include "..\X3DLib\X3DLib.h"

#ifndef _DEBUG
#define new DEBUG_NEW
#endif

CWinApp theApp;
using namespace std;

CX3DScene* m_pScene = NULL;

void changeSize(int w, int h) {
    if(h == 0)
        h = 1;

    float ratio = 1.0* w / h;

    // Reset the coordinate system before modifying
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h);

    // Set the correct perspective.
    gluPerspective(45,ratio,1,1000);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(5.0,5.0,5.0,
              0.0,0.0,-1.0,
              0.0F,1.0F,0.0F);
}

}
```

Include GLUT Library

Include X3D C++ Library

Window size changed

X3D C++ Binding Sample Source (Win32)

```
void DrawNode(CX3DNode *pNode)
{
    if(pNode==NULL)
        return;

    ::glPushMatrix();
    if (pNode->isType(NODE_SHAPE))
        ((CX3DShapeNode*)pNode)->geometry->Draw();
    ::glPopMatrix();
}

void renderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPushMatrix();

    int nCount = m_pScene->m_Objects.GetCount();
    for(int i=0; i<nCount; i++)
    {
        CX3DNode* pChild = (CX3DNode*)(m_pScene->m_Objects).GetAt(i);
        if (pChild)
            DrawNode(pChild);
    }

    glPopMatrix();
    glutSwapBuffers();
}

void initialize(void)
{
    m_pScene = new CX3DScene();

    CX3DShapeNode* shape = new CX3DShapeNode;
    CBox* box = new CBox;
    shape->setGeometry((CX3DGeometryNode*)box);

    m_pScene->AddNode(shape, NULL);
}
```

Draw a Box

Draw Child Nodes

Create a Box

X3D C++ Binding Sample Source (Win32)

```
int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;

    HMODULE hModule = ::GetModuleHandle(NULL);

    if (hModule != NULL)
    {
        if (!AfxWinInit(hModule, NULL, ::GetCommandLine(), 0))
        {
            _tprintf(_T("심각한 오류: MFC를 초기화하지 못했습니다.\n"));
            nRetCode = 1;
        }
        else
        {
            glutInit(&argc, argv);
            glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
            glutInitWindowPosition(100,100);
            glutInitWindowSize(320,320);
            glutCreateWindow("X3D Sample1");

            initialize();

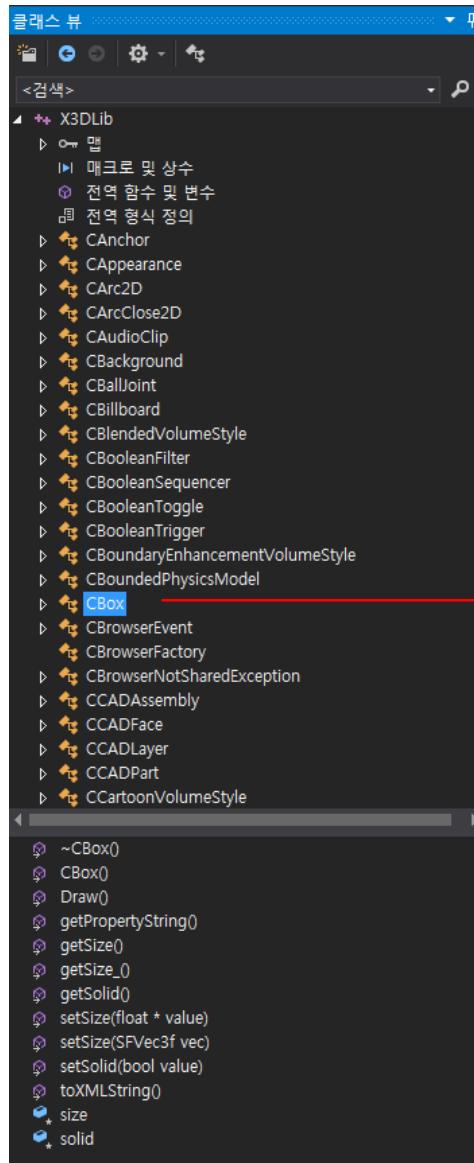
            glutDisplayFunc(renderScene);
            glutIdleFunc(renderScene);
            glutReshapeFunc(changeSize);

            glutMainLoop();
        }
    }
    else
    {
        _tprintf(_T("심각한 오류: GetModuleHandle 실패\n"));
        nRetCode = 1;
    }

    return nRetCode;
}
```

Create a Window and initialization

X3D C++ Library



Box Node CBox Class

CBox Class function

```
//C.3.16 Box
/** Box defines a concrete node interface that extends interface X3DGeometryNode. */

class AFX_EXT_CLASS CBox : public CX3DGeometryNode
{
    DECLARE_DYNAMIC(CBox);

public:
    CBox();
    virtual ~CBox();

//Implementation
public:
    virtual void Draw();
    virtual CString toXMLString();
    virtual CString getPropertyString();

    /** Return array of 3-tuple float results array [] from SFVec3f initializeOnly field named "size" */
    float* getSize();

    /** Assign 3-tuple float array [] to SFVec3f initializeOnly field named "size" */
    void setSize (float* value);

    /** Return boolean result from SFBool initializeOnly field named "solid" */
    bool getSolid();

    /** Assign boolean value to SFBool initializeOnly field named "solid" */
    void setSolid (bool value);

//Attributes
protected:
    SFVec3f    size;
    bool       solid;
};
```

X3D C++ Library

- ◀ + X3DLib
- ▷ ↵ 맵
 - ▶ 매크로 및 상수
 - ∅ 전역 함수 및 변수
 - ▣ 전역 형식 정의
- ▷ CAnchor
- ▷ CAppearance
- ▷ CArc2D
- ▷ CArcClose2D
- ▷ CAudioClip
- ▷ CBackground
- ▷ CBallJoint
- ▷ CBillboard
- ▷ CBlenDEDVolumeStyle
- ▷ CBooleanFilter
- ▷ CBooleanSequencer
- ▷ CBooleanToggle
- ▷ CBooleanTrigger
- ▷ CBoundaryEnhancementVolumeStyle
- ▷ CBoundedPhysicsModel
- ▷ CBox
- ▷ CBrowserEvent
- ▷ CBrowserFactory
- ▷ CBrowserNotSharedException
- ▷ CCADAssembly
- ▷ CCADFace
- ▷ CCADLayer
- ▷ CCADPart
- ▷ CCartoonVolumeStyle
- ▷ CCircle2D
- ▷ CClipPlane
- ▷ CCollidableOffset
- ▷ CCollidableShape
- ▷ CCollision
- ▷ CCollisionCollection
- ▷ CCollisionSensor

- ▷ CCollisionSpace
- ▷ CColor
- ▷ CColorChaser
- ▷ CColorDamper
- ▷ CColorInterpolator
- ▷ CColorRGBA
- ▷ CComponent
- ▷ CComponentInfo
- ▷ CComposedCubeMapTexture
- ▷ CComposedShader
- ▷ CComposedTexture3D
- ▷ CComposedVolumeStyle
- ▷ CCone
- ▷ CConeEmitter
- ▷ Cconnect
- ▷ CConnectionException
- ▷ CContact
- ▷ CContour2D
- ▷ CContourPolyline2D
- ▷ CCoordinate
- ▷ CCoordinateChaser
- ▷ CCoordinateDamper
- ▷ CCoordinateDouble
- ▷ CCoordinateInterpolator
- ▷ CCoordinateInterpolator2D
- ▷ CCylinder
- ▷ CCylinderSensor
- ▷ CDirectionalLight
- ▷ CDISEntityManager
- ▷ CDISEntityTypeMapping
- ▷ CDisk2D
- ▷ CDoubleAxisHingeJoint
- ▷ CEaseInEaseOut
- ▷ CEdgeEnhancementVolumeStyle
- ▷ CElevationGrid
- ▷ CEspduTransform

- ▷ CEventObject
- ▷ CExplosionEmitter
- ▷ CEXPORT
- ▷ CExternProtoDeclare
- ▷ CExtrusion
- ▷ Cfield
- ▷ CFieldValue
- ▷ CFillProperties
- ▷ CFloatVertexAttribute
- ▷ CFog
- ▷ CFogCoordinate
- ▷ CFontStyle
- ▷ CForcePhysicsModel
- ▷ CGeneratedCubeMapTexture
- ▷ CGeoCoordinate
- ▷ CGeoElevationGrid
- ▷ CGeoLocation
- ▷ CGeoLOD
- ▷ CGeoMetadata
- ▷ CGeoOrigin
- ▷ CGeoPositionInterpolator
- ▷ CGeoProximitySensor
- ▷ CGeoTouchSensor
- ▷ CGeoTransform
- ▷ CGeoViewpoint
- ▷ CGroup
- ▷ CHAnimDisplacer
- ▷ CHAnimHumanoid
- ▷ CHAnimJoint
- ▷ CHAnimSegment
- ▷ CHAnimSite
- ▷ Chead
- ▷ CImageCubeMapTexture
- ▷ CImageTexture
- ▷ CImageTexture3D
- ▷ CIMPORT

X3D C++ Library

- ▷ CImportedNodeException
- ▷ CIndexedFaceSet
- ▷ CIndexedLineSet
- ▷ CIndexedQuadSet
- ▷ CIndexedTriangleFanSet
- ▷ CIndexedTriangleSet
- ▷ CIndexedTriangleStripSet
- ▷ CInline
- ▷ CInsufficientCapabilitiesException
- ▷ CIntegerSequencer
- ▷ CIntegerTrigger
- ▷ CInvalidBrowserException
- ▷ CInvalidDocumentException
- ▷ CInvalidExecutionContextException
- ▷ CInvalidFieldException
- ▷ CInvalidFieldValueException
- ▷ CInvalidNodeException
- ▷ CInvalidOperationTimingException
- ▷ CInvalidProtoException
- ▷ CInvalidRouteException
- ▷ CInvalidURLException
- ▷ CInvalidX3DEception
- ▷ CIS
- ▷ CIsoSurfaceVolumeData
- ▷ CKeySensor
- ▷ CLayer
- ▷ CLayerSet
- ▷ CLayout
- ▷ CLayoutGroup
- ▷ CLayoutLayer
- ▷ CLinePickSensor
- ▷ CLineProperties
- ▷ CLineSet
- ▷ CLoadSensor
- ▷ CLocalFog
- ▷ CLOD

- ▷ CMaterial
- ▷ CMatrix3VertexAttribute
- ▷ CMatrix4VertexAttribute
- ▷ Cmeta
- ▷ CMetadataBoolean
- ▷ CMetadataDouble
- ▷ CMetadataFloat
- ▷ CMetadataInteger
- ▷ CMetadataSet
- ▷ CMetadataString
- ▷ CMFBool
- ▷ CMFColor
- ▷ CMFColorRGBA
- ▷ CMFDouble
- ▷ CMFFloat
- ▷ CMField
- ▷ CMFImage
- ▷ CMFInt32
- ▷ CMFMatrix3d
- ▷ CMFMatrix3f
- ▷ CMFMatrix4d
- ▷ CMFMatrix4f
- ▷ CMFRotation
- ▷ CMFString
- ▷ CMFTime
- ▷ CMFVec2d
- ▷ CMFVec2f
- ▷ CMFVec3d
- ▷ CMFVec3f
- ▷ CMFVec4d
- ▷ CMFVec4f
- ▷ CMotorJoint
- ▷ CMovieTexture
- ▷ CMultiTexture
- ▷ CMultiTextureCoordinate
- ▷ CMultiTextureTransform

- ▷ CNavigationInfo
- ▷ CNodeInUseException
- ▷ CNodeUnavailableException
- ▷ CNormal
- ▷ CNormalInterpolator
- ▷ CNoSUCHBrowserException
- ▷ CNurbsCurve
- ▷ CNurbsCurve2D
- ▷ CNurbsOrientationInterpolator
- ▷ CNurbsPatchSurface
- ▷ CNurbsPositionInterpolator
- ▷ CNurbsSet
- ▷ CNurbsSurfaceInterpolator
- ▷ CNurbsSweptSurface
- ▷ CNurbsSwungSurface
- ▷ CNurbsTextureCoordinate
- ▷ CNurbsTrimmedSurface
- ▷ COpacityMapVolumeStyle
- ▷ COrientationChaser
- ▷ COrientationDamper
- ▷ COrientationInterpolator
- ▷ COrthoViewpoint
- ▷ CPackagedShader
- ▷ CParticleSystem
- ▷ CPickableGroup
- ▷ CPixelTexture
- ▷ CPixelTexture3D
- ▷ CPlaneSensor
- ▷ CPointEmitter
- ▷ CPointLight
- ▷ CPointPickSensor
- ▷ CPointSet
- ▷ CPolyline2D
- ▷ CPolylineEmitter
- ▷ CPolypoint2D
- ▷ CPositionChaser

X3D C++ Library

- ▷ `CPositionChaser2D`
- ▷ `CPositionDamper`
- ▷ `CPositionDamper2D`
- ▷ `CPositionInterpolator`
- ▷ `CPositionInterpolator2D`
- ▷ `CPrimitivePickSensor`
- ▷ `CProfileInfo`
- ▷ `CProgramShader`
- ▷ `CProjectionVolumeStyle`
- ▷ `CProtoBody`
- ▷ `CProtoDeclare`
- ▷ `CProtoInstance`
- ▷ `CProtoInterface`
- ▷ `CProximitySensor`
- ▷ `CQuadSet`
- ▷ `CReceiverPdu`
- ▷ `CRectangle2D`
- ▷ `CRigidBody`
- ▷ `CRigidBodyCollection`
- ▷ `CRROUTE`
- ▷ `CScalarChaser`
- ▷ `CScalarDamper`
- ▷ `CScalarInterpolator`
- ▷ `CScene`
- ▷ `CSceneGraphStructureStatement`
- ▷ `CScreenFontStyle`
- ▷ `CScreenGroup`
- ▷ `CScript`
- ▷ `CSegmentedVolumeData`
- ▷ `CSFBool`
- ▷ `CSFColor`
- ▷ `CSFColorRGBA`
- ▷ `CSFDouble`
- ▷ `CSFFloat`
- ▷ `CSFImage`
- ▷ `CSFInt32`

- ▷ `CSFMatrix3d`
- ▷ `CSFMatrix3f`
- ▷ `CSFMatrix4d`
- ▷ `CSFMatrix4f`
- ▷ `CSFRotation`
- ▷ `CSFString`
- ▷ `CSFTime`
- ▷ `CSFVec2d`
- ▷ `CSFVec2f`
- ▷ `CSFVec3d`
- ▷ `CSFVec3f`
- ▷ `CSFVec4d`
- ▷ `CSFVec4f`
- ▷ `CShadedVolumeStyle`
- ▷ `CShaderPart`
- ▷ `CShaderProgram`
- ▷ `CShape`
- ▷ `CSignalPdu`
- ▷ `CSilhouetteEnhancementVolumeStyle`
- ▷ `CSingleAxisHingeJoint`
- ▷ `CSliderJoint`
- ▷ `CSound`
- ▷ `CSphere`
- ▷ `CSphereSensor`
- ▷ `CSplinePositionInterpolator`
- ▷ `CSplinePositionInterpolator2D`
- ▷ `CSplineScalarInterpolator`
- ▷ `CSpotLight`
- ▷ `CSquadOrientationInterpolator`
- ▷ `CStaticGroup`
- ▷ `CStdioFileEx`
- ▷ `CStringSensor`
- ▷ `CSurfaceEmitter`
- ▷ `CSwitch`
- ▷ `CTexCoordChaser2D`
- ▷ `CTexCoordDamper2D`

- ▷ `CText`
- ▷ `CTextureBackground`
- ▷ `CTextureCoordinate`
- ▷ `CTextureCoordinate3D`
- ▷ `CTextureCoordinate4D`
- ▷ `CTextureCoordinateGenerator`
- ▷ `CTextureProperties`
- ▷ `CTextureTransform`
- ▷ `CTextureTransform3D`
- ▷ `CTextureTransformMatrix3D`
- ▷ `CTimeSensor`
- ▷ `CTimeTrigger`
- ▷ `CToneMappedVolumeStyle`
- ▷ `CTouchSensor`
- ▷ `CTransform`
- ▷ `CTransformSensor`
- ▷ `CTransmitterPdu`
- ▷ `CTriangleFanSet`
- ▷ `CTriangleSet`
- ▷ `CTriangleSet2D`
- ▷ `CTriangleStripSet`
- ▷ `CTwoSidedMaterial`
- ▷ `Cunit`
- ▷ `CUniversalJoint`
- ▷ `CURLUnavailableException`
- ▷ `CViewpoint`
- ▷ `CViewpointGroup`
- ▷ `CViewport`
- ▷ `CVisibilitySensor`
- ▷ `CVolumeData`
- ▷ `CVolumeEmitter`
- ▷ `CVolumePickSensor`
- ▷ `CWindPhysicsModel`
- ▷ `CWorldInfo`
- ▷ `CX3D`
- ▷ `CX3DAppearanceChildNode`

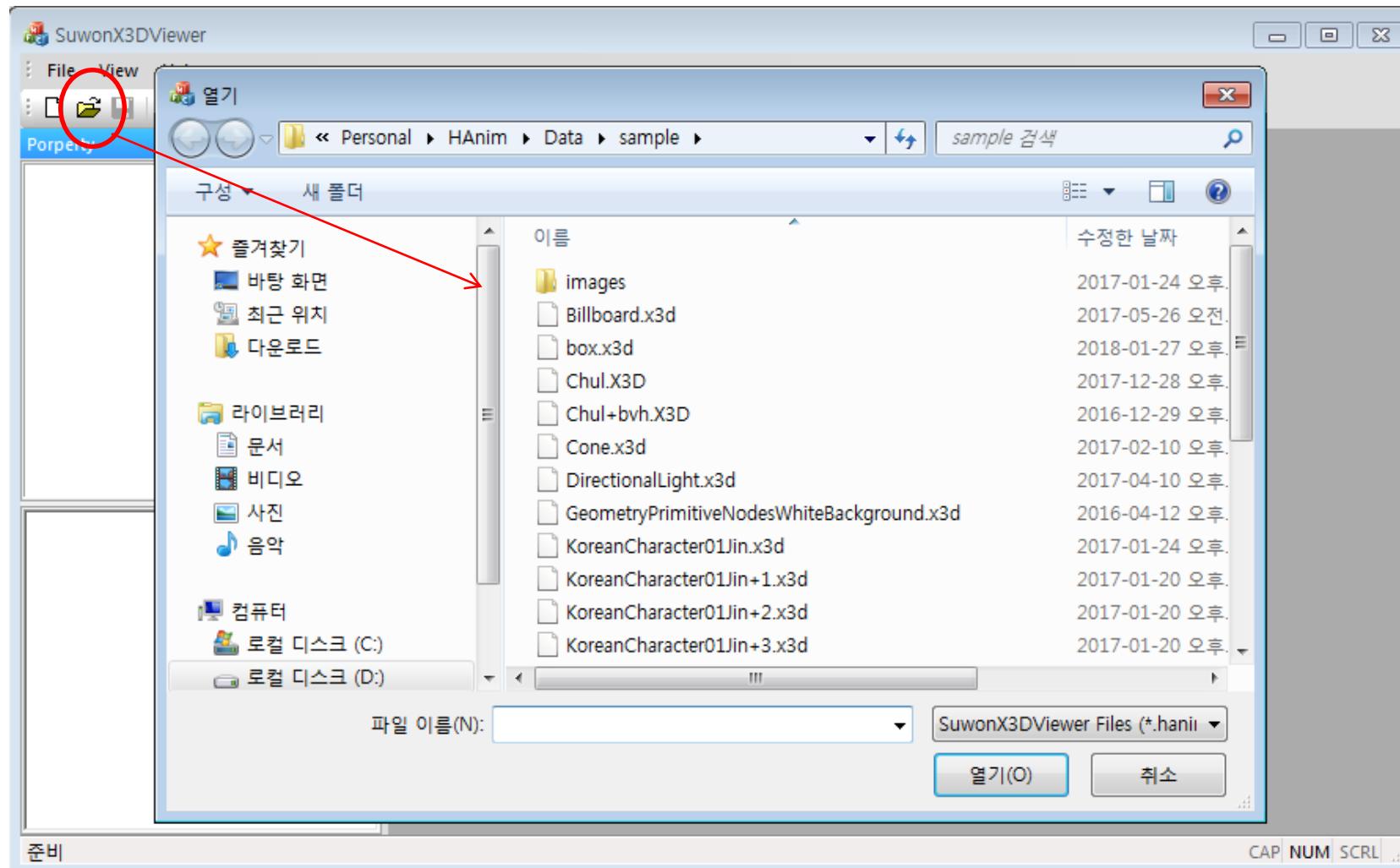
X3D C++ Library

- ▷ CX3DAppearanceNode
- ▷ CX3DArrayField
- ▷ CX3DBackgroundNode
- ▷ CX3DBindableNode
- ▷ CX3DBoundedObject
- ▷ CX3DChaserNode
- ▷ CX3DChildNode
- ▷ CX3DColorNode
- ▷ CX3DComposableVolumeRenderStyleNode
- ▷ CX3DComposedGeometryNode
- ▷ CX3DCoordinateNode
- ▷ CX3DDamperNode
- ▷ CX3DDragSensorNode
- ▷ CX3DEnvironmentalSensorNode
- ▷ CX3DEnvironmentTextureNode
- ▷ CX3DException
- ▷ CX3DField
- ▷ CX3DFieldDefinition
- ▷ CX3DFieldEvent
- ▷ CX3DFieldEventListener
- ▷ CX3DFogObject
- ▷ CX3DFollowerNode
- ▷ CX3DFontStyleNode
- ▷ CX3DGeometricPropertyNode
- ▷ CX3DGeometryNode
- ▷ CX3DGroupingNode
- ▷ CX3DInfoNode
- ▷ CX3DInterpolatorNode
- ▷ CX3DKeyDeviceSensorNode
- ▷ CX3DLayerNode
- ▷ CX3DLayoutNode
- ▷ CX3DLib
- ▷ CX3DLightNode
- ▷ CX3DMaterialNode
- ▷ CX3DMeta
- ▷ CX3DMetadataObject

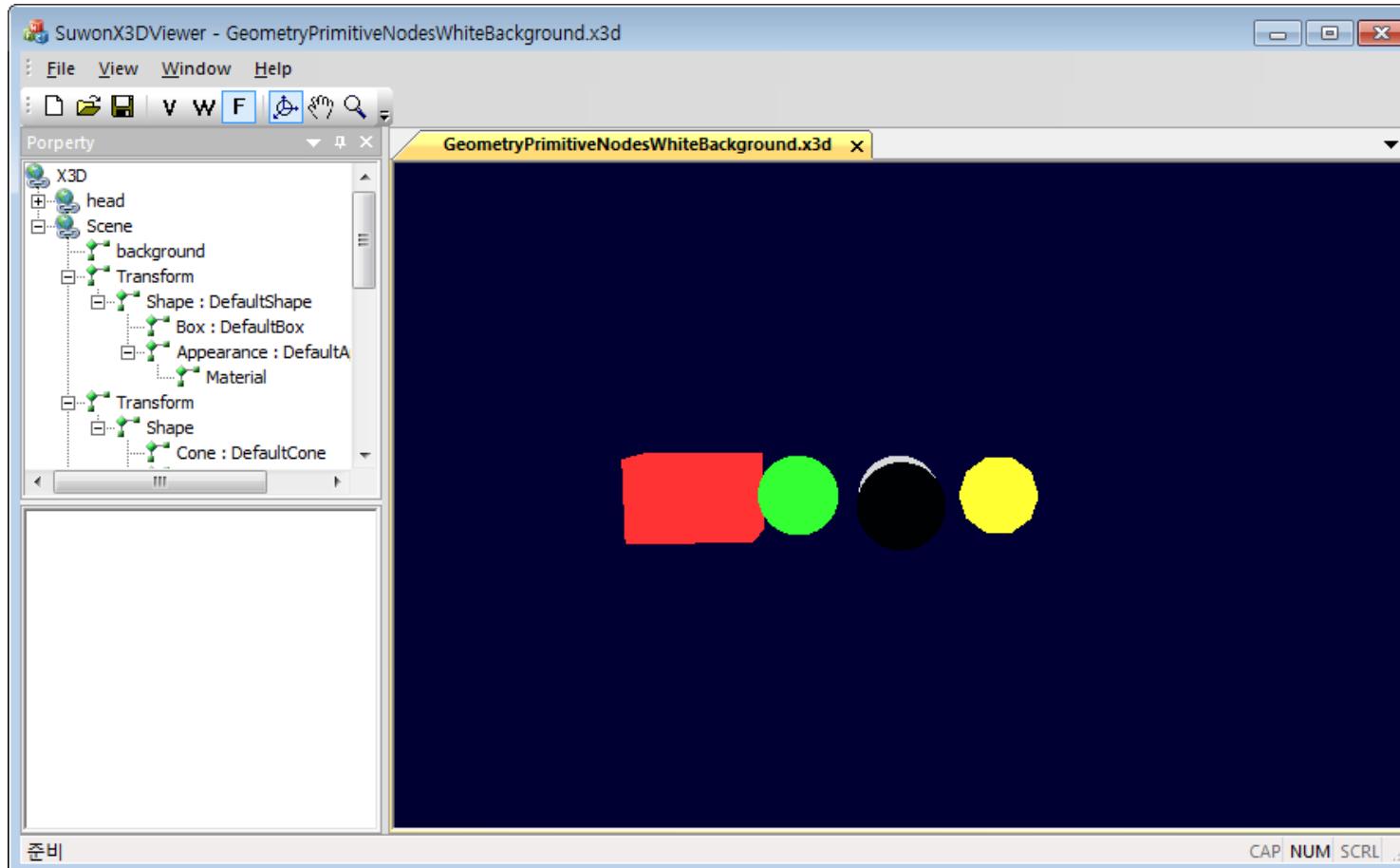
- ▷ CX3DNBodyCollidableNode
- ▷ CX3DNBodyCollisionSpaceNode
- ▷ CX3DNetworkSensorNode
- ▷ CX3DNode
- ▷ CX3DNodeArray
- ▷ CX3DNodeMixedContent
- ▷ CX3DNormalNode
- ▷ CX3DNurbsControlCurveNode
- ▷ CX3DNurbsSurfaceGeometryNode
- ▷ CX3DParametricGeometryNode
- ▷ CX3DParticleEmitterNode
- ▷ CX3DParticlePhysicsModelNode
- ▷ CX3DPickableObject
- ▷ CX3DPickSensorNode
- ▷ CX3DPointingDeviceSensorNode
- ▷ CX3DProductStructureChildNode
- ▷ CX3DProgrammableShaderObject
- ▷ CX3DPrototypeInstance
- ▷ CX3DRigidJointNode
- ▷ CX3DScene
- ▷ CX3DScriptNode
- ▷ CX3DSensorNode
- ▷ CX3DSequencerNode
- ▷ CX3DShaderNode
- ▷ CX3DShapeNode
- ▷ CX3DSoundNode
- ▷ CX3DSoundSourceNode
- ▷ CX3DTexture2DNode
- ▷ CX3DTexture3DNode
- ▷ CX3DTextureCoordinateNode
- ▷ CX3DTextureNode
- ▷ CX3DTextureTransformNode
- ▷ CX3DTimeDependentNode
- ▷ CX3DTouchSensorNode
- ▷ CX3DTriggerNode
- ▷ CX3DUrlObject

- ▷ CX3DVertexAttributeNode
- ▷ CX3DViewpointNode
- ▷ CX3DViewportNode
- ▷ CX3DVolumeDataNode
- ▷ CX3DVolumeRenderStyleNode
- ▷ CX3DNode
- ▷ Matrix3
- ▷ Matrix4

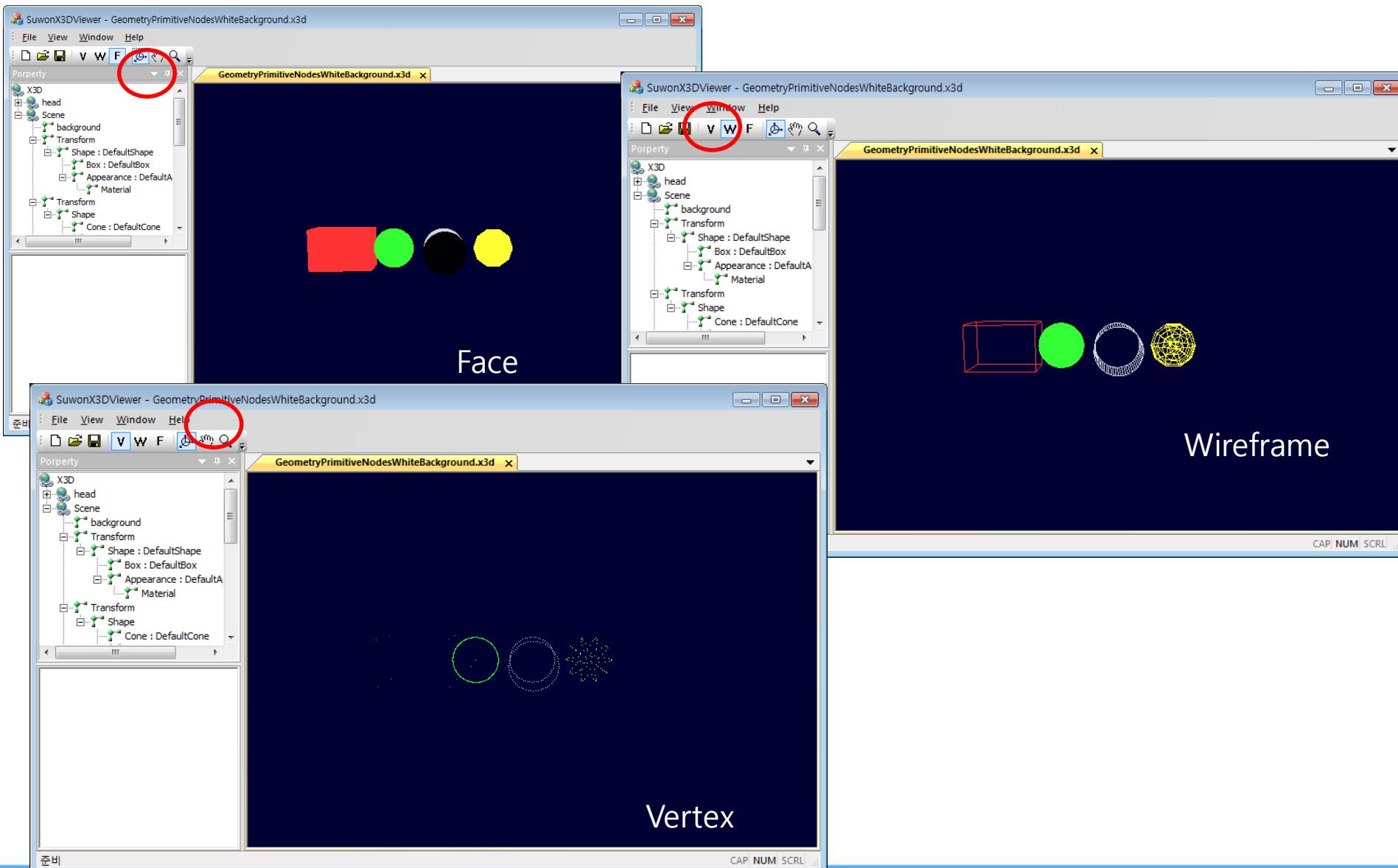
X3D C++ Binding Viewer (X3D File Open)



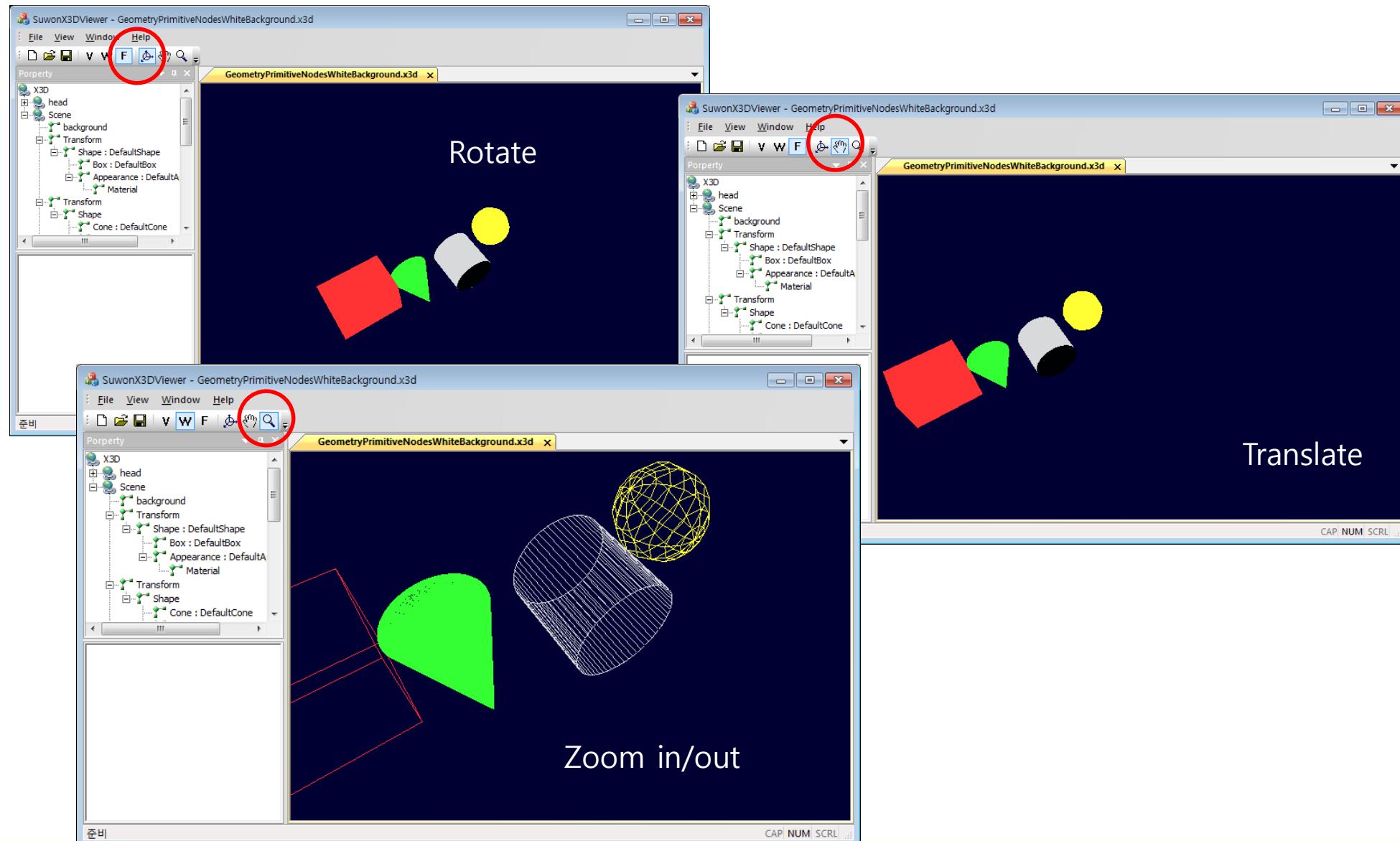
X3D C++ Binding Viewer (X3D Load)



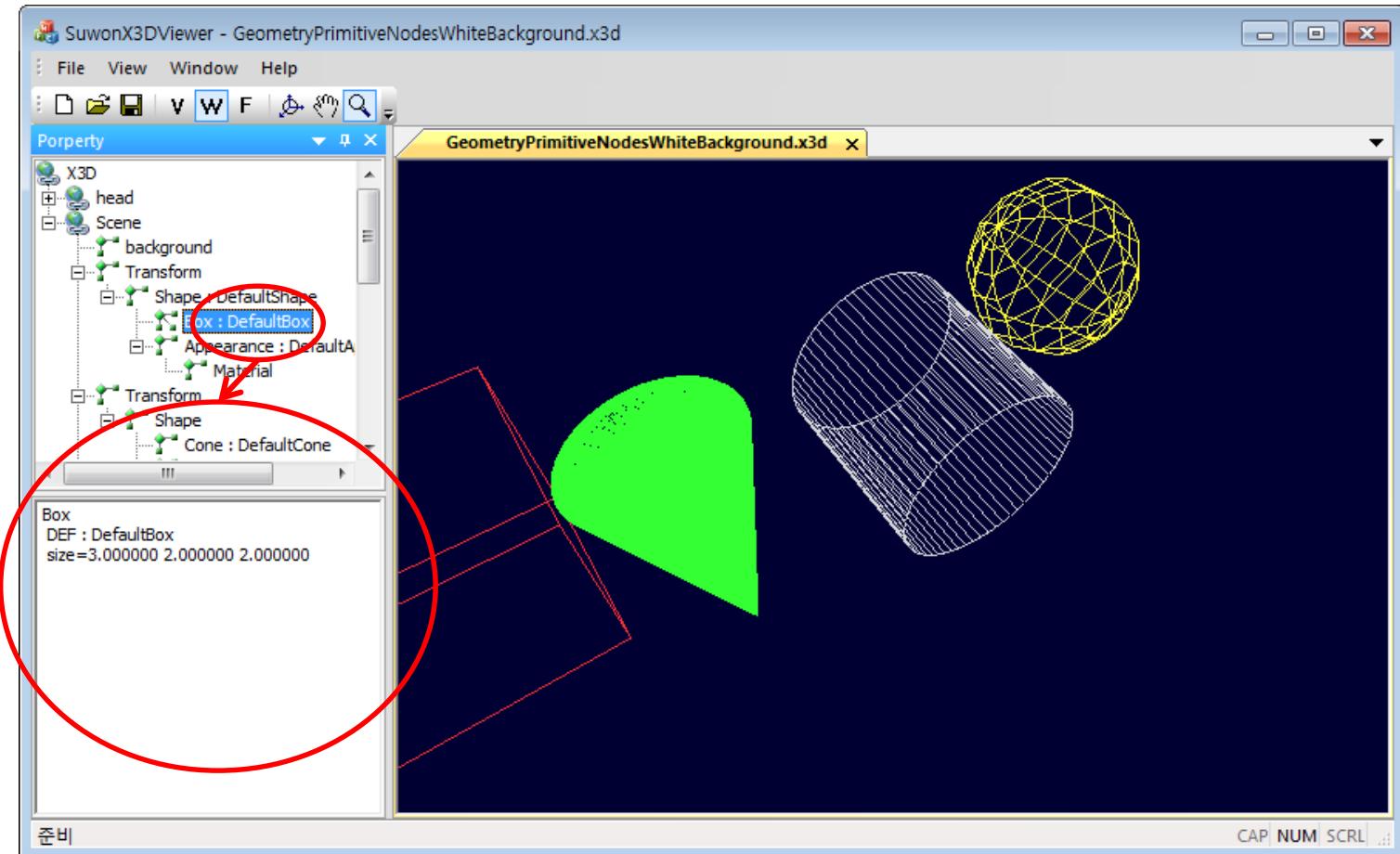
X3D C++ Binding Viewer (Viewer Mode)



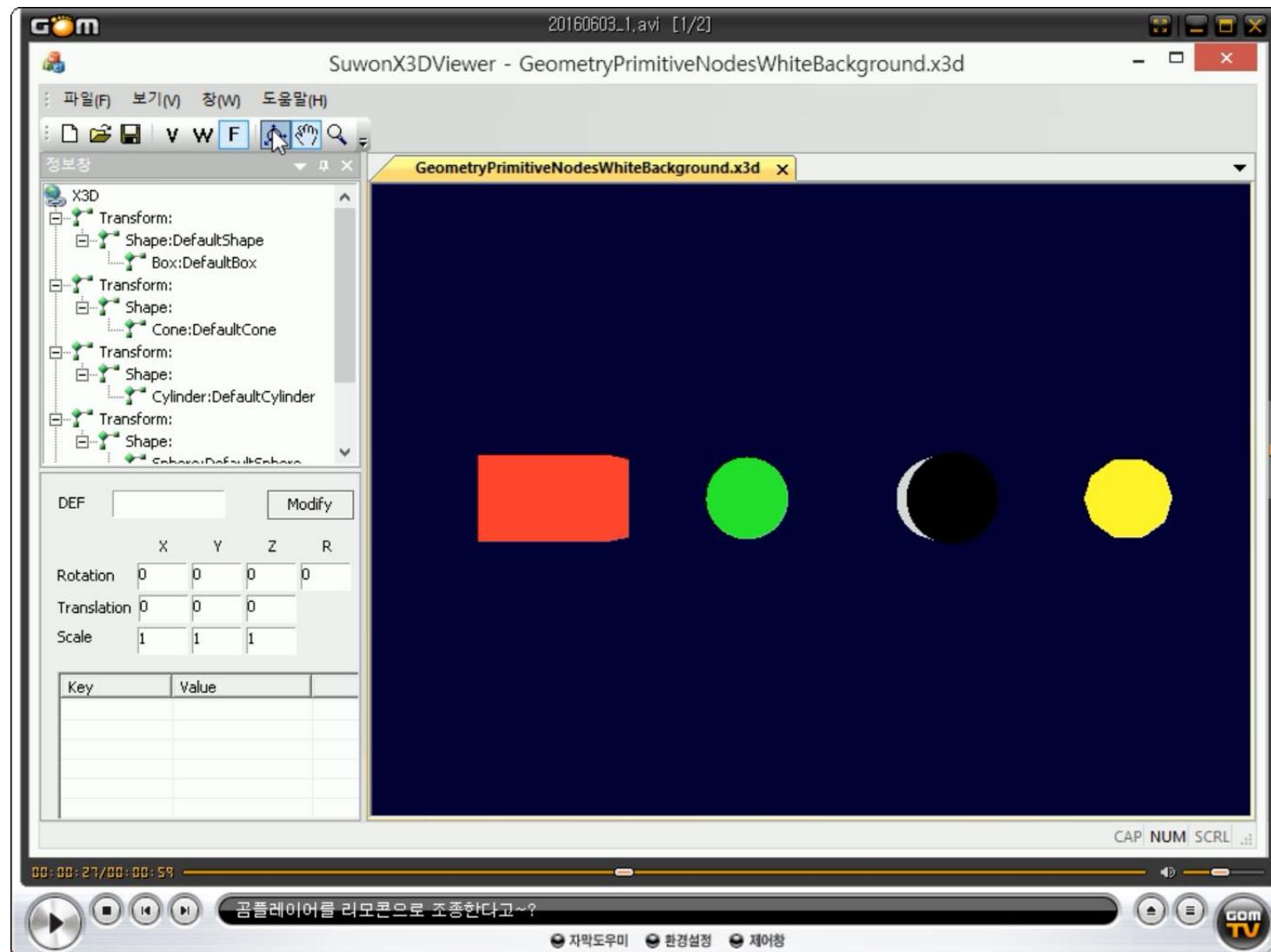
X3D C++ Binding Viewer (Viewer Mode)



X3D C++ Binding Viewer (Property Window)



X3D C++ Binding Viewer Demo



ISO/IEC NP 19777-4

X3D C++ Language Binding

Annex C Examples

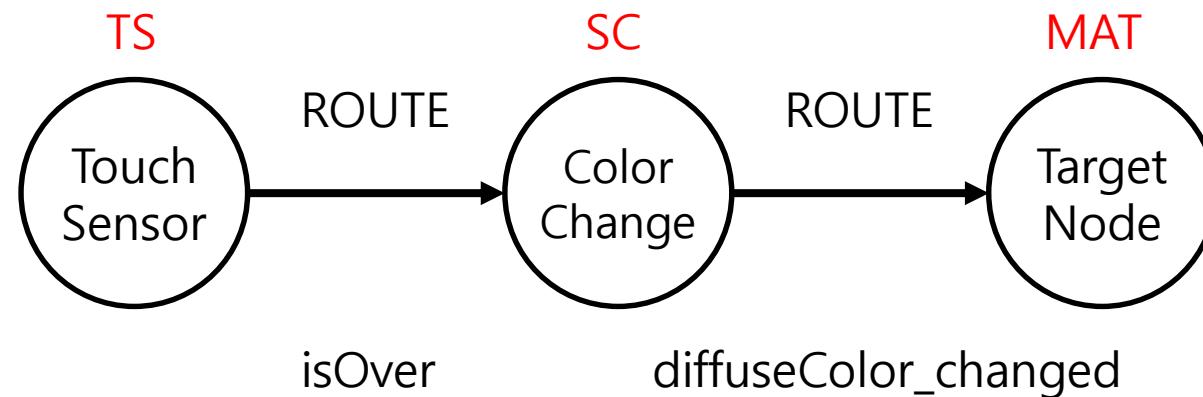
(Implementation)

Example 1. TouchSensor isOver event.x3d

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Immersive">
  <head>
    <meta content="TouchSensorIsOverEvent.x3d" name="filename"/>
    <meta content="Xeena VRML importer" name="translator"/>
    <meta content="23 February 2005" name="imported"/>
    <meta content="23 February 2005" name="revised"/>
    <meta
      content="X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-Edit.html" name="generator"/>
    <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html" name="generator"/>
  </head>
  <Scene>
    <Group>
      <Shape>
        <Appearance>
          <Material DEF="MAT" diffuseColor="0 0 1"/>
        </Appearance>
        <Box/>
      </Shape>
      <TouchSensor DEF="TS"/>
    </Group>
    <Script DEF="SC" url=""SAIExample1.class"">
      <field accessType="inputOnly" name="isOver" type="SFBool"/>
      <field accessType="outputOnly" name="diffuseColor_changed" type="SFColor"/>
    </Script>
    <ROUTE fromField="isOver" fromNode="TS" toField="isOver" toNode="SC"/>
    <ROUTE fromField="diffuseColor_changed" fromNode="SC"
      toField="set_diffuseColor" toNode="MAT"/>
  </Scene>
</X3D>
```

