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

SIGGRAPH Web3D Meetings
Los Angeles, CA, USA

July 31, 2017

Roy Walmsley and Myeong Won Lee

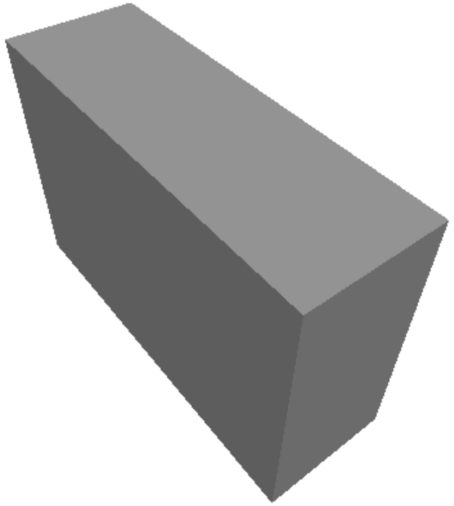
C++/C# Language Binding Concepts

- What is C++ / C# language binding?
  - X3D scene access using C++ and C# languages
  - Development of C++/C# programs using X3D data types and functions
  - X3D scene read, update, store, and exchange in 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 a C++ and C# API format for X3D scene access

ISO/IEC 19777-3 & 19777-4 WD Preparation

- ISO/IEC 19777-3 X3D C++ Language Binding WD
  - Working on clause 4 Concepts, clause 5 Tables, clause 6 Type definitions, clause 7 Function definitions, Annex A Abstract node interfaces, Annex B Concrete node interfaces
- ISO/IEC 19777-4 X3D C# Language Binding WD
  - Working on clause 4 Concepts, clause 5 Tables, clause 6 Type definitions, clause 7 Function definitions, Annex A Abstract node interfaces, Annex B Concrete node interfaces

A Simple Example of X3D Scene Access API



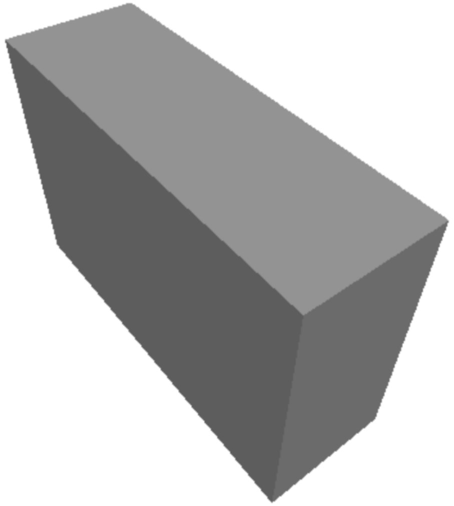
getX3D	setX3D
getScene	setScene
getBackground	setBackground
getViewpoint	setViewpoint
getShape	setShape
getBox	setBox
getApperance	setApperance
getMaterial	setMaterial

```
<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++)



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

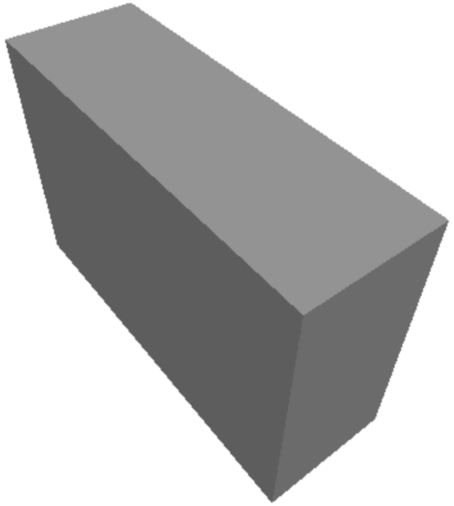
```
setX3D (pX3D)
setScene(pScene)
setBackground(pBackground)
setViewpoint(pViewpoint)
setShape(pShape)
setBox(pBox)
setApperance(pAppearance)
setMaterial(pMaterial)
```

X3D C++ encoding

getX3D	setX3D
getScene	setScene
getBackground	setBackground
getViewpoint	setViewpoint
getShape	setShape
getBox	setBox
getApperance	setApperance
getMaterial	setMaterial

X3D Scene Access Interface (SAI)

A Sample of X3D Scene Access API (C#)



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

getX3D	setX3D
getScene	setScene
getBackground	setBackground
getViewpoint	setViewpoint
getShape	setShape
getBox	setBox
getApperance	setApperance
getMaterial	setMaterial

X3D Scene Access Interface (SAI)

X3D C++ Binding Viewer Program Example

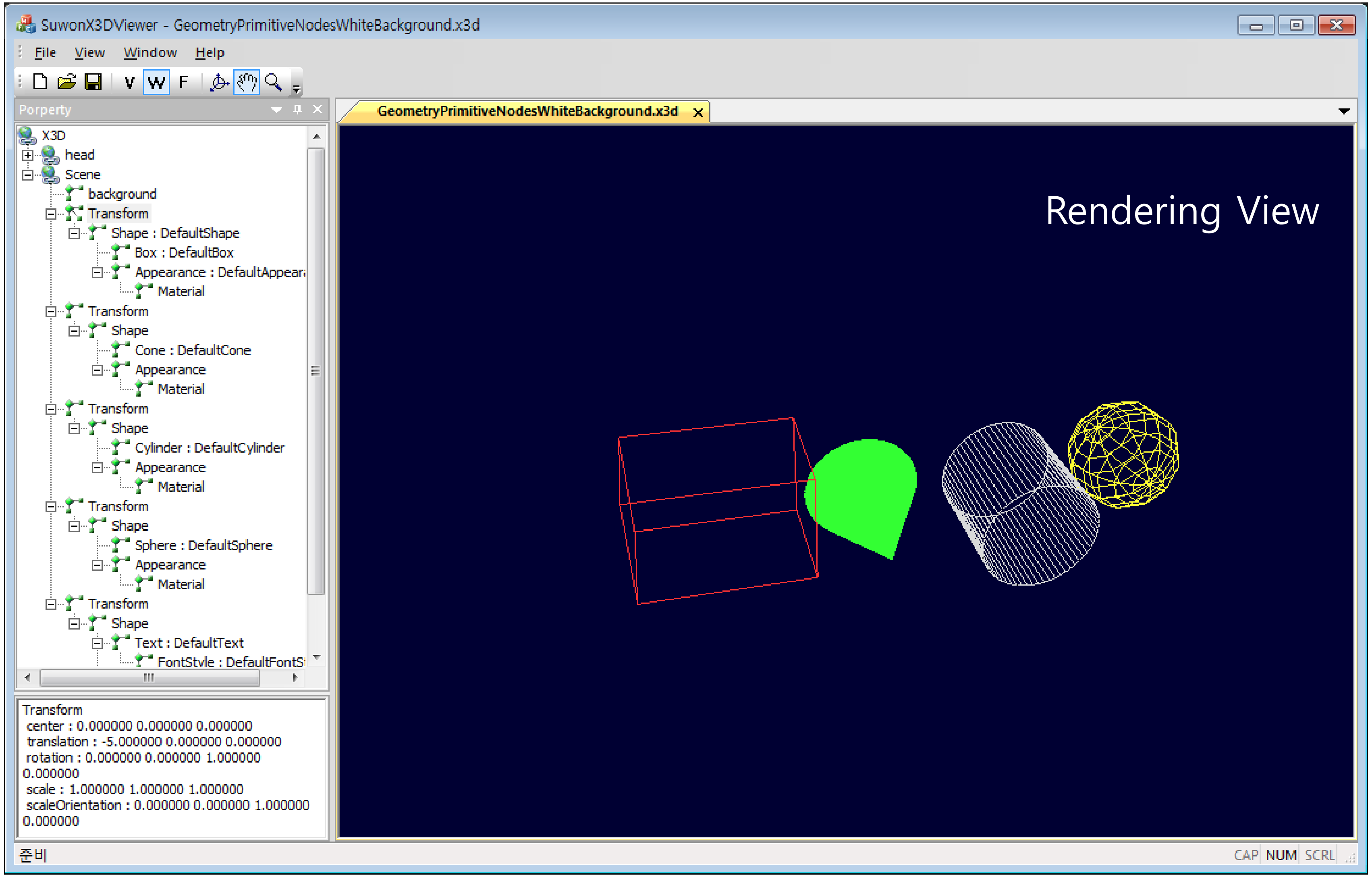
1. SuwonX3DBindingViewer
    1) Load X3DLib.dll
    2) Parse an X3D file with X3DLib
    3) Read, update, draw, and store the X3D file using X3DLib classes

2. X3DLib.dll
    1) Parse an X3D file
    2) Insert the parsed X3D into an internal class
    3) Provide an interface to read X3D

SuwonX3DBindingViewer (X3D C++ Binding Viewer)

X3D Tree View

Property View

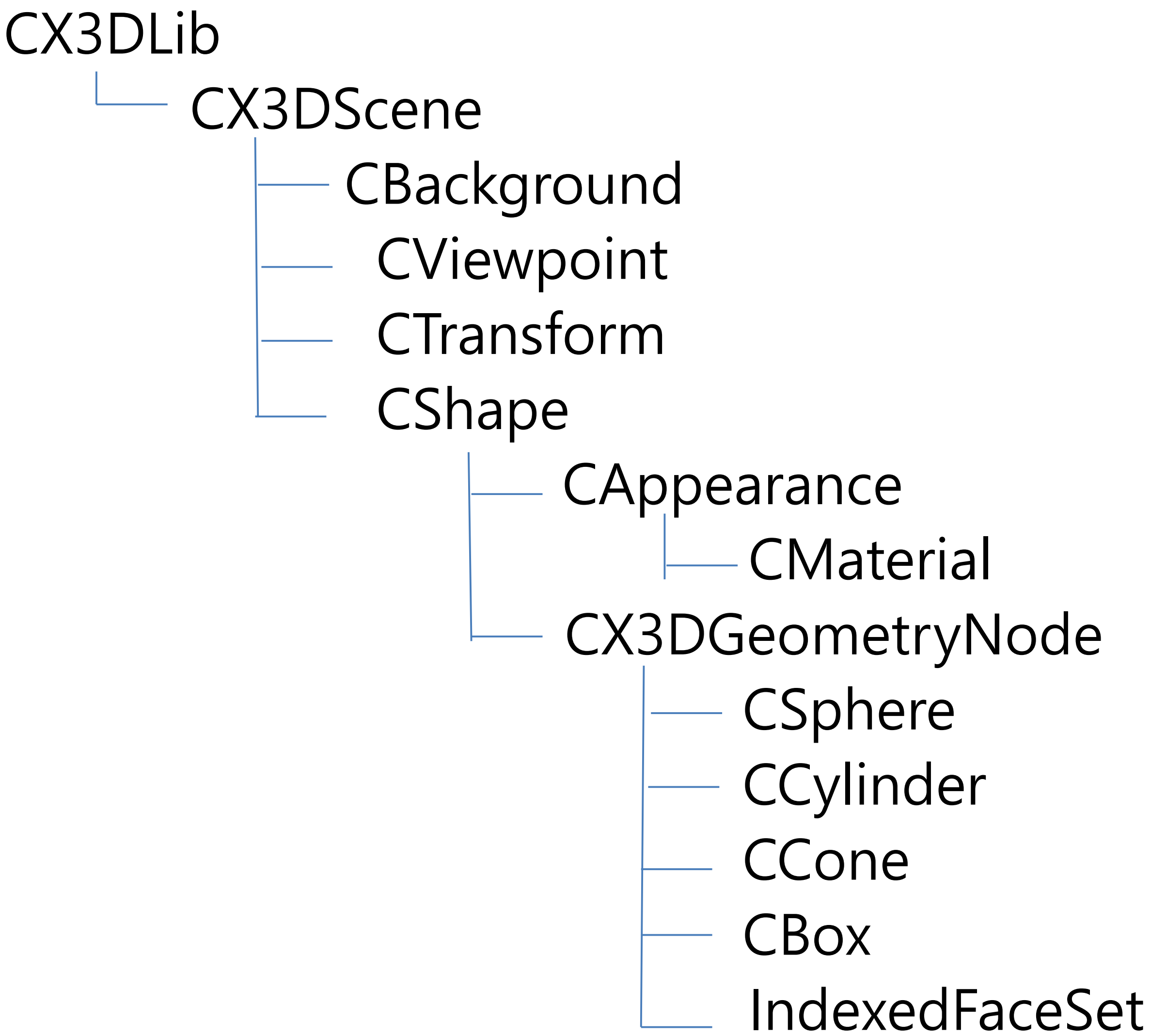

SuwonX3DViewer - GeometryPrimitiveNodesWhiteBackground.x3d
File
View
Window
Help
V
W
F
Porperty
GeometryPrimitiveNodesWhiteBackground.x3d
준비
CAP
NUM
SCRL

Rendering View

```
X3D
  head
  Scene
    background
    Transform
      Shape : DefaultShape
        Box : DefaultBox
        Appearance : DefaultAppear:
          Material
    Transform
      Shape
        Cone : DefaultCone
        Appearance
          Material
    Transform
      Shape
        Cylinder : DefaultCylinder
        Appearance
          Material
    Transform
      Shape
        Sphere : DefaultSphere
        Appearance
          Material
    Transform
      Shape
        Text : DefaultText
          FontStyle : DefaultFontS
```

