

**Introduction to
X3D - eXtensible 3D**

Content

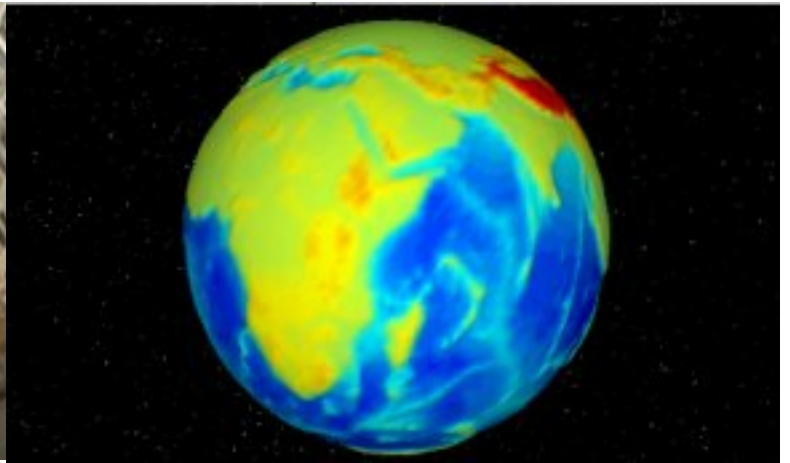
1. Introduction to X3D - eXtensible 3D	2
1.1. X3D - extensible 3D	3
1.1.1. Related technologies	4
1.1.2. Creation of X3D	5
1.2. X3D basics	6
1.2.1. Structure	7
1.2.2. Units	10
1.2.3. Simple 3D objects	11
1.2.4. Surface of objects	14
1.2.5. Transformation	18
1.3. X3D advanced geometries	21
1.3.1. Coordinate-based geometries	22
1.3.2. Further 3D objects	26
1.4. X3D scenes	29
1.4.1. Lighting	30
1.4.2. Background	32
1.4.3. Navigation	34
1.4.4. Viewpoints	35
1.5. Advanced X3D concepts	37
1.5.1. Level of Detail (LOD)	38
1.5.2. The Geospatial component of X3D	40
1.6. Bibliography	42

1. Introduction to X3D - eXtensible 3D

This lesson is a practical step-by-step introduction to X3D - extensible 3D.



(theMightyAtom 2007)



(TurriceN 2007)

Learning Objectives

- You are able to explain the definitions of existing X3D files and to view and navigate these files.
- You are able to create X3D virtual environments representing simple real world objects using geometric primitives, coordinate-based geometries, elevation grids, transformations and appearance definitions.
- You are able to improve X3D files by defining scene settings such as simple lighting, background, navigation and viewpoints.
- You are able to explain the levels of detail (LOD) and geo concepts of X3D.
- You are to apply the LOD concept to simple objects in a X3D virtual environment.

1.1. X3D - extensible 3D

What is X3D?

X3D - short for extensible 3D Graphics - is an open xml-based standard for (Web-enabled) interactive 3D content. The X3D standard is developed by the [Web3D Consortium](#) .

X3D supports...

- 3D graphics - Polygonal geometry, parametric geometry, hierarchical transformations, lighting, materials and multipass/ multi-stage texture mapping
- 2D graphics - Text, 2D vector and planar shapes displayed within the 3D transformation hierarchy
- Animation - Timers and interpolators to drive continuous animations; humanoid animation and morphing
- Spatialized audio and video - Audiovisual sources mapped onto geometry in the scene
- User interaction - Mouse-based picking and dragging; keyboard input
- Navigation - Cameras; user movement within the 3D scene; collision, proximity and visibility detection
- Networking - Ability to compose a single X3D scene out of assets located on a network; hyperlinking of objects to other scenes or assets located on the World Wide Web
- User-defined objects - Ability to extend built-in browser functionality by creating user-defined data types (PROTOS)
- Scripting - Ability to dynamically change the scene via programming and scripting languages
- Physical simulation - Humanoid animation; geospatial datasets; integration with Distributed Interactive Simulation (DIS) protocols

1.1.1. Related technologies

X3D is the successor of VRML, the virtual reality modelling language ([VRML in Wikipedia](#)). There exist related or similar technologies used for displaying 3D virtual environments. Some of them are listed below.