Example 1. TouchSensor isOver event.x3d (Implementation Overview)

```
<Material DEF="MAT" diffuseColor="0 0 1"/>
<TouchSensor DEF="TS"/>
<Script DEF="SC" url=""SAIExample1.class"">
  <field accessType="inputOnly" name="isOver" type="SFBool"/>
  <field accessType="outputOnly" name="diffuseColor_changed" type="SFColor"/>
</Script>
<ROUTE fromField="isOver" fromNode="TS" toField="isOver" toNode="SC"/>
<ROUTE fromField="diffuseColor_changed" fromNode="SC" toField="set diffuseColor" toNode="MAT"/>
```



Example 1. TouchSensor isOver event.x3d (Implementation) (1)

```
#include "stdafx.h"
#include "X3D_Sample2.h"

#include <glut.h>
#include "..\X3DLib\X3DLib.h"

#ifndef _DEBUG
#define new DEBUG_NEW
#endif

CWinApp theApp;
using namespace std;

CX3DScene m_pScene;
float ratio = 0;

/** The isOver field */
CSFBool* isOver=NULL;

/** The diffuseColor_changed field */
CSFColor* diffuseColor=NULL;

/** Color Constant, RED */
float RED[] = {1.0f, 0, 0};

/** Color Constant, BLUE */
float BLUE[] = {0, 0, 1.0f};

CX3DFieldEventLister lis;
```

Example 1. TouchSensor isOver event.x3d (Implementation) (2)

```
void initialize()
{
    /* specify clear values for the color buffers */
    glClearColor(0.0, 0.0, 0.0, 0.0);

    /* enable various capabilities */
    glEnable(GL_DEPTH_TEST);

    /* select flat or smooth shading */
    glShadeModel(GL_FLAT);

    /* specify mapping of depth values from normalized device coordinates to window coordinates */
    glDepthRange(0.0, 1.0);

    /* x3d File loading */
    m_pScene.createX3DFromFile(_T("2.x3d"));

    /**
     * Notification that the script has completed the setup and should go
     * about its own internal initialization.
     */
    isOver = (CSFBool*)(m_pScene.m_fields.get("isOver"));
    diffuseColor = (CSFColor*)(m_pScene.m_fields.get("diffuseColor_changed"));

    isOver->addX3DEventListener(lis);
}
```

Example 1. TouchSensor isOver event.x3d (Implementation) (3)

```
void display(void)
{
    /* sets the bitplane area of the window to values previously selected by glClearColor */
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    /* X3d Library Draw function Call */
    m_pScene.Draw();

    glFlush();
}

/**
 * Handle field changes.
 *
 * @param evt The field event
 */
void readableFieldChanged(CX3DFieldEvent* evt)
{
    if (evt->getSource() == isOver)
    {
        if (isOver->getValue() == true)
            diffuseColor->setValue(RED);
        else
            diffuseColor->setValue(BLUE);
    }
    else
    {
        printf("Unhandled event: %d", evt);
    }
}
```

Example 1. TouchSensor isOver event.x3d (Implementation) (4)

```
/* Mouse Hit Object Porcess */
void processHits(GLint hits, GLuint buffer[])
{
    unsigned int i, j;
    GLuint names, *ptr;

    ptr = (GLuint *) buffer;
    for (i = 0; i < hits; i++) { /* for each hit */
        names = *ptr;
        for (j = 0; j < names; j++) { /* for each name */
            diffuseColor->setValue(RED);
        }
    }
}

#define BUFSIZE 512
void pickRects(int button, int state, int x, int y)
{
    GLuint selectBuf[BUFSIZE];
    GLint hits;
    GLint viewport[4];

    if (button != GLUT_LEFT_BUTTON || state != GLUT_DOWN)
        return;

    /* return the value or values of a selected parameter */
    glGetIntegerv(GL_VIEWPORT, viewport);
```

Example 1. TouchSensor isOver event.x3d (Implementation) (5)

```
/* establish a buffer for selection mode values */
glSelectBuffer(BUFSIZE, selectBuf);

/* sets the rasterization mode */
glRenderMode(GL_SELECT);
/* initializes the name stack */
glInitNames();

/* push and pop the name stack */
glPushName(0);

glMatrixMode(GL_PROJECTION);
glPushMatrix();
glLoadIdentity();

/* forces execution of OpenGL functions in finite time */
glFlush();

hits = glRenderMode(GL_RENDER);
processHits(hits, selectBuf);
}
```

Example 1. TouchSensor isOver event.x3d (Implementation) (6)

```
void reshape(int w, int h)
{
    if(h == 0)
        h = 1;
    ratio = 1.0* w / h;

    /* set the viewport */
    glViewport(0, 0, w, h);

    /* defines a picking region */
    gluPickMatrix((GLdouble) x, (GLdouble) (viewport[3] - y),
                  5.0, 5.0, viewport);

    /* set up a perspective projection matrix */
    gluPerspective(45, ratio, 1, 1000);

    m_pScene.Draw(GL_SELECT);

    glPopMatrix();
```

Example 1. TouchSensor isOver event.x3d (Implementation) (7)

```
/* specify which matrix is the current matrix */
glMatrixMode(GL_PROJECTION);

/* replace the current matrix with the identity matrix */
glLoadIdentity();

/* set up a perspective projection matrix */
gluPerspective(45,ratio,1,1000);

glMatrixMode(GL_MODELVIEW);

glLoadIdentity();

/* defines a viewing transformation */
gluLookAt(5.0,5.0,5.0,
          0.0,0.0,-1.0,
          0.0f,1.0f,0.0f);

}

int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;

    /* Retrieves a module handle for the specified module. The module must have been loaded by
the calling process */
    HMODULE hModule = ::GetModuleHandle(NULL);
```

Example 1. TouchSensor isOver event.x3d (Implementation) (8)

```
if (hModule != NULL)
{
    // to initialize MFC
    if (!AfxWinInit(hModule, NULL, ::GetCommandLine(), 0))
    {
        _tprintf(_T("Error: Initialize failed.\n"));
        nRetCode = 1;
    }
    else
    {
        /* Initializes the COM library on the current thread and identifies the concurrency
model as single-thread apartment */
        HRESULT hr = CoInitialize(0);

        /* initialize the GLUT library */
        glutInit(&argc, argv);

        /* sets the initial display mode */
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);

        /* set the initial window position and size respectively */
        glutInitWindowSize (200, 200);
        glutInitWindowPosition (100, 100);

        /* creates a top-level window. */
        glutCreateWindow(argv[0]);
    }
}
```

Example 1. TouchSensor isOver event.x3d (Implementation) (9)

```
        /* sets the mouse callback for the current window. */
        glutMouseFunc(pickRects);

        /* sets the reshape callback for the current window. */
        glutReshapeFunc(reshape);

        /* sets the display callback for the current window. */
        glutDisplayFunc(display);

        /* enters the GLUT event processing loop. */
        glutMainLoop();
    }

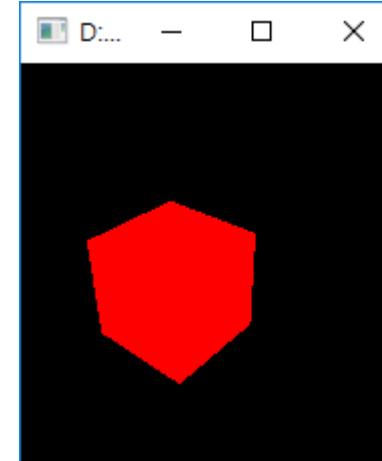
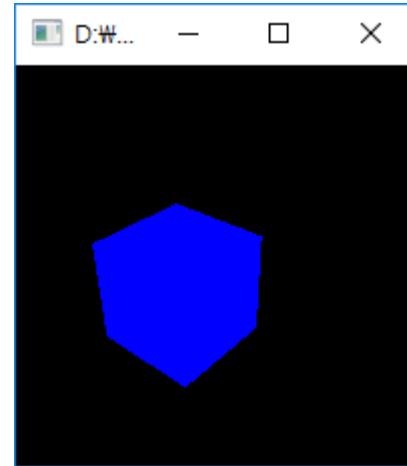
    else
    {
        _tprintf(_T("Error: GetModuleHandle failed\n"));
        nRetCode = 1;
    }
}

return nRetCode;
}
```

Example 1. TouchSensor isOver event.x3d (Implementation) (10)

On click, an event is generated

- On click, the color of the box is changed from blue to red.
- On click in the outside of the box, the color of the box is changed to blue.

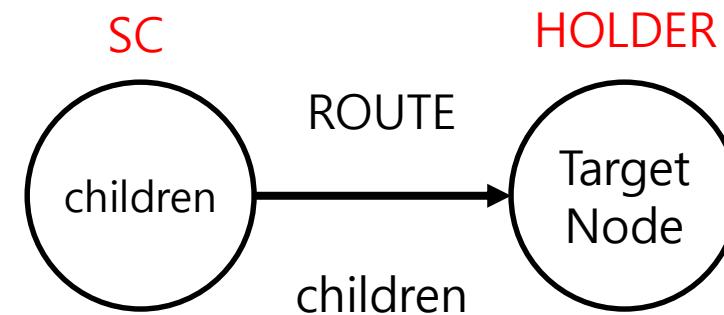


Example 2. Create nodes.x3d

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Immersive">
  <head>
    <meta content="CreateNodes.x3d" name="filename"/>
    <meta content="Xeena VRML importer" name="translator"/>
    <meta content="*enter date of initial version here*" name="created"/>
    <meta content="23 February 2005" name="imported"/>
    <meta content="23 February 2005" name="revised"/>
    <meta content="X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-
      Edit.html" name="generator"/>
    <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html" name="generator"/>
  </head>
  <Scene>
    <Transform DEF="HOLDER" translation="-2.0 0.0 0.0"/>
    <Script DEF="SC" url="SAIExample.class">
      <field accessType="outputOnly" name="children" type="MFNode"/>
    </Script>
    <ROUTE fromField="children" fromNode="SC" toField="children" toNode="HOLDER"/>
  </Scene>
</X3D>
```

Example 2. Create nodes.x3d (Implementation Overview)

```
<Transform DEF="HOLDER" translation="-2.0 0.0 0.0"/>
  <Script DEF="SC" url="SAIExample.class">
    <field accessType="outputOnly" name="children" type="MFNode"/>
  </Script>
  <ROUTE fromField="children" fromNode="SC" toField="children"
toNode="HOLDER"/>
```



Example 2. Create nodes.x3d (Implementation) (1)

```
// X3D_Sample3.cpp : 콘솔 응용 프로그램에 대한 진입점을 정의합니다.  
//  
  
#include "stdafx.h"  
#include "X3D_Sample3.h"  
  
/* include glut library */  
#include <glut.h>  
  
/* include X3D library */  
#include "..\WX3DLib\WX3DLib.h"  
  
#ifdef _DEBUG  
#define new DEBUG_NEW  
#endif  
  
/* Global variable */  
CWinApp theApp;  
using namespace std;  
  
/* X3D Library pointer */  
CX3DScape m_pScene;  
float ratio = 0;  
  
/**  
 * Notification that the script has completed the setup and should go  
 * about its own internal initialization.  
 */
```

Example 2. Create nodes.x3d (Implementation) (2)

```
void initialize()
{
    /* specify clear values for the color buffers */
    glClearColor(0.0, 0.0, 0.0, 0.0);

    /* enable various capabilities */
    glEnable(GL_DEPTH_TEST);

    /* select flat or smooth shading */
    glShadeModel(GL_FLAT);

    /* specify mapping of depth values from normalized device coordinates to window coordinates */
    glDepthRange(0.0, 1.0); /* The default z mapping */

    /* x3d File loading */
    m_pScene.createX3DFromFile(_T("3.x3d"));

    /**
     * Notification that the script has completed the setup and should go
     * about its own internal initialization.
     */
    CShape* shape = (CShape*) m_pScene.createNode("Shape");
    CBox* box = (CBox*) m_pScene.createNode("Box");
    shape->setGeometry(box);
    m_pScene.addRootNode(shape);
}
```

Example 2. Create nodes.x3d (Implementation) (3)

```
void display(void)
{
    /* sets the bitplane area of the window to values previously selected by glClearColor */
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    /* X3d Library Draw function Call */
    m_pScene.Draw();

    /* forces execution of OpenGL functions in finite time */
    glFlush();
}

void reshape(int w, int h)
{
    if(h == 0)
        h = 1;

    ratio = 1.0* w / h;

    /* set the viewport */
    glViewport(0, 0, w, h);

    /* GL_PROJECTION specify which matrix is the current matrix */
    glMatrixMode(GL_PROJECTION);

    /* replace the current matrix with the identity matrix */
    glLoadIdentity();
```

Example 2. Create nodes.x3d (Implementation) (4)

```
/* set up a perspective projection matrix */
gluPerspective(45,ratio,1,1000);

/* GL_MODELVIEW specify which matrix is the current matrix */
glMatrixMode(GL_MODELVIEW);

/* replace the current matrix with the identity matrix */
glLoadIdentity();

/* defines a viewing transformation */
gluLookAt(5.0,5.0,5.0,
          0.0,0.0,-1.0,
          0.0f,1.0f,0.0f);

}

int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;

    /* Retrieves a module handle for the specified module. The module must have been loaded by the calling
process */
    HMODULE hModule = ::GetModuleHandle(NULL);

    if (hModule != NULL)
    {
```

Example 2. Create nodes.x3d (Implementation) (5)

```
// to initialize MFC
if (!AfxWinInit(hModule, NULL, ::GetCommandLine(), 0))
{
    _tprintf(_T("Error: Initialize failed.\n"));
    nRetCode = 1;
}
else
{
    /* Initializes the COM library on the current thread and identifies the concurrency
model as single-thread apartment */
    HRESULT hr = CoInitialize(0);

    /* initialize the GLUT library */
    glutInit(&argc, argv);

    /* sets the initial display mode */
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);

    /* set the initial window position and size respectively */
    glutInitWindowSize (200, 200);

    glutInitWindowPosition (100, 100);

    /* creates a top-level window. */
    glutCreateWindow(argv[0]);
```

Example 2. Create nodes.x3d (Implementation) (6)

```
/* Data Initialize */
initialize();

/* sets the reshape callback for the current window.*/
glutReshapeFunc(reshape);

/* sets the display callback for the current window.*/
glutDisplayFunc(display);

/* enters the GLUT event processing loop.*/
glutMainLoop();
}

}

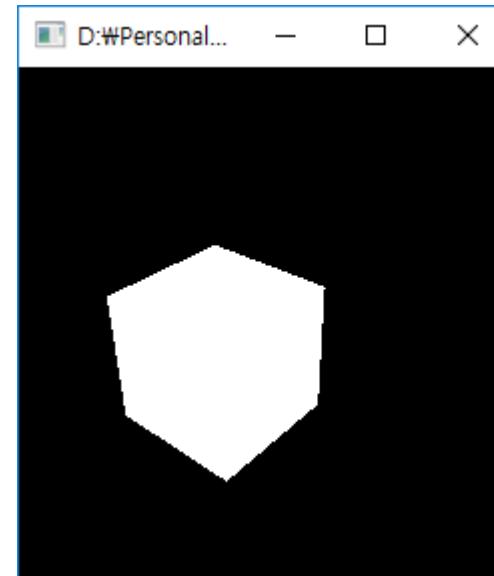
else
{
    _tprintf(_T("Error: GetModuleHandle failed\n"));
    nRetCode = 1;
}

return nRetCode;
}
```

Example 2. Create nodes.x3d (Implementation Results)

In the initialization stage,

- A Box under Shape is created in the Scene

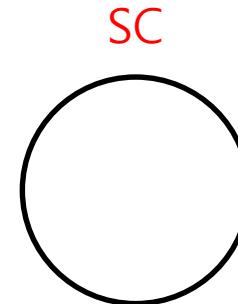


Example 3. Per frame notification.x3d

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "http://www.web3d.org/specifications/x3d-3.0.dtd"
          "file:///www.web3d.org/TaskGroups/x3d/translation/x3d-3.0.dtd">
<X3D profile="Immersive">
  <head>
    <meta content="PerFrameNotification.x3d" name="filename"/>
    <meta content="Xeena VRML importer" name="translator"/>
    <meta content="23 February 2005" name="imported"/>
    <meta content="23 February 2005" name="revised"/>
    <meta
      content="X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-Edit.html"
      name="generator"/>
    <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html" name="generator"/>
  </head>
  <Scene>
    <Script DEF="SC" url=""SAIExample3.class" "/>
  </Scene>
</X3D>
```

~~Example 3. Per frame notification.x3d (Implementation Overview)~~

```
<Script DEF="SC" url=""SAIExample3.class" "/>
```



Initialize(): Initialize Time value

prepareEvents(): Calculate and output FPS

Example 3. Per frame notification.x3d (Implementation) (1)

```
#include "stdafx.h"
#include "X3D_Sample4.h"

/* include glut library */
#include <glut.h>

/* include X3D library */
#include "..\X3DLib\X3DLib.h"

#ifndef _DEBUG
#define new DEBUG_NEW
#endif

/* Global variable */
CWinApp theApp;
using namespace std;

/* X3D Library pointer */
CX3DScene m_pScene;
float wratio = 0;

int frame,currentTime,lastStartTime=0;
char s[30];
```

Example 3. Per frame notification.x3d (Implementation) (2)

```
/**  
 * Notification that the script has completed the setup and should go  
 * about its own internal initialization.  
 */  
  
void initialize()  
{  
    /* specify clear values for the color buffers */  
    glClearColor(0.0, 0.0, 0.0, 0.0);  
  
    /* enable various capabilities */  
    glEnable(GL_DEPTH_TEST);  
  
    /* select flat or smooth shading */  
    glShadeModel(GL_FLAT);  
  
    /* specify mapping of depth values from normalized device coordinates to window coordinates */  
    glDepthRange(0.0, 1.0); /* The default z mapping */  
  
    /* x3d File loading */  
    m_pScene.createX3DFromFile(_T("4.x3d"));  
  
    // current time initialize  
    currentTime=glutGet(GLUT_ELAPSED_TIME);  
}
```

Example 3. Per frame notification.x3d (Implementation) (3)

```
void display(void)
{
    /* sets the bitplane area of the window to values previously selected by glClearColor */
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    /* X3d Library Draw function Call */
    m_pScene.Draw();

    /* forces execution of OpenGL functions in finite time */
    glFlush();
}

/**
 * Start of frame notification.
 */
void prepareEvents()
{
    /* sets the bitplane area of the window to values previously selected by glClearColor */
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    /* FPS (frames per second) calculate */
    frame++;

    /* retrieves simple GLUT state represented by integers */
    currentTime=glutGet(GLUT_ELAPSED_TIME);
```

Example 3. Per frame notification.x3d (Implementation) (4)

```
if (currentTime - lastStartTime > 1000) {  
    sprintf_s(s,"FPS:%4.2f", (frame * 1000.0 / (currentTime-lastStartTime)));  
    lastStartTime = currentTime;  
    frame = 0;  
}  
  
/* RGB color Sets the current color */  
glColor3f(1.0f,1.0f,1.0f);  
  
glPushMatrix();  
  
char *c;  
/* specify the raster position for pixel operations */  
glRasterPos2f(-2, -2);  
  
for (c=s; *c != '\0'; c++) {  
    /* Draw a bitmapped character. */  
    glutBitmapCharacter(GLUT_BITMAP_8_BY_13, *c);  
}  
glPopMatrix();  
  
/* forces execution of OpenGL functions in finite time */  
glFlush();  
}
```

Example 3. Per frame notification.x3d (Implementation) (5)

```
void reshape(int w, int h)
{
    if(h == 0)
        h = 1;

    wratio = 1.0* w / h;

    /* set the viewport */
    glViewport(0, 0, w, h);

    /* GL_PROJECTION specify which matrix is the current matrix */
    glMatrixMode(GL_PROJECTION);

    /* replace the current matrix with the identity matrix */
    glLoadIdentity();

    /* set up a perspective projection matrix */
    gluPerspective(45,wratio,1,1000);

    /* GL_MODELVIEW specify which matrix is the current matrix */
    glMatrixMode(GL_MODELVIEW);

    /* replace the current matrix with the identity matrix */
    glLoadIdentity();
```

Example 3. Per frame notification.x3d (Implementation) (6)

```
/* defines a viewing transformation */
gluLookAt(5.0,5.0,5.0,
0.0,0.0,-1.0,
0.0f,1.0f,0.0f);

}

int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;

    /* Retrieves a module handle for the specified module. The module must have been loaded by the calling
process */

    HMODULE hModule = ::GetModuleHandle(NULL);

    if (hModule != NULL)
    {
        // to initialize MFC
        if (!AfxWinInit(hModule, NULL, ::GetCommandLine(), 0))
        {
            _tprintf(_T("Error: Initialze failed.\n"));
            nRetCode = 1;
        }
        else
        {
```

Example 3. Per frame notification.x3d (Implementation) (7)

```
/* Initializes the COM library on the current thread and identifies the concurrency
model as single-thread apartment */
HRESULT hr = CoInitialize(0); // 캠 객체 초기화..

/* initialize the GLUT library */
glutInit(&argc, argv);

/* sets the initial display mode */
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);

/* set the initial window position and size respectively */
glutInitWindowSize (200, 200);
glutInitWindowPosition (100, 100);

/* creates a top-level window. */
glutCreateWindow(argv[0]);

/* Data Initialize */
initialize();

/* sets the reshape callback for the current window. */
glutReshapeFunc(reshape);

/* sets the display callback for the current window. */
glutDisplayFunc(display);

/* sets the global idle callback. */
glutIdleFunc(prepareEvents);
```

Example 3. Per frame notification.x3d (Implementation) (8)