```
Transform
 center : 0.000000 0.000000 0.000000
 translation : -5.000000 0.000000 0.000000
 rotation : 0.000000 0.000000 1.000000
0.000000
 scale : 1.000000 1.000000 1.000000
 scaleOrientation : 0.000000 0.000000 1.000000
0.000000
```

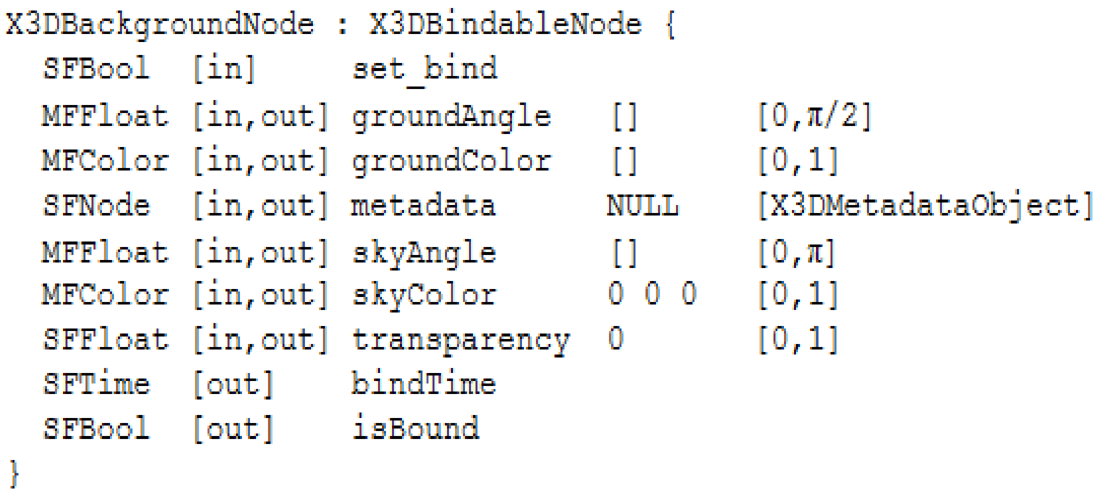
X3D C++ Binding Class Structure (Partial)



- CX3DLib
  - CX3DScene
    - CBackground
    - CViewpoint
    - CTransform
    - CShape
      - CAppearance
        - CMaterial
      - CX3DGeometryNode
        - CSphere
        - CCylinder
        - CCone
        - CBox
        - IndexedFaceSet

Background

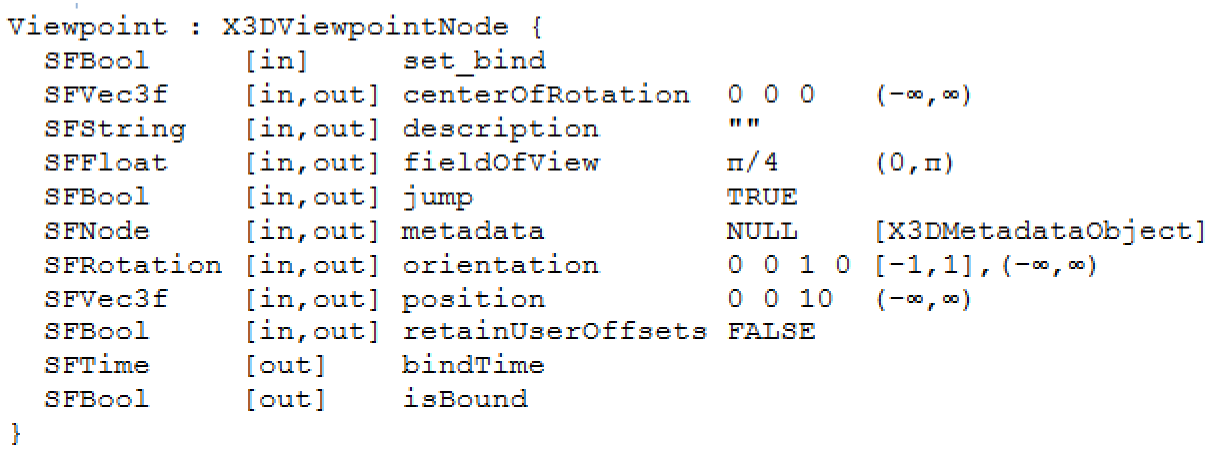
```
class AFX_EXT_CLASS CX3DBackgroundNode : public CX3DBindableNode
{
public:
    int      getNumSkyAngle();
    float*   getSkyAngle();
    void     setSkyAngle(float* angles, int size);
    void     setSkyAngle(float angle);
    int      getNumGroundAngle();
    float*   getGroundAngle();
    void     setGroundAngle(float* angles, int size);
    void     setGroundAngle(float angle);
    int      getNumSkyColor();
    float*   getSkyColor();
    void     setSkyColor(float* colors, int size);
    void     setSkyColor(float color);
    int      getNumGroundColor();
    float*   getGroundColor();
    void     setGroundColor(float* colors, int size);
    void     setGroundColor(float color);
    float    getTransparency();
    void     setTransparency(float value);

protected:
    MFFloat  groundAngle;
    MFColor  groundColor;
    MFFloat  skyAngle;
    MFColor  skyColor;
    SFFloat  transparency;
};
```



```
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
}
```

Viewpoint

```
class AFX_EXT_CLASS CViewpoint : public CX3DViewpointNode
{
public:
      void       setFieldOfView(SFFloat fov);
      SFFloat    getFieldOfView();
protected:
      SFFloat    fieldOfView;
};
```

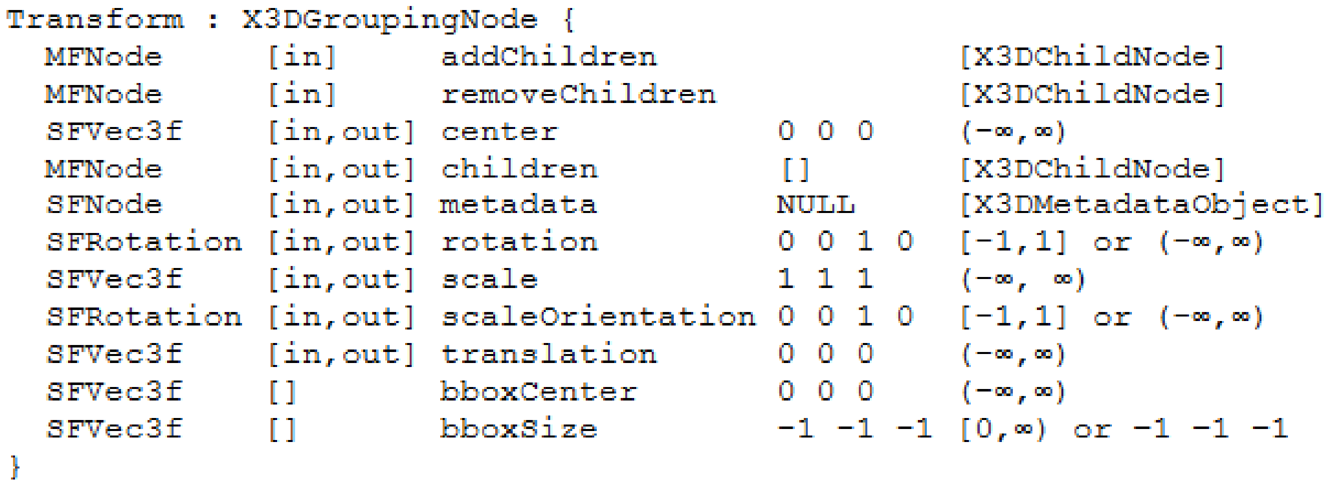


```
Viewpoint : X3DViewpointNode {
  SFBool     [in]     set_bind
  SFVec3f    [in,out] centerOfRotation  0 0 0   (-∞,∞)
  SFString   [in,out] description       ""
  SFFloat    [in,out] fieldOfView       π/4     (0,π)
  SFBool     [in,out] jump              TRUE
  SFNode     [in,out] metadata          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
}
```

```
class AFX_EXT_CLASS CX3DViewpointNode : public CX3DBindableNode
{
public:
      void    setDescription(CString strText);
      void    setPosition(SFVec3f pos);
      void    setOrientation(SFVec4f ori);
      void    setCenterOfRotaion(SFVec3f cen);
      void    setJump(BOOL bBool);
      void    setRetainUserOffsets(BOOL bBool);
      CString       getDescription();
      SFVec3f       getPosition();
      SFRotation    getOrientation();
      SFVec3f       getCenterOfRotaion();
      BOOL          isJump();
      BOOL          isRetainUserOffsets();
protected:
      CString       description;
      SFBool        jump;
      SFVec3f       position;
      SFVec3f       centerOfRotation;
      SFRotation    orientation;
      SFBool        retainUserOffsets;
};
```

Transform

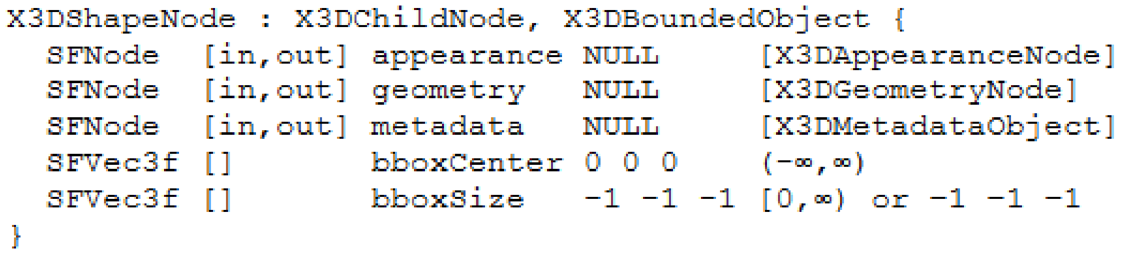
```
class AFX_EXT_CLASS CTransform : public CX3DGroupingNode
{
public:
     void setCenter(SFVec3f val);
     void setRotation(SFVec4f val);
     void setScale(SFVec3f val);
     void setScaleOrientation(SFVec4f val);
     void setTranslation(SFVec3f val);

     SFVec3f getCenter();
     SFRotation getRotation();
     SFVec3f getScale();
     SFRotation getScaleOrientation();
     SFVec3f getTranslation();

public:
      SFVec3f      center;
      SFRotation   rotation;
      SFVec3f      scale;
      SFRotation   scaleOrientation;
      SFVec3f      translation;

};
```



```
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
}
```

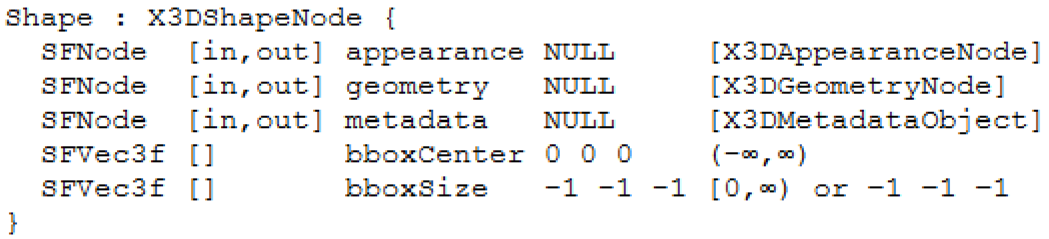
Shape

```
class AFX_EXT_CLASS CX3DShape : public CX3DChildNode , public CX3DBoundedObject
{
public:
    void setGeometry(CX3DGeometryNode* pNode);
    void setAppearance(CAppearance* pNode);

public:
    CX3DGeometryNode*   geometry;
    CAppearance*        appearance;
};
```



```
X3DShapeNode : X3DChildNode, X3DBoundedObject {
  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
}
```

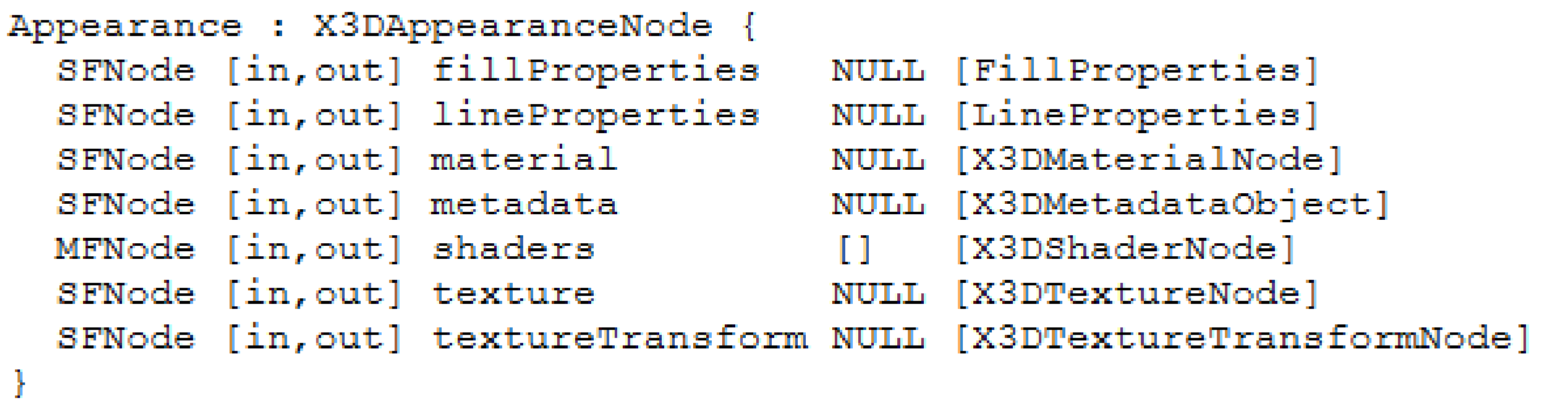
```
class AFX_EXT_CLASS CShape : public CX3DShape
{
}
```



```
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
}
```

Appearance

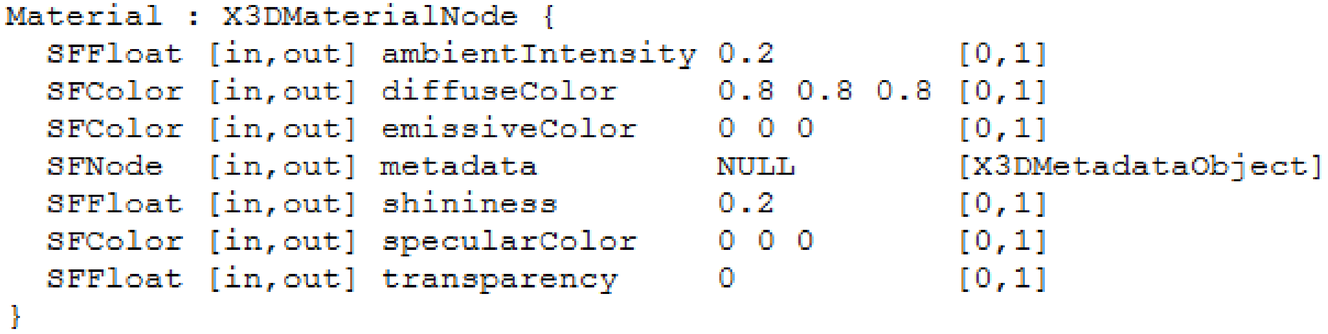
```
class AFX_EXT_CLASS CAppearance : public CX3DAppearanceNode
{
public:
     void setMaterial(CMaterial*Node);
     void setImageTexture(CImageTexture*Node);

public:
      CMaterial*        material;
      CImageTexture* imagetexture;
};
```