- Collada - <http://www.collada.org>
- U3D (Universal 3D) - <http://www.ecma-international.org>
- 3DML - <http://www.flatland.com/>

1.1.2. Creation of X3D

X3D file creation

X3D is XML-based (XML = Extensible Markup Language, find more Information on <http://www.w3.org/XML/>) and thus a X3D file is a text file structured by the XML elements that are defined in the X3D XMLSchema.

To create an X3D file you can use any text editor. But it is more comfortable to use an XML editor (see the list below) or at least a text editor with syntax highlighting. Using a specific XML editor allows attaching the X3D XMLSchema and checking the X3D file for well-formedness and validity against the XMLSchema.

Download the [X3D XMLSchemas](#) to create your own X3D files. Unzip them at a convenient location and reference the file `x3d-3.0.xsd` in the X3D files you are going to create. The file suffix of X3D files is `.x3d`.

Some XML editors

- Cooktop [free] - <http://www.xmlcooktop.com/>
- Oxygen [free trial, ~50\$ for the academic version] - <http://www.oxygenxml.com/>
- XMLSpy [free trial, rather expensive] - <http://www.altova.com/>

X3D editors

Another possibility to create X3D scenes is to use a special X3D editor (e.g. [Flux Studio](#) or [X3D Edit](#)). Some of them allow creating scenes within a WYSIWYG environment. There are also tools available that convert 3D formats (e.g. Sketchup or 3D Studio Max) in X3D or some 3D modelling packages may offer X3D export functionality.

X3D file viewing

To view the X3D text file in a 3D window you need a special viewer. Most viewer function as plugins for browsers. The following link allows you to check what plugins already are installed for your browser.

Click the following link to check plugin installations - <http://cic.nist.gov/vrml/vbdetect.html>

If you do not already have a X3D viewer installed on your computer you can install one of the following free viewers to view the X3D files we are going to create during this lesson.

- Flux viewer [recommended] - <http://www.mediamachines.com>
- Octaga viewer - <http://www.octaga.com/>

After installing an X3D viewer you should be able to see the simple X3D virtual environment showing a red box below.

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Click [redbox.x3d](#) to view and navigate the above X3D scene in a separate window.

Use the help (right click into the 3D scene, select 'Help/User Guide' - or <http://www.mediamachines.com/userguide.php>) to get accustomed to the navigation possibilities in the Flux Player. Use the respective user manual if you are using a X3D viewer different from the Flux player.

1.2. X3D basics

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

Hi,
my name is Wörml. This name comes from the pronunciation of VRML, the preceding 3D standard. It is my plan to guide you through this introduction to X3D and to teach you some of the basic concepts of this **v**irtual **r**eality **m**odeling **l**anguage X3D (that's where VRML was abbreviated from



).

Hope you have fun and enjoy learning X3D!

Remark: You might find the first couple of pages not too interesting. But read through them they lay the base and it is soon going to be more interesting.

The links marked with the book icon



link you to the official



[ISO/IEC 19775:2004 Extensible 3D \(X3D\) specification](#). There you find more information and details about the nodes discussed in each section and also about the nodes and concepts not discussed in this introductory lesson to X3D.

1.2.1. Structure

X3D header

As already mentioned in the previous unit, X3D is an XML-based language for the description of Virtual Reality environments. The X3D XML file contains the root node <X3D>. Its child nodes are <head> (optional) and <Scene>. The latter contains all the definitions and descriptions of the scene content as we will see below and in the further units of this lesson.

```
<X3D xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\spec\x3d-3.0.xsd" profile="Full" version="3.0">
  <head>
  |
  | <meta content="Susanne Bleisch" name="Author"/>
  | <meta content="10. September 2007" name="CreationDate"/>
  |
  </head>
  <Scene>
  |
  | ...
  |
  </Scene>
</X3D>
```








Example X3D file header



[X3D specification - AbstractX3DStructure explanations](#)

Root node attributes

The root node has two attributes that allow specifying the profile and version of X3D used for this file.

profile	Core	 X3D specification - Core profile
	Interchange	 X3D specification - Interchange profile
	Interactive	 X3D specification - Interactive profile
	MPEG4Interactive	 X3D specification - Interactive profile
	Immersive	 X3D specification - MPEG-4 interactive profile
	Full	 X3D specification - Immersive profile  X3D specification - Full profile

The different profiles define subsets of the X3D elements to be used for specific applications. The full profile includes all X3D elements.

version	3.0	The current version of X3D is 3.0.
---------	-----	------------------------------------

Meta information

The element <head> may contain meta information about the file. Several (as many as needed) <meta> elements with the attributes 'name' (for the description of the information) and 'content' (for the actual information) can be defined to describe the file or the virtual environment created.