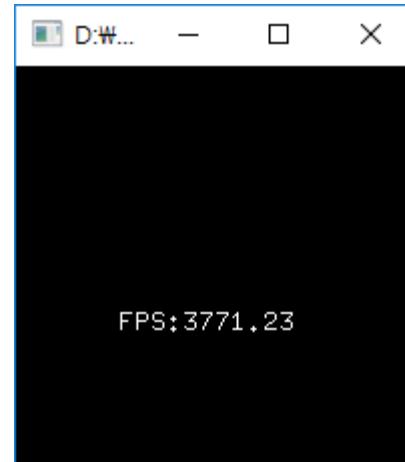
```
        /* enters the GLUT event processing loop. */
        glutMainLoop();
    }
} else
{
    _tprintf(_T("Error: GetModuleHandle failed\n"));
    nRetCode = 1;
}

return nRetCode;
}
```

Example 3. Per frame notification.x3d (Implementation Results)

In the prepareEvents()

- FPS is calculated and displayed

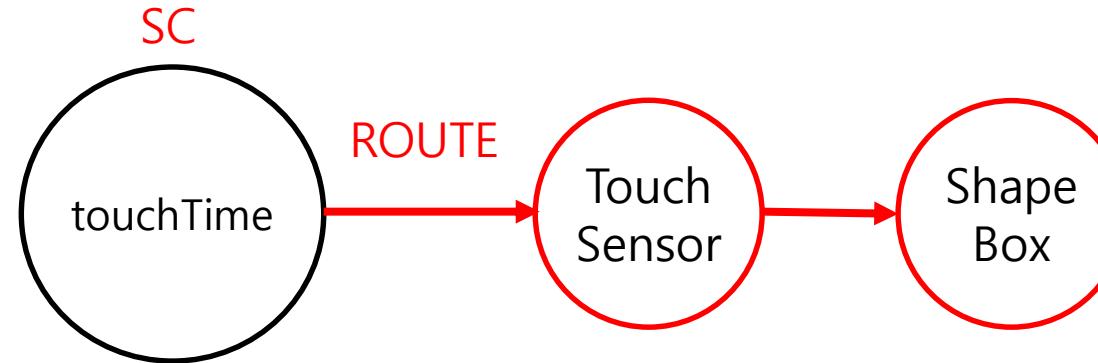


Example 4. Add dynamic routes.x3d

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
  "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Immersive">
  <Scene>
    <Script DEF="SC" url="SAIExample4.class">
      <field accessType="inputOnly" name="touchTime" type="SFTime"/>
    </Script>
  </Scene>
</X3D>
```

Example 4. Add dynamic routes.x3d (Implementation Overview)

```
<Script DEF="SC" url="SAIExample4.class">
    <field accessType="inputOnly" name="touchTime" type="SFTime"/>
</Script>
```



Initialize() :

- Read "touchTime" field value
- Create Shape(Box) and TouchSesor under Group in the Scene
- Create "touchTime" ROUTE in the scene, connect the TouchSensor to the scene

readableFieldChanged()

- On the event of TouchSensor, "Poke!" is displayed

Example 4. Add dynamic routes.x3d (Implementation) (1)

```
#include "stdafx.h"
#include "X3D_Sample5.h"

/* include glut library */
#include <glut.h>

/* include X3D library */
#include "..\WX3DLib\WX3DLib.h"

#ifndef _DEBUG
#define new DEBUG_NEW
#endif

/* Global variable */
CWinApp theApp;
using namespace std;

/* X3D Library pointer */
CX3DScene m_pScene;
float ratio = 0;

CSFTime* touchTime = NULL;
```

Example 4. Add dynamic routes.x3d (Implementation) (2)

```
/**  
 * Notification that the script has completed the setup and should go  
 * about its own internal initialization.  
 */  
void initialize()  
{  
    /* specify clear values for the color buffers */  
    glClearColor(0.0, 0.0, 0.0, 0.0);  
  
    /* enable various capabilities */  
    glEnable(GL_DEPTH_TEST);  
  
    /* select flat or smooth shading */  
    glShadeModel(GL_FLAT);  
  
    /* specify mapping of depth values from normalized device coordinates to window coordinates */  
    glDepthRange(0.0, 1.0); /* The default z mapping */  
  
    /* x3d File loading */  
    m_pScene.createX3DFromFile(_T("5.x3d"));
```

Example 4. Add dynamic routes.x3d (Implementation) (3)

```
/**  
 * Notification that the script has completed the setup and should go  
 * about its own internal initialization.  
 */  
  
touchTime = (CSFTime*)(m_pScene.m_fields.get("touchTime"));  
  
CShape* shape = (CShape*) m_pScene.createNode("Shape");  
CBox* box = (CBox*) m_pScene.createNode("Box");  
CX3DNode* touchSensor = m_pScene.createNode("TouchSensor");  
  
shape->setGeometry(box);  
  
CGroup* group = (CGroup*)m_pScene.createNode("Group");  
  
// Add the shape and sensor to the group  
group->addChild(shape);  
group->addChild(touchSensor);  
  
m_pScene.addRootNode(group);  
  
CX3DScriptNode* selfRef =(CX3DScriptNode*)m_pScene.getNode("SC");  
// Get a handle to the toplevel execution context  
m_pScene.addRoute(touchSensor, "touchTime", selfRef, "touchTime");  
}
```

Example 4. Add dynamic routes.x3d (Implementation) (4)

```
void display(void)
{
    /* sets the bitplane area of the window to values previously selected by glClearColor */
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    /* X3d Library Draw function Call */
    m_pScene.Draw();

    /* forces execution of OpenGL functions in finite time */
    glFlush();
}

/* Mouse Hit Object Porcess */
void processHits(GLint hits, GLuint buffer[])
{
    unsigned int i, j;
    GLuint names, *ptr;

    //printf("hits = %d\n", hits);
    ptr = (GLuint *) buffer;
    for (i = 0; i < hits; i++) { /* for each hit */
        names = *ptr;
        // printf(" number of names for hit = %d\n", names); ptr++;
        // printf(" z1 is %g;", (float) *ptr/0x7fffffff); ptr++;
        // printf(" z2 is %g\n", (float) *ptr/0x7fffffff); ptr++;
        // printf(" the name is ");
    }
}
```

Example 4. Add dynamic routes.x3d (Implementation) (5)

```
        for (j = 0; j < names; j++) { /* for each name */
            printf("Poke!"); ptr++;
        }
        printf("\n");
    }

#define BUFSIZE 512

void pickRects(int button, int state, int x, int y)
{
    GLuint selectBuf[BUFSIZE];
    GLint hits;
    GLint viewport[4];

    if (button != GLUT_LEFT_BUTTON || state != GLUT_DOWN)
        return;

    /* return the value or values of a selected parameter */
    glGetIntegerv(GL_VIEWPORT, viewport);

    /* establish a buffer for selection mode values */
    glSelectBuffer(BUFSIZE, selectBuf);

    /* sets the rasterization mode */
    glRenderMode(GL_SELECT);
```

Example 4. Add dynamic routes.x3d (Implementation) (6)

```
/* defines a picking region */
gluPickMatrix(GLdouble x, (GLdouble) (viewport[3] - y),
               5.0, 5.0, viewport);

/* set up a perspective projection matrix */
gluPerspective(45, ratio, 1, 1000);

/* X3d Library Draw function Call */
m_pScene.Draw(GL_SELECT);

glPopMatrix();

/* forces execution of OpenGL functions in finite time */
glFlush();

hits = glRenderMode(GL_RENDER);
processHits(hits, selectBuf);
}

void reshape(int w, int h)
{
    if(h == 0)
        h = 1;

    ratio = 1.0* w / h;

    /* set the viewport */
    glViewport(0, 0, w, h);
```

Example 4. Add dynamic routes.x3d (Implementation) (7)

```
/* specify which matrix is the current matrix */
glMatrixMode(GL_PROJECTION);

/* replace the current matrix with the identity matrix */
glLoadIdentity();

/* set up a perspective projection matrix */
gluPerspective(45,ratio,1,1000);

glMatrixMode(GL_MODELVIEW);

glLoadIdentity();

/* defines a viewing transformation */
gluLookAt(5.0,5.0,5.0,
          0.0,0.0,-1.0,
          0.0f,1.0f,0.0f);

}

int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;

    /* Retrieves a module handle for the specified module. The module must have been loaded by the calling
process */
    HMODULE hModule = ::GetModuleHandle(NULL);
```

Example 4. Add dynamic routes.x3d (Implementation) (8)

```
if (hModule != NULL)
{
    // to initialize MFC
    if (!AfxWinInit(hModule, NULL, ::GetCommandLine(), 0))
    {
        _tprintf(_T("Error: Initialze failed.\n"));
        nRetCode = 1;
    }
    else
    {
        /* Initializes the COM library on the current thread and identifies the
concurrency model as single-thread apartment */
        HRESULT hr = CoInitialize(0); // 캠 객체 초기화..

        /* initialize the GLUT library */
        glutInit(&argc, argv);

        /* sets the initial display mode */
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);

        /* set the initial window position and size respectively */
        glutInitWindowSize (200, 200);
        glutInitWindowPosition (100, 100);

        /* creates a top-level window. */
        glutCreateWindow(argv[0]);
    }
}
```

Example 4. Add dynamic routes.x3d (Implementation) (9)

```
/* Data Initialize */
initialize();

/* sets the mouse callback for the current window.*/
glutMouseFunc(pickRects);

/* sets the reshape callback for the current window.*/
glutReshapeFunc(reshape);

/* sets the display callback for the current window.*/
glutDisplayFunc(display);

/* enters the GLUT event processing loop.*/
glutMainLoop();
}

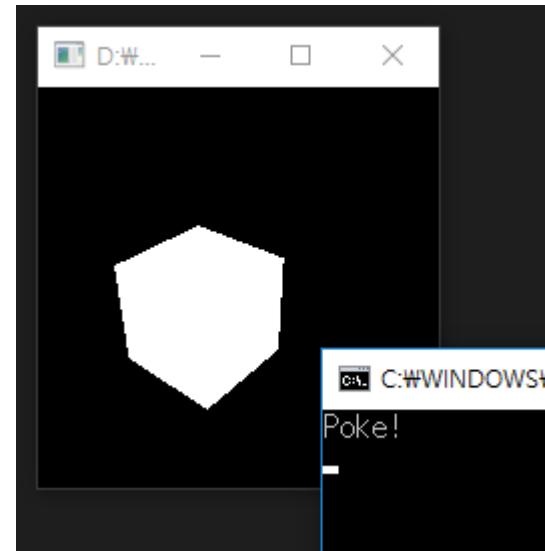
}

else
{
    _tprintf(_T("Error: GetModuleHandle failed\n"));
    nRetCode = 1;
}

return nRetCode;
}
```

Example 4. Add dynamic routes.x3d (Implementation Results)

- In the initialization
Create Scene > Group > Shape > Box > TouchSensor
- On clicking Box, "Poke!" is displayed



Multi Texture File Binding

```
void CX3DScene::BindingTextue()
{
    CString strValue;
    for (int i=0; i < m_arrTextureName.GetCount(); i++)
    {
        strValue = m_arrTextureName.GetAt(i);
        LoadTexture(strValue, i+1);
    }
}
```

Texture File Loading

```
GLuint CX3DScene::LoadTexture(CString file_name, int nIdx)
{
    GLuint texName = nIdx; //OpenGL texture to create and return

    // load texture image for ...
    Bitmap* pBitmap = Bitmap::FromFile(file_name.AllocSysString());
    if (pBitmap->GetLastStatus() != Gdiplus::Ok) {
        return 0;
    }

    pBitmap->RotateFlip(RotateFlipType::RotateNoneFlipY);
    BitmapData bitmap_data;
    Rect rect(0, 0, pBitmap->GetWidth(), pBitmap->GetHeight());

    // make texture...
    glBindTexture(GL_TEXTURE_2D, texName);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, pBitmap->GetWidth(), pBitmap->GetHeight(), 0, GL_BGRA_EXT,
                 GL_UNSIGNED_BYTE, (GLvoid*)bitmap_data.Scan0);

    GLenum x = glGetError();
    pBitmap->UnlockBits(&bitmap_data);
    delete pBitmap;

    return texName;
}
```

Draw Texture File

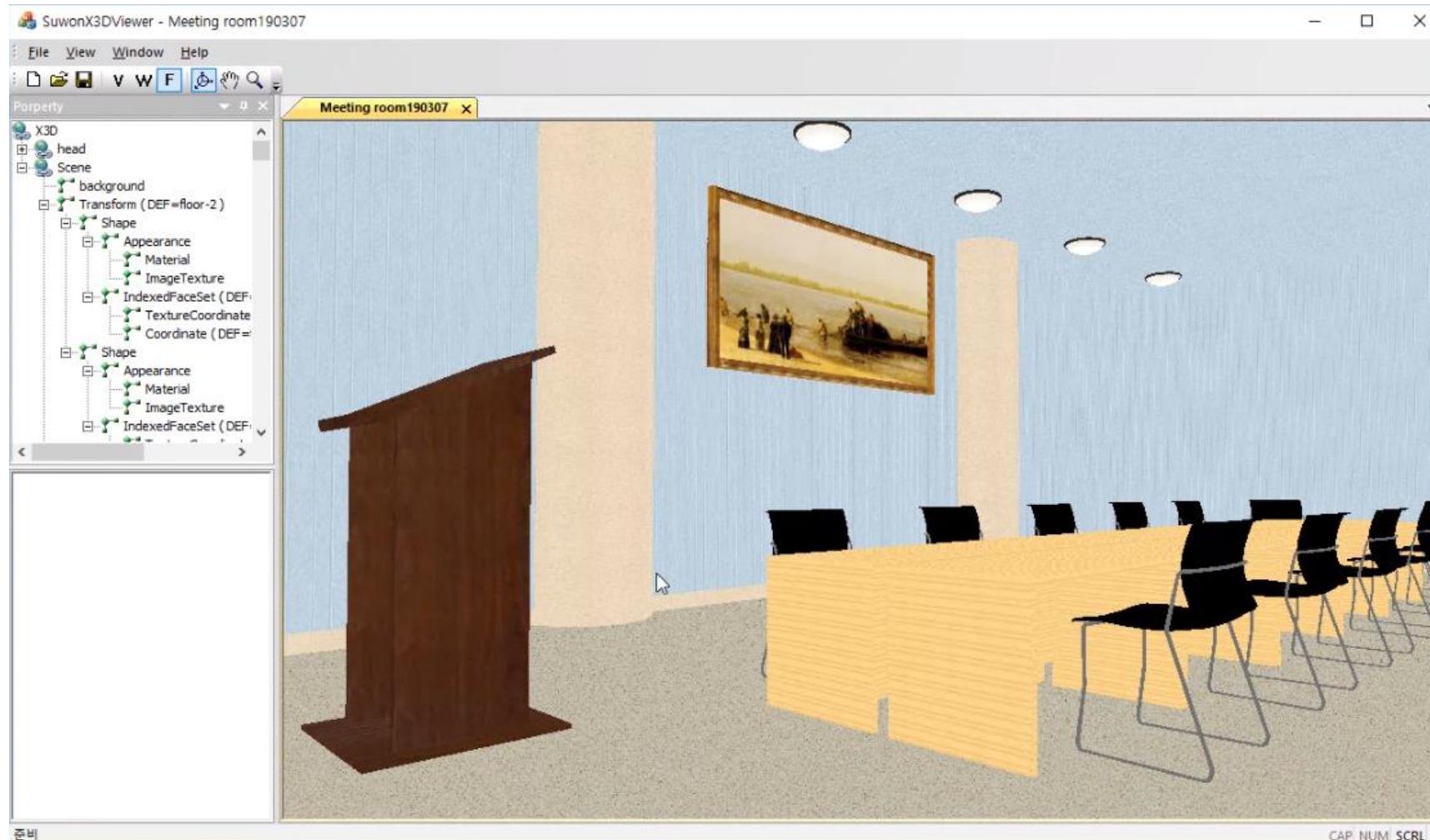
```
if (pNode->isnodeName(_T("Box")) || pNode->isnodeName(_T("Cone")) || pNode->isnodeName(_T("Cylinder")) ||
    pNode->isnodeName(_T("Sphere")) || pNode->isnodeName(_T("IndexedFaceSet"))
    || pNode->isnodeName(_T("Material")) ||
    pNode->isnodeName(_T("ImageTexture")) )
{
    if (mode == GL_SELECT)
    {
        glBindName(pNode->getID());
    }