```
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 [X3DTextureTransformNode]
}
```

Material

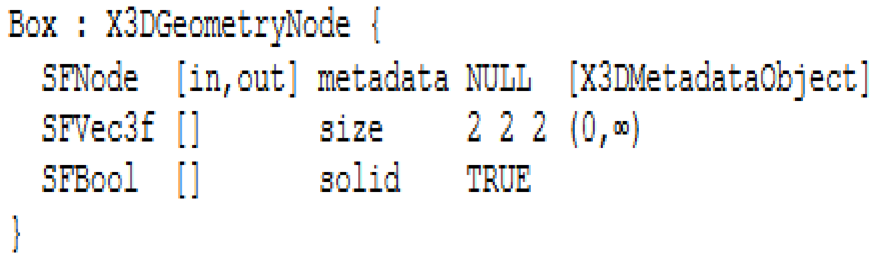
```
class AFX_EXT_CLASS CMaterial : public CX3DMaterialNode
{
public:
      void setAmbientIntensity(SFFloat val);
      void setDiffuseColor(SFVec3f val);
      void setEmissiveColor(SFVec3f val);
      void setShininess(SFFloat val);
      void setSpecularColor(SFVec3f val);
      void setTransparency(SFFloat val);
      SFFloat getAmbientIntensity();
      SFColor getDiffuseColor();
      SFColor getEmissiveColor();
      SFFloat getShininess();
      SFColor getSpecularColor();
      SFFloat getTransparency();
public:
      SFFloat   ambientIntensity;
      SFColor   diffuseColor;
      SFColor   emissiveColor;
      SFFloat   shininess;
      SFColor   specularColor;
      SFFloat   transparency;
};
```



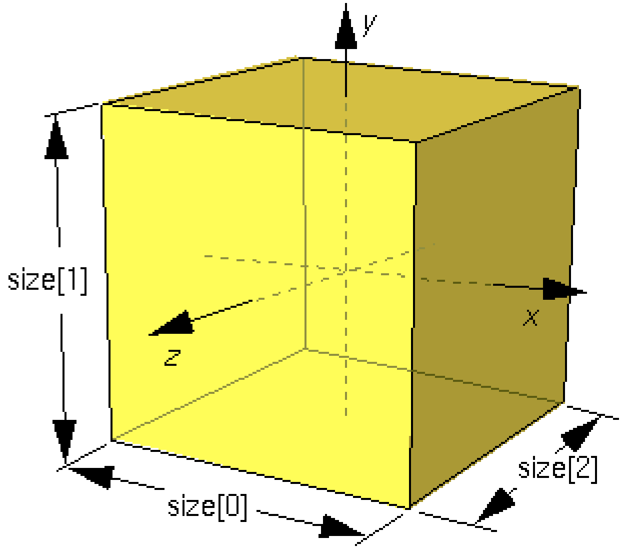
```
Material : X3DMaterialNode {
  SFFloat [in,out] ambientIntensity 0.2         [0,1]
  SFColor [in,out] diffuseColor     0.8 0.8 0.8 [0,1]
  SFColor [in,out] emissiveColor    0 0 0       [0,1]
  SFNode  [in,out] metadata         NULL        [X3DMetadataObject]
  SFFloat [in,out] shininess        0.2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

Box

```
class AFX_EXT_CLASS CBox : public CX3DGeometryNode
{
public:
        SFVec3f     getSize();
        void        setSize(SFVec3f vec);
        BOOL        getSolid();
        void        setSolid(BOOL value);

protected:
        SFVec3f     size;
        SFBool      solid;
}
```

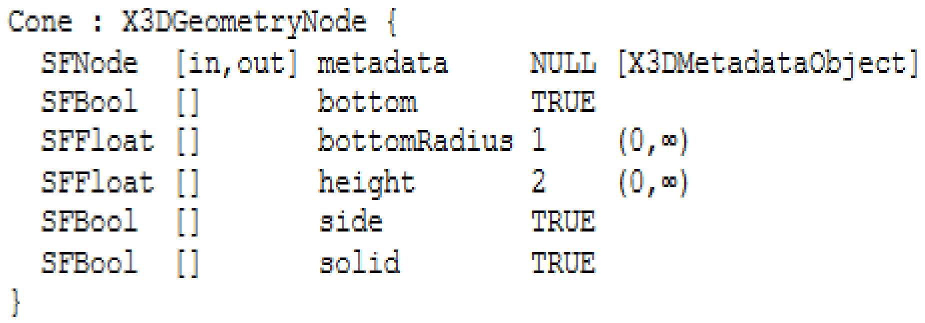


```
Box : X3DGeometryNode {
  SFNode  [in,out] metadata NULL  [X3DMetadataObject]
  SFVec3f []       size     2 2 2 (0,∞)
  SFBool  []       solid    TRUE
}
```

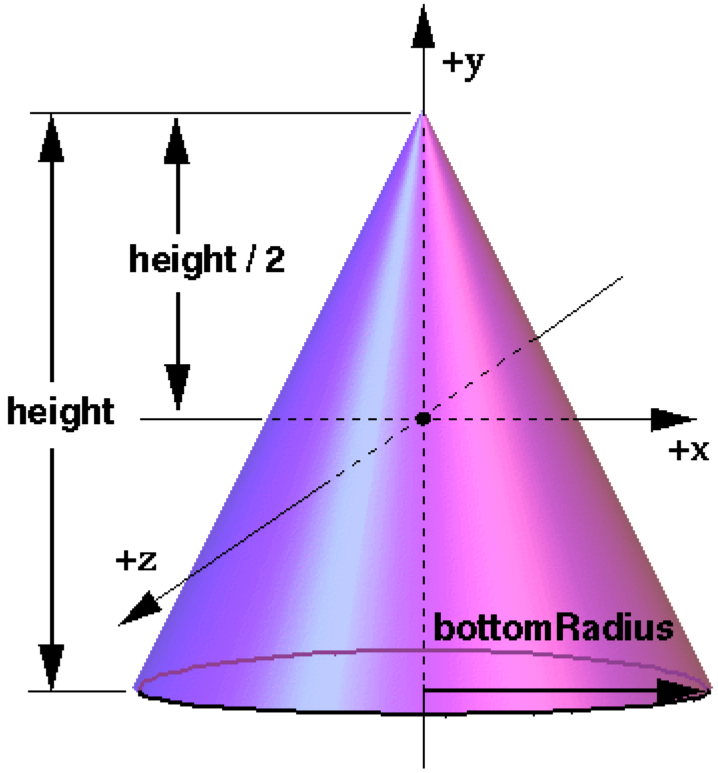


Cone

```
class AFX_EXT_CLASS CCone : public CX3DGeometryNode
{
public:
        void     setHeight(SFFloat fHeight);
        void     setBottomRadius(SFFloat fRadius);
        void     setBottom(SFBool bBottom);
        void     setSide(SFBool bSide);
        SFFloat getHeight();
        SFFloat getBottomRadius();
        SFBool isBottom();
        SFBool isSide();
        BOOL   getSolid();
        void     setSolid(BOOL value);
protected:
         SFFloat    height;
         SFFloat    bottomRadius;
         SFBool     bottom;
         SFBool     side;
         SFBool     solid;
}
```

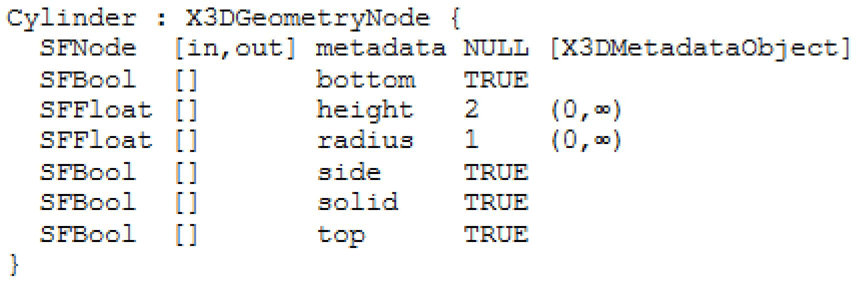


```
Cone : X3DGeometryNode {
  SFNode  [in,out] metadata     NULL [X3DMetadataObject]
  SFBool  []       bottom       TRUE
  SFFloat []       bottomRadius 1    (0,∞)
  SFFloat []       height       2    (0,∞)
  SFBool  []       side         TRUE
  SFBool  []       solid        TRUE
}
```

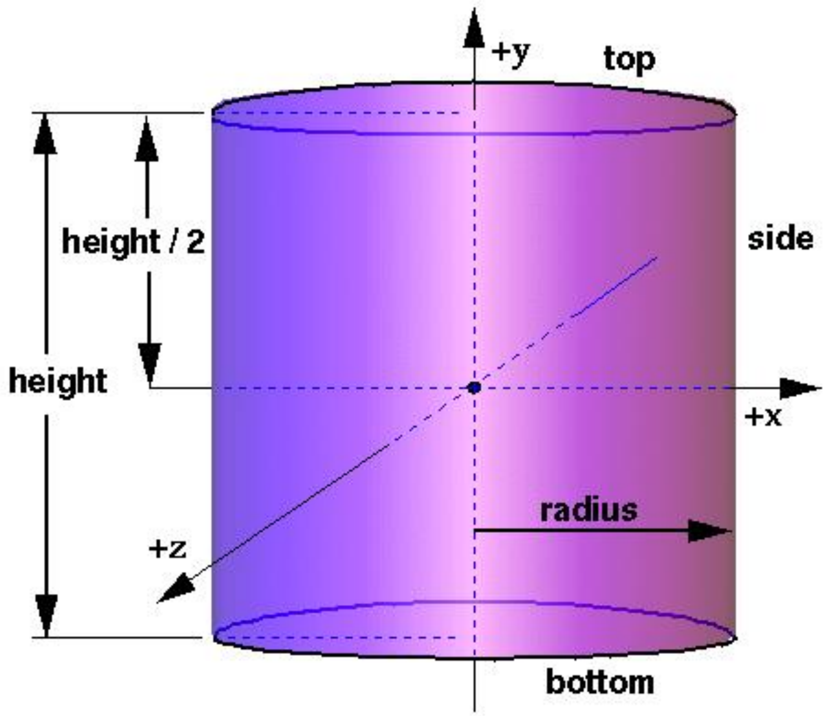


Cylinder

```
class AFX_EXT_CLASS CCylinder : public CX3DGeometryNode
{
public:
        void     setHeight(SFFloat fHeight);
        void     setRadius(SFFloat fRadius);
        void     setBottom(SFBool bBottom);
        void     setSide(SFBool bSide);
        void     setTop(SFBool bTop);
        SFFloat getHeight();
        SFFloat getRadius();
        SFBool isBottom();
        SFBool isSide();
        SFBool isTop();
        BOOL   getSolid();
        void     setSolid(BOOL value);
protected:
        SFFloat     height;
        SFFloat     radius;
        SFBool      bottom;
        SFBool      side;
        SFBool      top;
        SFBool     solid;
};
```

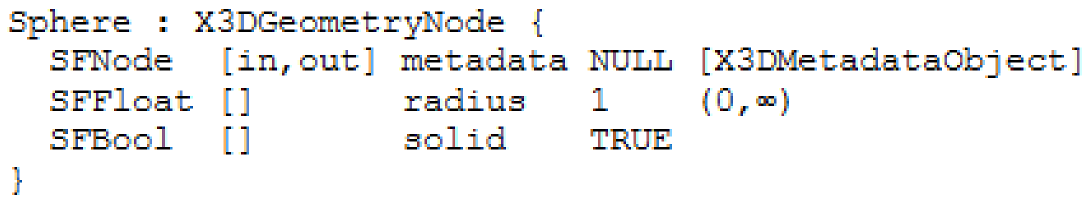


```
Cylinder : X3DGeometryNode {
  SFNode  [in,out] metadata NULL [X3DMetadataObject]
  SFBool  []       bottom   TRUE
  SFFloat []       height   2    (0,∞)
  SFFloat []       radius   1    (0,∞)
  SFBool  []       side     TRUE
  SFBool  []       solid    TRUE
  SFBool  []       top      TRUE
}
```

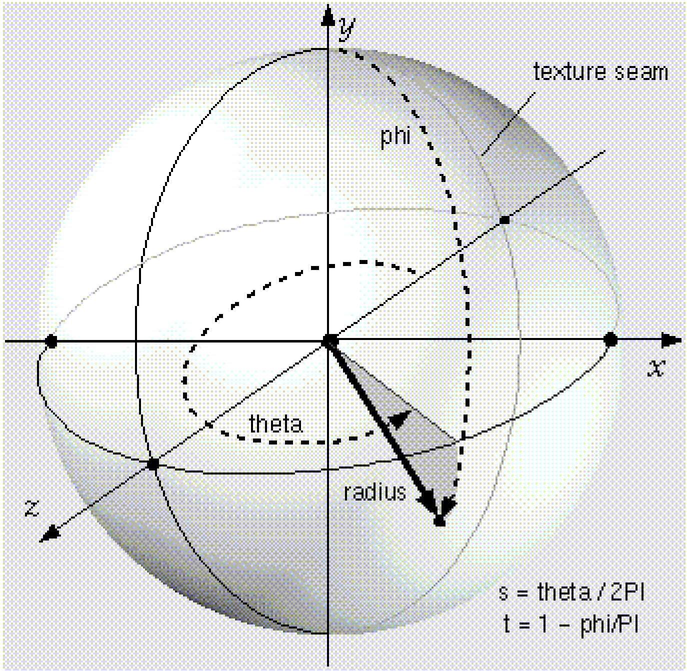


Sphere

```
class AFX_EXT_CLASS CSphere : public CX3DGeometryNode
{
public:
      void      setRadius(SFFloat fRadius);
      SFFloat getRadius();
      BOOL    getSolid();
      void      setSolid(BOOL value);
protected:
      SFFloat  radius;
      SFBool   solid;
};
```