Scene graph

The scene graph or the element <Scene> in the X3D file contains the definition of the virtual environment with all its geometries, characteristics, animations, background and so on. You will learn about possible content in the following units. For the moment it is sufficient to know that hierarchical structure of the nodes in a X3D file is called the scene graph. Within a scene graph every node has one or several ancestors (nodes that are higher up in the hierarchy, e.g. parent node) and possible one or several descendants (nodes that are further down in the scene graph hierarchy, e.g. child nodes).



[X3D specification - Scene graph](#)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D SYSTEM "../schema/x3d-3.0.dtd">
3 <X3D>
4   <!-- General information about the file and its content -->
5   <head>
6     <meta name="filename" content="exampleBoxCone.x3d"/>
7     <meta name="author" content="Abt. VGI, Fachhochschule beider Basel"/>
8     <meta name="created" content="29 June 2005"/>
9     <meta name="description" content="Simple X3D example"/>
10  </head>
11  <Scene>
12    <!-- Cone -->
13    <Transform translation="2 0 0">
14      <Shape>
15        <Cone bottomRadius="1" height="3"/>
16        <Appearance>
17          <Material emissiveColor="0 0.3 0"/>
18        </Appearance>
19      </Shape>
20    </Transform>
21    <!-- Box -->
22    <Transform translation="-1 0 0">
23      <Shape>
24        <Box size="2 2 2"/>
25        <Appearance>
26          <Material diffuseColor="0.3 0.4 0.8"/>
27        </Appearance>
28      </Shape>
29    </Transform>
30  </Scene>
31 </X3D>
```

Example of a simple X3D file

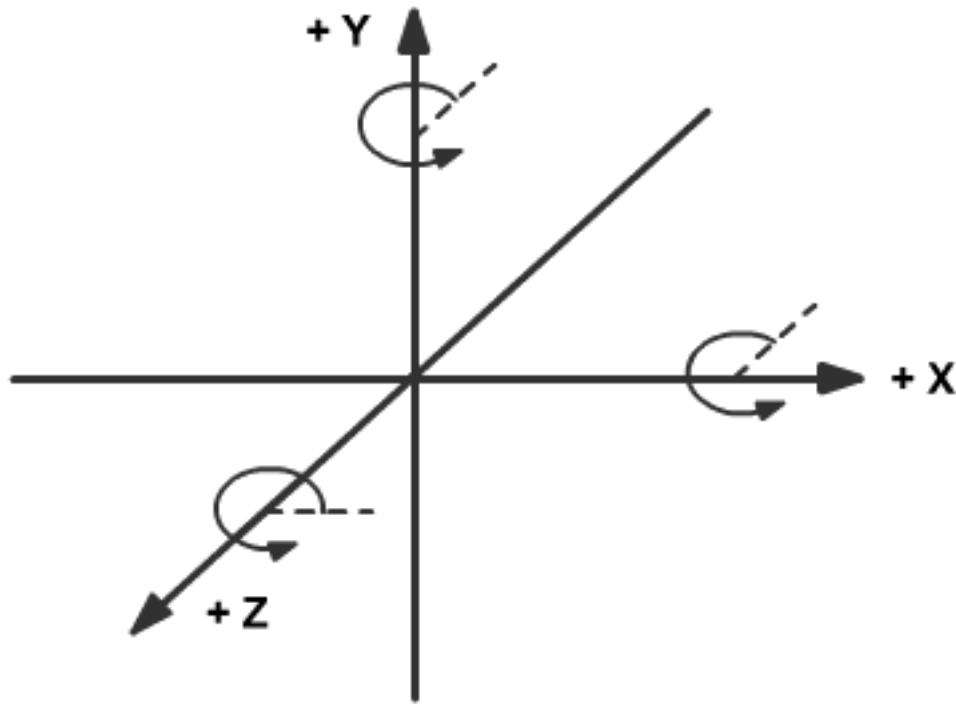
Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

1.2.2. Units

Coordinate system

X3D uses a right-handed Cartesian coordinate system (see figure below). The system consists of three spatial axes: x, y and z. Definitions of coordinates or sizes are made in this order. Expressed as dimensions, x means width, y means height and z means depth.

When creating a X3D scene the coordinate origin is 0, 0, 0 (x, y, z) and lies in the centre of the scene. Every element is placed at the coordinate origin if it is not translated.



X3D coordinate system

Note that the positive x-axis points to the front of the scene.

Units

The basic unit of measure used in X3D scenes is meters. The following list shows the other standard units employed in X3D scenes.

Category	Unit
Linear distance	Metres
Angles	Radians
Time	Seconds
Colour space	RGB ([0.,1.], [0.,1.], [0., 1.])

Standard units (ISO/IEC 19775 2004)

The units of a X3D scene are defined implicitly. Thus, if you want to use millimetres instead of metres you can do so. Just look at the entered values as millimetres instead of as metres. You do not need to define anything.

1.2.3. Simple 3D objects

Shape node

The basic building blocks of a virtual world are objects or solids which have a shape or form and consist of a specific material. Quite similarly to the real world. The characteristics of the material of the object, such as colour or texture are defined as surface characteristics.

The X3D Shape nodes combines the geometry of an object with its material characteristics. It has two subnodes: a geometry node (see below) and the Appearance node.