    pNode->Draw();
}

void CImageTexture::Draw()
{
    glBindTexture(GL_TEXTURE_2D, m_nID);
}
```

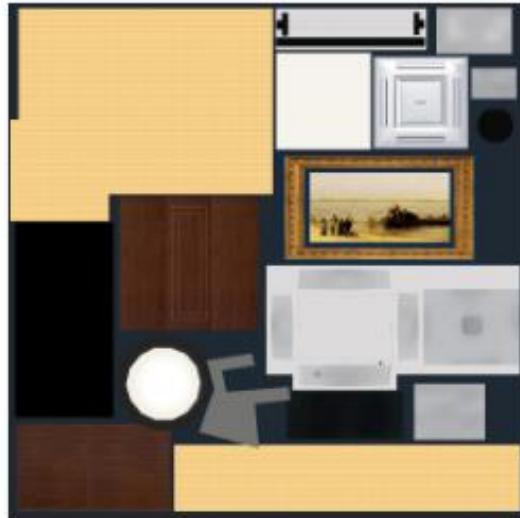
Example – Meeting room

Meeting room

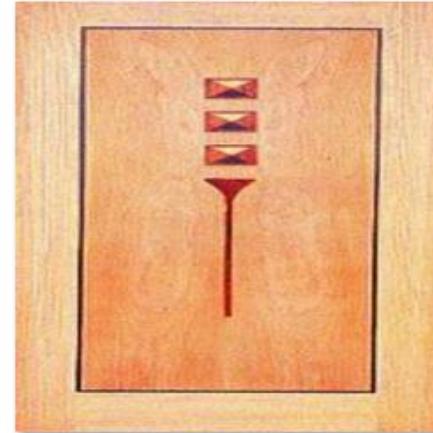


Example – Meeting room

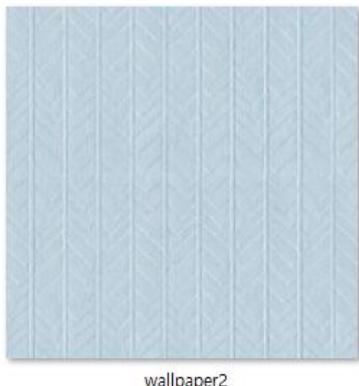
Images



room
wallpaper



door



wallpaper2



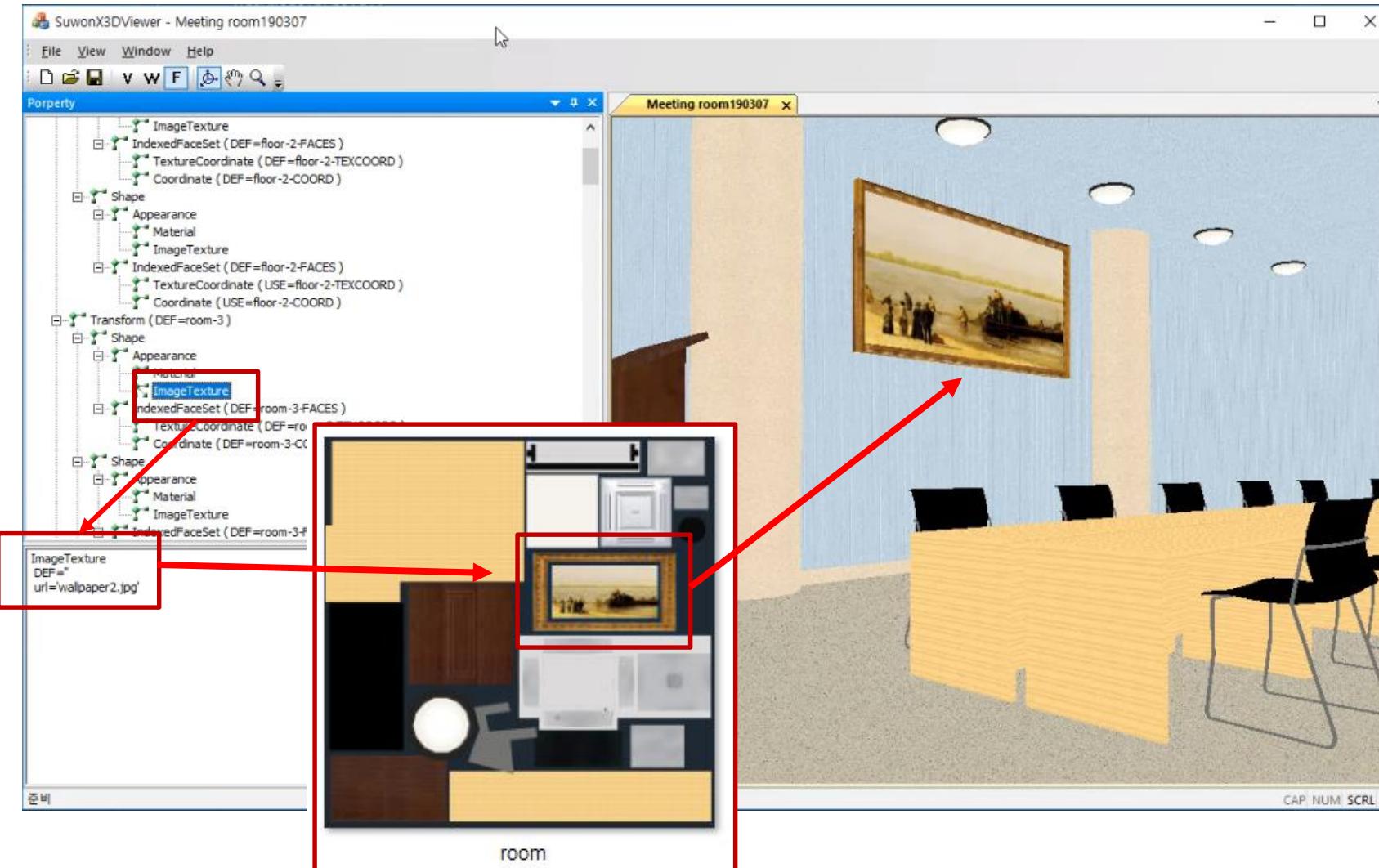
floor



wallpaper

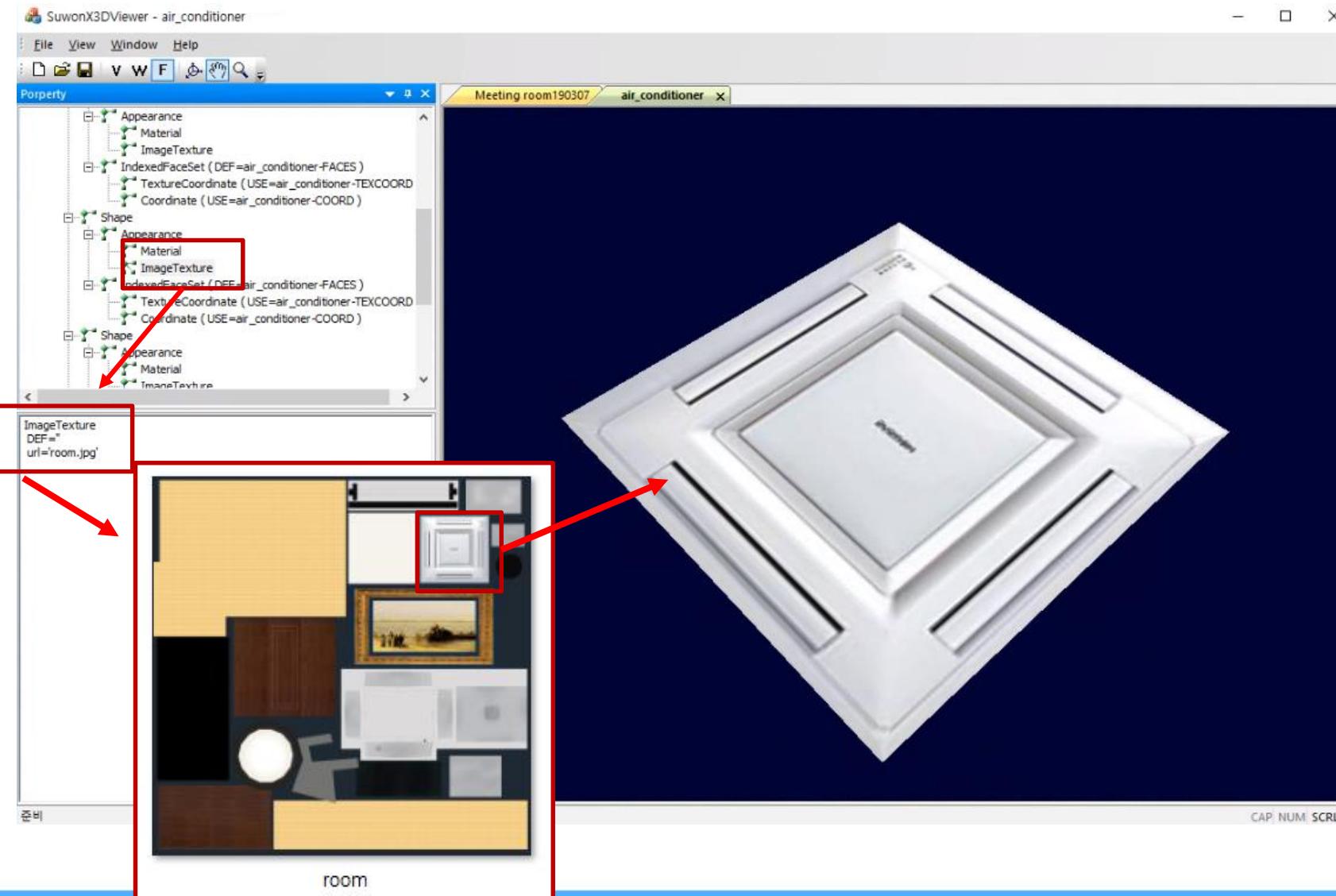
Example – Meeting room

Image Processing



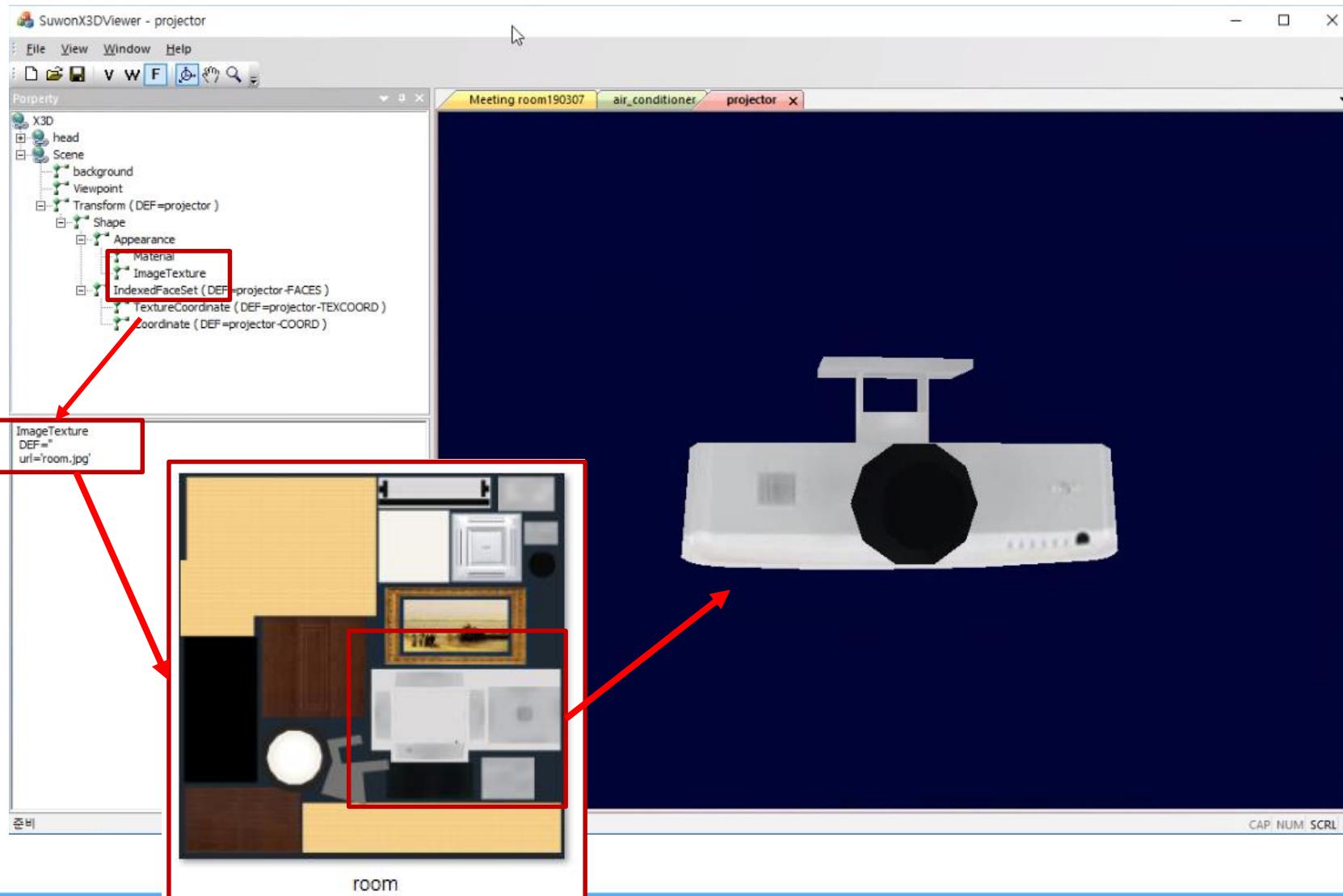
Example – Air Conditioner

Air Conditioner



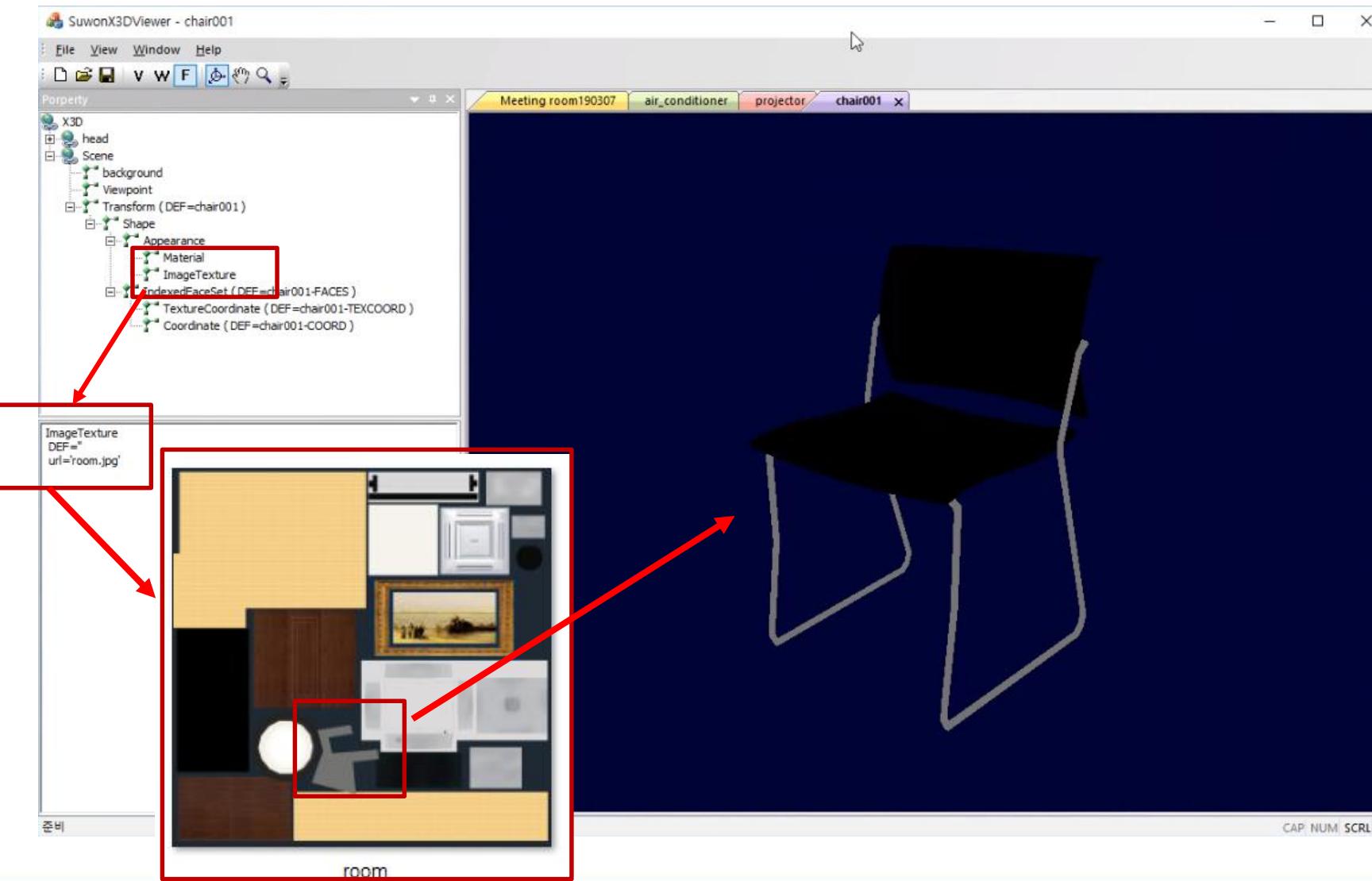
Example – Projector

Projector



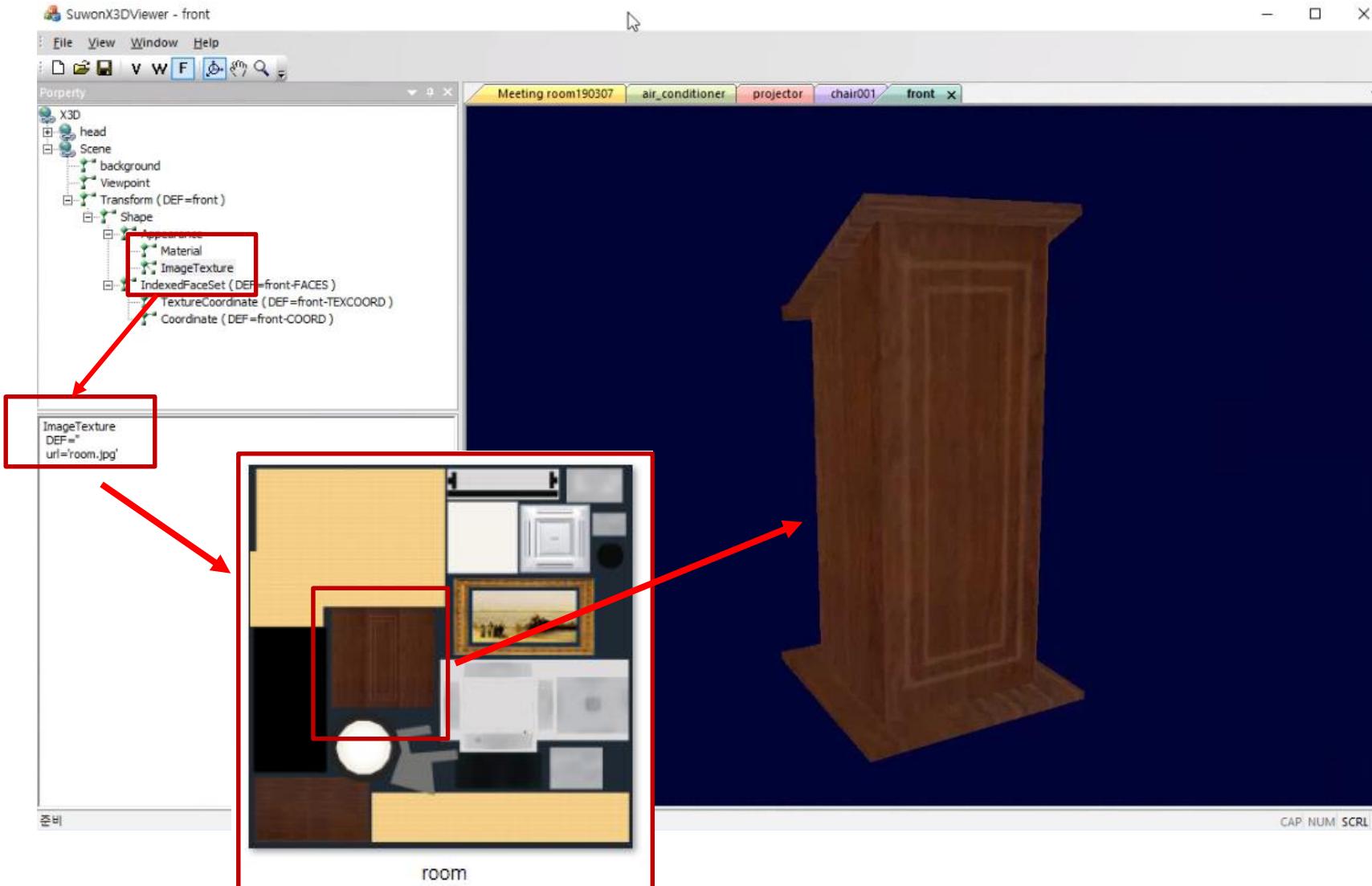
Example – chair

chair

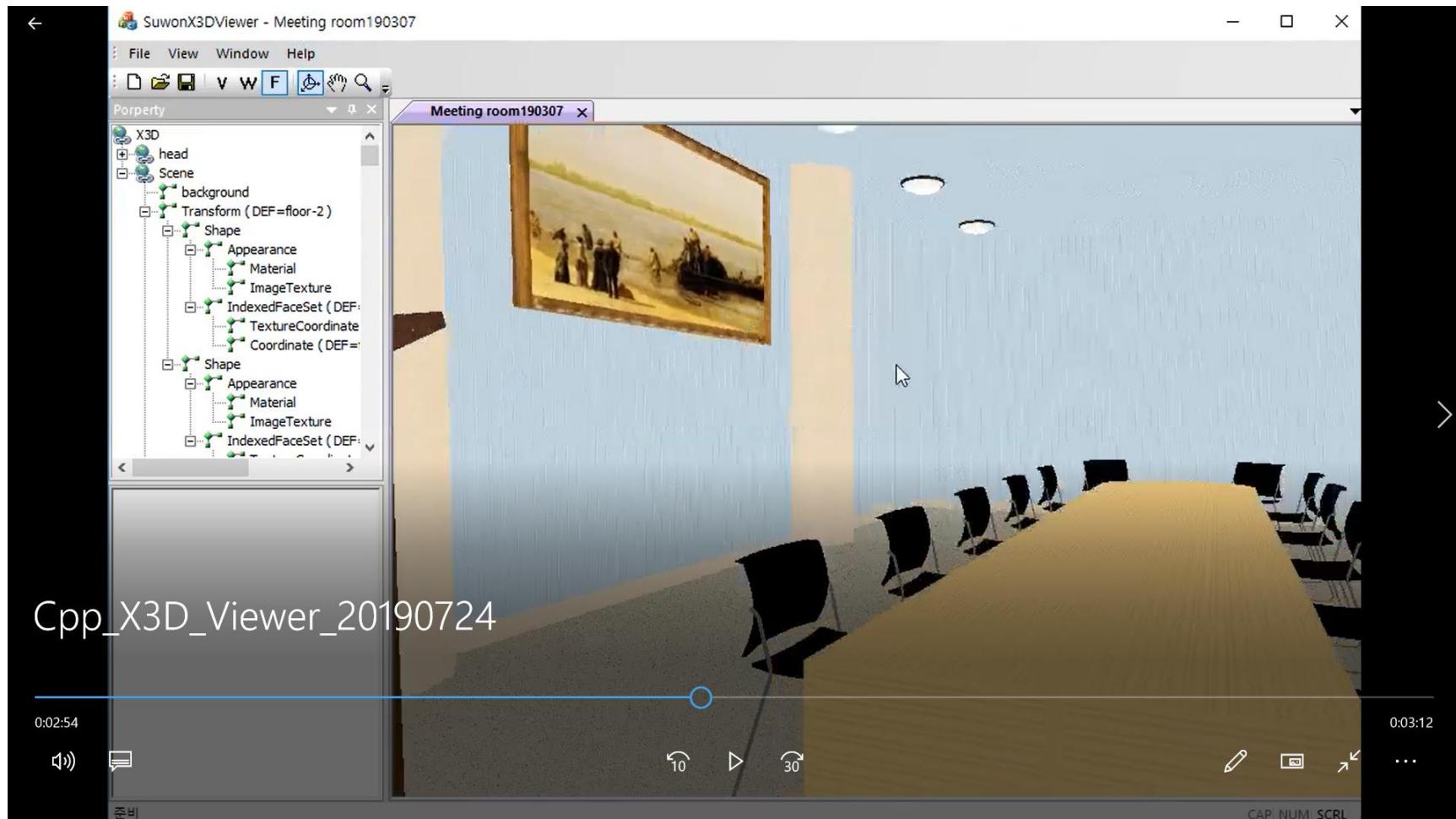


Example – Front

Front



X3D C++ Binding Viewer



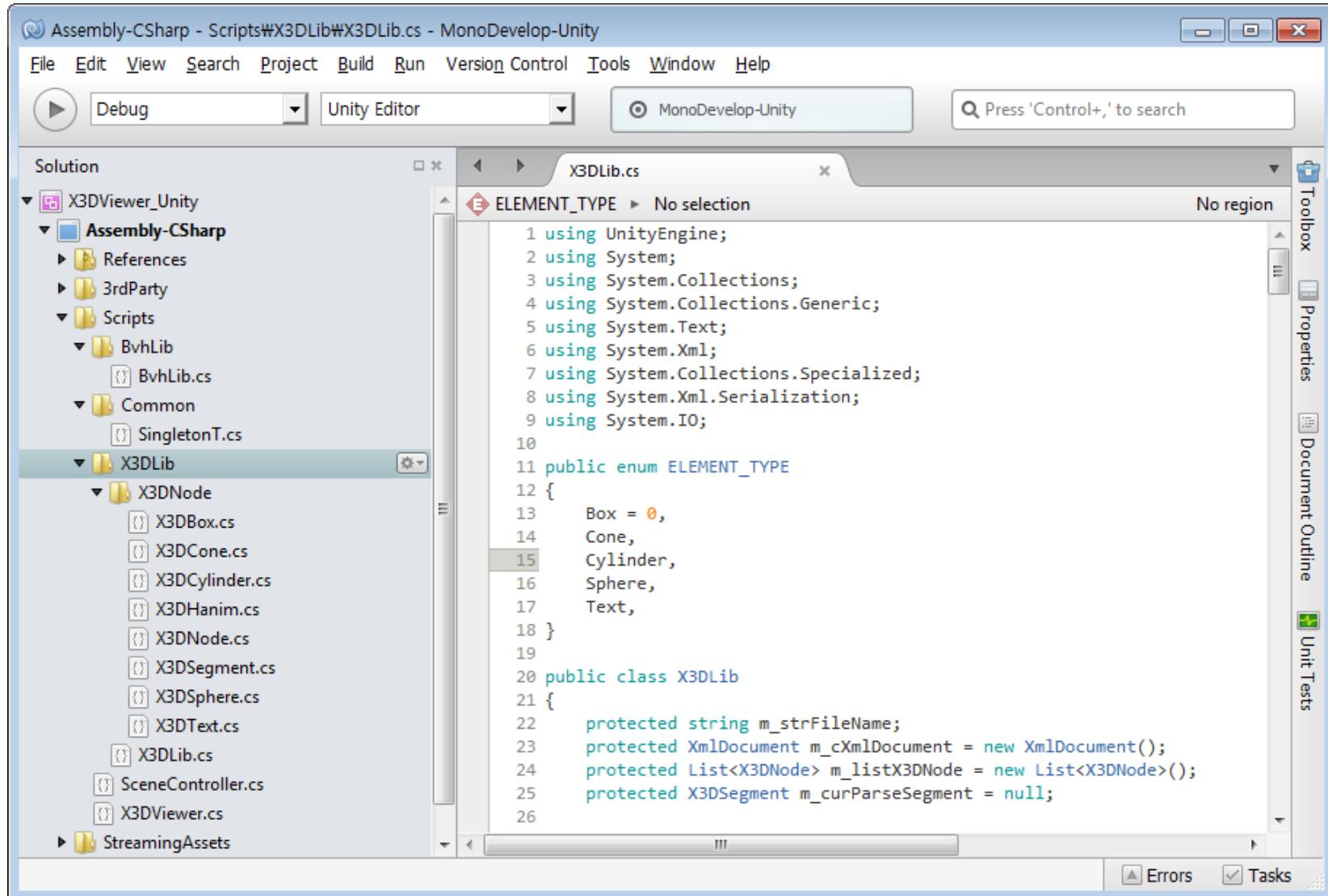
ISO/IEC NP 19777-5

X3D C# Language Binding

Annex C Examples

(Implementation)

X3D C# Binding Viewer (Unity)



The screenshot shows the MonoDevelop-Unity IDE interface. The title bar reads "Assembly-CSharp - Scripts\X3DLib\X3DLib.cs - MonoDevelop-Unity". The menu bar includes File, Edit, View, Search, Project, Build, Run, Version Control, Tools, Window, and Help. The toolbar has buttons for Debug, Unity Editor, and MonoDevelop-Unity. A search bar says "Press 'Control+, to search". The left sidebar is the Solution Explorer showing the project structure:

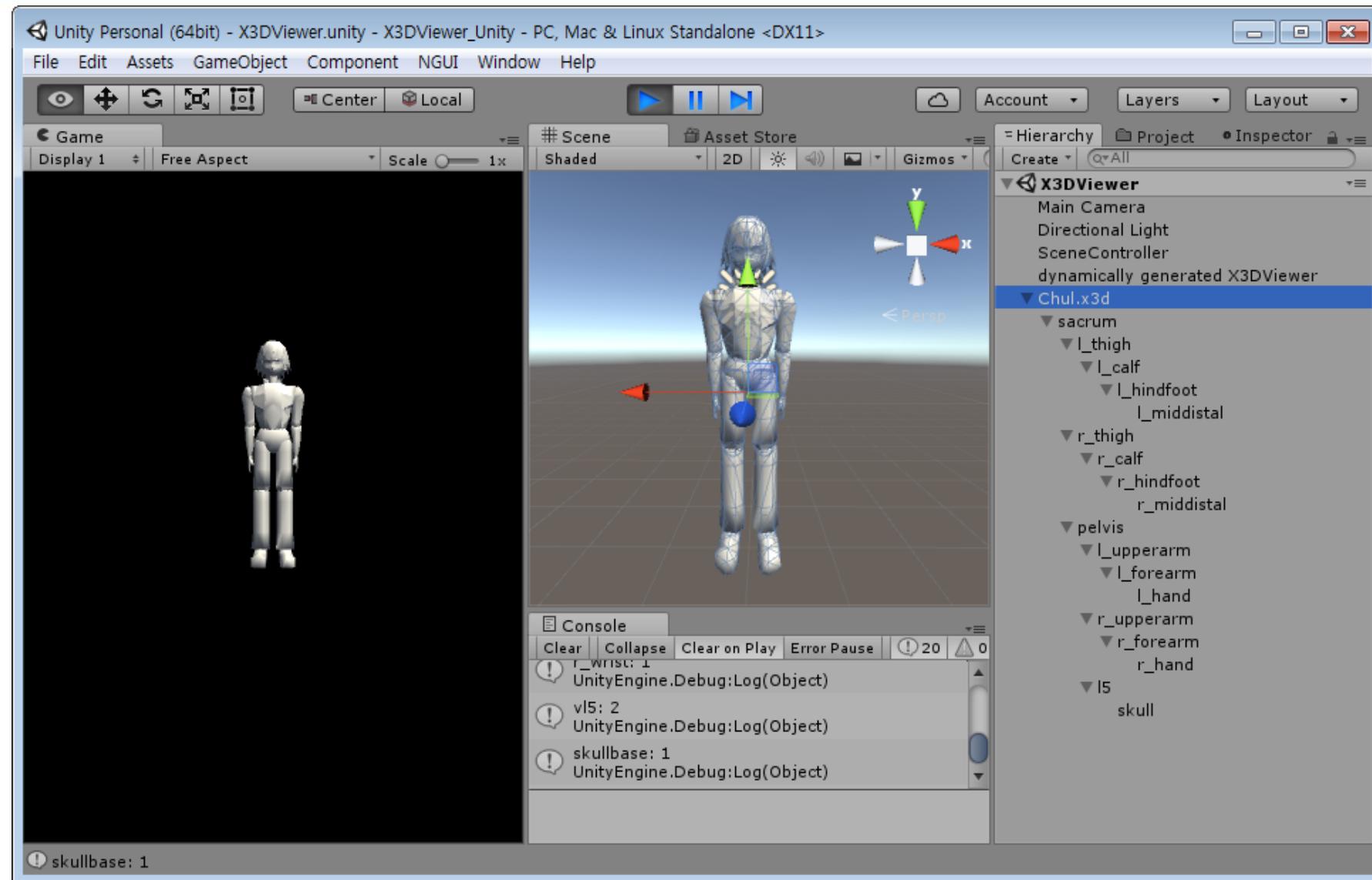
- X3DViewer_Unity
- Assembly-CSharp
 - References
 - 3rdParty
 - Scripts
 - BvhLib
 - BvhLib.cs
 - Common
 - SingletonT.cs
 - X3DLib
 - X3DNode
 - X3DBox.cs
 - X3DCone.cs
 - X3DCylinder.cs
 - X3DAnim.cs
 - X3DNode.cs
 - X3DSegment.cs
 - X3DSphere.cs
 - X3DText.cs
 - X3DLib.cs
 - SceneController.cs
 - X3DViewer.cs
 - StreamingAssets

The main editor window displays the X3DLib.cs file content:

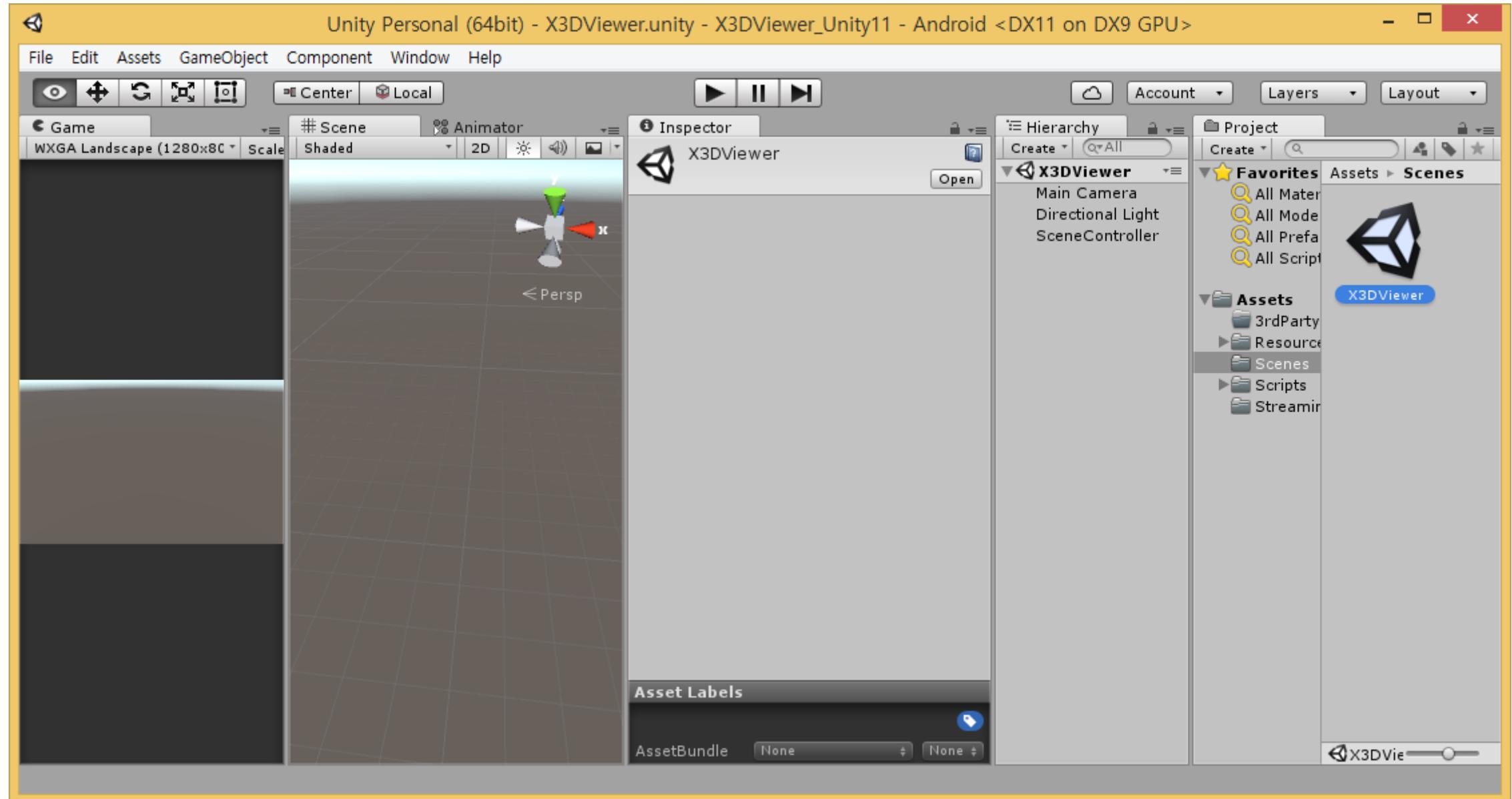
```
1 using UnityEngine;
2 using System;
3 using System.Collections;
4 using System.Collections.Generic;
5 using System.Text;
6 using System.Xml;
7 using System.Collections.Specialized;
8 using System.Xml.Serialization;
9 using System.IO;
10
11 public enum ELEMENT_TYPE
12 {
13     Box = 0,
14     Cone,
15     Cylinder,
16     Sphere,
17     Text,
18 }
19
20 public class X3DLib
21 {
22     protected string m_strFileName;
23     protected XmlDocument m_c XmlDocument = new XmlDocument();
24     protected List<X3DNode> m_listX3DNode = new List<X3DNode>();
25     protected X3DSegment m_curParseSegment = null;
26 }
```

The right sidebar contains tabs for Toolbox, Properties, Document Outline, and Unit Tests. The bottom status bar shows Errors and Tasks.

X3D C# Binding Viewer (Unity)



X3D C# Binding Viewer (Unity)



Lib Class

X3DViewer_Unity11 - Microsoft Visual Studio

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(I) 테스트(S) 분석(N) 창(W) 도움말(H)

빠른 실행(Ctrl+Q) 로그인

서버 탐색기 솔루션 탐색기 검색(Ctrl+.)

도구 상자

X3DNode.cs X3DHanim.cs X3DCylinder.cs X3DCone.cs X3DViewer.cs X3DBox.cs X3DLib.cs

Text

```
1  using UnityEngine;
2  using System;
3  using System.Collections;
4  using System.Collections.Generic;
5  using System.Text;
6  using System.Xml;
7  using System.Collections.Specialized;
8  using System.Xml.Serialization;
9  using System.IO;
10
11 public enum ELEMENT_TYPE
12 {
13     Box = 0,
14     Cone,
15     Cylinder,
16     Sphere,
17     Text,
18 }
19
20 public class X3DLib
21 {
```

100 %

오류 목록 출력 찾기 결과 1 기호 찾기 결과

준비 줄: 17 열: 10 문자: 10 INS ↑ 게시 ▲

Base Node

X3DViewer_Unity11 - Microsoft Visual Studio

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(I) 테스트(S) 분석(N) 창(W) 도움말(H)

빠른 실행(Ctrl+Q) 로그인

솔루션 탐색기 X3DNode.cs X3DHanim.cs X3DCylinder.cs X3DCone.cs X3DViewer.cs SceneController.cs X3DBox.cs

솔루션 탐색기 검색(Ctrl+) BvhLib Common X3DLib X3DNode X3DBox.cs X3DCone.cs X3DCylinder.cs X3DHanim.cs X3DNode.cs X3DSegment.cs X3DSphere.cs X3DText.cs X3Dx3d.cs X3DLib.cs CameraManager.cs SceneController.cs X3DViewer.cs

```
1 using UnityEngine;
2 using System;
3 using System.Collections;
4 using System.Collections.Generic;
5 using System.Text;
6
7 public class X3DNode
8 {
9     protected Vector3 m_vecTranslation;
10    protected Vector4 m_vecRotation;
11    protected Vector3 m_vecScale;
12    protected Vector3 m_vecDiffuseColor;
13
14    public void SetTranslation( Vector3 vec )
15    {
16        m_vecTranslation = vec;
17    }
18    public Vector3 GetTranslation()
19    {
20        return m_vecTranslation;
21    }
22}
```

100 %

오류 목록 출력 찾기 결과 1 기호 찾기 결과

준비 줄: 13 열: 5 문자: 5 INS ↑ 게시 ▲

Box Class

The screenshot shows the Microsoft Visual Studio interface with the title bar "X3DViewer_Unity11 - Microsoft Visual Studio". The menu bar includes File, Edit, View, Project, Build, Debug, Tools, Test, Analyze, Windows, and Help. The toolbar has icons for Undo, Redo, Cut, Copy, Paste, Find, Replace, and others. The status bar at the bottom shows "준비" (Ready), "줄: 12" (Line: 12), "열: 25" (Column: 25), "문자: 25" (Characters: 25), and "INS".

The Solution Explorer on the left shows the project structure:

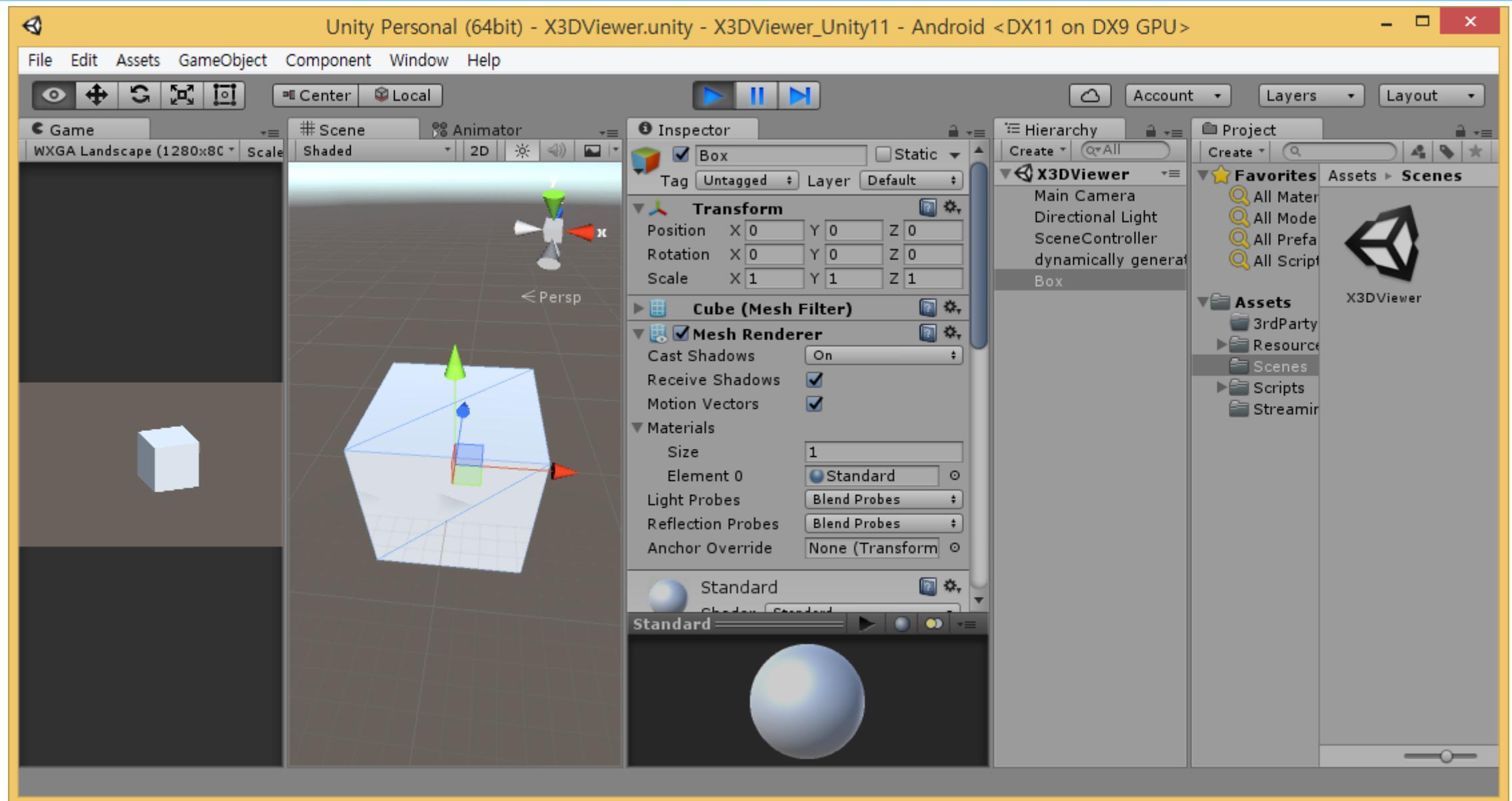
- 解决方案 'X3DViewer_Unity11' (1개 프로젝트)
- ▶ X3DViewer_Unity11
 - ▶ 참조
 - ▶ Assets
 - ▶ Scripts
 - ▶ BvhLib
 - ▶ BvhLib.cs
 - ▶ Common
 - ▶ X3DLib
 - ▶ X3DNode
 - ▶ X3DBox.cs
 - ▶ X3DCone.cs
 - ▶ X3DCylinder.cs
 - ▶ X3DHanim.cs
 - ▶ X3DNode.cs
 - ▶ X3DSegment.cs
 - ▶ X3DSphere.cs
 - ▶ X3DText.cs
 - ▶ X3DView.cs

Box Class

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the project structure for "X3DViewer_Unity11". The "Assets/Scripts/X3DLib/X3DNode" folder contains several C# files: BvhLib.cs, Common.cs, X3DBox.cs, X3DCone.cs, X3DCylinder.cs, X3DHanim.cs, X3DNode.cs, X3DSegment.cs, X3DSphere.cs, X3DTText.cs, and X3DXmlNode.cs. The "X3DBox.cs" file is currently selected.
- Code Editor:** Displays the content of the X3DBox.cs file. The code defines a class X3DBox that creates a 3D box object with specified dimensions. It uses Unity's MeshFilter and MeshRenderer components. The code also initializes vertex and triangle arrays for the box's geometry.
- Status Bar:** At the bottom, it shows "준비" (Ready), "줄: 12" (Line: 12), "열: 25" (Column: 25), "문자: 25" (Characters: 25), and "INS" (Insert mode).

Box Class



H-Anim Class

X3DViewer_Unity11 - Microsoft Visual Studio

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(I) 테스트(S) 분석(N) 창(W) 도움말(H)

빠른 실행(Ctrl+Q) 로그인

솔루션 탐색기 X3DSegment.cs X3DSphere.cs X3DNode.cs X3DHanim.cs X3DCylinder.cs X3DCone.cs X3DBox.cs

솔루션 탐색기 검색(Ctrl+.)

BvhLib.cs Common.cs X3DLib.cs X3DNode.cs X3DSegment.cs X3DSphere.cs X3DText.cs X3Dx3d.cs X3DLib.cs CameraManager.cs SceneController.cs X3DViewer.cs