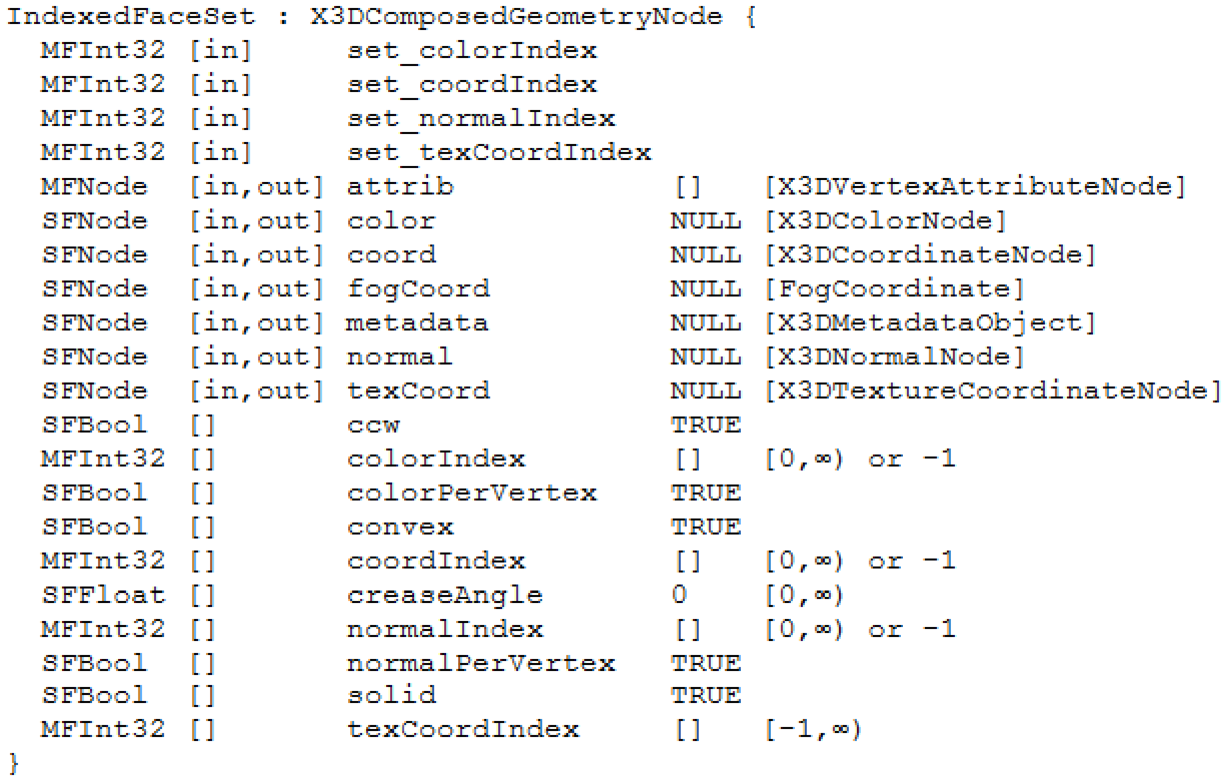


```
Sphere : X3DGeometryNode {
  SFNode  [in,out] metadata NULL [X3DMetadataObject]
  SFFloat []       radius   1    (0,∞)
  SFBool  []       solid    TRUE
}
```



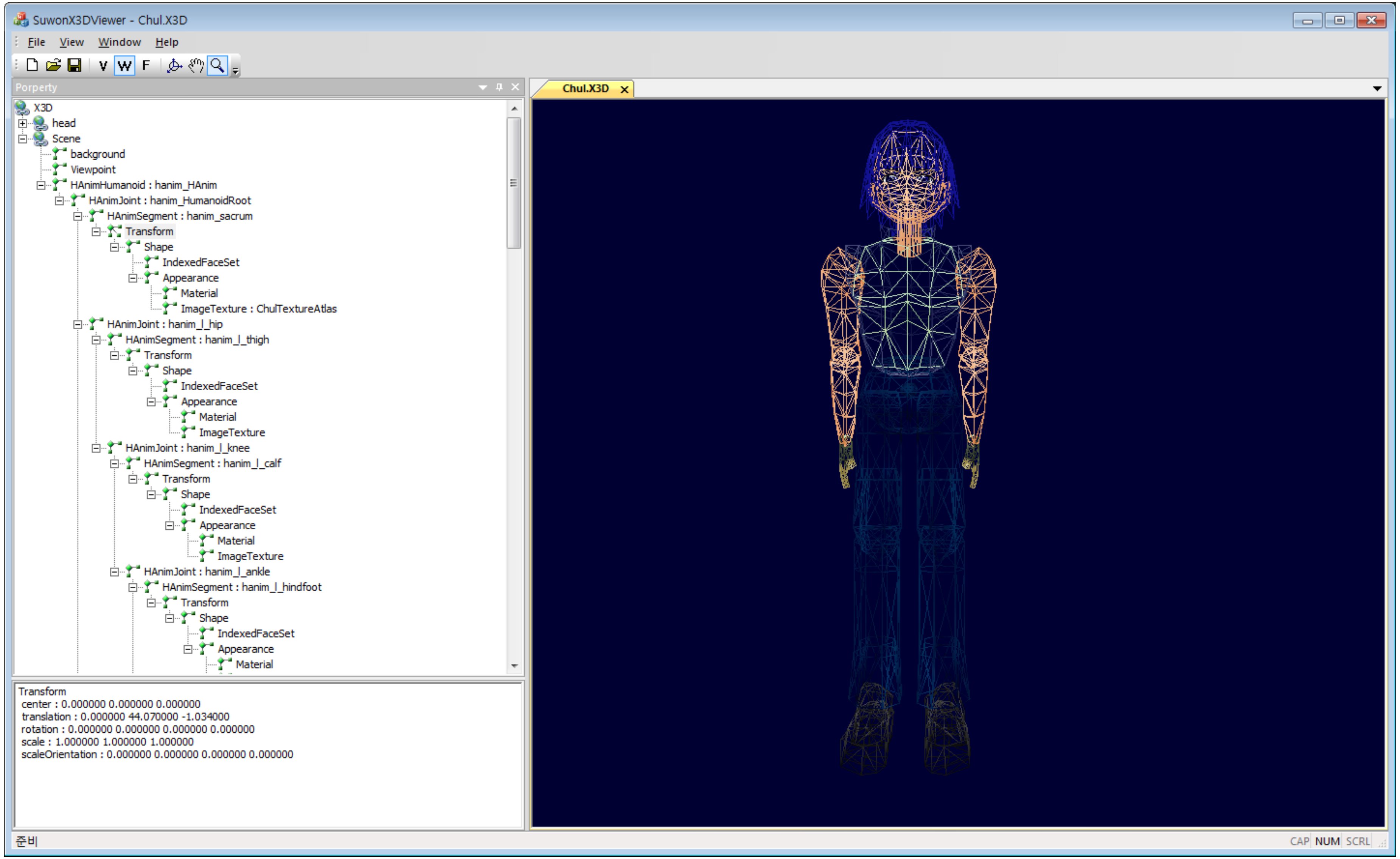
IndexedFaceSet

```
class AFX_EXT_CLASS CIndexedFaceSet : public CX3DComposedGeometryNode
{
public:
    void setCreaseAngle(SFFloat angle);
    SFFloat getCreaseAngle();
    void set_colorIndex(CString strIndex);
    void set_coordIndex(CString strIndex);
    void set_normalIndex(CString strIndex);
    void set_texCoordIndex(CString strIndex);
    void setIndex(CString strIndex, int nType);

protected:
    MFInt32  colorIndex;
    MFInt32  coordIndex;
    MFInt32 normalIndex;
    MFInt32 texCoordIndex;
};
```



```
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 Sample



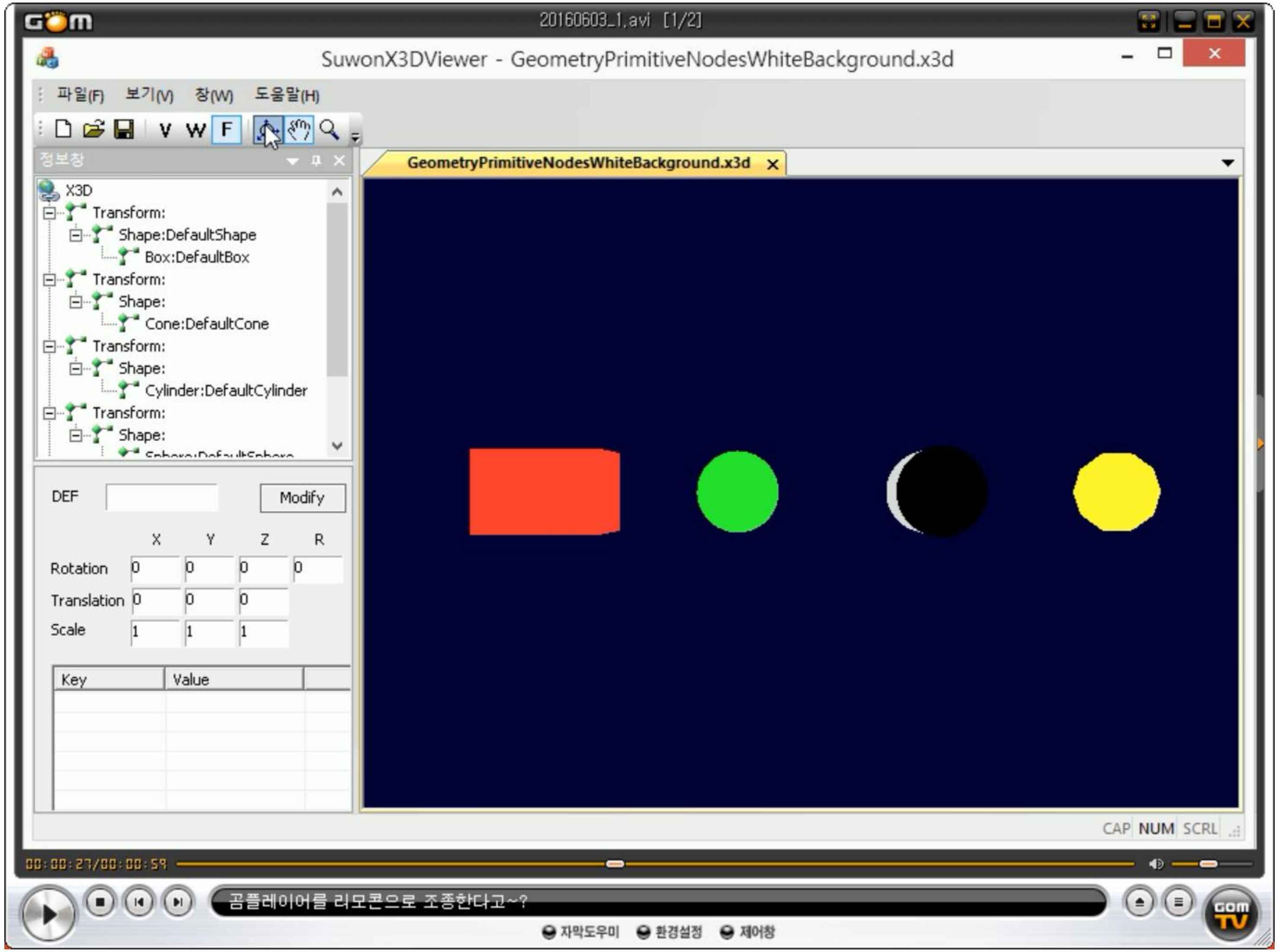
```
X3D
  head
  Scene
    background
    Viewpoint
    HAnimHumanoid : hanim_HAnim
      HAnimJoint : hanim_HumanoidRoot
        HAnimSegment : hanim_sacrum
          Transform
            Shape
              IndexedFaceSet
              Appearance
                Material
                ImageTexture : ChulTextureAtlas
        HAnimJoint : hanim_l_hip
          HAnimSegment : hanim_l_thigh
            Transform
              Shape
                IndexedFaceSet
                Appearance
                  Material
                  ImageTexture
          HAnimJoint : hanim_l_knee
            HAnimSegment : hanim_l_calf
              Transform
                Shape
                  IndexedFaceSet
                  Appearance
                    Material
                    ImageTexture
            HAnimJoint : hanim_l_ankle
              HAnimSegment : hanim_l_hindfoot
                Transform
                  Shape
                    IndexedFaceSet
                    Appearance
                      Material
```

```
Transform
 center : 0.000000 0.000000 0.000000
 translation : 0.000000 44.070000 -1.034000
 rotation : 0.000000 0.000000 0.000000 0.000000
 scale : 1.000000 1.000000 1.000000
 scaleOrientation : 0.000000 0.000000 0.000000 0.000000
```

X3DComposedGeometryNode (IndexedFaceSet Public Node)

```
class AFX_EXT_CLASS CX3DComposedGeometryNode : public CX3DGeometryNode
{
public:
    void     setCcw(SFBool bBool);
    SFBool   getCcw();
    void      setColorPerVertex(SFBool bBool);
    SFBool    getColorPerVertex();
    void      setConvex(SFBool bBool);
    SFBool    getConvex();
    void      setNormalPerVertex(SFBool bBool);
    SFBool    getNormalPerVertex();
    void      setSolid(SFBool bBool);
    SFBool    getSolid();
    CCoordinate*   getCoord();
    void      setCoord(CX3DCoordinateNode* node);
    void      setCoord(CX3DPrototypeInstance protoInstance);
    CFogCoordinate*   getFogCoord();
    void       setFogCoord(CFogCoordinate* node);
    void       setFogCoord(CX3DPrototypeInstance protoInstance);
    CColor*    getColor();
    void       setColor(CX3DColorNode * color);
    void       setColor(CX3DPrototypeInstance protoInstance);
        CNormal*   getNormal();
        voidsetNormal(CX3DNormalNode * node);
    voidsetNormal(CX3DPrototypeInstance protoInstance);