```
<Shape>  
  <Box size="2 2 2"/>  
  <Appearance>  
    <Material diffuseColor="1 0 0"/>  
  </Appearance>  
</Shape>
```

Example of the Shape node

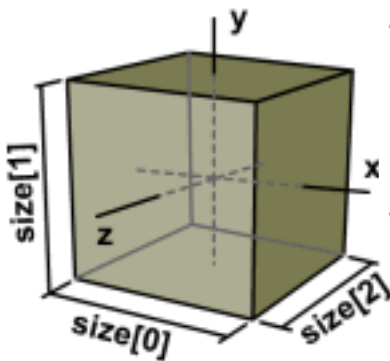


[X3D specification - Shape node](#)

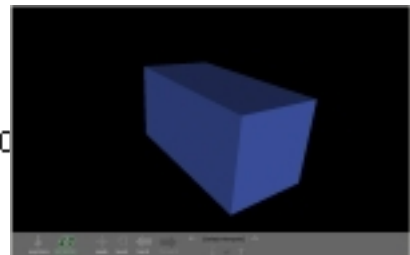
Geometry nodes

The term geometry node stands for a selection of nodes describing the geometry of simple boxes to complex spatial objects. X3D offers a number of predefined geometries, so-called geometric primitives such as box, cone, cylinder or sphere but also objects with irregular surfaces and elevation models. The following sections describe the geometric primitives.

Box node

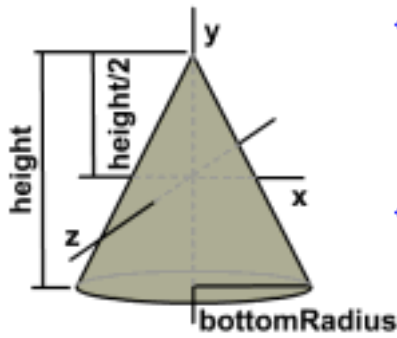


```
<Shape>  
  <Box size="1 1 2"/>  
  <Appearance>  
    <Material diffuseColor="0.3 0.4 0"/>  
  </Appearance>  
</Shape>
```



[X3D specification - Box node](#)

Cone node

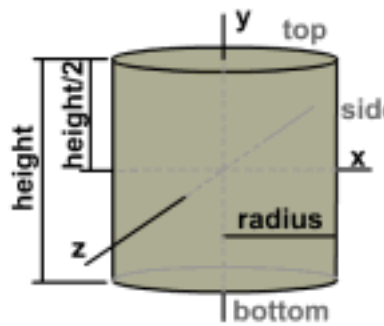


```
<Shape>  
  <Cone bottomRadius="1" height="3"  
  <Appearance>  
    <Material emissiveColor="0 0.3 0"  
  </Appearance>  
</Shape>
```

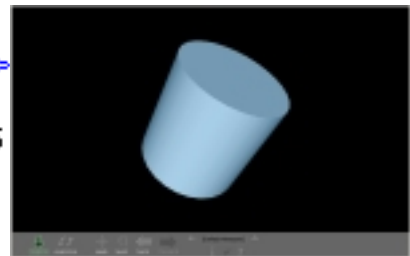


X3D specification - Cone node

Cylinder node

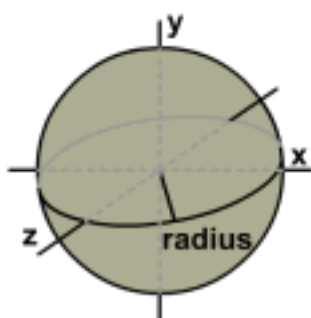