```
1 using UnityEngine;
2 using System;
3 using System.Collections;
4 using System.Collections.Generic;
5 using System.Text;
6 using System.IO;
7
8 public class X3DHanim : X3DNode
9 {
10     protected BvhLib m_cBvhLib = new BvhLib();
11
12     protected List<X3DSegment> m_listSegment = new List<X3DSegment>();
13     protected string m_strFileName;
14     protected string m_strBvhFileName;
15     protected GameObject m_goRootParent;
16     protected Texture2D m_texture;
17
18     protected bool m_bAnimationPlay = false;
19     protected int m_nAnimationFrame;
20
21     public X3DHanim( string strFileName, string strBvhFileName )
```

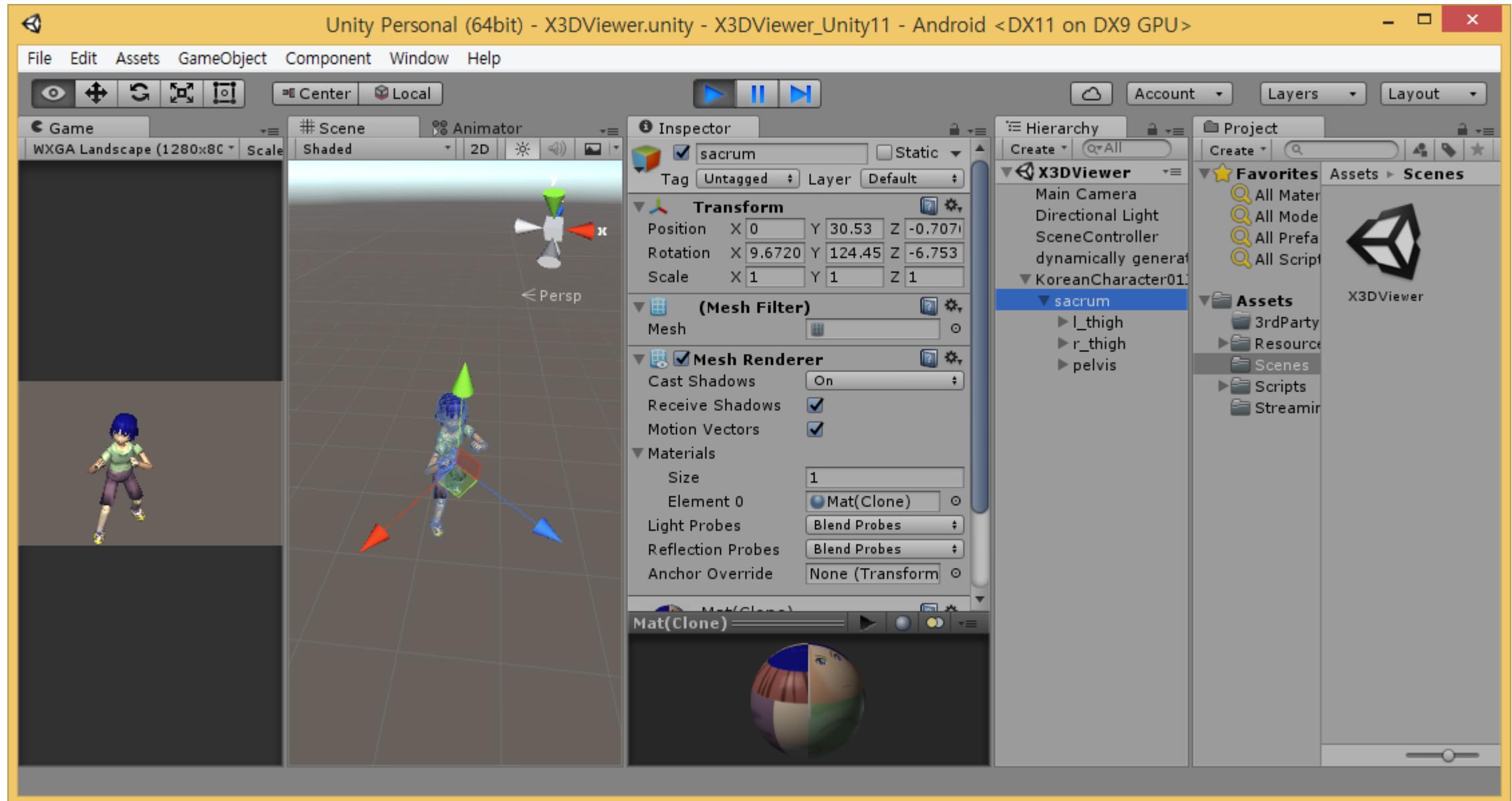
준비 줄: 1 열: 1 문자: 1 INS ↑ 게시 ▾

H-Anim Class

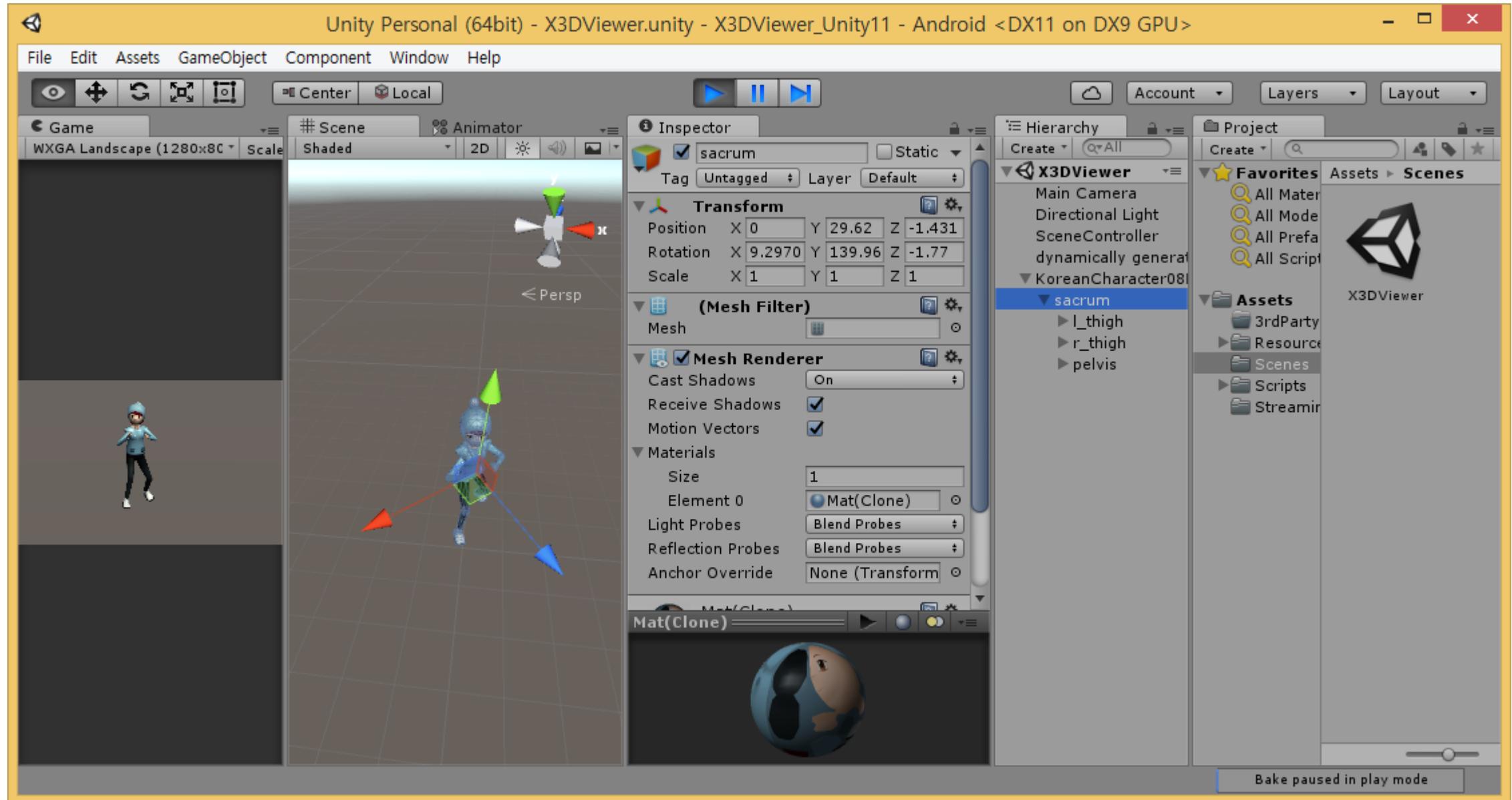
The screenshot shows the Microsoft Visual Studio interface with the project "X3DViewer_Unity11" open. The Solution Explorer on the left lists various C# files under the "X3DLib" and "X3DNode" namespaces. The "X3DHanim.cs" file is selected in the code editor, and its content is displayed below:

```
165     public override void Draw()
166     {
167         if (m_listSegment.Count == 0)
168             return;
169
170         m_goRootParent = new GameObject();
171         m_goRootParent.name = m_strFileName;
172         List<GameObject> listParent = new List<GameObject>();
173
174         //X3DViewer.Instance.m_listDebug.Add("segCount: " + m_listSegment.Count.ToString() );
175
176         for (int i = 0; i < m_listSegment.Count; ++i)
177         {
178             GameObject cGo = new GameObject();
179             m_listSegment[i].SetGoSegment(cGo);
180             cGo.name = m_listSegment[i].GetSegment();
181
182             if (listParent.Count > 0)
183             {
184                 cGo.transform.parent = listParent[listParent.Count - 1].transform;
185                 listParent.RemoveAt(listParent.Count - 1);
186             }
187             else
```

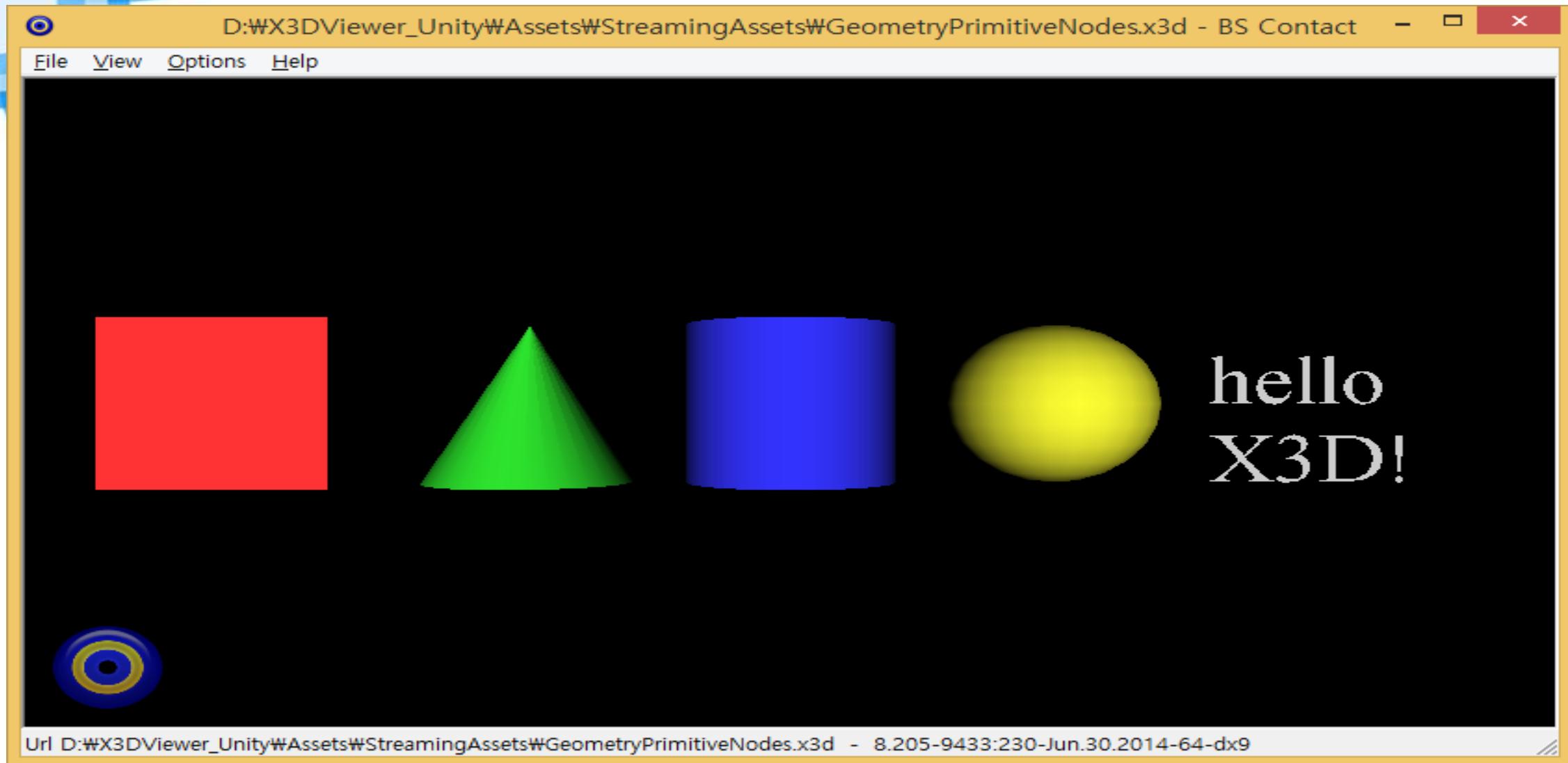
H-Anim Character Animation



H-Anim Character Animation (Video)



GeometryPrimitiveNodes.x3d



Box Parsing

```
case "Box":  
    X3DBox *x3dBox = new X3DBox();  
    x3dBox->SetSize(Parse_Vector3(Parse_AttributeValue(xnRoot, "size")));  
  
    x3dBox->SetTranslation(m_vecParseTranslation);  
    x3dBox->SetRotation(m_vecParseRotation);  
    x3dBox->SetScale(m_vecParseScale);  
    m_listX3DNode.Add(x3dBox);  
    break;
```

Box Class

```
public class X3DBox : X3DNode
{
    protected Vector3 m_vecSize;

    참조 1개
    public void SetSize( Vector3 vec )
    {
        m_vecSize = vec;
    }
    참조 0개
    public void GetSize( ref Vector3 vec )
    {
        vec = m_vecSize;
    }

    참조 9개
    public override IEnumerator LoadEndAction()
    {
        Draw();
        yield return null;
    }

    참조 14개
    public override void Draw()
    {
        GameObject goBox = GameObject.Instantiate(Resources.Load("Box") as GameObject);
        goBox.transform.localPosition = m_vecTranslation;
        goBox.transform.localRotation = Quaternion.Euler(GetRotation());
        goBox.transform.localScale = m_vecSize;

        goBox.GetComponent<MeshRenderer>().material.SetColor("_Color", new Color(m_vecDiffuseColor.x, m_vecDiffuseColor.y, m_vecDiffuseColor.z));
    }
}
```

Box Unity



Cone Parsing

```
case "Cone":  
    X3DCone x3dCone = new X3DCone();  
    x3dCone.SetBottom(Parse_Boolean(Parse_AttributeValue(xnRoot, "bottom")));  
    x3dCone.SetBottomRadius(Parse_Float(Parse_AttributeValue(xnRoot, "bottomRadius")));  
    x3dCone.setHeight(Parse_Float(Parse_AttributeValue(xnRoot, "height")));  
    x3dCone.SetSide(Parse_Boolean(Parse_AttributeValue(xnRoot, "side")));  
  
    x3dCone.SetTranslation(m_vecParseTranslation);  
    x3dCone.SetRotation(m_vecParseRotation);  
    x3dCone.setScale(m_vecParseScale);  
    m_listX3DNode.Add(x3dCone);  
    break;
```

Cone Class

```
public class X3DCone : X3DNode
{
    protected bool m_bBottom;
    protected float m_fBottomRadius;
    protected float m_fHeight;
    protected bool m_bSide;

    참조 1개
    public void SetBottom(bool bBottom)
    {
        m_bBottom = bBottom;
    }
    참조 0개
    public void GetBottom(ref bool bBottom)
    {
        bBottom = m_bBottom;
    }
    2
    참조 1개
    public void SetBottomRadius(float fBottomRadius)
    {
        m_fBottomRadius = fBottomRadius;
    }
    참조 0개
    public void GetBottomRadius(ref float fBottomRadius)
    {
        fBottomRadius = m_fBottomRadius;
    }

    참조 1개
    public void SetHeight(float fHeight)
    {
        m_fHeight = fHeight;
    }
    ...
}
```

Cone Class

```
public void GetHeight(ref float fHeight)
{
    fHeight = m_fHeight;
}

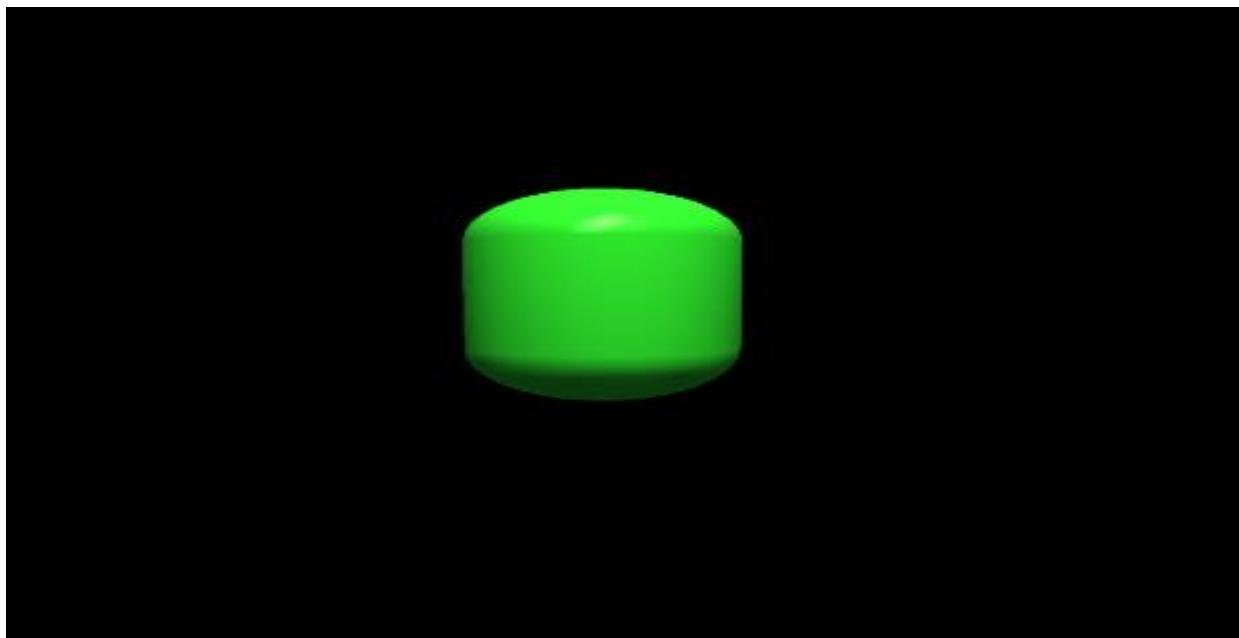
참조 1개
public void SetSide(bool bSide)
{
    m_bSide = bSide;
}
참조 0개
public void GetSide(ref bool bSide)
{
    bSide = m_bSide;
}

참조 9개
public override IEnumerator LoadEndAction()
{
    Draw();
    yield return null;
}

참조 14개
public override void Draw()
{
    GameObject goCone = GameObject.Instantiate(Resources.Load("Cone") as GameObject);
    goCone.transform.localPosition = m_vecTranslation;
    goCone.transform.localRotation = Quaternion.Euler(GetRotation());
    goCone.transform.localScale = new Vector3(m_fBottomRadius * 2f, m_fHeight*0.5f, m_fBottomRadius * 2f);

    goCone.GetComponent<MeshRenderer>().material.SetColor("_Color", new Color(m_vecDiffuseColor.x, m_vecDiffuseColor.y, m_vecDiffuseCol
```

Cone Unity



Cylinder Parsing

```
case "Cylinder":  
    X3DCylinder X3DCylinder = new X3DCylinder();  
    X3DCylinder.SetBottom(Parse_Boolean(Parse_AttributeValue(xnRoot, "bottom")));  
    X3DCylinder.SetRadius(Parse_Float(Parse_AttributeValue(xnRoot, "radius")));  
    X3DCylinder.setHeight(Parse_Float(Parse_AttributeValue(xnRoot, "height")));  
    X3DCylinder.SetSide(Parse_Boolean(Parse_AttributeValue(xnRoot, "side")));  
    X3DCylinder.SetTop(Parse_Boolean(Parse_AttributeValue(xnRoot, "top")));  
  
    X3DCylinder.SetTranslation(m_vecParseTranslation);  
    X3DCylinder.SetRotation(m_vecParseRotation);  
    X3DCylinder.setScale(m_vecParseScale);  
    m_listX3DNode.Add(X3DCylinder);  
    break;
```

Cylinder Class

```
public class X3DCylinder : X3DNode
{
    protected bool m_bBottom;
    protected float m_fRadius;
    protected float m_fHeight;
    protected bool m_bSide;
    protected bool m_bTop;
    2
    참조 1개
    public void SetBottom(bool bBottom)
    {
        m_bBottom = bBottom;
    }
    참조 0개
    public void GetBottom(ref bool bBottom)
    {
        bBottom = m_bBottom;
    }

    참조 1개
    public void SetRadius(float fRadius)
    {
        m_fRadius = fRadius;
    }
    참조 0개
    public void GetRadius(ref float fRadius)
    {
        fRadius = m_fRadius;
    }
}
```

Cylinder Class

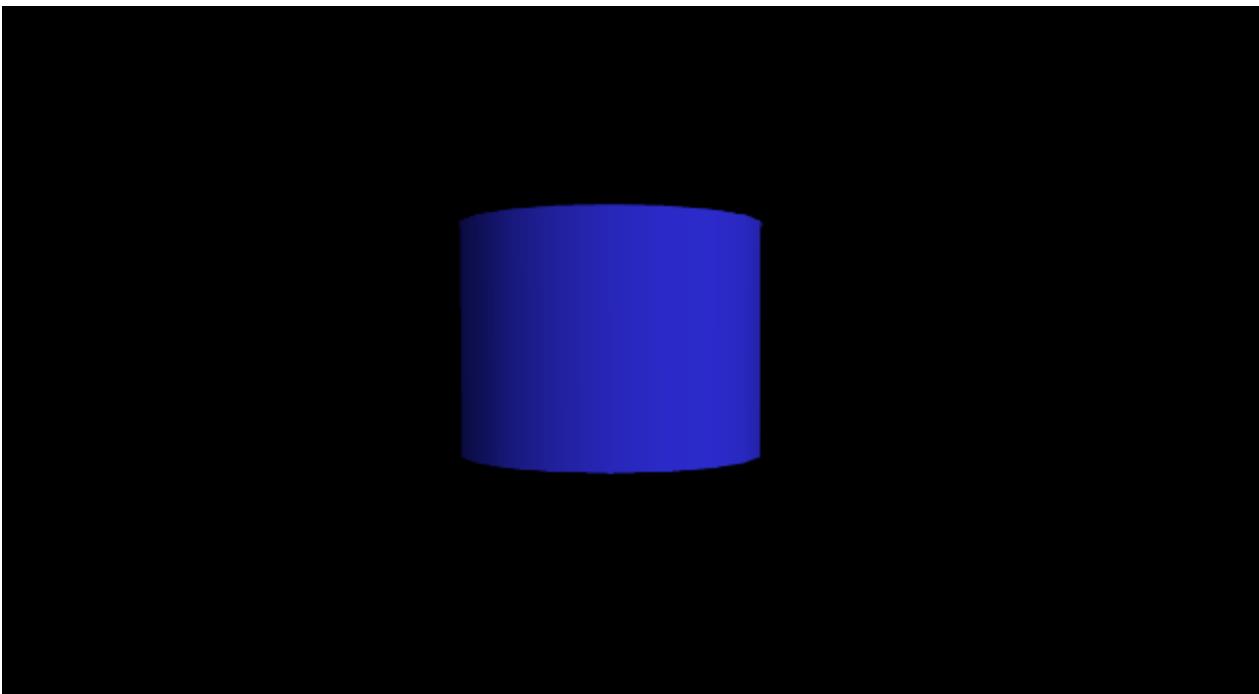
```
public void GetSide(ref bool bSide)
{
    bSide = m_bSide;
}

참조 1개
public void SetTop(bool bTop)
{
    m_bTop = bTop;
}
참조 0개
public void GetTop(ref bool bTop)
{
    bTop = m_bTop;
}

참조 9개
public override IEnumerator LoadEndAction()
{
    Draw();
    yield return null;
}
2
참조 14개
public override void Draw()
{
    GameObject goCylinder = GameObject.Instantiate(Resources.Load("Cylinder") as GameObject);
    goCylinder.transform.localPosition = m_vecTranslation;
    goCylinder.transform.localRotation = Quaternion.Euler(GetRotation());
    goCylinder.transform.localScale = new Vector3(m_fRadius*2f, m_fHeight*0.5f, m_fRadius * 2f);

    goCylinder.GetComponent<MeshRenderer>().materialSetColor("_Color", new Color(m_vecDiffuseColor.x, m_vecDiffuseColor.y,
}
```

Cylinder Unity



Sphere Parsing

```
case "Sphere":  
    X3DSphere *x3dSphere = new X3DSphere();  
    x3dSphere.SetRadius(Parse_Float(Parse_AttributeValue(xnRoot, "radius")));  
  
    x3dSphere.SetTranslation(m_vecParseTranslation);  
    x3dSphere.SetRotation(m_vecParseRotation);  
    x3dSphere.setScale(m_vecParseScale);  
    m_listX3DNode.Add(x3dSphere);  
    break;
```

Sphere Class

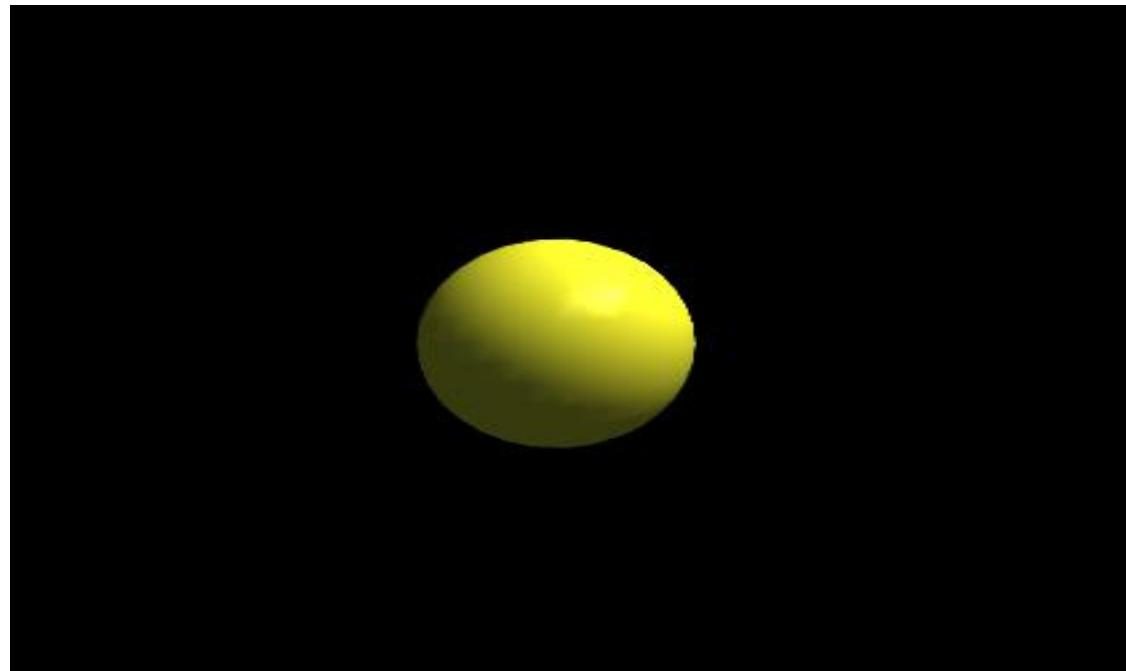
```
public class X3DSphere : X3DNode
{
    protected float m_fRadius;

    참조 1개
    public void SetRadius(float fRadius)
    {
        m_fRadius = fRadius;
    }
    참조 0개
    public void GetRadius(ref float fRadius)
    {
        fRadius = m_fRadius;
    }
    2
    참조 9개
    public override IEnumerator LoadEndAction()
    {
        Draw();
        yield return null;
    }

    참조 14개
    public override void Draw()
    {
        GameObject goSphere = GameObject.Instantiate(Resources.Load("Sphere") as GameObject);
        goSphere.transform.localPosition = m_vecTranslation;
        goSphere.transform.localRotation = Quaternion.Euler(GetRotation());
        goSphere.transform.localScale = new Vector3(m_fRadius * 2f, m_fRadius * 2f, m_fRadius * 2f);

        goSphere.GetComponent<MeshRenderer>().material.SetColor("_Color", new Color(m_vecDiffuseColor.x, m_vecDiffuseColor.y
    }
}
```

Sphere Unity



Text Parsing

```
case "Text":
    X3DText x3dText = new X3DText();
    List<string> listString = Parse_ListString(Parse_AttributeValue(xnRoot, "string"));
    string strText = "";
    for(int i = 0; i < listString.Count; ++i)
    {
        if(i != 0)
        {
            strText += "#n";
        }

        strText += listString[i];
    }
    strText.Replace("#n", "");
    x3dText.SetString(strText);

    x3dText.SetTranslation(m_vecParseTranslation);
    x3dText.SetRotation(m_vecParseRotation);
    x3dText.setScale(m_vecParseScale);
    m_listX3DNode.Add(x3dText);
break;
```

Text Class

```
public class X3DText : X3DNode
{
    protected string m_strString;

    [SerializeField]
    public void SetString(string strString)
    {
        m_strString = strString;
    }

    [SerializeField]
    public void GetString(ref string strString)
    {
        strString = m_strString;
    }

    [SerializeField]
    public override IEnumerator LoadEndAction()
    {
        Draw();
        yield return null;
    }

    [SerializeField]
    public override void Draw()
    {
        GameObject goText = GameObject.Instantiate(Resources.Load("Text") as GameObject);
        goText.transform.localPosition = m_vecTranslation;
        goText.transform.localRotation = Quaternion.Euler(GetRotation());
        goText.transform.localScale = GetScale();

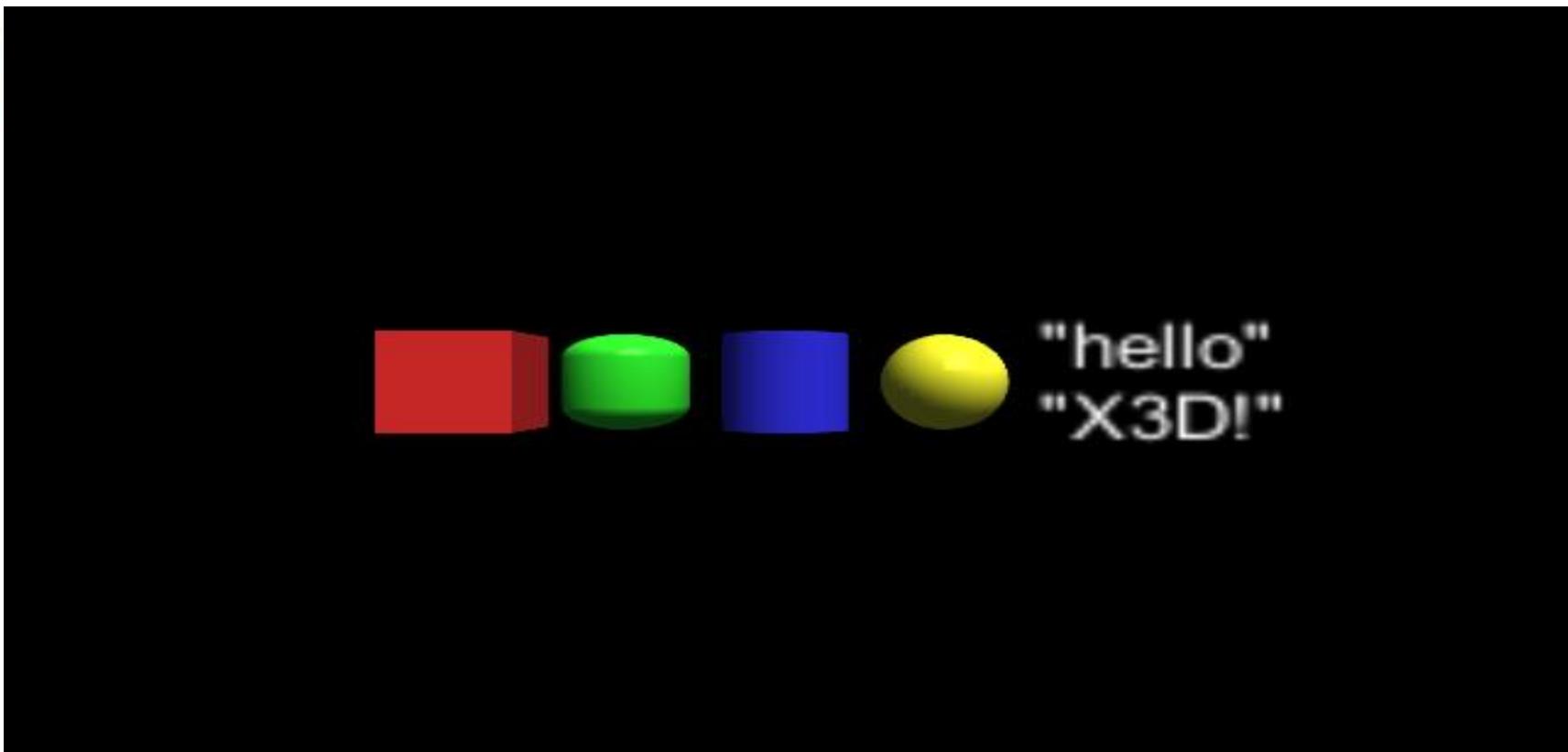
        goText.GetComponent<TextMesh>().text = m_strString;
    }
}
```

Text Unity

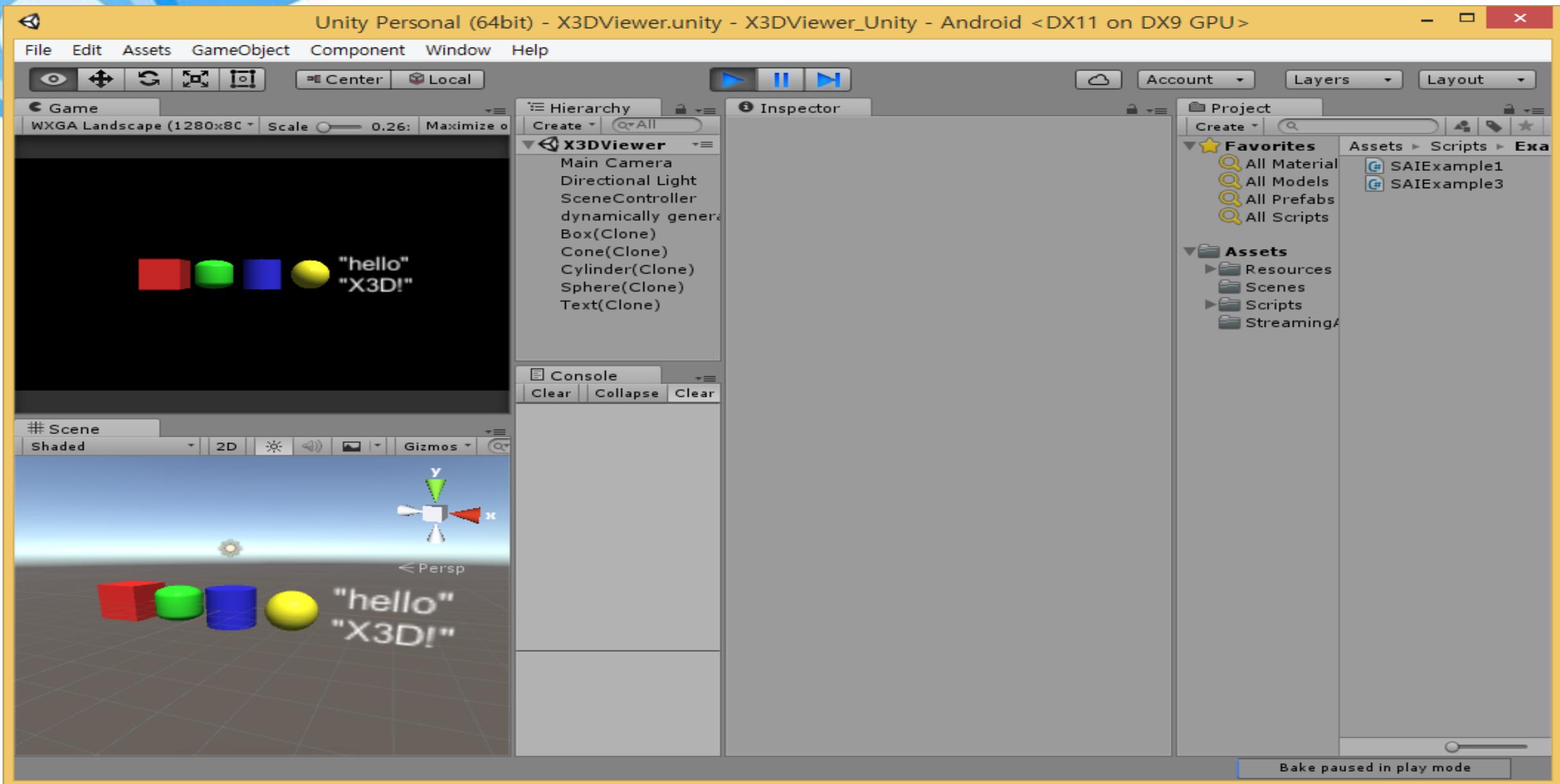


"hello"
"X3D!"

GeometryPrimitiveNodes.x3d Unity



GeometryPrimitiveNodes.x3d Unity

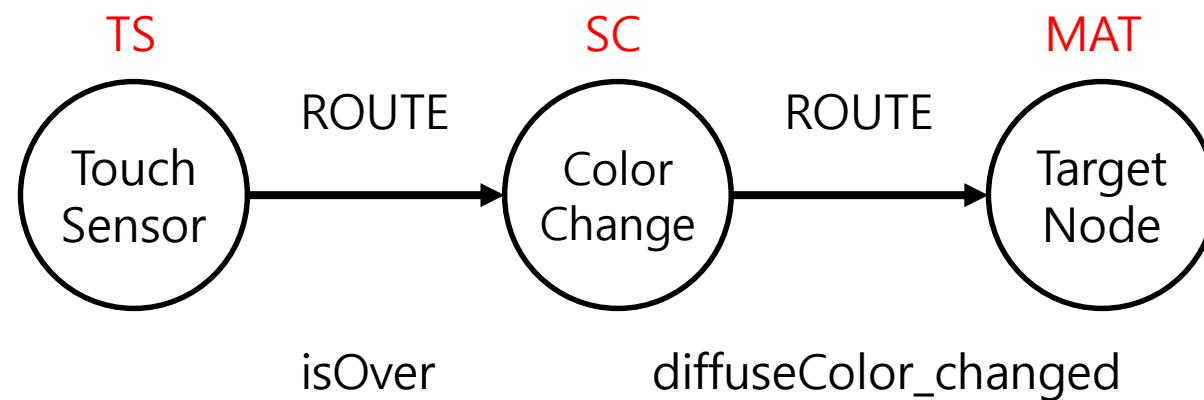


Example 1. TouchSensor isOver event

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Immersive">
  <head>
    <meta content="TouchSensorIsOverEvent.x3d" name="filename"/>
    <meta content="Xeena VRML importer" name="translator"/>
    <meta content="23 February 2005" name="imported"/>
    <meta content="23 February 2005" name="revised"/>
    <meta
content="X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-Edit.html" name="generator"/>
    <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html" name="generator"/>
  </head>
  <Scene>
    <Group>
      <Shape>
        <Appearance>
          <Material DEF="MAT" diffuseColor="0 0 1"/>
        </Appearance>
        <Box/>
      </Shape>
      <TouchSensor DEF="TS"/>
    </Group>
    <Script DEF="SC" url=""SAIExample1.class"">
      <field accessType="inputOnly" name="isOver" type="SFBool"/>
      <field accessType="outputOnly" name="diffuseColor_changed" type="SFColor"/>
    </Script>
    <ROUTE fromField="isOver" fromNode="TS" toField="isOver" toNode="SC"/>
    <ROUTE fromField="diffuseColor_changed" fromNode="SC"
      toField="set_diffuseColor" toNode="MAT"/>
  