    CTextureCoordinate* getTexCoord();
    voidsetTexCoord(CTextureCoordinate* node);
    voidsetTexCoord(CX3DPrototypeInstance protoInstance);

    //Attributes
    protected:
    SFBoolccw;
    SFBoolcolorPerVertex;
    SFBoolconvex;
    SFBoolnormalPerVertex;
    SFBoolsolid;

    CColorcolor;
    CCoordinatecoord;
    CFogCoordinate  fogCoord;
    CNormalnormal;
    CTextureCoordinate texCoord;
    };
```

X3D C++ Binding Viewer Demo


20160603_1.avi [1/2]
SuwonX3DViewer - GeometryPrimitiveNodesWhiteBackground.x3d
파일(F)
보기(V)
창(W)
도움말(H)
V
W
F
정보창
GeometryPrimitiveNodesWhiteBackground.x3d
Modify
CAP NUM SCRL
00:00:27/00:00:59
곰플레이어를 러모콘으로 조종한다고~?
자막도우미
환경설정
제어창

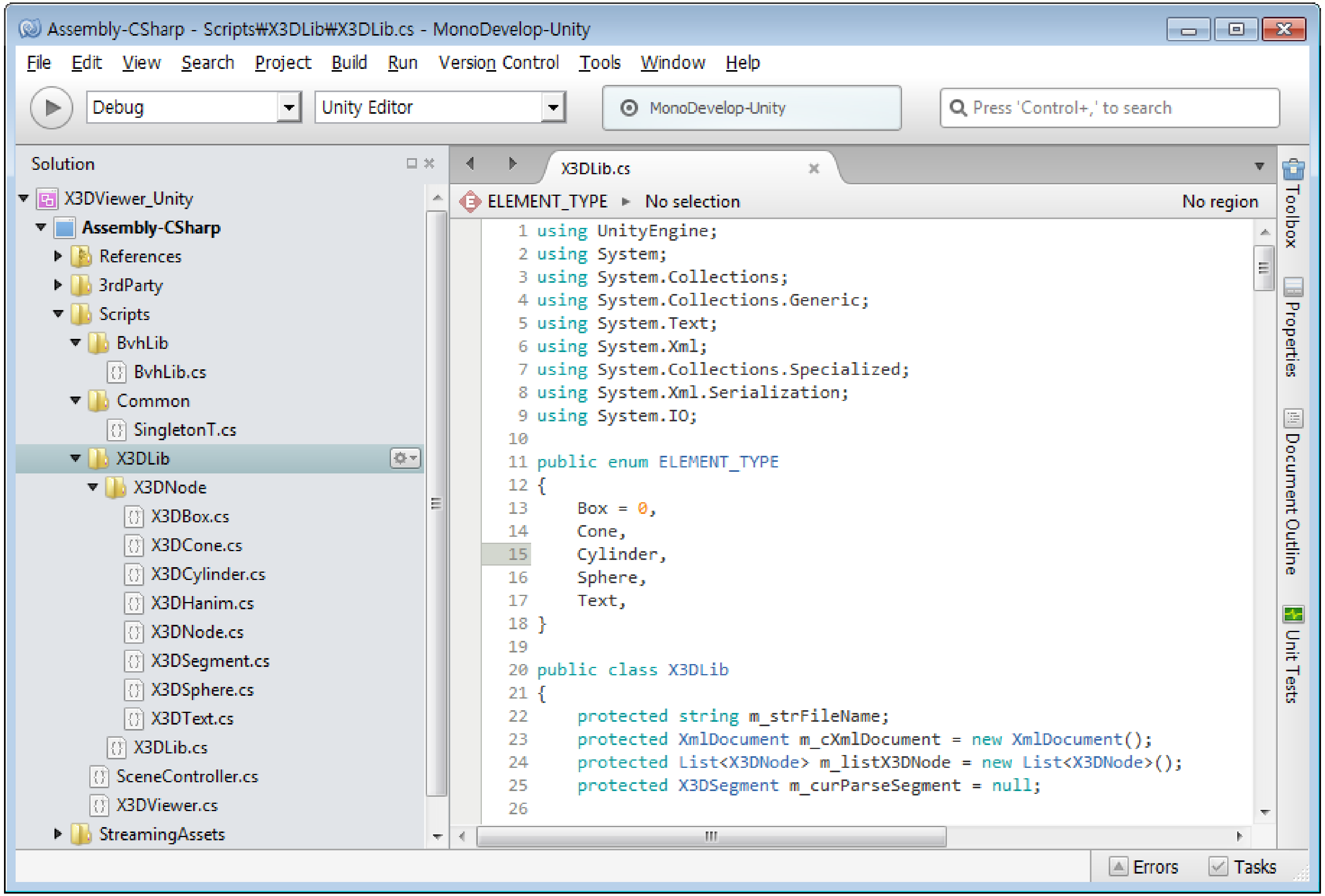
- X3D
  - Transform:
    - Shape:DefaultShape
      - Box:DefaultBox
  - Transform:
    - Shape:
      - Cone:DefaultCone
  - Transform:
    - Shape:
      - Cylinder:DefaultCylinder
  - Transform:
    - Shape:
      - Sphere:DefaultSphere

DEF

	X	Y	Z	R
Rotation	0	0	0	0
Translation	0	0	0	
Scale	1	1	1	

Key	Value

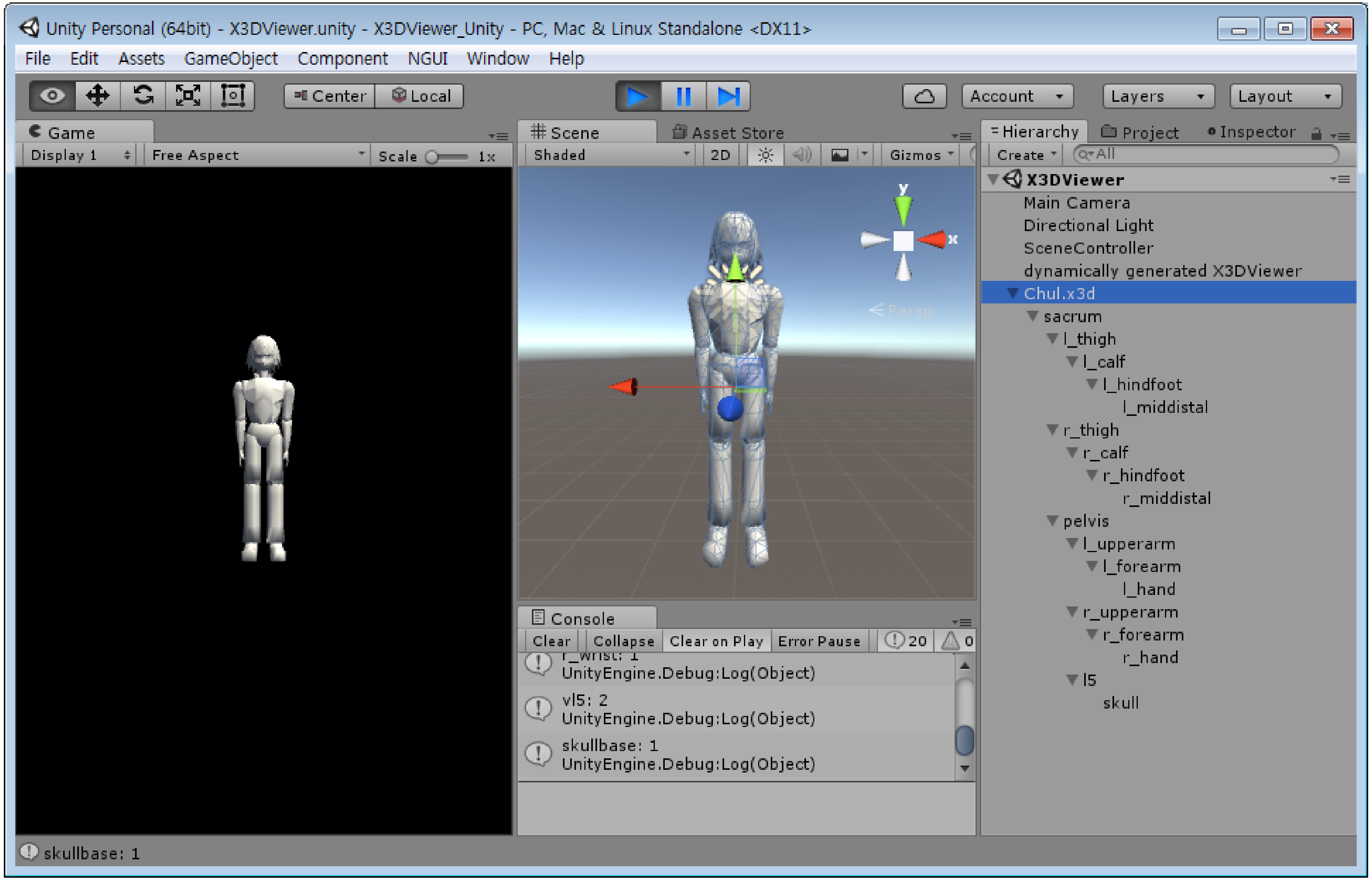
X3D C# Binding Viewer (Unity)



Assembly-CSharp - Scripts₩X3DLib₩X3DLib.cs - MonoDevelop-Unity

```
using UnityEngine;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.Collections.Specialized;
using System.Xml.Serialization;
using System.IO;

public enum ELEMENT_TYPE
{
    Box = 0,
    Cone,
    Cylinder,
    Sphere,
    Text,
}

public class X3DLib
{
    protected string m_strFileName;
    protected XmlDocument m_cXmlDocument = new XmlDocument();
    protected List<X3DNode> m_listX3DNode = new List<X3DNode>();
    protected X3DSegment m_curParseSegment = null;

```

X3D C# Binding Viewer (Unity)

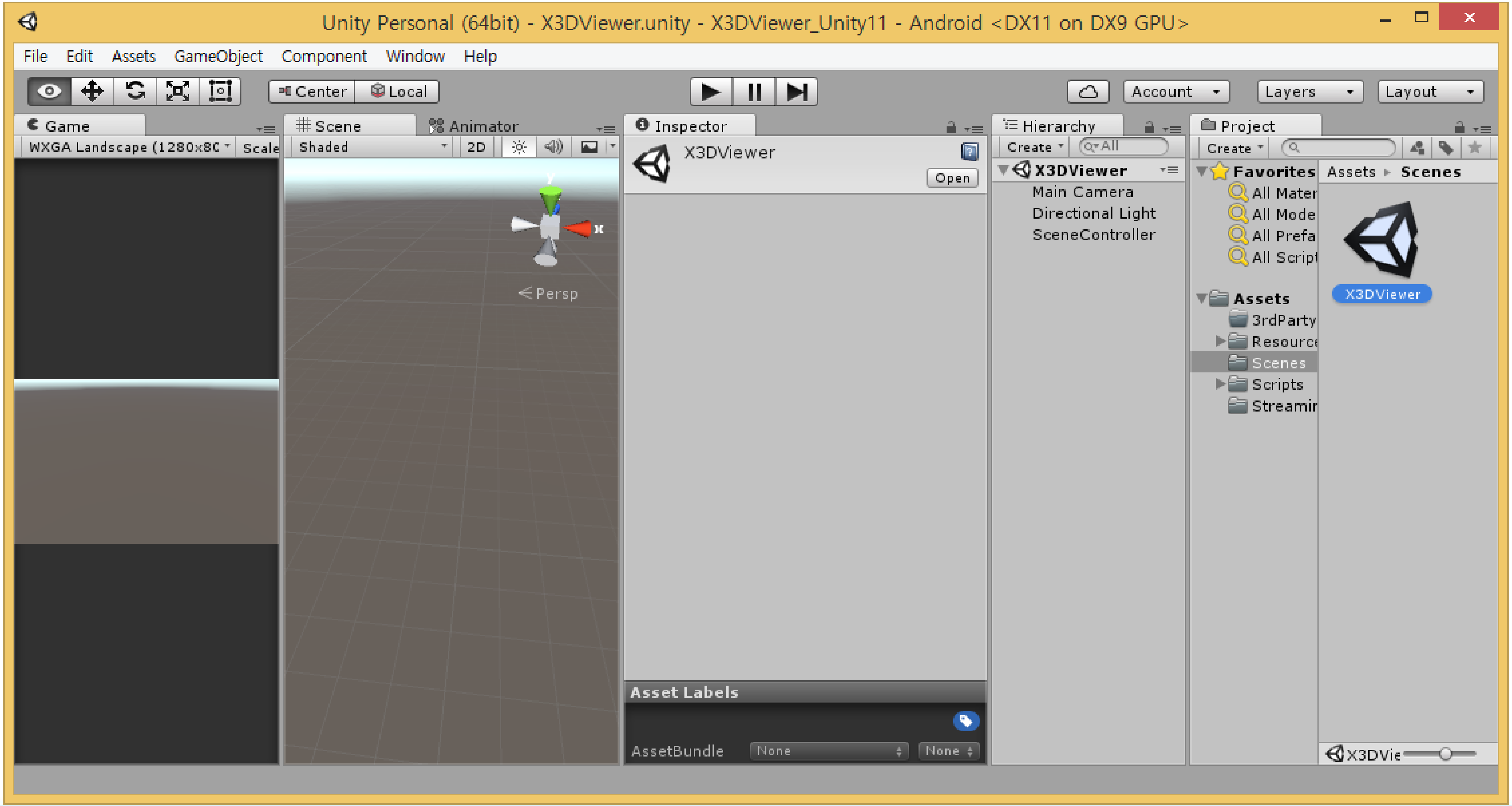