```
<Shape>  
  <Cylinder height="3" radius="1.5"/>  
  <Appearance>  
    <Material emissiveColor="0 0.15"  
  </Appearance>  
</Shape>
```



X3D specification - Cylinder node

Sphere node



```
<Shape>  
  <Sphere radius="2.75"/>  
  <Appearance>  
    <Material emissiveColor="0.8 0.1"  
  </Appearance>  
</Shape>
```



X3D specification - Sphere node

Wörmls hat

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Hi,

below you learn how my hat is constructed in X3D. The colour is missing yet. We will learn about material definitions, including colours, in the next section.

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Full" version="3.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\_eLML_25\X3D\X3D.dtd">
  <Scene>
    <Shape>
      <Cone bottomRadius="0.7" height="2"/>
    </Shape>
  </Scene>
</X3D>
```

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Task: Wörmls head

Create an X3D file representing Wörmls head (without any colour).

Hint: Wörmls head has the size 2 (diameter).

[Solution](#)

1.2.4. Surface of objects

Appearance node

The Appearance node is a child-node of Shape. The subnodes of Appearance are Material and a number of different texture nodes. For the construction of the hat and head of Wörml we have not used the Appearance node. Thus, the objects are displayed in the default appearance with colour white. But normally the Appearance node is used to define the display of the objects, their colour or texture.

```
<Appearance>
... <Material diffuseColor="0.8 0 0.8" specularColor="0 0 0" transparency="0.3"></Material>
</Appearance>
```

Example of the Appearance node



[X3D specification - Appearance node](#)

Material node

The Material node is used to specify the surface material properties or mainly the colour of an object. The fields in the Material node determine how light reflects of an object to create colour. There are six fields which can contain single values between 0 and 1 or arrays of values between 0 and 1.

ambientIntensity	The ambientIntensity field specifies how much ambient light from light sources this surface shall reflect.	<code>ambientIntensity="0.2"</code>
diffuseColor	The diffuseColor field reflects all X3D light sources depending on the angle of the surface with respect to the light source. The more directly the surface faces the light, the more diffuse light reflects.	<code>diffuseColor="0.3 0.4 0.8"</code>
emissiveColor	The emissiveColor field models "glowing" objects.	<code>emissiveColor="0.4 0.8 0.3"</code>
specularColor and shininess	The specularColor and shininess fields determine the specular highlights (e.g., the shiny spots on an apple). Lower shininess values produce soft glows, while higher values result in sharper, smaller highlights.	<code>shininess="0.2" specularColor="0 0 0"</code>

transparency	The transparency field specifies how "clear" an object is, with 1.0 being completely transparent, and 0.0 completely opaque.	<code>transparency="0.5"</code>
--------------	--	---------------------------------

Most simply you define a diffuseColor of an object. X3D uses scaled RGB colours. Meaning that you define the three components of RGB but in value ranges from 0 to 1. The value 114 is then 0.447 or 1/255*114. For example red is defined as diffuseColor="1 0 0" or yellow as diffuseColor="1 1 0".



[X3D specification - Material node](#)

Texture nodes

Beside defining colours, X3D allows using different textures to model the appearance of objects. You have the choice between using static (image) and dynamic (movie) textures.

<p>ImageTexture</p> <p>ImageTexture node</p>	The ImageTexture node defines a texture map by specifying an image file and general parameters for mapping to geometry. The texture is read from the URL specified by the url field.	<code><ImageTexture url="texture.jpg"/></code>
<p>MovieTexture</p> <p>MovieTexture node</p>	The MovieTexture node defines a time dependent texture map (contained in a movie file) and parameters for controlling the movie and the texture mapping. A MovieTexture node can also be used as the source of sound data for a Sound node. In this special case, the MovieTexture node is not used for rendering.	<code><MovieTexture url="movie.avi"/></code>
<p>MultiTexture</p> <p>MultiTexture node</p>	MultiTexture enables the application of several individual textures to a 3D object to achieve a more complex visual effect.	<pre> <MultiTexture mode="ADD"> <ImageTexture url="texture1.jpg"/> <ImageTexture url="texture2.jpg"/> </MultiTexture> </pre>