```

Example 1. TouchSensor isOver event (Implementation Overview)

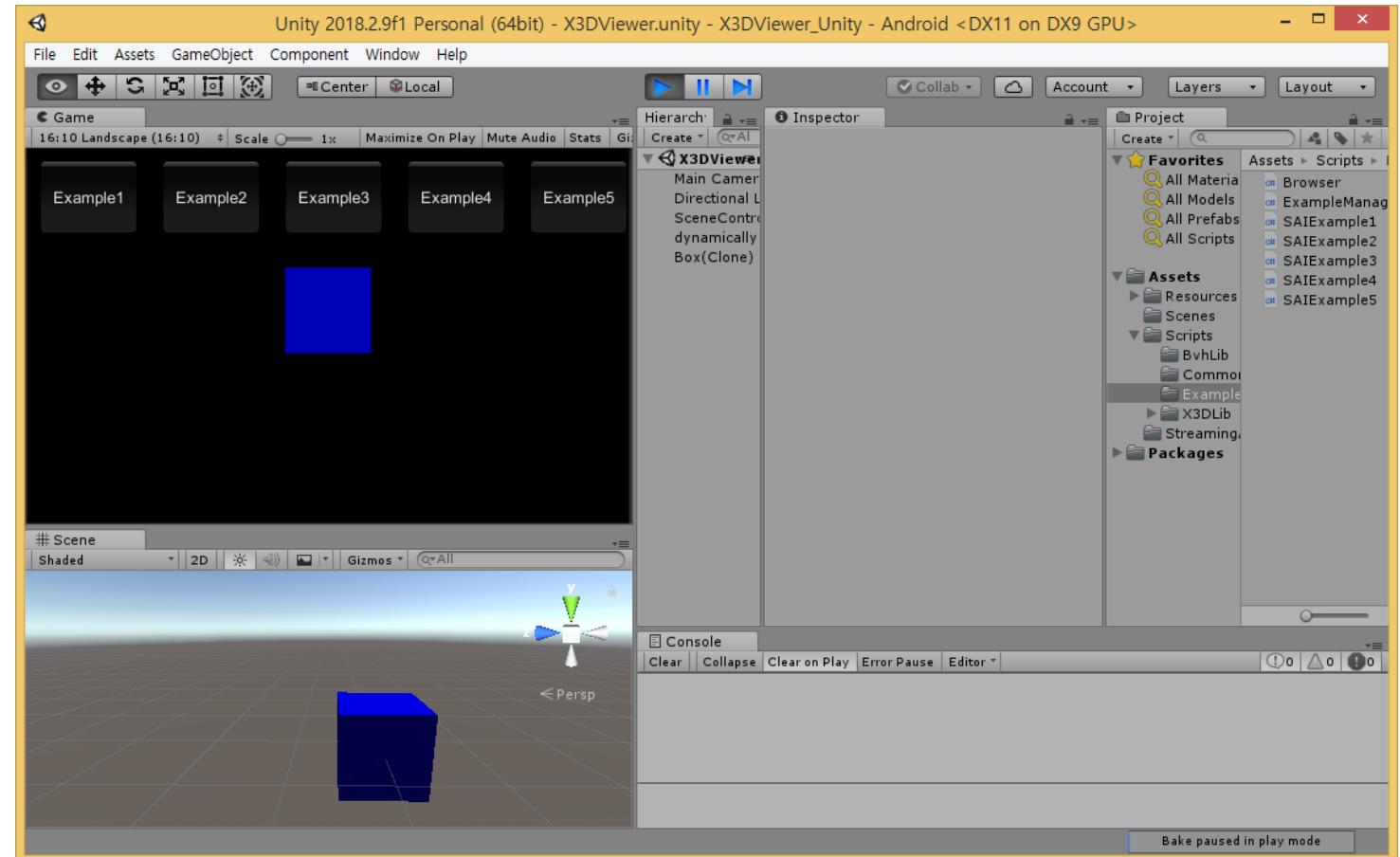
```
<Material DEF="MAT" diffuseColor="0 0 1"/>
<TouchSensor DEF="TS"/>
<Script DEF="SC" url=""SAIExample1.class" ">
  <field accessType="inputOnly" name="isOver" type="SFBool"/>
  <field accessType="outputOnly" name="diffuseColor_changed" type="SFColor"/>
</Script>
<ROUTE fromField="isOver" fromNode="TS" toField="isOver" toNode="SC"/>
<ROUTE fromField="diffuseColor_changed" fromNode="SC" toField="set diffuseColor" toNode="MAT"/>
```



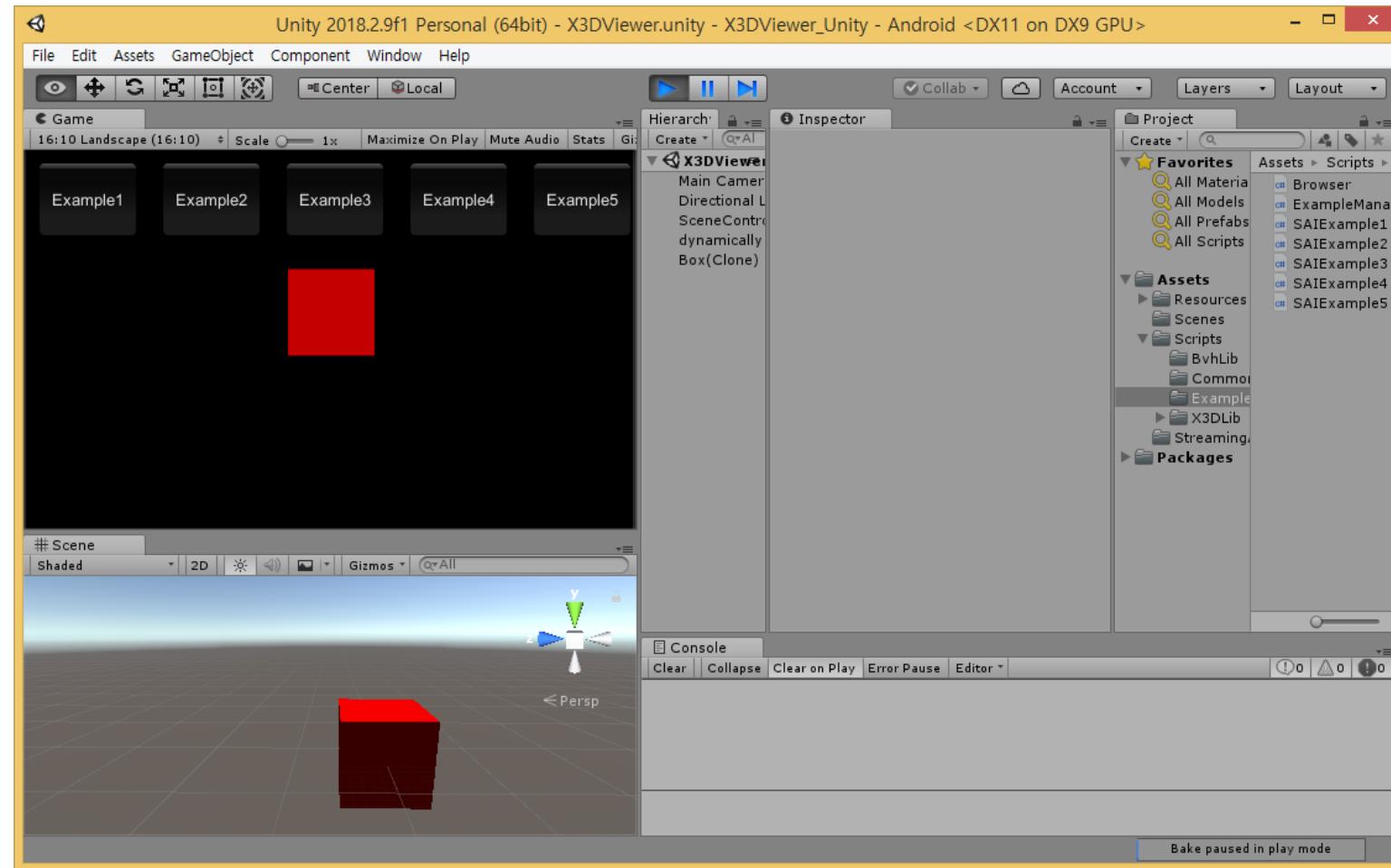
Example 1. TouchSensor isOver event (Implementation) (1)

On click, an event is generated

- On click, the color of the box is changed from blue to red.
- On click in the outside of the box, the color of the box is changed to blue.



Example 1. TouchSensor isOver event (Implementation) (2)



Example 1. TouchSensor isOver event (Implementation) (3)

- Define and create Map, Browser, SFBool classes
- Defined similarly as Java examples
- Unity 3D was used for rendering geometries such as Box

Example 1. TouchSensor isOver event (Implementation) (4)

```
•    using UnityEngine;  
•    using System.Collections;  
•    using System.Collections.Generic;  
  
•    public class SAIExample1 : X3DScriptImplementation, X3DFieldEventListener  
•    {  
•        /** Color Constant, RED */  
•        private static float[] RED = new float[] { 1.0f, 0, 0 };  
  
•        /** Color Constant, BLUE */  
•        private static float[] BLUE = new float[] { 0, 0, 1.0f };  
  
•        /** A mapping for fieldName(String) to an X3DField object */  
•        private Map fields;  
  
•        /** The isOver field */  
•        private SFBool isOver;  
  
•        /** The diffuseColor_changed field */  
•        private Color diffuseColor;  
  
•        private GameObject goBox;
```

Example 1. TouchSensor isOver event (Implementation) (5)

```
• //-----
•     // Methods from the X3DScriptImplementation interface.
•     //-----
•     /**
•      * Set the browser instance to be used by this script implementation.
•      *
•      * @param browser The browser reference to keep
•      */
•     public void setBrowser(Browser browser)
•     {
•     }
•
•     /**
•      * Set the listing of fields that have been declared in the file for
•      * this node. .
•      *
•      * @param The external view of ourselves, so you can add routes to yourself
•      * using the standard API calls
•      * @param fields The mapping of field names to instances
•      */
•     public void setFields(X3DScriptNode externalView, Map fields)
•     {
•         this.fields = fields;
•     }
```

Example 1. TouchSensor isOver event (Implementation) (6)

```
• /**
•  * Notification that the script has completed the setup and should go
•  * about its own internal initialization.
•  */
• public void initialize()
• {
•     isOver = (SFBool)fields.get("isOver");
•     diffuseColor = (Color)fields.get("diffuseColor_changed");
•
•     // Listen to events on isOver
•     isOver.addX3DEventListener(this);
• }
•
• /**
•  * Notification that this script instance is no longer in use by the
•  * scene graph and should now release all resources.
•  */
• public void shutdown()
• {
• }
```

Example 1. TouchSensor isOver event (Implementation) (7)

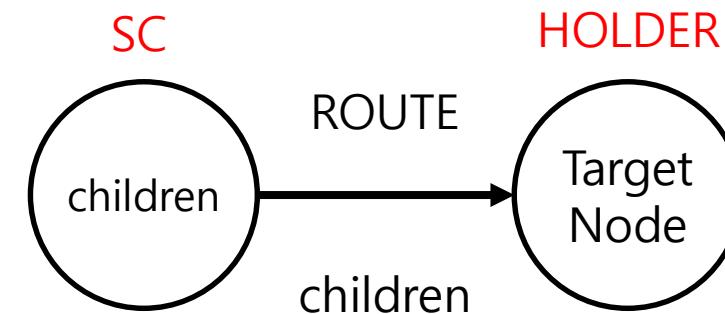
```
•     //-----  
•     // Methods from the X3DFieldEventListener interface.  
•     //-----  
  
•     /**  
•      * Handle field changes.  
•      *  
•      * @param evt The field event  
•      */  
•     public void readableFieldChanged(X3DFieldEvent evt)  
•     {  
•         if (evt.getSource() == isOver)  
•         {  
•             if (isOver.getValue() == true)  
•                 diffuseColor = new Color(RED[0], RED[1], RED[2]);  
•             else  
•                 diffuseColor = new Color(BLUE[0], BLUE[1], BLUE[2]);  
•         }  
•         else  
•         {  
•             Debug.Log("Unhandled event: " + evt);  
•         }  
•     }  
• }
```

Example 2. Create Nodes

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Immersive">
  <head>
    <meta content="CreateNodes.x3d" name="filename"/>
    <meta content="Xeena VRML importer" name="translator"/>
    <meta content="*enter date of initial version here*" name="created"/>
    <meta content="23 February 2005" name="imported"/>
    <meta content="23 February 2005" name="revised"/>
    <meta content="X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-Edit.html" name="generator"/>
    <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html" name="generator"/>
  </head>
  <Scene>
    <Transform DEF="HOLDER" translation="-2.0 0.0 0.0"/>
    <Script DEF="SC" url="SAIExample.class">
      <field accessType="outputOnly" name="children" type="MFNode"/>
    </Script>
    <ROUTE fromField="children" fromNode="SC" toField="children" toNode="HOLDER"/>
  </Scene>
</X3D>
```

Example 2. Create Nodes (Implementation Overview)

```
<Transform DEF="HOLDER" translation="-2.0 0.0 0.0"/>
<Script DEF="SC" url="SAIExample.class">
  <field accessType="outputOnly" name="children" type="MFNode"/>
</Script>
<ROUTE fromField="children" fromNode="SC" toField="children"
toNode="HOLDER"/>
```



Example 2. Create Nodes (Implementation) (1)

```
•    using UnityEngine;  
•    using System.Collections;  
•    using System.Collections.Generic;  
  
•    public class SAIExample2 : X3DScriptImplementation  
•    {  
  
•        /** Color Constant, RED */  
•        private static float[] RED = new float[] { 1.0f, 0, 0 };  
  
•        /** A mapping for fieldName(String) to an X3DField object */  
•        private Map fields;  
  
•        /** A reference to the browser */  
•        private Browser browser;  
  
•        /** The field to place the generated nodes via createX3DFromString */  
•        private MFNode children;
```

Example 2. Create Nodes (Implementation) (2)

```
• //-----
•     // Methods from the X3DScriptImplementation interface.
•     //-----
•     /**
•      * Set the browser instance to be used by this script implementation.
•      *
•      * @param browser The browser reference to keep
•      */
•     public void setBrowser(Browser browser)
•     {
•         this.browser = browser;
•     }
•
•     /**
•      * Set the listing of fields that have been declared in the file for
•      * this node. .
•      *
•      * @param The external view of ourselves, so you can add routes to yourself
•      *       using the standard API calls
•      * @param fields The mapping of field names to instances
•      */
•     public void setFields(X3DScriptNode externalView, Map fields)
•     {
•         this.fields = fields;
•     }
```

Example 2. Create Nodes (Implementation) (3)

```
• /**
•  * Notification that the script has completed the setup and should go
•  * about its own internal initialization.
•  */
• public void initialize()
• {
•     children = (MFNode)fields.get("children");
•
•     // Create nodes directly in the parent scene
•     X3DScene scene = (X3DScene)browser.getExecutionContext();
•
•     X3DShapeNode shape = (X3DShapeNode)scene.createNode("Shape");
•     X3DGeometryNode box = (X3DGeometryNode)scene.createNode("Box");
•
•     shape.setGeometry(box);
•     scene.addRootNode(shape);
•
•     // Create children using the createX3DFromString service
•     string vrmlCmd = "PROFILE Interchange Shape { geometry Sphere{} }";
•
•     X3DScene tmpScene = browser.createX3DFromString(vrmlCmd);
•     X3DNode[] nodes = tmpScene.getRootNodes();
```

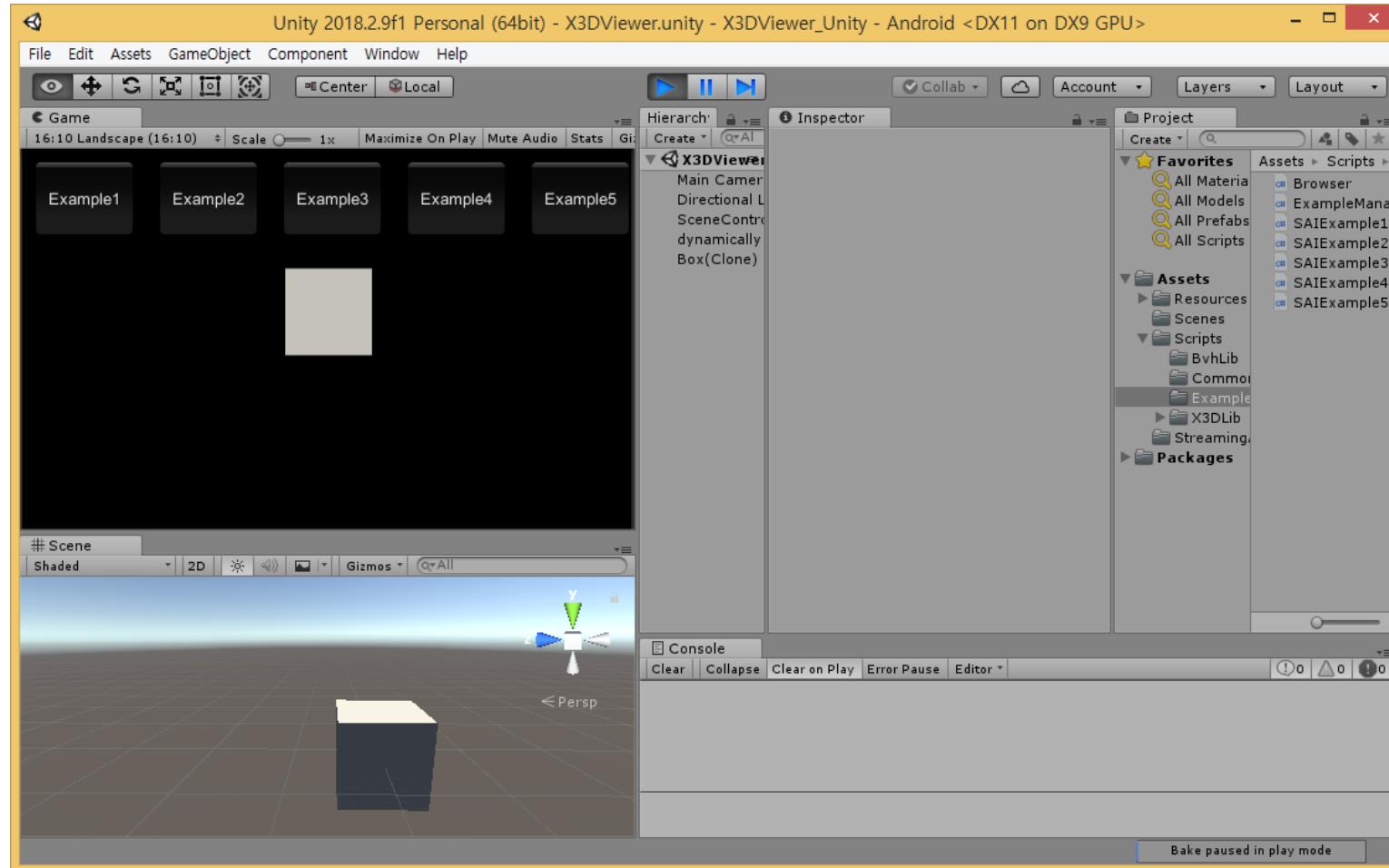
Example 2. Create Nodes (Implementation) (4)

```
• /**
•  * Notification that this script instance is no longer in use by the
•  * scene graph and should now release all resources.
•  */
• public void shutdown()
• {
• }

• /**
•  * Notification that all the events in the current cascade have finished
•  * processing.
•  */
• public void eventsProcessed()
• {
• }
```

Example 2. Create Nodes (Implementation) (5)

프로그램 기동시 초기화 처리에서
Scene에 Shape>Box생성

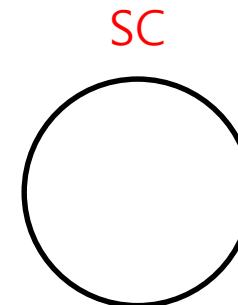


Example 3. Per frame notification

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "http://www.web3d.org/specifications/x3d-3.0.dtd"
  "file:///www.web3d.org/TaskGroups/x3d/translation/x3d-3.0.dtd">
<X3D profile="Immersive">
  <head>
    <meta content="PerFrameNotification.x3d" name="filename"/>
    <meta content="Xeena VRML importer" name="translator"/>
    <meta content="23 February 2005" name="imported"/>
    <meta content="23 February 2005" name="revised"/>
    <meta
      content="X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-Edit.html" name="generator"/>
    <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html" name="generator"/>
  </head>
  <Scene>
    <Script DEF="SC" url=""SAIExample3.class" "/>
  </Scene>
</X3D>
```

Example 3. Per frame notification (Implementation Overview)

```
<Script DEF="SC" url=""SAIExample3.class" "/>
```



Initialize() : Initialize Time value

prepareEvents() : FPS is calculated and displayed

Example 3. Per frame notification (Implementation) (1)

```
•    using UnityEngine;  
•    using System.Collections;  
•    using System.Collections.Generic;  
  
•    public class SAIExample3 : X3DPerFrameObserverScript  
•    {  
•        /** When did the last frame start */  
•        private float lastStartTime;  
  
•        //-----  
•        // Methods from the X3DScriptImplementation interface.  
•        //-----  
•        /**  
•         * Set the browser instance to be used by this script implementation.  
•         *  
•         * @param browser The browser reference to keep  
•         */  
•        public void setBrowser(Browser browser)  
•        {  
•        }
```

Example 3. Per frame notification (Implementation) (2)

```
• /**
•  * Set the listing of fields that have been declared in the file for
•  * this node. .
•  *
•  * @param The external view of ourselves, so you can add routes to yourself
•  * using the standard API calls
•  * @param fields The mapping of field names to instances
•  */
• public void setFields(X3DScriptNode externalView, Map fields)
• {
• }

• /**
•  * Notification that the script has completed the setup and should go
•  * about its own internal initialization.
•  */
• public void initialize()
• {
•     lastStartTime = Time.deltaTime;
• }
```

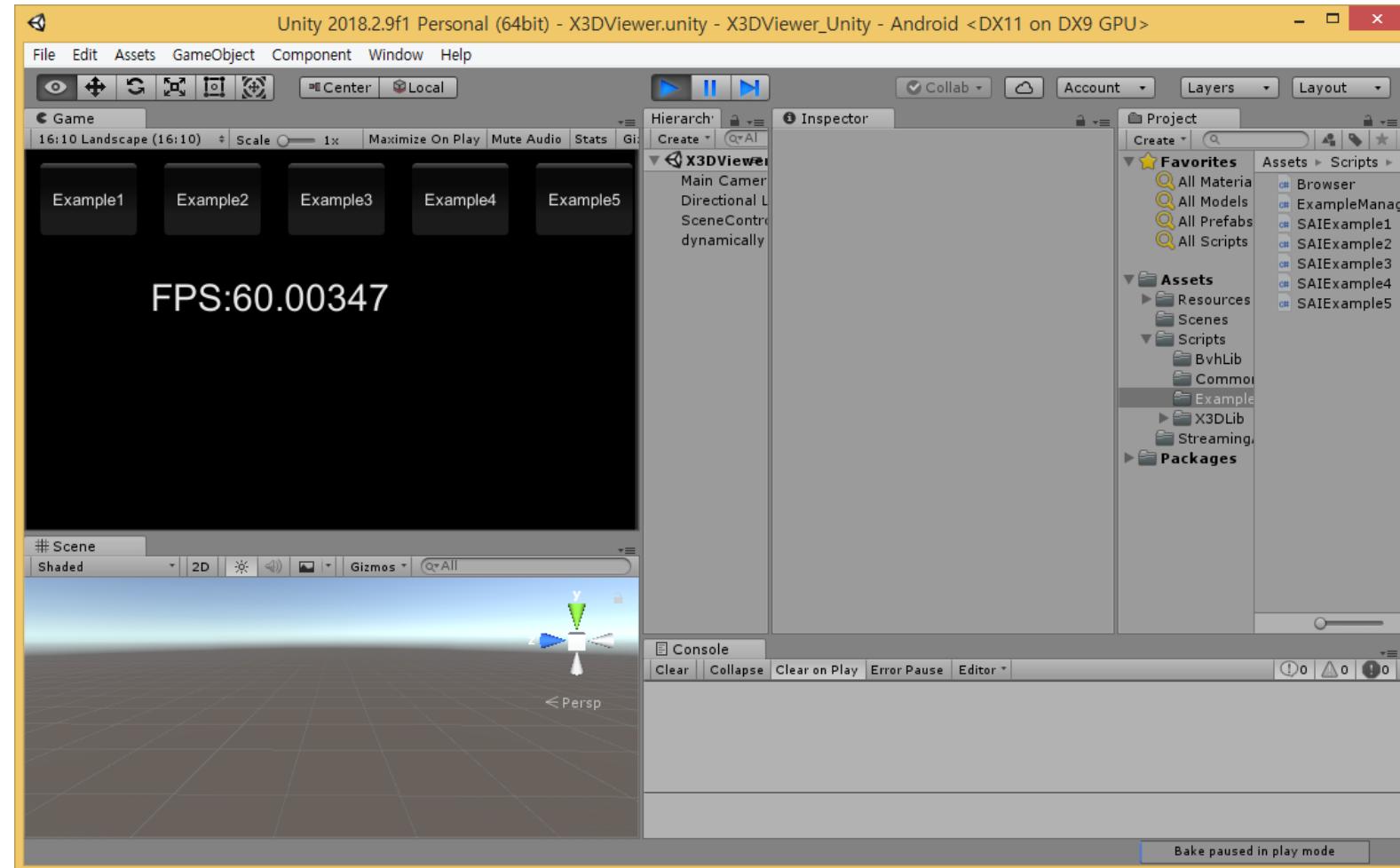
Example 3. Per frame notification (Implementation) (3)

```
• //-----
• // Methods from the X3DPerFrameObserver interface.
• //-----
• 
• /**
• * Start of frame notification.
• */
• public void prepareEvents()
• {
•     float frameTime = Time.deltaTime;
•     float fps = 1.0f / frameTime;
• 
•     GUI.skin.label.fontSize = 30;
•     GUI.Label(new Rect(100f, 100f, 1000f, 1000f), "FPS:" + fps);
• }
```

Example 3. Per frame notification (Implementation Results)

prepareEvents()

FPS is calculated and displayed

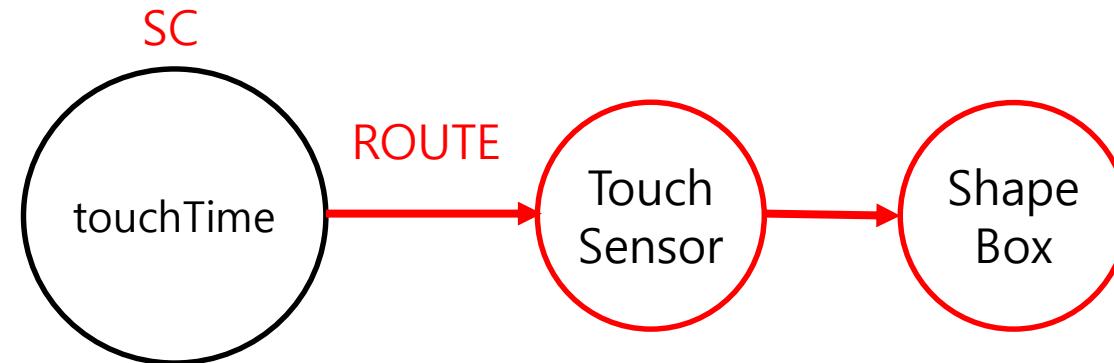


Example 4. Add dynamic routes

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Immersive">
<Scene>
<Script DEF="SC" url="SAIExample4.class">
  <field accessType="inputOnly" name="touchTime" type="SFTime"/>
</Script>
</Scene>
</X3D>
```

Example 4. Add dynamic routes (Implementation Overview)

```
<Script DEF="SC" url="SAIExample4.class">
    <field accessType="inputOnly" name="touchTime" type="SFTime"/>
</Script>
```



Initialize :

Read "touchTime" field.

Create Shape(Box), TouchSensor under Group in the scene

Create "touchTime" ROUTE

Connect TouchSensor to the scene

readableFieldChanged()

When TouchSensor is generated, "Poke!" is displayed.

Example 4. Add dynamic routes (Implementation) (1)

```
• using UnityEngine;  
• using System.Collections;  
• using System.Collections.Generic;  
  
• public class SAIExample4 : X3DScriptImplementation, X3DFieldEventListener  
• {  
•     /** A mapping for fieldName(String) to an X3DField object */  
•     private Map fields;  
  
•     /** A reference to the browser */  
•     private Browser browser = new Browser();  
  
•     /** inputOnly touchTime */  
•     private SFTime touchTime;  
  
•     /** initializeOnly selfRef */  
•     private X3DScriptNode selfRef;  
  
•     private GameObject goBox;
```

Example 4. Add dynamic routes (Implementation) (2)

```
• //-----
•     // Methods from the X3DScriptImplementation interface.
•     //-----
•     /**
•      * Set the browser instance to be used by this script implementation.
•      *
•      * @param browser The browser reference to keep
•      */
•     public void setBrowser(Browser browser)
•     {
•         this.browser = browser;
•     }
•
•     /**
•      * Set the listing of fields that have been declared in the file for
•      * this node. .
•      *
•      * @param The external view of ourselves, so you can add routes to yourself
•      *       using the standard API calls
•      * @param fields The mapping of field names to instances
•      */
•     public void setFields(X3DScriptNode externalView, Map fields)
•     {
•         this.fields = fields;
•         selfRef = externalView;
•     }
```

Example 4. Add dynamic routes (Implementation) (3)

```
• /**
•  * Notification that the script has completed the setup and should go
•  * about its own internal initialization.
•  */
• public void initialize()
• {
•     touchTime = (SFTime)fields.get("touchTime");
•
•     // Listen to events on touchTime
•     touchTime.addX3DEventListener(this);
•
•     // Create nodes directly in the parent scene
•     X3DScene scene = (X3DScene)browser.getExecutionContext();
•
•     X3DShapeNode shape = (X3DShapeNode)scene.createNode("Shape");
•     X3DGeometryNode box = (X3DGeometryNode)scene.createNode("Box");
•     X3DNode touchSensor = scene.createNode("TouchSensor");
•
•     shape.setGeometry(box);
•
•     // Create a Group to hold the nodes
•     X3DGroupingNode group = (X3DGroupingNode)scene.createNode("Group");
```

Example 4. Add dynamic routes (Implementation) (4)