```
X3DViewer
  Main Camera
  Directional Light
  SceneController
  dynamically generated X3DViewer
  Chul.x3d
    sacrum
      l_thigh
        l_calf
          l_hindfoot
            l_middistal
      r_thigh
        r_calf
          r_hindfoot
            r_middistal
      pelvis
        l_upperarm
          l_forearm
            l_hand
        r_upperarm
          r_forearm
            r_hand
        l5
          skull
```

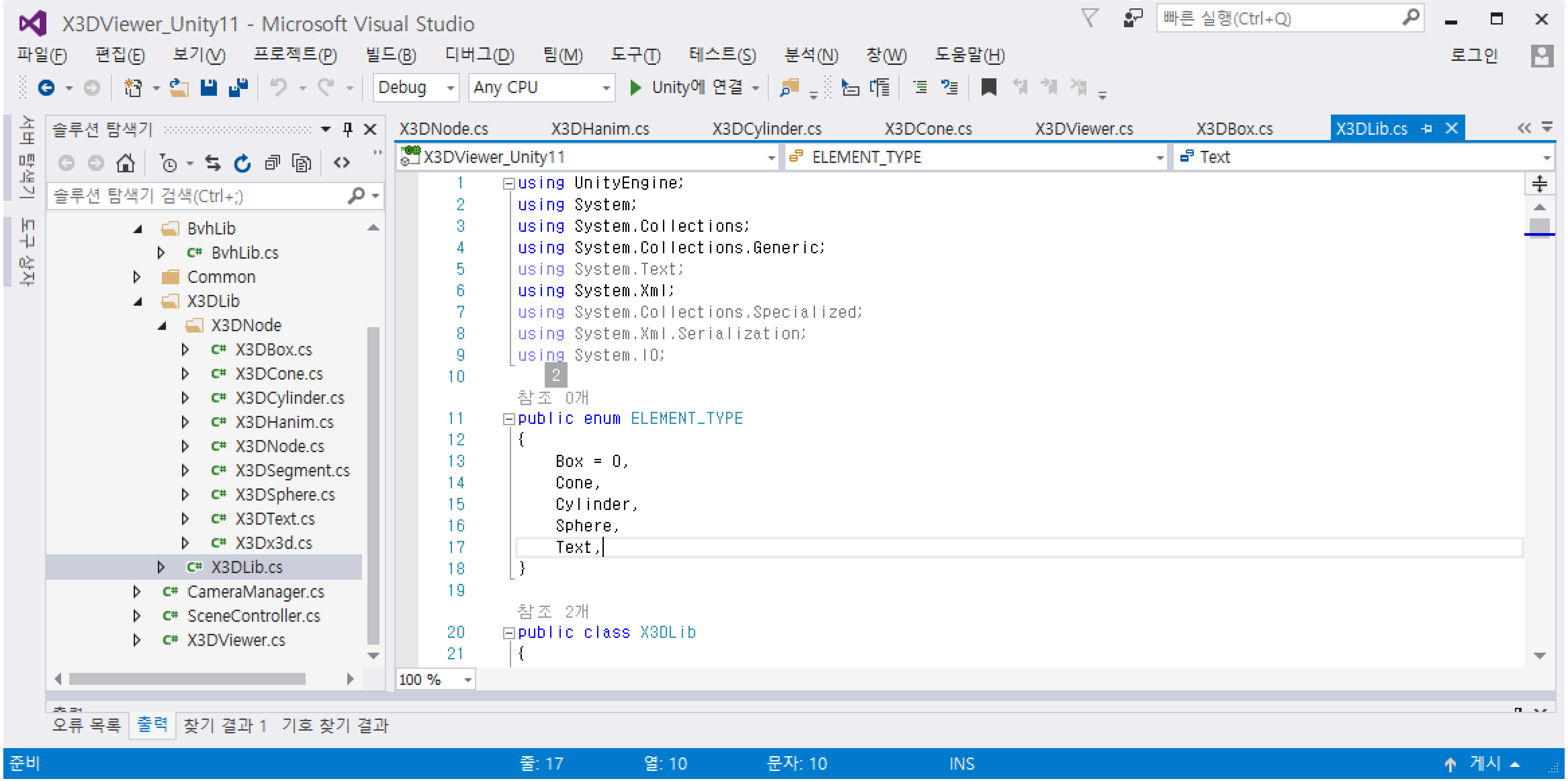
```
r_wrist: 1
UnityEngine.Debug:Log(Object)
vl5: 2
UnityEngine.Debug:Log(Object)
skullbase: 1
UnityEngine.Debug:Log(Object)
```

```
skullbase: 1
```

X3D C# Binding Viewer (Unity)



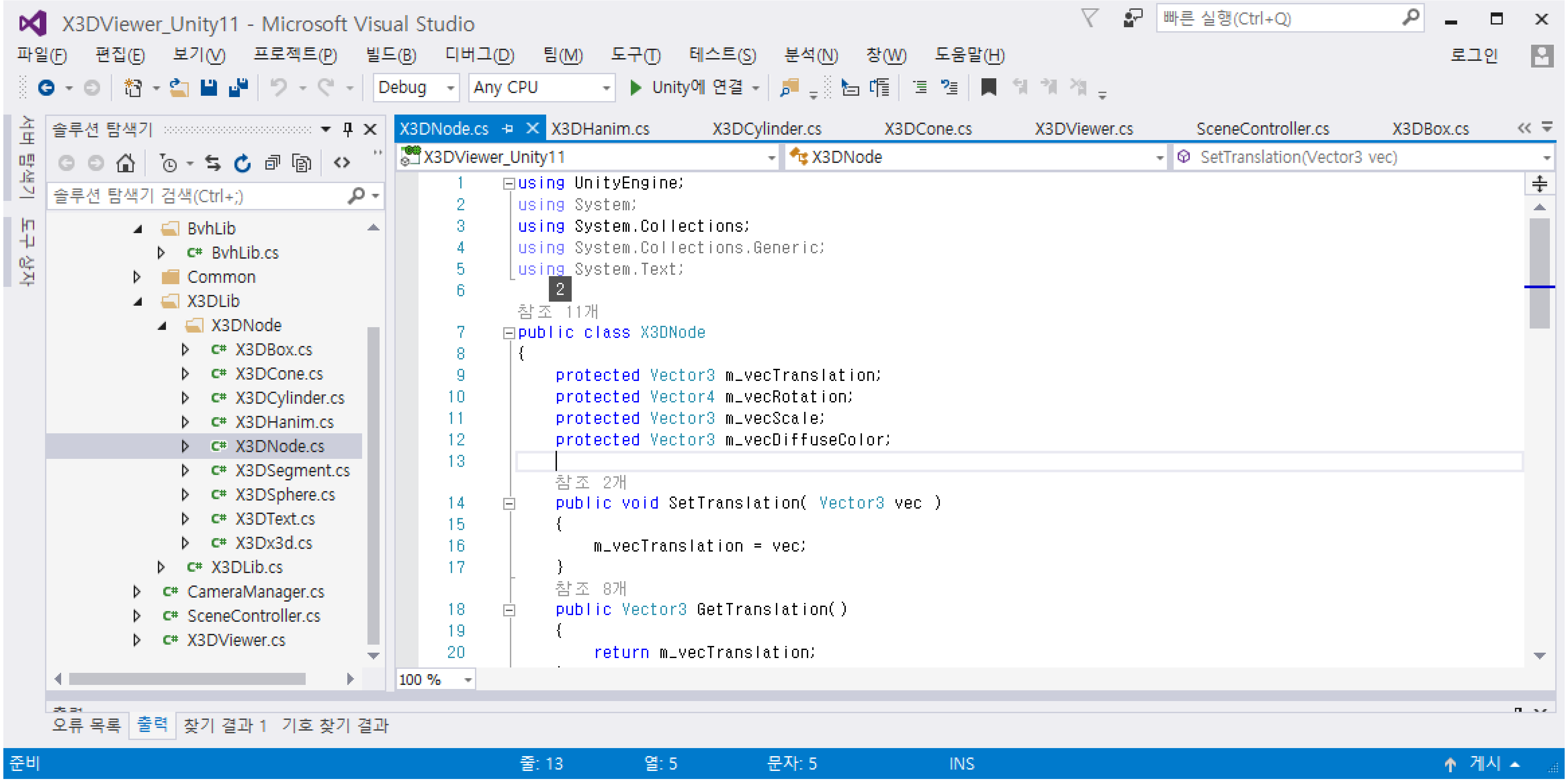
Lib Class



X3DViewer_Unity11 - Microsoft Visual Studio
빠른 실행(Ctrl+Q)
파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H)
로그인
Debug
Any CPU
Unity에 연결
서버 탐색기
도구 상자
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
BvhLib
BvhLib.cs
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
X3DNode.cs X3DHanim.cs X3DCylinder.cs X3DCone.cs X3DViewer.cs X3DBox.cs X3DLib.cs
X3DViewer_Unity11
ELEMENT_TYPE
Text
100 %
오류 목록 출력 찾기 결과 1 기호 찾기 결과
준비
줄: 17 열: 10 문자: 10 INS
게시

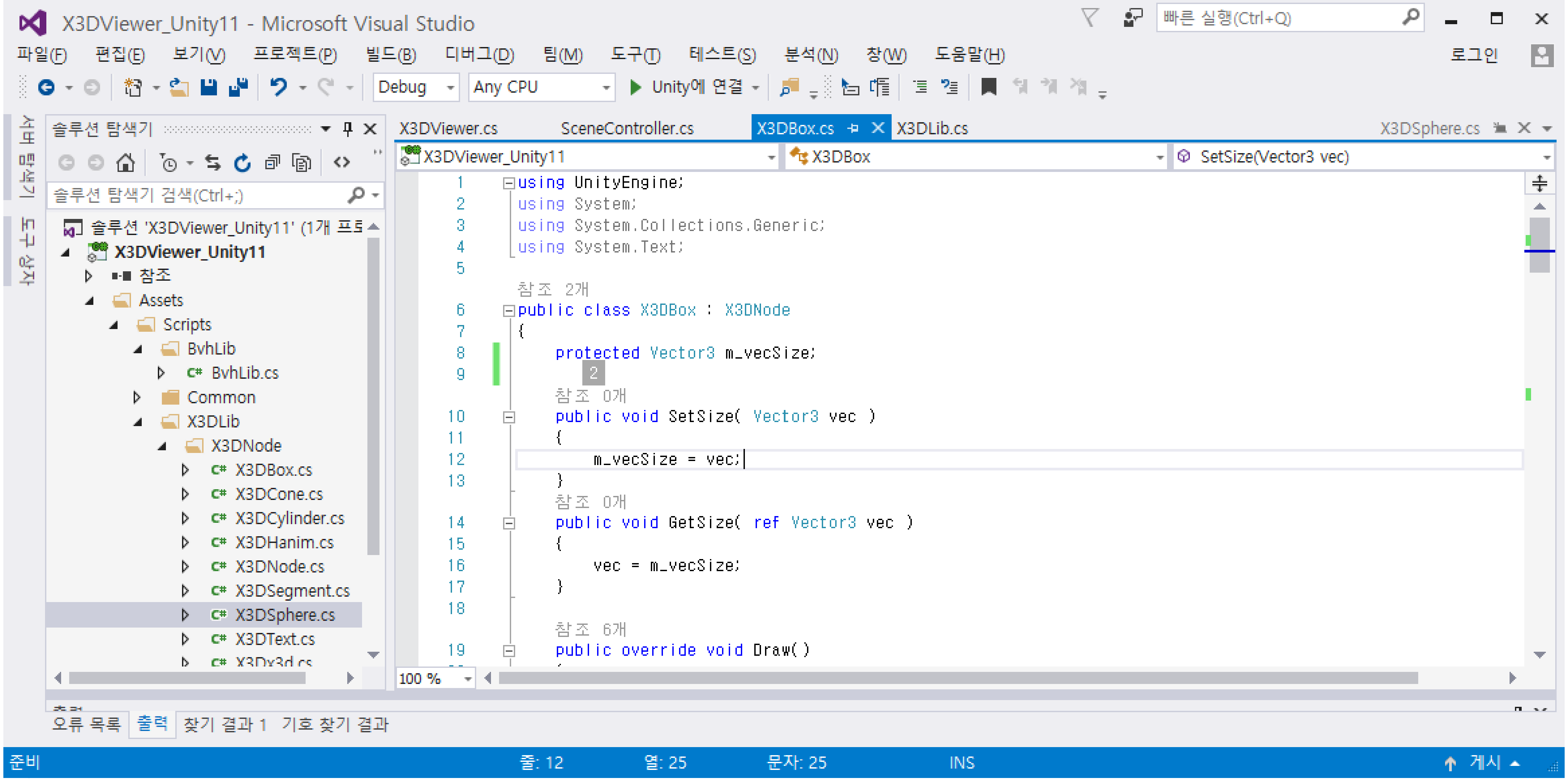
```
using UnityEngine;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.Collections.Specialized;
using System.Xml.Serialization;
using System.IO;

참조 0개
public enum ELEMENT_TYPE
{
    Box = 0,
    Cone,
    Cylinder,
    Sphere,
    Text,
}

참조 2개
public class X3DLib
{
```

Base Node



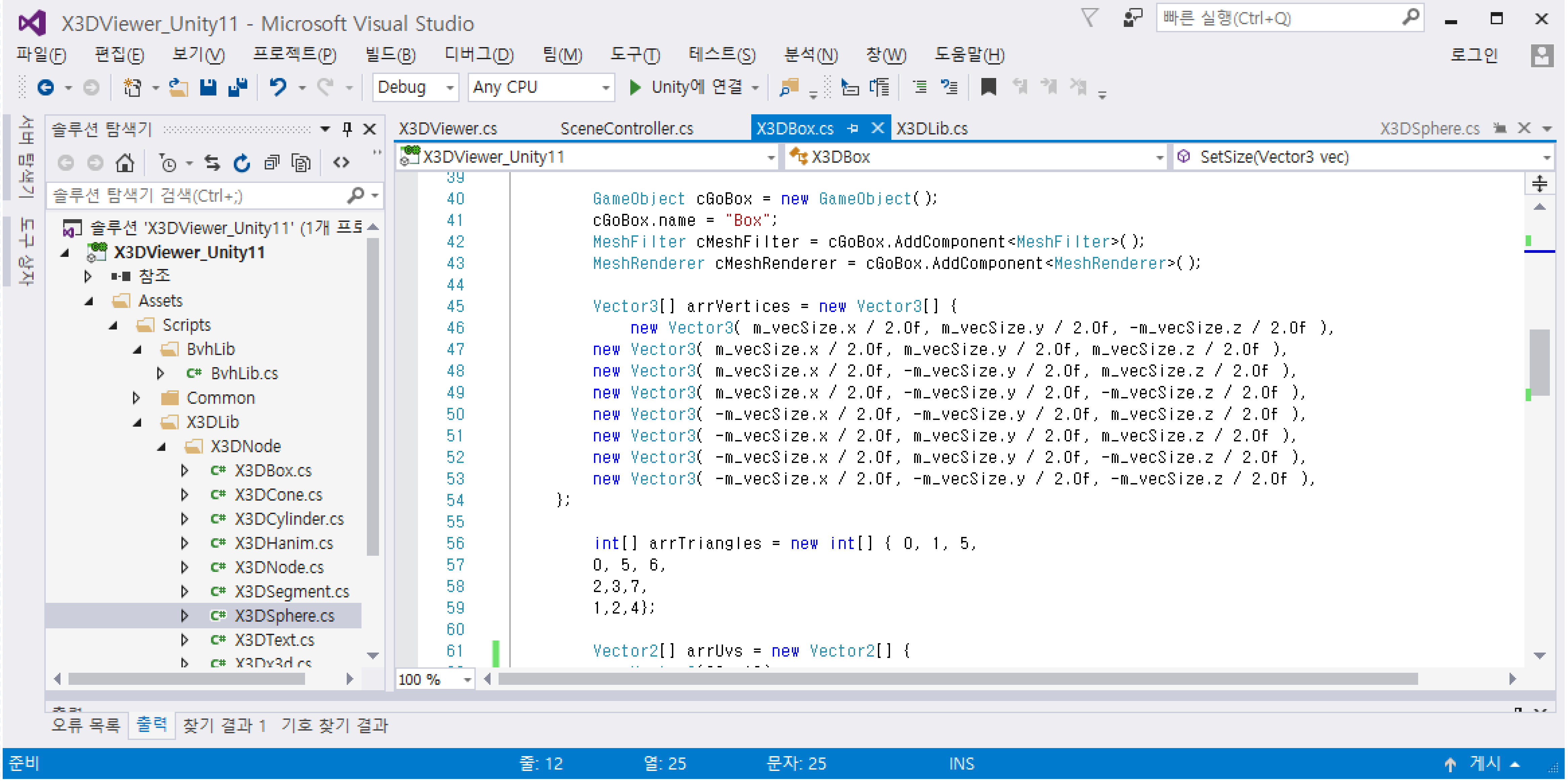
```
using UnityEngine;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;

참조 11개
public class X3DNode
{
    protected Vector3 m_vecTranslation;
    protected Vector4 m_vecRotation;
    protected Vector3 m_vecScale;
    protected Vector3 m_vecDiffuseColor;

    참조 2개
    public void SetTranslation( Vector3 vec )
    {
        m_vecTranslation = vec;
    }
    참조 8개
    public Vector3 GetTranslation( )
    {
        return m_vecTranslation;
```

Box Class



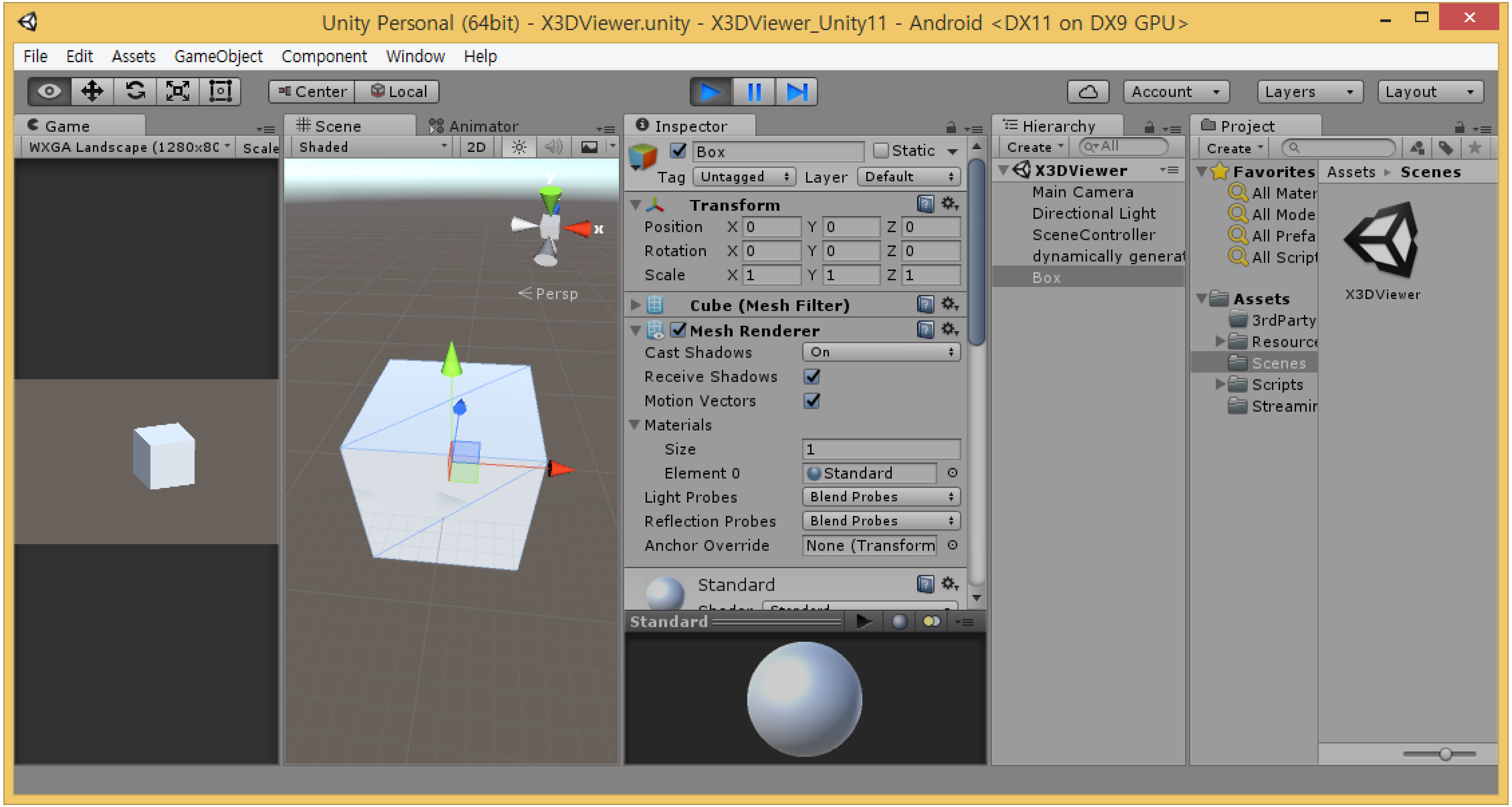
```
using UnityEngine;
using System;
using System.Collections.Generic;
using System.Text;

참조 2개
public class X3DBox : X3DNode
{
    protected Vector3 m_vecSize;

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

    참조 6개
    public override void Draw( )
```

Box Class



```
        GameObject cGoBox = new GameObject();
        cGoBox.name = "Box";
        MeshFilter cMeshFilter = cGoBox.AddComponent<MeshFilter>();
        MeshRenderer cMeshRenderer = cGoBox.AddComponent<MeshRenderer>();

        Vector3[] arrVertices = new Vector3[] {
            new Vector3( m_vecSize.x / 2.0f, m_vecSize.y / 2.0f, -m_vecSize.z / 2.0f ),
        new Vector3( m_vecSize.x / 2.0f, m_vecSize.y / 2.0f, m_vecSize.z / 2.0f ),
        new Vector3( m_vecSize.x / 2.0f, -m_vecSize.y / 2.0f, m_vecSize.z / 2.0f ),
        new Vector3( m_vecSize.x / 2.0f, -m_vecSize.y / 2.0f, -m_vecSize.z / 2.0f ),
        new Vector3( -m_vecSize.x / 2.0f, -m_vecSize.y / 2.0f, m_vecSize.z / 2.0f ),
        new Vector3( -m_vecSize.x / 2.0f, m_vecSize.y / 2.0f, m_vecSize.z / 2.0f ),
        new Vector3( -m_vecSize.x / 2.0f, m_vecSize.y / 2.0f, -m_vecSize.z / 2.0f ),
        new Vector3( -m_vecSize.x / 2.0f, -m_vecSize.y / 2.0f, -m_vecSize.z / 2.0f ),
    };

        int[] arrTriangles = new int[] { 0, 1, 5,
        0, 5, 6,
        2,3,7,
        1,2,4};