```
• //-----
• // Methods from the X3DFieldEventListener interface.
• //-----
• 
• /**
• * Handle field changes.
• *
• * @param evt The field event
• */
• public void readableFieldChanged(X3DFieldEvent evt)
• {
•     if (evt.getSource() == touchTime)
•     {
•         GUI.skin.label.fontSize = 30;
•         GUI.Label(new Rect(100f, 100f, 1000f, 1000f), "Poke!");
•     }
•     else
•     {
•         Debug.Log("Unhandled event: " + evt);
•     }
• }
```

Create nodes from a prototype

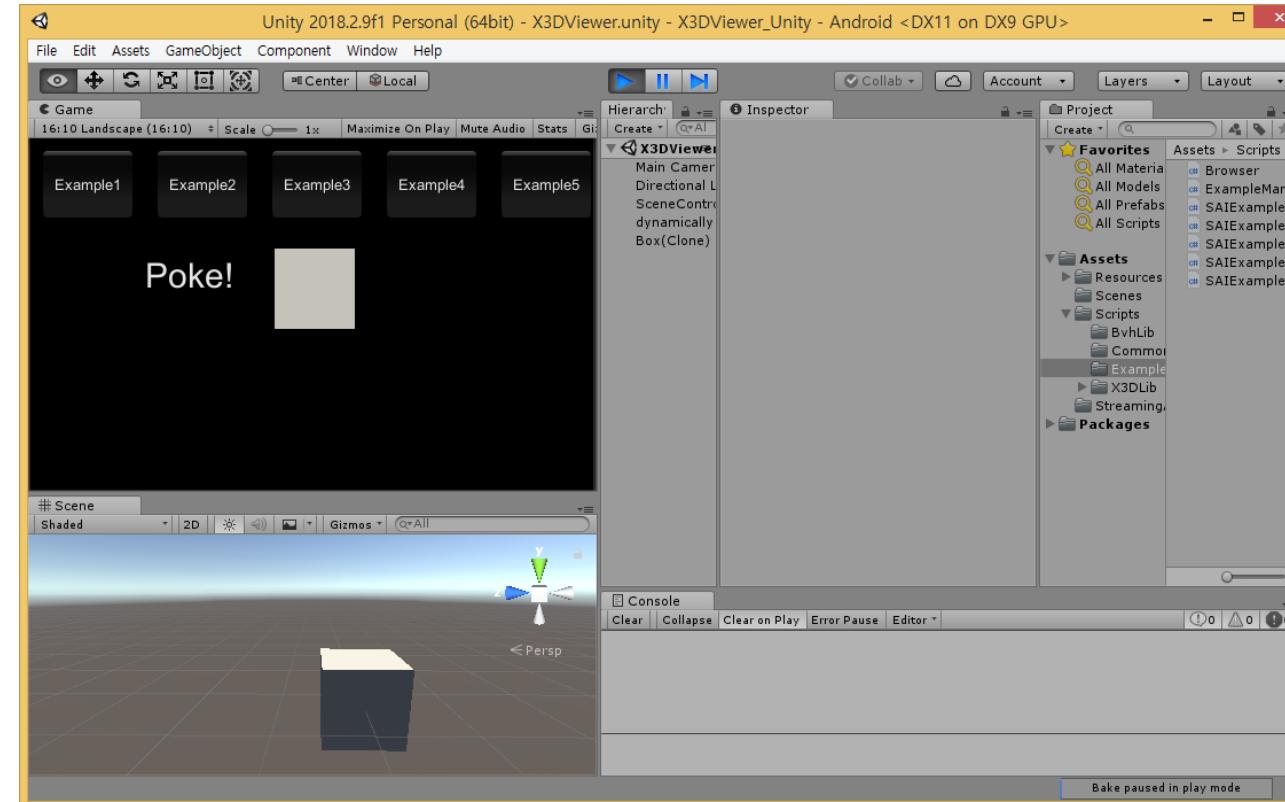
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Immersive">
<Scene>
  <ProtoDeclare name="ColoredSphere">
    <ProtoInterface>
      <field accessType="initializeOnly" name="color" type="SFColor" value="0.0 0.0 0.0"/>
    </ProtoInterface>
    <ProtoBody>
      <Shape>
        <Appearance>
          <Material>
            <IS>
              <connect nodeField="diffuseColor" protoField="color"/>
            </IS>
          </Material>
        </Appearance>
        <Sphere/>
      </Shape>
    </ProtoBody>
  </ProtoDeclare>
  <Script DEF="SC" url="SAIExample5.class"/>
</Scene>
</X3D>
```

Example 4. Add dynamic routes (Implementation Results)

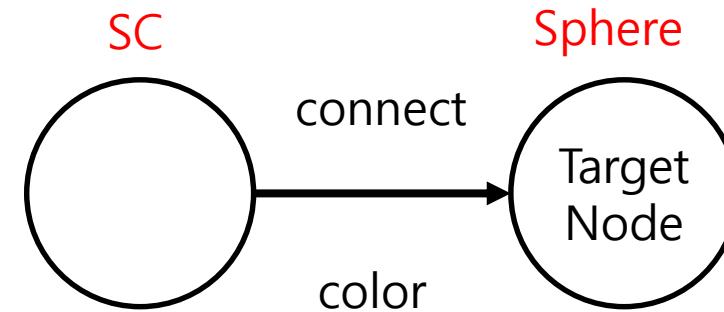
In the initialization

Create Scene > Group > Shape > Box > TouchSensor

On clicking Box, "Poke!" is displayed



Example 5. Create nodes from a prototype (Implementation Overview)



Initialize() :

Read “color” field under ProtoDeclare and ProtolInterface
Set RED value to diffuseColor of Material

Example 5. Create nodes from a prototype (Implementation) (1)

```
• using UnityEngine;  
• using System.Collections;  
• using System.Collections.Generic;  
  
• public class SAIExample5 : X3DScriptImplementation  
• {  
•     /** Color Constant, RED */  
•     private static float[] RED = new float[] { 1.0f, 0, 0 };  
  
•     /** A mapping for fieldName(String) to an X3DField object */  
•     private Map fields;  
  
•     /** A reference to the browser */  
•     private Browser browser;  
  
•     /** The field to place the generated nodes via createX3DFromString */  
•     private MFNode children;
```

Example 5. Create nodes from a prototype (Implementation) (2)

```
• //-----
•     // Methods from the X3DScriptImplementation interface.
•     //-----
•     /**
•      * Set the browser instance to be used by this script implementation.
•      *
•      * @param browser The browser reference to keep
•      */
•     public void setBrowser(Browser browser)
•     {
•         this.browser = browser;
•     }
•
•     /**
•      * Set the listing of fields that have been declared in the file for
•      * this node. .
•      *
•      * @param The external view of ourselves, so you can add routes to yourself
•      *       using the standard API calls
•      * @param fields The mapping of field names to instances
•      */
•     public void setFields(X3DScriptNode externalView, Map fields)
•     {
•         this.fields = fields;
•     }
```

Example 5. Create nodes from a prototype (Implementation) (3)

```
• /**
•  * Notification that the script has completed the setup and should go
•  * about its own internal initialization.
•  */
• public void initialize()
• {
•     // Create nodes directly in the parent scene
•     X3DScene scene = (X3DScene)browser.getExecutionContext();
•
•     // Create protoInstance nodes
•
•     // Get the proto declaration declared in the main scene
•     X3DProtoDeclaration protoDecl = scene.getProtoDeclaration("ColoredSphere");
•
•     // Create a new instance of this proto
•     X3DProtoInstance instance = protoDecl.createInstance();
•
•     // Get the color field and set it to red
•     SFColor color = (SFColor)instance.getField("color");
•     color.setValue(RED);
•
•     // Add the created proto instance to the scene
•     scene.addRootNode(instance);
• }
```

Example 5. Create nodes from a prototype (Implementation) (4)

```
• /**
•  * Notification that this script instance is no longer in use by the
•  * scene graph and should now release all resources.
•  */
• public void shutdown()
• {
• }

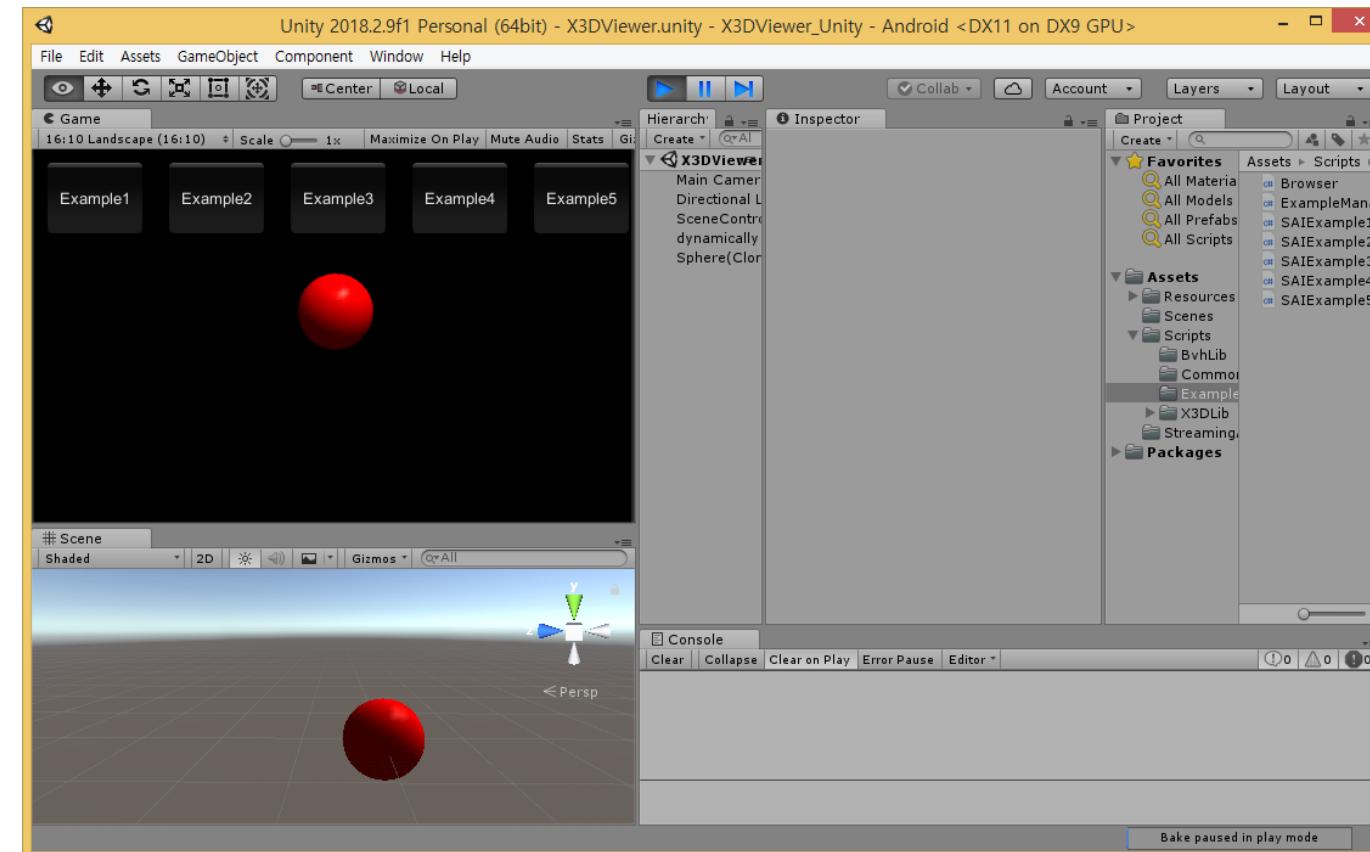
• /**
•  * Notification that all the events in the current cascade have finished
•  * processing.
•  */
• public void eventsProcessed()
• {
• }
```

Example 5. Create nodes from a prototype (Implementation Results)

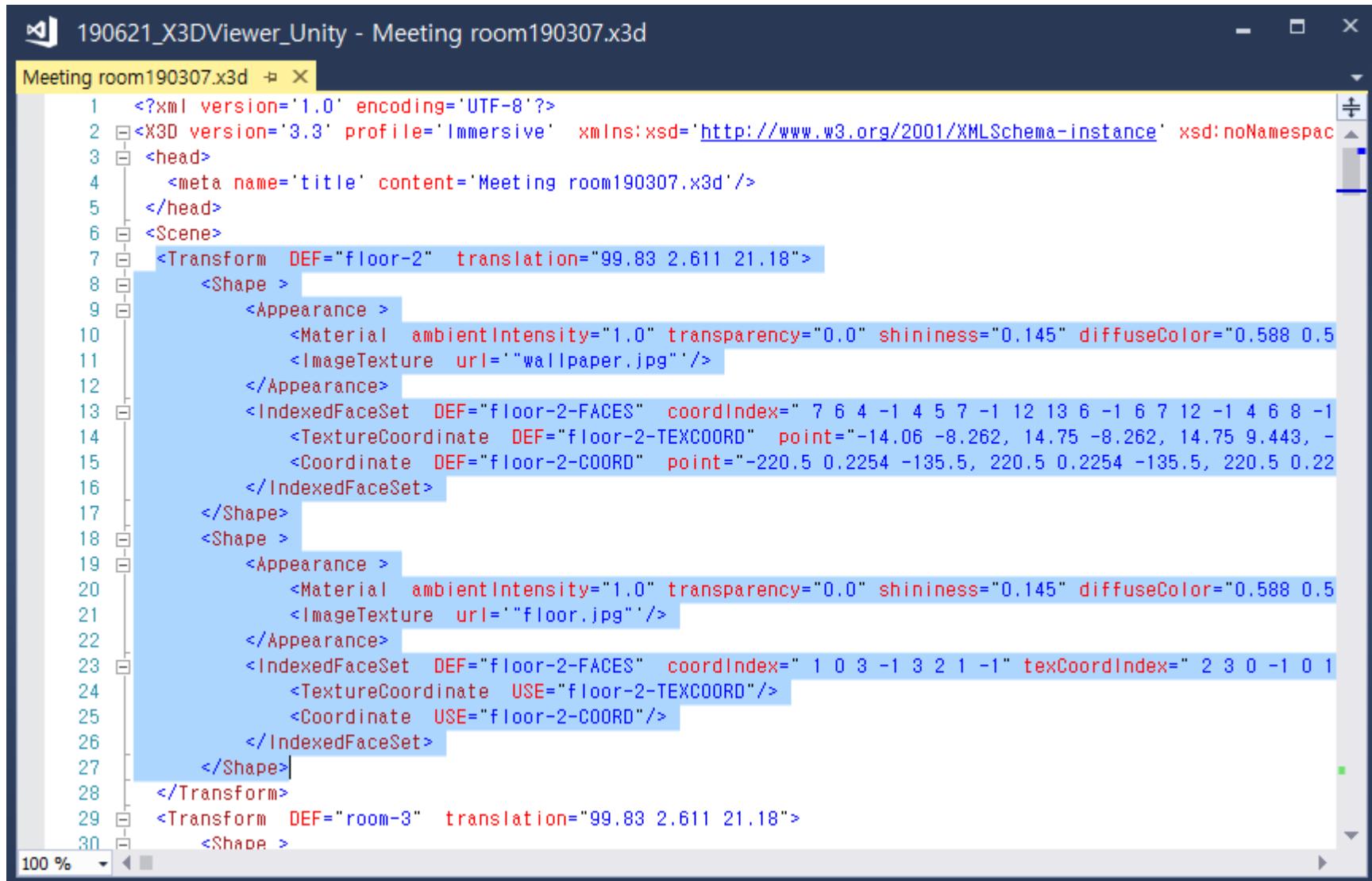
In the initialization, read a ColoredSphere in the scene.

Read "color" field.

Set RED to "color"



Multi-Objects Structure (1)



The screenshot shows the X3DViewer_Unity application window with the title "190621_X3DViewer_Unity - Meeting room190307.x3d". The main area displays the XML code for a 3D scene named "Meeting room190307.x3d". The code defines two main objects: "floor-2" and "room-3".

```
<?xml version='1.0' encoding='UTF-8'?>
<X3D version='3.3' profile='Immersive' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.3.xsd'>
  <head>
    <meta name='title' content='Meeting room190307.x3d'/>
  </head>
  <Scene>
    <Transform DEF="floor-2" translation="99.83 2.611 21.18">
      <Shape>
        <Appearance>
          <Material ambientIntensity="1.0" transparency="0.0" shininess="0.145" diffuseColor="0.588 0.588 0.588">
            <ImageTexture url='wallpaper.jpg'/>
          </Material>
          <IndexedFaceSet DEF="floor-2-FACES" coordIndex=" 7 6 4 -1 4 5 7 -1 12 13 6 -1 6 7 12 -1 4 6 8 -1 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 
```

Multi-Objects Structure (2)

190621_X3DViewer_Unity - Meeting room190307.x3d

Meeting room190307.x3d

```
23     <IndexedFaceSet DEF="floor-2-FACES" coordIndex=" 1 0 3 -1 3 2 1 -1" texCoordIndex=" 2 3 0 -1 0 1
24         <TextureCoordinate USE="floor-2-TEXCOORD"/>
25         <Coordinate USE="floor-2-COORD"/>
26     </IndexedFaceSet>
27   </Shape>
28 </Transform>
29 <Transform DEF="room-3" translation="99.83 2.611 21.18">
30   <Shape >
31     <Appearance >
32       <Material ambientIntensity="1.0" transparency="0.0" shininess="0.145" diffuseColor="0.588 0.5
33       <ImageTexture url='wallpaper2.jpg' />
34     </Appearance>
35     <IndexedFaceSet DEF="room-3-FACES" coordIndex=" 6 5 2 -1 2 0 6 -1 15 14 7 -1 7 3 15 -1 1 3 7 -1
36         <TextureCoordinate DEF="room-3-TEXCOORD" point="-4.886 -7.471, -4.886 -5.474, -6.769 -5.474,
37         <Coordinate DEF="room-3-COORD" point="-220.5 143.7 -135.5, 220.5 143.7 -135.5, -220.5 143.7
38     </IndexedFaceSet>
39   </Shape>
40   <Shape >
41     <Appearance >
42       <Material ambientIntensity="1.0" transparency="0.0" shininess="0.145" diffuseColor="0.588 0.5
43       <ImageTexture url='wallpaper.jpg' />
44     </Appearance>
45     <IndexedFaceSet DEF="room-3-FACES" coordIndex=" 52 59 60 -1 60 53 52 -1 53 60 61 -1 61 54 53 -1
46         <TextureCoordinate USE="room-3-TEXCOORD"/>
47         <Coordinate USE="room-3-COORD"/>
48     </IndexedFaceSet>
49   </Shape>
50   <Shape >
51     <Appearance >
52       <Material ambientIntensity="1.0" transparency="0.0" shininess="0.145" diffuseColor="0.588 0.5
```

Multi-Objects Implementation

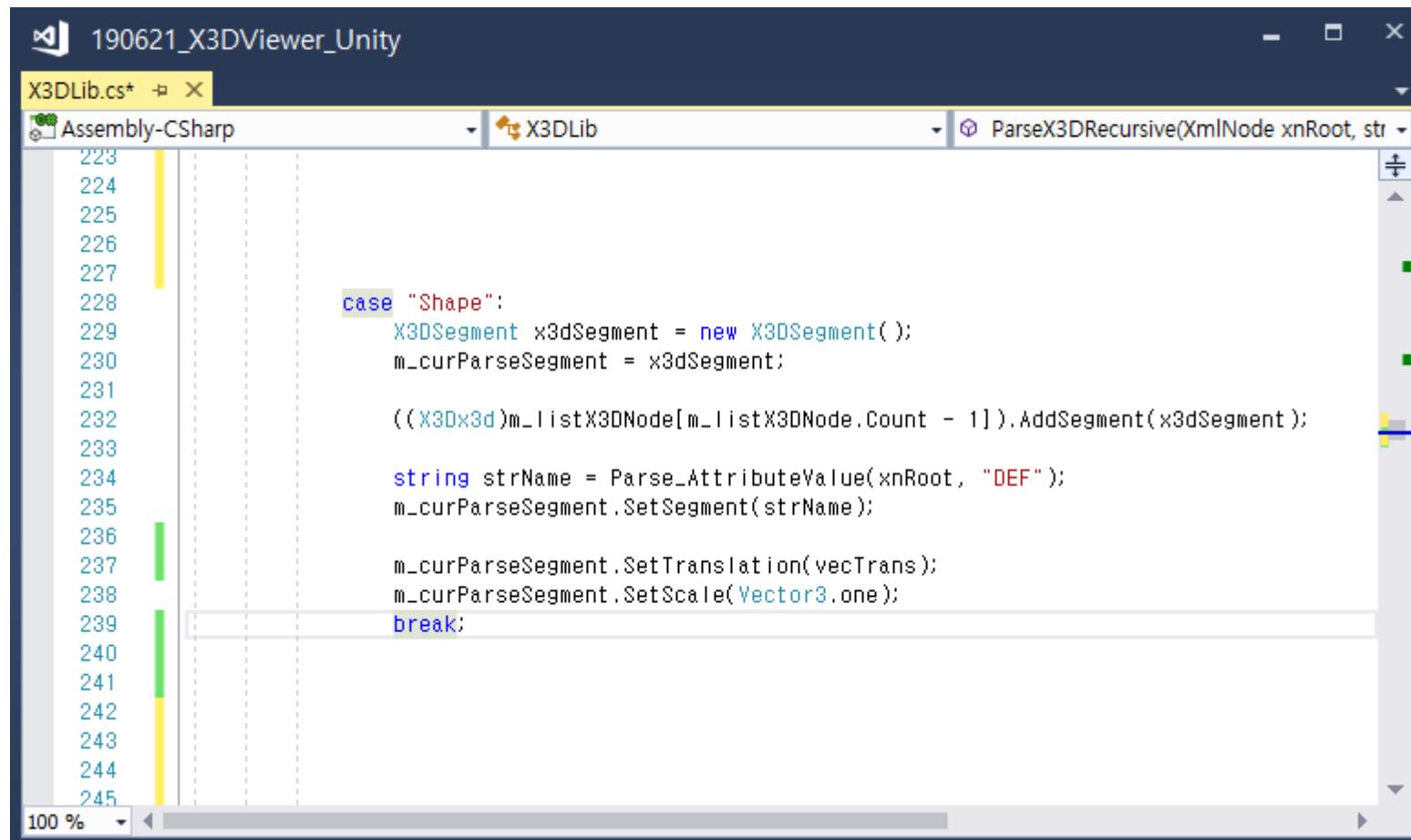
190621_X3DViewer_Unity

X3Dx3d.cs X

Assembly-CSharp X3Dx3d m_listSegment

```
1  using UnityEngine;
2  using System;
3  using System.Collections;
4  using System.Collections.Generic;
5  using System.Text;
6  using System.IO;
7
8  public class X3Dx3d : X3DNode
9  {
10     protected List<X3DSegment> m_listSegment = new List<X3DSegment>();
11     protected string m_strFileName;
12     protected GameObject m_goRootParent;
13
14
15     public void SetFileName(string strFileName)
16     {
17         m_strFileName = strFileName;
18     }
19
20     public override void Destroy()
21     {
22         base.Destroy();
23
24         GameObject.Destroy(m_goRootParent);
25
26         m_listSegment.Clear();
27     }
28
29     public void AddSegment(X3DSegment seg)
30     {
```

Multi-Objects Implementation



The screenshot shows a Microsoft Visual Studio IDE window titled "190621_X3DViewer_Unity". The active tab is "X3DLib.cs*". The code editor displays a portion of the "X3DLib" class, specifically the "ParseX3DRecursive(XmlNode xnRoot, string strName)" method. The code handles the "Shape" node type, creating a new "X3DSegment" object, setting it as the current parse segment, adding it to the list of segments for the current node, and then setting its name and translation/scale. A "break;" statement is present at line 239.

```
223  
224  
225  
226  
227  
228     case "Shape":  
229         X3DSegment x3dSegment = new X3DSegment();  
230         m_curParseSegment = x3dSegment;  
231  
232         ((X3Dx3d)m_listX3DNode[m_listX3DNode.Count - 1]).AddSegment(x3dSegment);  
233  
234         string strName = Parse_AttributeValue(xnRoot, "DEF");  
235         m_curParseSegment.SetSegment(strName);  
236  
237         m_curParseSegment.SetTranslation(vecTrans);  
238         m_curParseSegment.setScale(Vector3.one);  
239         break;  
240  
241  
242  
243  
244  
245
```

Multi-Textures Structure (1)

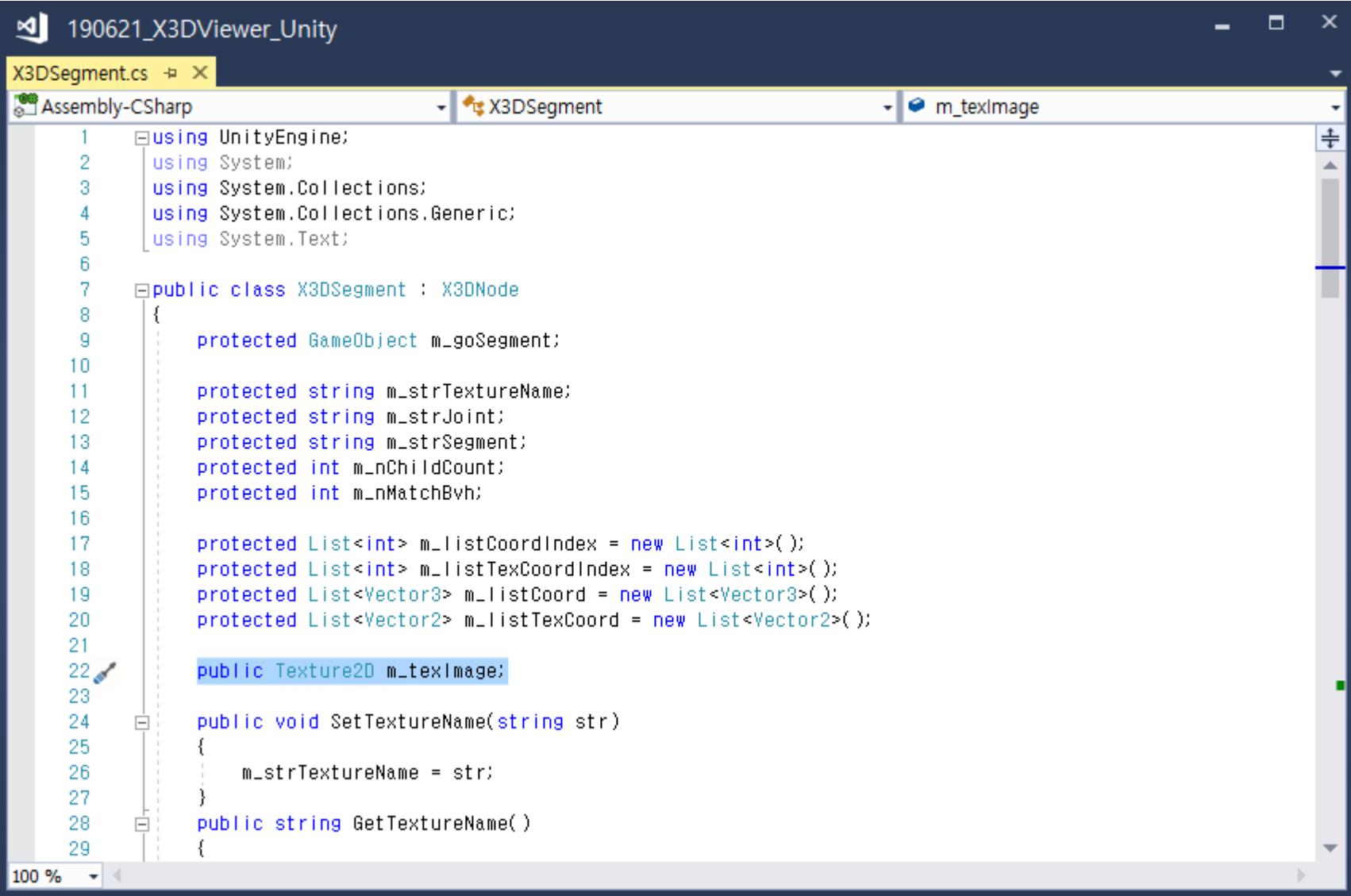
190621_X3DViewer_Unity

Meeting room190307.x3d

```
1  <?xml version='1.0' encoding='UTF-8'?>
2  <X3D version='3.3' profile='Immersive' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.3.xsd'>
3      <head>
4          <meta name='title' content='Meeting room190307.x3d'/>
5      </head>
6      <Scene>
7          <Transform DEF="floor-2" translation="99.83 2.611 21.18">
8              <Shape>
9                  <Appearance>
10                     <Material ambientIntensity="1.0" transparency="0.0" shininess="0.145" diffuseColor="0.588 0.0 0.0">
11                         <ImageTexture url='"/wallpaper.jpg"/>
12                     </Appearance>
13                     <IndexedFaceSet DEF="floor-2-FACES" coordIndex=" 7 6 4 -1 4 5 7 -1 12 13 6 -1 6 7 12 -1 4 6 8 -1 13 12 11 10 9 8 7 6 5 4 3 2 1 0 ">
14                         <TextureCoordinate DEF="floor-2-TEXCOORD" point="-14.06 -8.262, 14.75 -8.262, 14.75 9.443, -14.06 8.262, 14.75 8.262, 14.75 9.443, -14.06 14.75 -8.262, 14.75 14.75 -8.262, 14.75 14.75 9.443, -14.06 14.75 8.262, 14.75 14.75 8.262, 14.75 14.75 9.443, -14.06 20.5 -8.262, 14.75 20.5 -8.262, 14.75 20.5 9.443, -14.06 20.5 8.262, 14.75 20.5 8.262, 14.75 20.5 9.443, -14.06 26.25 -8.262, 14.75 26.25 -8.262, 14.75 26.25 9.443, -14.06 26.25 8.262, 14.75 26.25 8.262, 14.75 26.25 9.443, -14.06 32.0 -8.262, 14.75 32.0 -8.262, 14.75 32.0 9.443, -14.06 32.0 8.262, 14.75 32.0 8.262, 14.75 32.0 9.443, -14.06 37.75 -8.262, 14.75 37.75 -8.262, 14.75 37.75 9.443, -14.06 37.75 8.262, 14.75 37.75 8.262, 14.75 37.75 9.443, -14.06 43.5 -8.262, 14.75 43.5 -8.262, 14.75 43.5 9.443, -14.06 43.5 8.262, 14.75 43.5 8.262, 14.75 43.5 9.443, -14.06 49.25 -8.262, 14.75 49.25 -8.262, 14.75 49.25 9.443, -14.06 49.25 8.262, 14.75 49.25 8.262, 14.75 49.25 9.443, -14.06 55.0 -8.262, 14.75 55.0 -8.262, 14.75 55.0 9.443, -14.06 55.0 8.262, 14.75 55.0 8.262, 14.75 55.0 9.443, -14.06 60.75 -8.262, 14.75 60.75 -8.262, 14.75 60.75 9.443, -14.06 60.75 8.262, 14.75 60.75 8.262, 14.75 60.75 9.443, -14.06 66.5 -8.262, 14.75 66.5 -8.262, 14.75 66.5 9.443, -14.06 66.5 8.262, 14.75 66.5 8.262, 14.75 66.5 9.443, -14.06 72.25 -8.262, 14.75 72.25 -8.262, 14.75 72.25 9.443, -14.06 72.25 8.262, 14.75 72.25 8.262, 14.75 72.25 9.443, -14.06 78.0 -8.262, 14.75 78.0 -8.262, 14.75 78.0 9.443, -14.06 78.0 8.262, 14.75 78.0 8.262, 14.75 78.0 9.443, -14.06 83.75 -8.262, 14.75 83.75 -8.262, 14.75 83.75 9.443, -14.06 83.75 8.262, 14.75 83.75 8.262, 14.75 83.75 9.443, -14.06 89.5 -8.262, 14.75 89.5 -8.262, 14.75 89.5 9.443, -14.06 89.5 8.262, 14.75 89.5 8.262, 14.75 89.5 9.443, -14.06 95.25 -8.262, 14.75 95.25 -8.262, 14.75 95.25 9.443, -14.06 95.25 8.262, 14.75 95.25 8.262, 14.75 95.25 9.443, -14.06 101.0 -8.262, 14.75 101.0 -8.262, 14.75 101.0 9.443, -14.06 101.0 8.262, 14.75 101.0 8.262, 14.75 101.0 9.443, -14.06 106.75 -8.262, 14.75 106.75 -8.262, 14.75 106.75 9.443, -14.06 106.75 8.262, 14.75 106.75 8.262, 14.75 106.75 9.443, -14.06 112.5 -8.262, 14.75 112.5 -8.262, 14.75 112.5 9.443, -14.06 112.5 8.262, 14.75 112.5 8.262, 14.75 112.5 9.443, -14.06 118.25 -8.262, 14.75 118.25 -8.262, 14.75 118.25 9.443, -14.06 118.25 8.262, 14.75 118.25 8.262, 14.75 118.25 9.443, -14.06 124.0 -8.262, 14.75 124.0 -8.262, 14.75 124.0 9.443, -14.06 124.0 8.262, 14.75 124.0 8.262, 14.75 124.0 9.443, -14.06 129.75 -8.262, 14.75 129.75 -8.262, 14.75 129.75 9.443, -14.06 129.75 8.262, 14.75 129.75 8.262, 14.75 129.75 9.443, -14.06 135.5 -8.262, 14.75 135.5 -8.262, 14.75 135.5 9.443, -14.06 135.5 8.262, 14.75 135.5 8.262, 14.75 135.5 9.443, -14.06 141.25 -8.262, 14.75 141.25 -8.262, 14.75 141.25 9.443, -14.06 141.25 8.262, 14.75 141.25 8.262, 14.75 141.25 9.443, -14.06 147.0 -8.262, 14.75 147.0 -8.262, 14.75 147.0 9.443, -14.06 147.0 8.262, 14.75 147.0 8.262, 14.75 147.0 9.443, -14.06 152.75 -8.262, 14.75 152.75 -8.262, 14.75 152.75 9.443, -14.06 152.75 8.262, 14.75 152.75 8.262, 14.75 152.75 9.443, -14.06 158.5 -8.262, 14.75 158.5 -8.262, 14.75 158.5 9.443, -14.06 158.5 8.262, 14.75 158.5 8.262, 14.75 158.5 9.443, -14.06 164.25 -8.262, 14.75 164.25 -8.262, 14.75 164.25 9.443, -14.06 164.25 8.262, 14.75 164.25 8.262, 14.75 164.25 9.443, -14.06 170.0 -8.262, 14.75 170.0 -8.262, 14.75 170.0 9.443, -14.06 170.0 8.262, 14.75 170.0 8.262, 14.75 170.0 9.443, -14.06 175.75 -8.262, 14.75 175.75 -8.262, 14.75 175.75 9.443, -14.06 175.75 8.262, 14.75 175.75 8.262, 14.75 175.75 9.443, -14.06 181.5 -8.262, 14.75 181.5 -8.262, 14.75 181.5 9.443, -14.06 181.5 8.262, 14.75 181.5 8.262, 14.75 181.5 9.443, -14.06 187.25 -8.262, 14.75 187.25 -8.262, 14.75 187.25 9.443, -14.06 187.25 8.262, 14.75 187.25 8.262, 14.75 187.25 9.443, -14.06 193.0 -8.262, 14.75 193.0 -8.262, 14.75 193.0 9.443, -14.06 193.0 8.262, 14.75 193.0 8.262, 14.75 193.0 9.443, -14.06 198.75 -8.262, 14.75 198.75 -8.262, 14.75 198.75 9.443, -14.06 198.75 8.262, 14.75 198.75 8.262, 14.75 198.75 9.443, -14.06 204.5 -8.262, 14.75 204.5 -8.262, 14.75 204.5 9.443, -14.06 204.5 8.262, 14.75 204.5 8.262, 14.75 204.5 9.443, -14.06 210.25 -8.262, 14.75 210.25 -8.262, 14.75 210.25 9.443, -14.06 210.25 8.262, 14.75 210.25 8.262, 14.75 210.25 9.443, -14.06 216.0 -8.262, 14.75 216.0 -8.262, 14.75 216.0 9.443, -14.06 216.0 8.262, 14.75 216.0 8.262, 14.75 216.0 9.443, -14.06 221.75 -8.262, 14.75 221.75 -8.262, 14.75 221.75 9.443, -14.06 221.75 8.262, 14.75 221.75 8.262, 14.75 221.75 9.443, -14.06 227.5 -8.262, 14.75 227.5 -8.262, 14.75 227.5 9.443, -14.06 227.5 8.262, 14.75 227.5 8.262, 14.75 227.5 9.443, -14.06 233.25 -8.262, 14.75 233.25 -8.262, 14.75 233.25 9.443, -14.06 233.25 8.262, 14.75 233.25 8.262, 14.75 233.25 9.443, -14.06 239.0 -8.262, 14.75 239.0 -8.262, 14.75 239.0 9.443, -14.06 239.0 8.262, 14.75 239.0 8.262, 14.75 239.0 9.443, -14.06 244.75 -8.262, 14.75 244.75 -8.262, 14.75 244.75 9.443, -14.06 244.75 8.262, 14.75 244.75 8.262, 14.75 244.75 9.443, -14.06 250.5 -8.262, 14.75 250.5 -8.262, 14.75 250.5 9.443, -14.06 250.5 8.262, 14.75 250.5 8.262, 14.75 250.5 9.443, -14.06 256.25 -8.262, 14.75 256.25 -8.262, 14.75 256.25 9.443, -14.06 256.25 8.262, 14.75 256.25 8.262, 14.75 256.25 9.443, -14.06 262.0 -8.262, 14.75 262.0 -8.262, 14.75 262.0 9.443, -14.06 262.0 8.262, 14.75 262.0 8.262, 14.75 262.0 9.443, -14.06 267.75 -8.262, 14.75 267.75 -8.262, 14.75 267.75 9.443, -14.06 267.75 8.262, 14.75 267.75 8.262, 14.75 267.75 9.443, -14.06 273.5 -8.262, 14.75 273.5 -8.262, 14.75 273.5 9.443, -14.06 273.5 8.262, 14.75 273.5 8.262, 14.75 273.5 9.443, -14.06 279.25 -8.262, 14.75 279.25 -8.262, 14.75 279.25 9.443, -14.06 279.25 8.262, 14.75 279.25 8.262, 14.75 279.25 9.443, -14.06 285.0 -8.262, 14.75 285.0 -8.262, 14.75 285.0 9.443, -14.06 285.0 8.262, 14.75 285.0 8.262, 14.75 285.0 9.443, -14.06 290.75 -8.262, 14.75 290.75 -8.262, 14.75 290.75 9.443, -14.06 290.75 8.262, 14.75 290.75 8.262, 14.75 290.75 9.443, -14.06 296.5 -8.262, 14.75 296.5 -8.262, 14.75 296.5 9.443, -14.06 296.5 8.262, 14.75 296.5 8.262, 14.75 296.5 9.443, -14.06 302.25 -8.262, 14.75 302.25 -8.262, 14.75 302.25 9.443, -14.06 302.25 8.262, 14.75 302.25 8.262, 14.75 302.25 9.443, -14.06 308.0 -8.262, 14.75 308.0 -8.262, 14.75 308.0 9.443, -14.06 308.0 8.262, 14.75 308.0 8.262, 14.75 308.0 9.443, -14.06 313.75 -8.262, 14.75 313.75 -8.262, 14.75 313.75 9.443, -14.06 313.75 8.262, 14.75 313.75 8.262, 14.75 313.75 9.443, -14.06 319.5 -8.262, 14.75 319.5 -8.262, 14.75 319.5 9.443, -14.06 319.5 8.262, 14.75 319.5 8.262, 14.75 319.5 9.443, -14.06 325.25 -8.262, 14.75 325.25 -8.262, 14.75 325.25 9.443, -14.06 325.25 8.262, 14.75 325.25 8.262, 14.75 325.25 9.443, -14.06 331.0 -8.262, 14.75 331.0 -8.262, 14.75 331.0 9.443, -14.06 331.0 8.262, 14.75 331.0 8.262, 14.75 331.0 9.443, -14.06 336.75 -8.262, 14.75 336.75 -8.262, 14.75 336.75 9.443, -14.06 336.75 8.262, 14.75 336.75 8.262, 14.75 336.75 9.443, -14.06 342.5 -8.262, 14.75 342.5 -8.262, 14.75 342.5 9.443, -14.06 342.5 8.262, 14.75 342.5 8.262, 14.75 342.5 9.443, -14.06 348.25 -8.262, 14.75 348.25 -8.262, 14.75 348.25 9.443, -14.06 348.25 8.262, 14.75 348.25 8.262, 14.75 348.25 9.443, -14.06 354.0 -8.262, 14.75 354.0 -8.262, 14.75 354.0 9.443, -14.06 354.0 8.262, 14.75 354.0 8.262, 14.75 354.0 9.443, -14.06 359.75 -8.262, 14.75 359.75 -8.262, 14.75 359.75 9.443, -14.06 359.75 8.262, 14.75 359.75 8.262, 14.75 359.75 9.443, -14.06 365.5 -8.262, 14.75 365.5 -8.262, 14.75 365.5 9.443, -14.06 365.5 8.262, 14.75 365.5 8.262, 14.75 365.5 9.443, -14.06 371.25 -8.262, 14.75 371.25 -8.262, 14.75 371.25 9.443, -14.06 371.25 8.262, 14.75 371.25 8.262, 14.75 371.25 9.443, -14.06 377.0 -8.262, 14.75 377.0 -8.262, 14.75 377.0 9.443, -14.06 377.0 8.262, 14.75 377.0 8.262, 14.75 377.0 9.443, -14.06 382.75 -8.262, 14.75 382.75 -8.262, 14.75 382.75 9.443, -14.06 382.75 8.262, 14.75 382.75 8.262, 14.75 382.75 9.443, -14.06 388.5 -8.262, 14.75 388.5 -8.262, 14.75 388.5 9.443, -14.06 388.5 8.262, 14.75 388.5 8.262, 14.75 388.5 9.443, -14.06 394.25 -8.262, 14.75 394.25 -8.262, 14.75 394.25 9.443, -14.06 394.25 8.262, 14.75 394.25 8.262, 14.75 394.25 9.443, -14.06 400.0 -8.262, 14.75 400.0 -8.262, 14.75 400.0 9.443, -14.06 400.0 8.262, 14.75 400.0 8.262, 14.75 400.0 9.443, -14.06 405.75 -8.262, 14.75 405.75 -8.262, 14.75 405.75 9.443, -14.06 405.75 8.262, 14.75 405.75 8.262, 14.75 405.75 9.443, -14.06 411.5 -8.262, 14.75 411.5 -8.262, 14.75 411.5 9.443, -14.06 411.5 8.262, 14.75 411.5 8.262, 14.75 411.5 9.443, -14.06 417.25 -8.262, 14.75 417.25 -8.262, 14.75 417.25 9.443, -14.06 417.25 8.262, 14.75 417.25 8.262, 14.75 417.25 9.443, -14.06 423.0 -8.262, 14.75 423.0 -8.262, 14.75 423.0 9.443, -14.06 423.0 8.262, 14.75 423.0 8.262, 14.75 423.0 9.443, -14.06 428.75 -8.262, 14.75 428.75 -8.262, 14.75 428.75 9.443, -14.06 428.75 8.262, 14.75 428.75 8.262, 14.75 428.75 9.443, -14.06 434.5 -8.262, 14.75 434.5 -8.262, 14.75 434.5 9.443, -14.06 434.5 8.262, 14.75 434.5 8.262, 14.75 434.5 9.443, -14.06 440.25 -8.262, 14.75 440.25 -8.262, 14.75 440.25 9.443, -14.06 440.25 8.262, 14.75 440.25 8.262, 14.75 440.25 9.443, -14.06 446.0 -8.262, 14.75 446.0 -8.262, 14.75 446.0 9.443, -14.06 446.0 8.262, 14.75 446.0 8.262, 14.75 446.0 9.443, -14.06 451.75 -8.262, 14.75 451.75 -8.262, 14.75 451.75 9.443, -14.06 451.75 8.262, 14.75 451.75 8.262, 14.75 451.75 9.443, -14.06 457.5 -8.262, 14.75 457.5 -8.262, 14.75 457.5 9.443, -14.06 457.5 8.262, 14.75 457.5 8.262, 14.75 457.5 9.443, -14.06 463.25 -8.262, 14.75 463.25 -8.262, 14.75 463.25 9.443, -14.06 463.25 8.262, 14.75 463.25 8.262, 14.75 463.25 9.443, -14.06 469.0 -8.262, 14.75 469.0 -8.262, 14.75 469.0 9.443, -14.06 469.0 8.262, 14.75 469.0 8.262, 14.75 469.0 9.443, -14.06 474.75 -8.262, 14.75 474.75 -8.262, 14.75 474.75 9.443, -14.06 474.75 8.262, 14.75 474.75 8.262, 14.75 474.75 9.443, -14.06 480.5 -8.262, 14.75 480.5 -8.262, 14.75 480.5 9.443, -14.06 480.5 8.262, 14.75 480.5 8.262, 14.75 480.5 9.443, -14.06 486.25 -8.262, 14.75 486.25 -8.262, 14.75 486.25 9.443, -14.06 486.25 8.262, 14.75 486.25 8.262, 14.75 486.25 9.443, -14.06 492.0 -8.262, 14.75 492.0 -8.262, 14.75 492.0 9.443, -14.06 492.0 8.262, 14.75 492.0 8.262, 14.75 492.0 9.443, -14.06 497.75 -8.262, 14.75 497.75 -8.262, 14.75 497.75 9.443, -14.06 497.75 8.262, 14.75 497.75 8.262, 14.75 497.75 9.443, -14.06 503.5 -8.262, 14.75 503.5 -8.262, 14.75 503.5 9.443, -14.06 503.5 8.262, 14.75 503.5 8.262, 14.75 503.5 9.443, -14.06 509.25 -8.262, 14.75 509.25 -8.262, 14.75 509.25 9.443, -14.06 509.25 8.262, 14.75 509.25 8.262, 14.75 509.25 9.443, -14.06 515.0 -8.262, 14.75 515.0 -8.262, 14.75 515.0 9.443, -14.06 515.0 8.262, 14.75 515.0 8.262, 14.75 515.0 9.443, -14.06 520.75 -8.262, 14.75 520.75 -8.262, 14.75 520.75 9.443, -14.06 520.75 8.262, 14.75 520.75 8.262, 14.75 520.75 9.443, -14.06 526.5 -8.262, 14.75 526.5 -8.262, 14.75 526.5 9.443, -14.06 526.5 8.262, 14.75 526.5 8.262, 14.75 526.5 9.443, -14.06 532.25 -8.262, 14.75 532.25 -8.262, 14.75 532.25 9.443, -14.06 532.25 8.262, 14.75 532.25 8.262, 14.75 532.25 9.443, -14.06 538.0 -8.262, 14.75 538.0 -8.262, 14.75 538.0 9.443, -14.06 538.0 8.262, 14.75 538.0 8.262, 14.75 538.0 9.443, -14.06 543.75 -8.262, 14.75 543.75 -8.262, 14.75 543.75 9.443, -14.06 543.75 8.262, 14.75 543.75 8.262, 14.75 543.75 9.443, -14.06 549.5 -8.262, 14.75 549.5 -8.262, 14.75 549.5 9.443, -14.06 549.5 8.262, 14.75 549.5 8.262, 14.75 549.5 9.443, -14.06 555.25 -8.262, 14.75 555.25 -8.262, 14.75 555.25 9.443, -14.06 555.25 8.262, 14.75 555.25 8.262, 14.75 555.25 9.443, -14.06 561.0 -8.262, 14.75 561.0 -8.262, 14.75 561.0 9.443, -14.06 561.0 8.262, 14.75 561.0 8.262, 14.75 561.0 9.443, -14.06 566.75 -8.262, 14.75 566.75 -8.262, 14.75 566.75 9.443, -14.06 566.75 8.262, 14.75 566.75 8.262, 14.75 566.75 9.443, -14.06 572.5 -8.262, 14.75 572.5 -8.262, 14.75 572.5 9.443, -14.06 572.5 8.262, 14.75 572.5 8.262, 14.75 572.5 9.443, -14.06 578.25 -8.262, 14.75 578.25 -8.262, 14.75 578.25 9.443, -14.06 578.25 8.262, 14.75 578.25 8.262, 14.75 578.25 9.443, -14.06 584.0 -8.262, 14.75 584.0 -8.262, 14.75 584.0 9.443, -14.06 584.0 8.262, 14.75 584.0 8.262, 14.75 584.0 9.443, -14.06 589.75 -8.262, 14.75 589.75 -8.262, 14.75 589.75 9.443, -14.06 589.75 8.262, 14.75 589.75 8.262, 14.75 589.75 9.443, -14.06 595.5 -8.262, 14.75 595.5 -8.262, 14.75 595.5 9.443, -14.06 595.5 8.262, 14.75 595.5 8.262, 14.75 595.5 9.443, -14.06 601.25 -8.262,
```

Multi-Textures Structure (2)

Multi-Textures Implementation (1)



The screenshot shows a code editor window titled "190621_X3DViewer_Unity". The tab bar indicates the file is "X3DSegment.cs". The code itself is a C# class definition for "X3DSegment" which inherits from "X3DNode". The class contains several protected member variables including "m_goSegment", "m_strTextureName", "m_strJoint", "m_strSegment", "m_nChildCount", "m_nMatchBvh", and lists for coordinate and texture coordinates. It also includes public properties for "m_texImage" and methods for setting and getting the texture name.

```
1  using UnityEngine;
2  using System;
3  using System.Collections;
4  using System.Collections.Generic;
5  using System.Text;
6
7  public class X3DSegment : X3DNode
8  {
9      protected GameObject m_goSegment;
10
11     protected string m_strTextureName;
12     protected string m_strJoint;
13     protected string m_strSegment;
14     protected int m_nChildCount;
15     protected int m_nMatchBvh;
16
17     protected List<int> m_listCoordIndex = new List<int>();
18     protected List<int> m_listTexCoordIndex = new List<int>();
19     protected List<Vector3> m_listCoord = new List<Vector3>();
20     protected List<Vector2> m_listTexCoord = new List<Vector2>();
21
22     public Texture2D m_texImage;
23
24     public void SetTextureName(string str)
25     {
26         m_strTextureName = str;
27     }
28
29     public string GetTextureName()
29     {
```

Multi-Textures Implementation (2)



The screenshot shows a code editor window titled "190621_X3DViewer_Unity" with the file "X3Dx3d.cs" open. The code implements a coroutine for loading multiple textures in parallel. The editor interface includes tabs for "Assembly-CSharp" and "X3Dx3d", and a status bar at the bottom.

```
190621_X3DViewer_Unity
X3Dx3d.cs
Assembly-CSharp X3Dx3d CoroutineLoadImage()

40
47
48     public override IEnumerator LoadEndAction()
49     {
50         yield return CoroutineLoadImage();
51
52         Draw();
53     }
54
55     public IEnumerator CoroutineLoadImage()
56     {
57         if(m_listSegment.Count == 0)
58         {
59             yield break;
60         }
61
62         for(int i = 0; i < m_listSegment.Count; i++)
63         {
64             yield return m_listSegment[i].CoroutineLoadImage();
65         }
66     }
67
68     public override void Draw()
69     {
70         if (m_listSegment.Count == 0)
71             return;
72
73         m_goRootParent = new GameObject();
74
75         //YODM:lower Instance m_listDebug.Add("endCount": " + m_listSegment.Count.ToString());
    }
```

Multi-Textures Implementation (3)



The screenshot shows a code editor window titled "190621_X3DViewer_Unity". The active tab is "X3DSegment.cs". The code is a C# script for a class named "X3DSegment". The script contains several methods: "GetCoord()", "SetTexCoord()", "GetTexCoord()", and "CoroutineLoadImage()". The "CoroutineLoadImage()" method uses a yield break and a WWW object to load a texture from a file.

```
190621_X3DViewer_Unity
X3DSegment.cs

public List<Vector3> GetCoord()
{
    return m_listCoord;
}

public void SetTexCoord( List<Vector2> list )
{
    m_listTexCoord = list;
}

public List<Vector2> GetTexCoord()
{
    return m_listTexCoord;
}

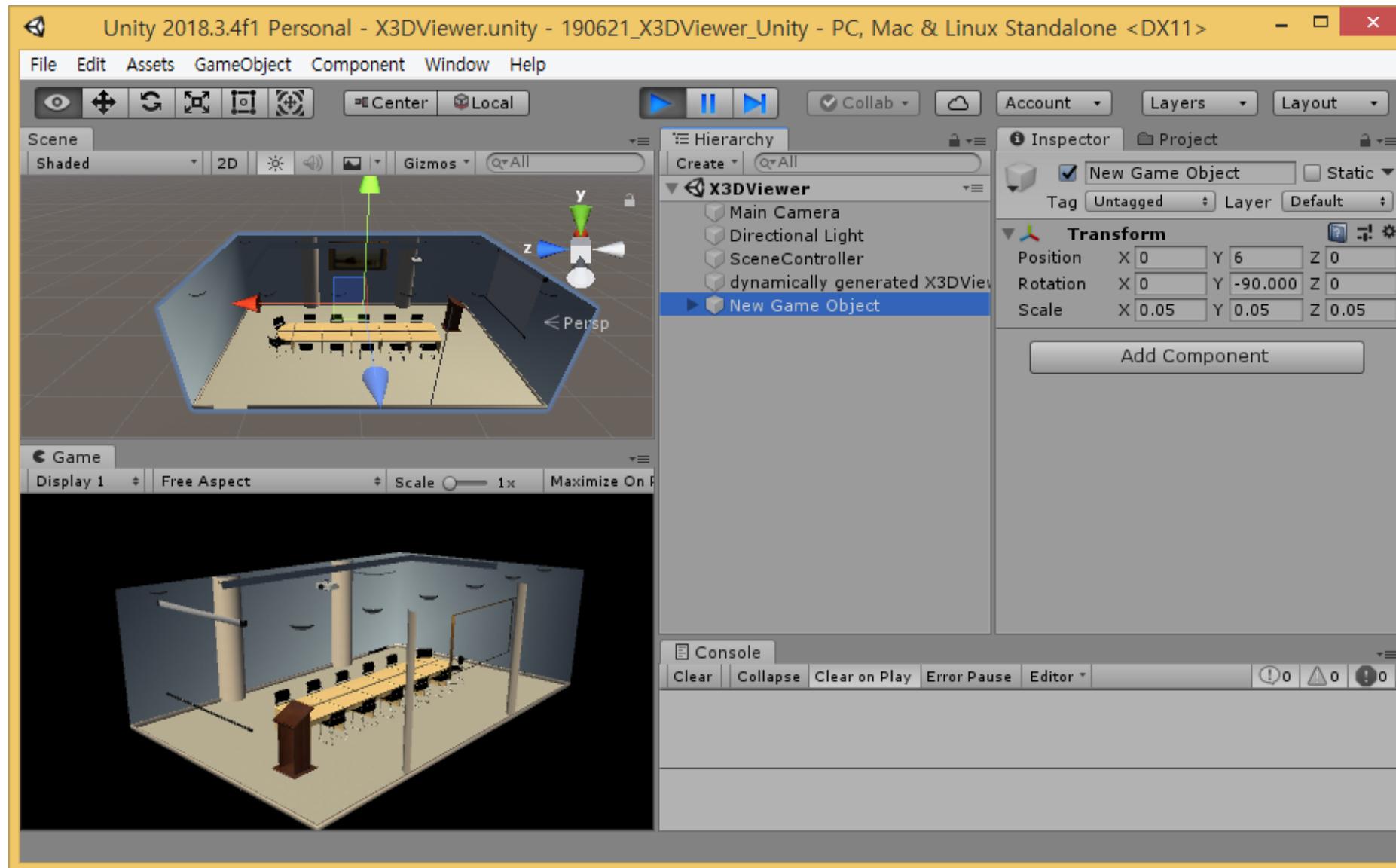
public IEnumerator CoroutineLoadImage()
{
    if (string.IsNullOrEmpty(GetTextureName()) == true)
    {
        yield break;
    }

    WWW www = UtilityClass.LoadFile(GetTextureName());

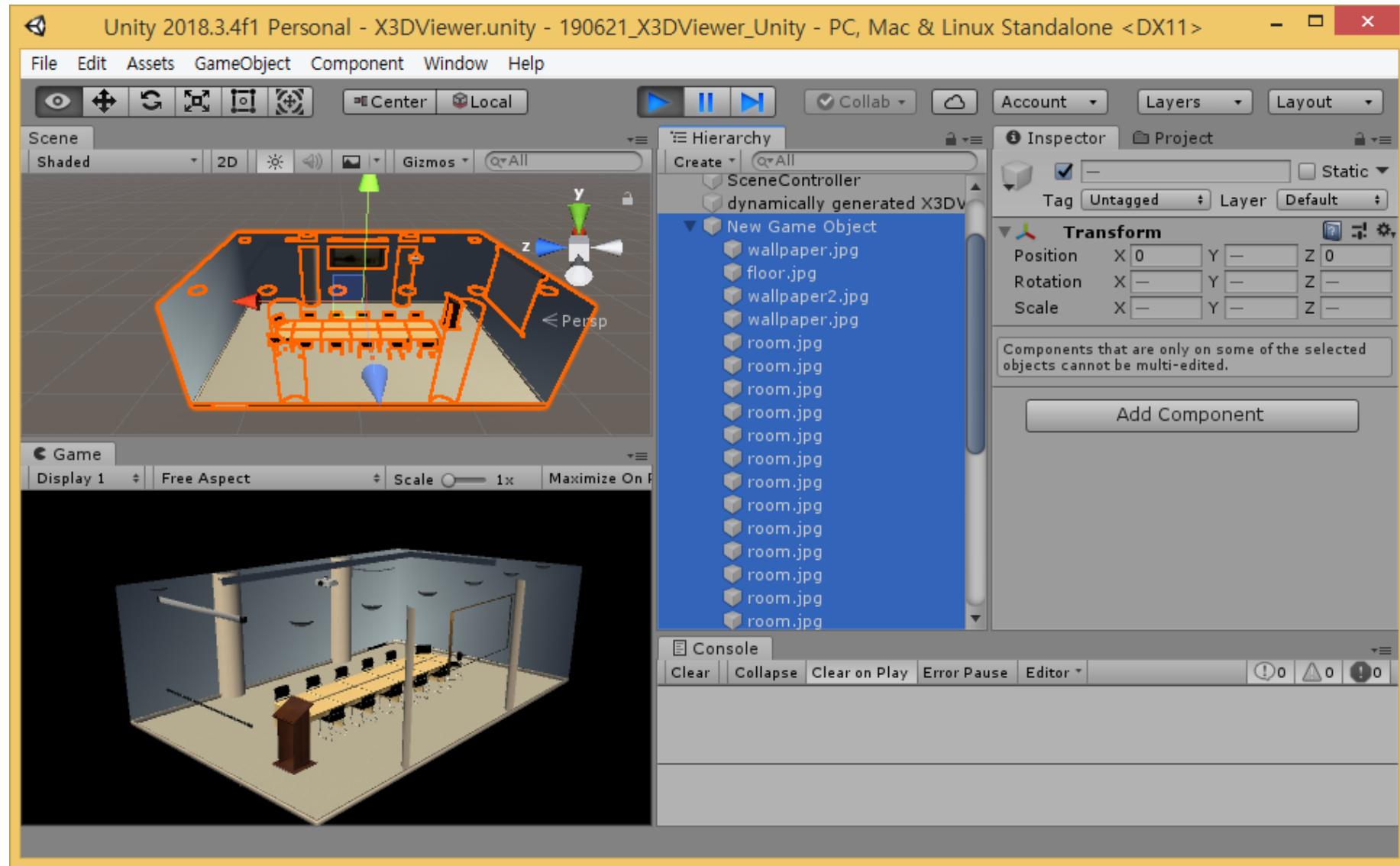
    yield return www;

    m_texture = www.texture;
}
```

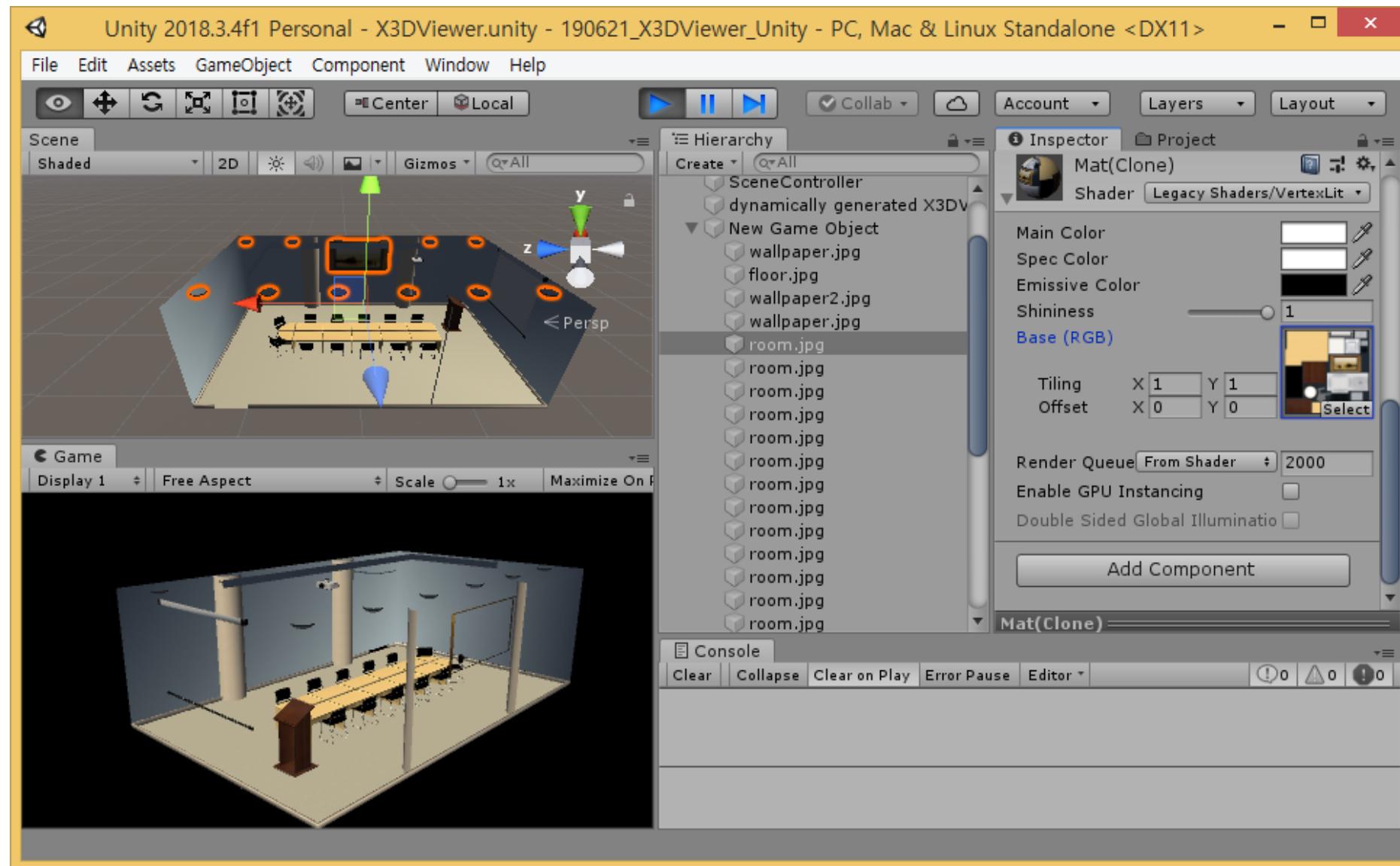
Implementation Results (1)



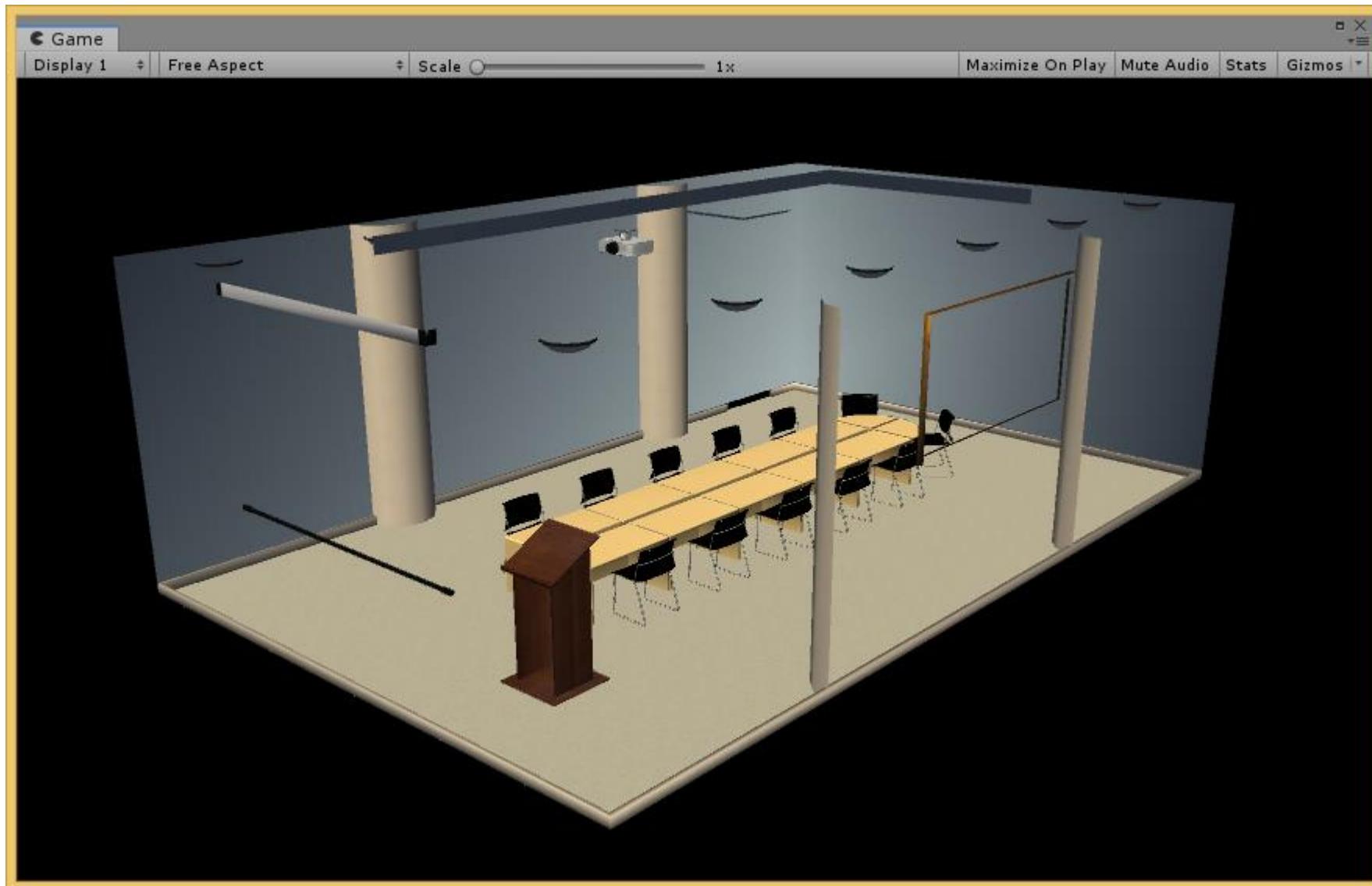
Implementation Results (2)



Implementation Results (3)



Implementation Results (4)



Work in Progress

- 19777-3, 19777-4, 19777-5 CD preparation
- Implementation of C, C++ and C# language bindings
 - 19777-3 X3D scene access interface definition using C
 - Visual C++ and OpenGL
 - 19777-4 X3D scene access interface definition using C++
 - Visual C++ and OpenGL
 - 19777-5 X3D scene access interface definition using C#
 - Unity 3D and C#
- Developing X3D Binding viewer programs with C, C++ and C# binding capability based on X3D SAI