        Vector2[] arrUvs = new Vector2[] {
```

Box Class



Unity Personal (64bit) - X3DViewer.unity - X3DViewer_Unity11 - Android <DX11 on DX9 GPU>

Inspector

☑ Box □ Static

Tag Untagged Layer Default

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Cube (Mesh Filter)

☑ Mesh Renderer

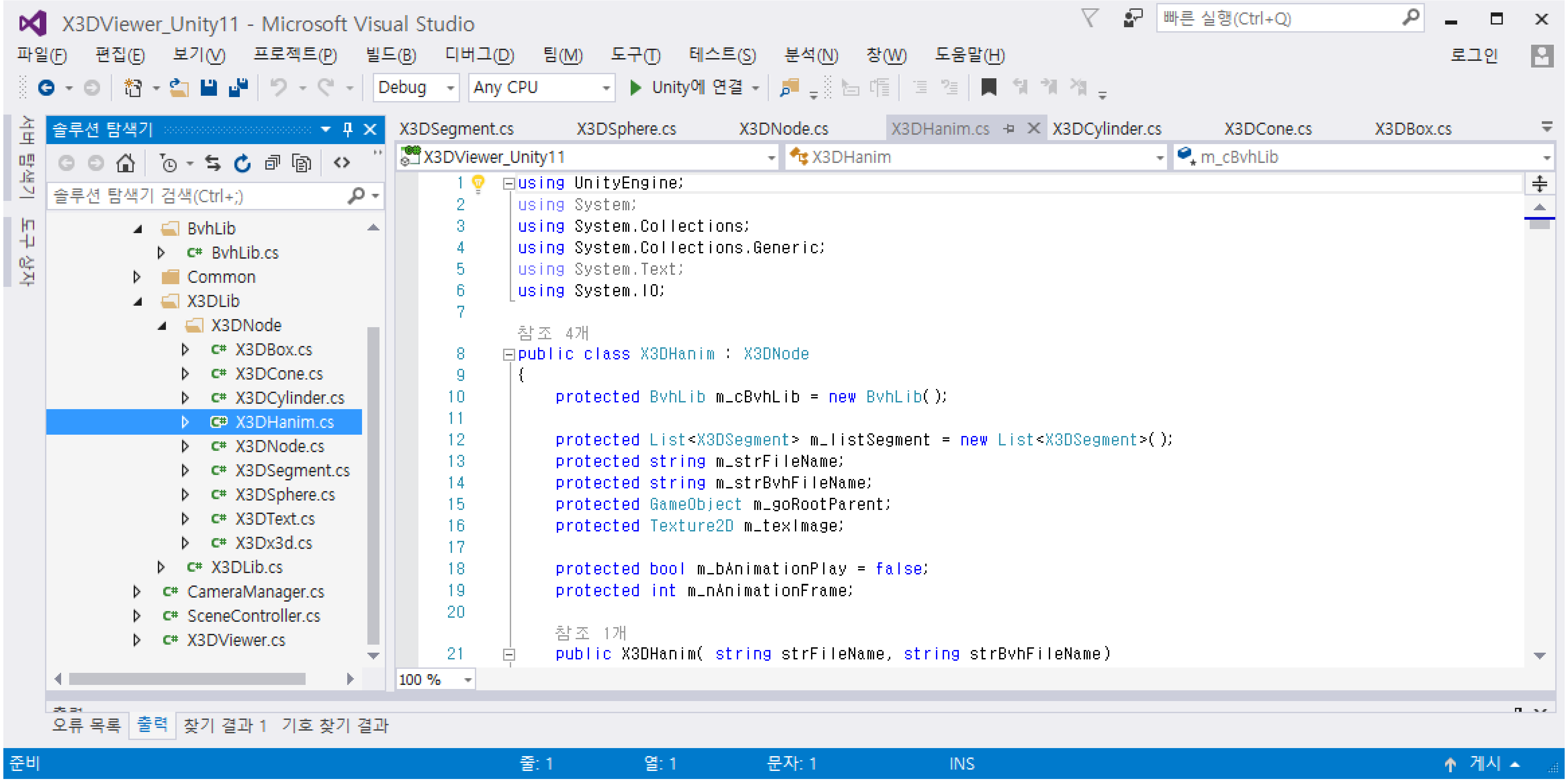
Cast Shadows	On
Receive Shadows	☑
Motion Vectors	☑
Materials	
Size	1
Element 0	Standard
Light Probes	Blend Probes
Reflection Probes	Blend Probes
Anchor Override	None (Transform

Standard

Hierarchy

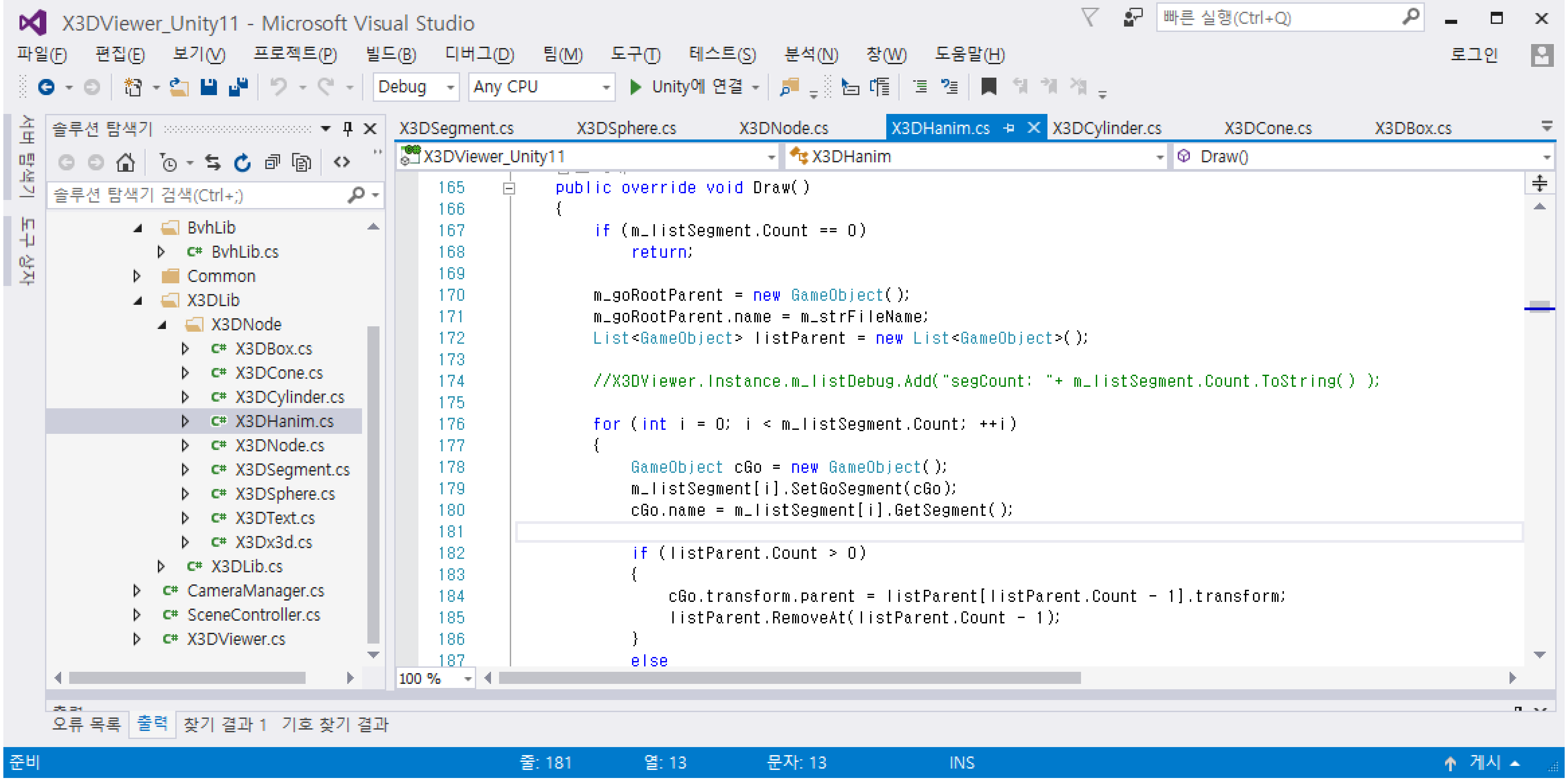
X3DViewer
- Main Camera
- Directional Light
- SceneController
- dynamically generat
- Box

H-Anim Class



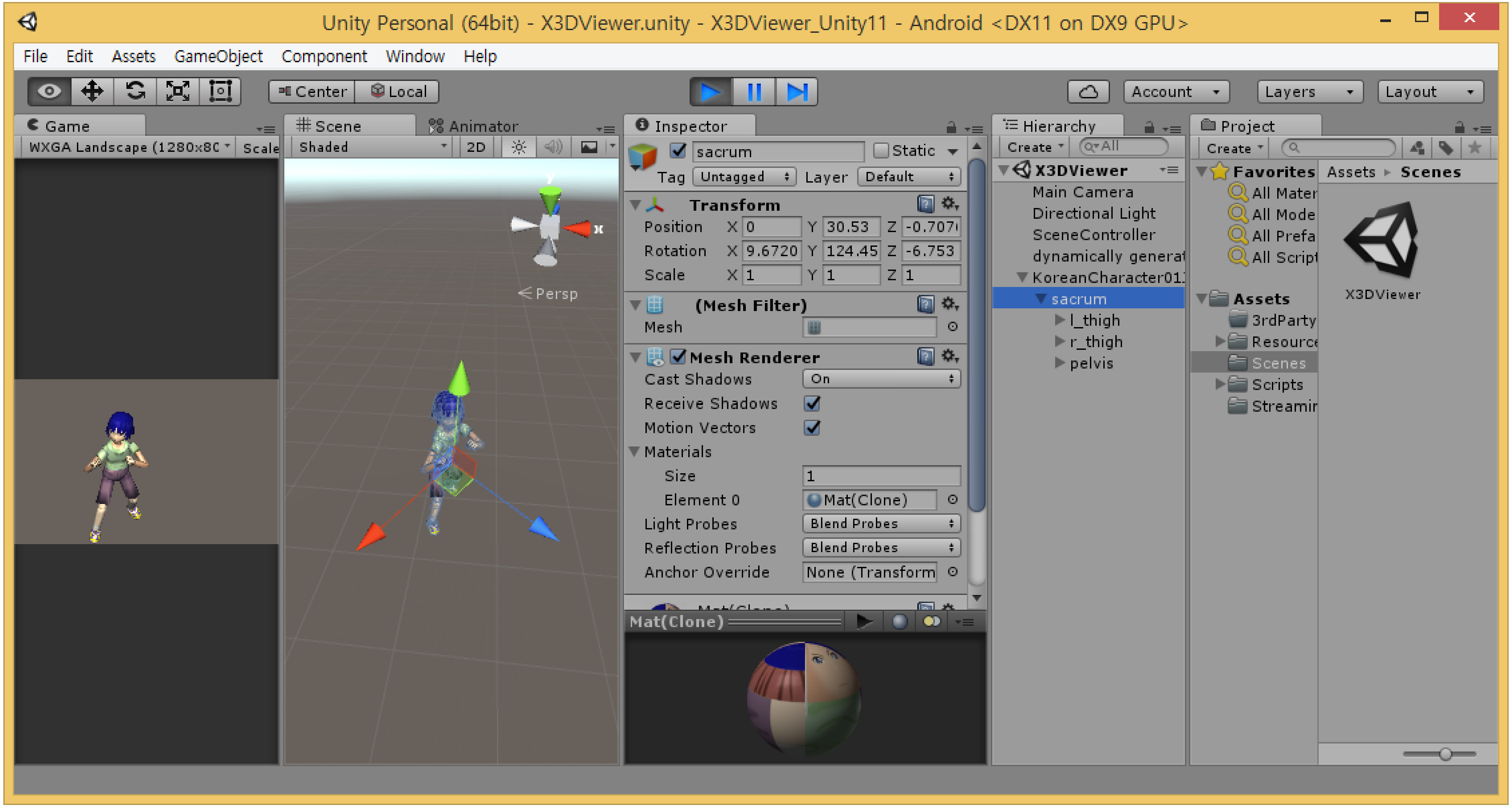
```
using UnityEngine;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;
using System.IO;

참조 4개
public class X3DHanim : X3DNode
{
    protected BvhLib m_cBvhLib = new BvhLib();

    protected List<X3DSegment> m_listSegment = new List<X3DSegment>();
    protected string m_strFileName;
    protected string m_strBvhFileName;
    protected GameObject m_goRootParent;
    protected Texture2D m_texImage;

    protected bool m_bAnimationPlay = false;
    protected int m_nAnimationFrame;

    참조 1개
    public X3DHanim( string strFileName, string strBvhFileName)
```

H-Anim Class



```
public override void Draw()
{
    if (m_listSegment.Count == 0)
        return;

    m_goRootParent = new GameObject();
    m_goRootParent.name = m_strFileName;
    List<GameObject> listParent = new List<GameObject>();

    //X3DViewer.Instance.m_listDebug.Add("segCount: "+ m_listSegment.Count.ToString() );

    for (int i = 0; i < m_listSegment.Count; ++i)
    {
        GameObject cGo = new GameObject();
        m_listSegment[i].SetGoSegment(cGo);
        cGo.name = m_listSegment[i].GetSegment();

        if (listParent.Count > 0)
        {
            cGo.transform.parent = listParent[listParent.Count - 1].transform;
            listParent.RemoveAt(listParent.Count - 1);
        }
        else
```

H-Anim Character Animation



Unity Personal (64bit) - X3DViewer.unity - X3DViewer_Unity11 - Android <DX11 on DX9 GPU>

Inspector

☑ sacrum □ Static

Tag Untagged Layer Default

Transform

	X	Y	Z
Position	0	30.53	-0.707(
Rotation	9.6720	124.45	-6.753
Scale	1	1	1

(Mesh Filter)

Mesh

☑ **Mesh Renderer**

Cast Shadows: On

Receive Shadows: ☑

Motion Vectors: ☑

Materials

Size: 1

Element 0: Mat(Clone)

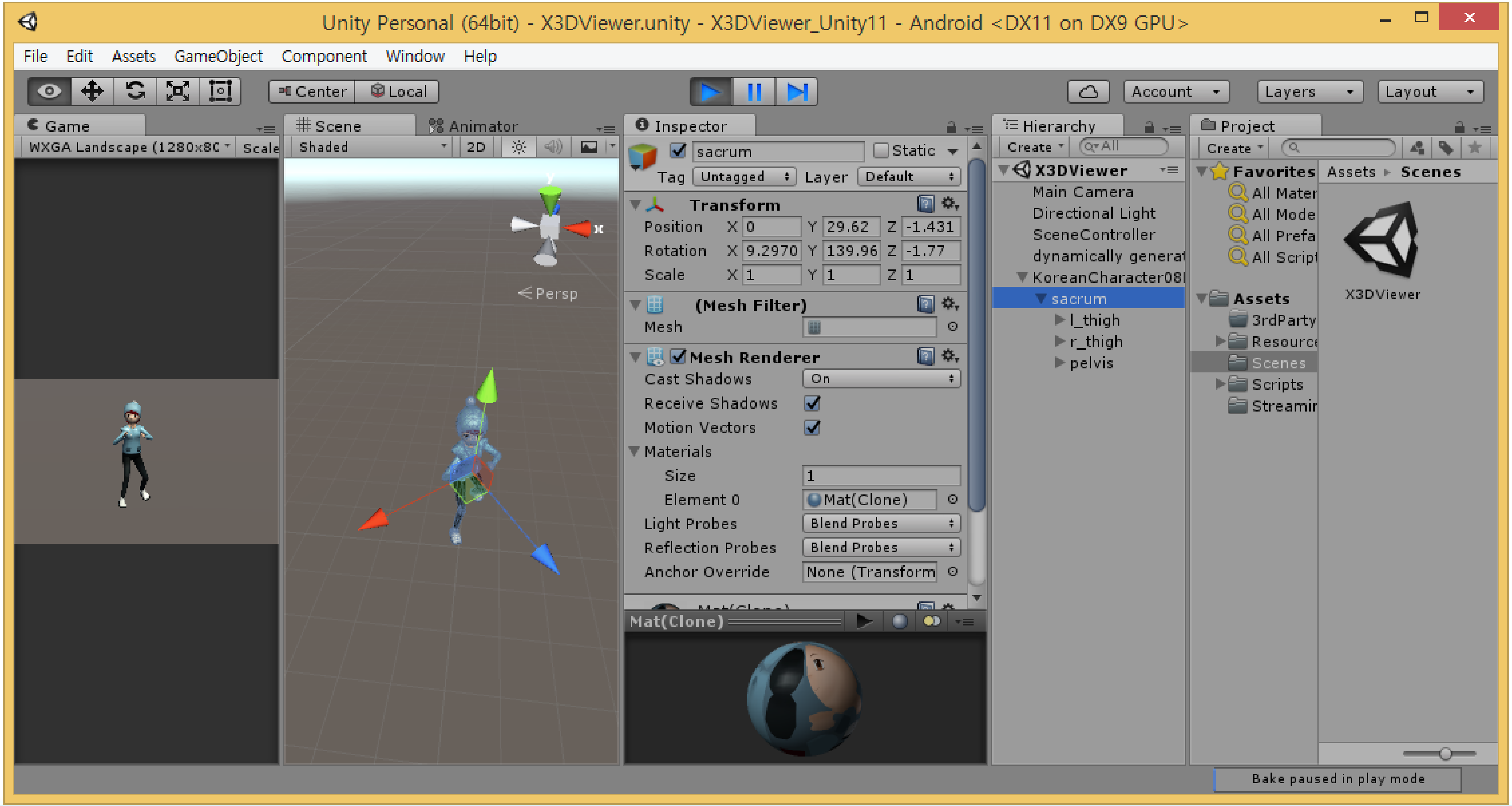
Light Probes: Blend Probes

Reflection Probes: Blend Probes

Anchor Override: None (Transform

Mat(Clone)

H-Anim Character Animation (Video)



Unity Personal (64bit) - X3DViewer.unity - X3DViewer_Unity11 - Android <DX11 on DX9 GPU>

Inspector

☑ sacrum □ Static

Tag Untagged Layer Default

Transform

Position	X	0	Y	29.62	Z	-1.431
Rotation	X	9.2970	Y	139.96	Z	-1.77
Scale	X	1	Y	1	Z	1

(Mesh Filter)

Mesh

☑ Mesh Renderer

Cast Shadows	On
Receive Shadows	☑
Motion Vectors	☑
Materials	
Size	1
Element 0	Mat(Clone)
Light Probes	Blend Probes
Reflection Probes	Blend Probes
Anchor Override	None (Transform

Mat(Clone)

Hierarchy

- X3DViewer
  - Main Camera
  - Directional Light
  - SceneController
  - dynamically generat
  - KoreanCharacter08
    - sacrum
      - l_thigh
      - r_thigh
      - pelvis

Bake paused in play mode

Work in Progress

- Preparation of working drafts for NWIP submission
- Specification: Co-editing with Roy Walmsley
  - The first Monday of each month at 09:30 UTC (10:30 BST 18:30 KST)
  - Via Web3D teleconference
- Developing C++ and C# binding examples
- Developing X3DBindingViewer programs with C++ and C# binding capability