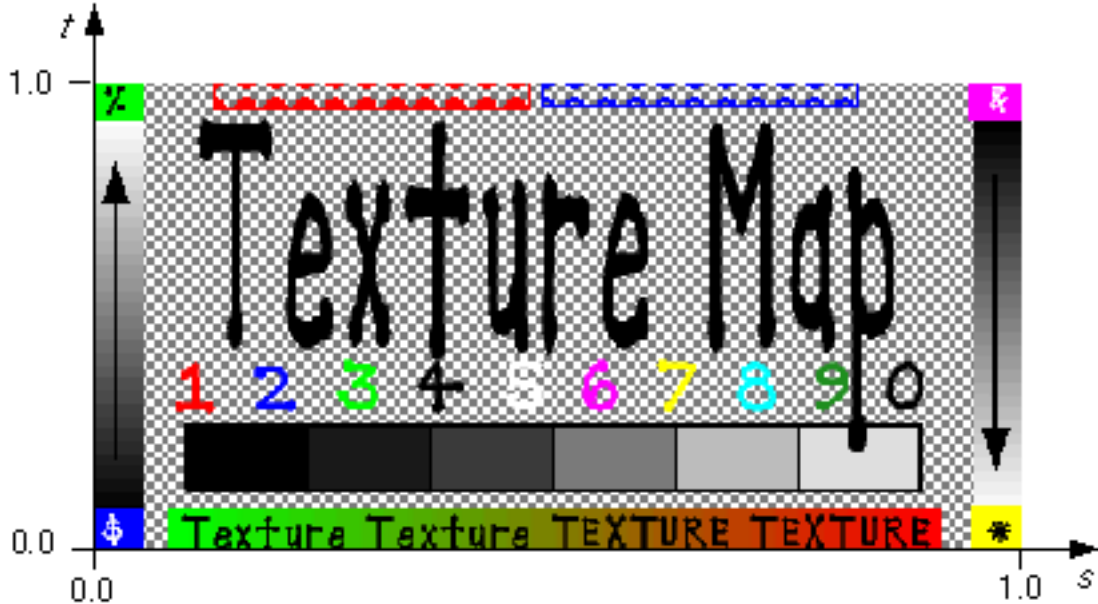
Additionally, you can use PixelTexture (



[PixelTexture node](#)) or extend this set of texture nodes by creating new nodes derived from the abstract X3DTextureNode base class.

Texture coordinates

Texture maps are defined in a 2D coordinate system (s, t) that ranges from [0.0, 1.0] in both directions. The bottom edge of the image corresponds to the S-axis of the texture map, and left edge of the image corresponds to the T-axis of the texture map. The lower-left pixel of the image corresponds to $s=0, t=0$, and the top-right pixel of the image corresponds to $s=1, t=1$.



Texture map coordinate system

Wörmls coloured hat

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

Hi,
below you learn how my hat gets its colour. For that I use the already existing X3D-file with the geometry definition of my hat. Then I just need to add the material node with the colour definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Full" version="3.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="..\spect
  <Scene>
    <Shape>
      <Cone bottomRadius="0.7" height="2"/>
      <Appearance>
        <Material diffuseColor="0.38 0.12 0.40"/>
      </Appearance>
    </Shape>
  </Scene>
</X3D>
```

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Task: Wörmls coloured head

Create an X3D file applying a texture to Wörmls head.

Hint: The texture file is [test.jpg](#)

[Solution](#)

1.2.5. Transformation

Transform node

The Transform node enables positioning of objects at different locations in the X3D scene. It is possible to move (translation), rotate (rotation) or scale (scale) objects. Basically, the Transform node is a grouping node that defines a coordinate system for its children that is relative to the coordinate systems of its ancestors. The most used fields of the Transform node are described in the table below.

Transformation parameters

translation	Distances along the three axes (x, y, z).	<code>translation="1 2 1"</code>
rotation	Axis of rotation and angle of rotation.	<code>rotation="0 1 1 0.5"</code>
scale	Scale factor along one or several of the three axes (x, y, z).	<code>scale="0.5 0.5 0.25"</code>

Transform node example

The following example shows the same (except the colour) box twice. The blue original box is at position [0 0 0] in the coordinate system. The red box is the same box as the blue one but translated, rotated and scaled.

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Full" version="3.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="..lspec"
  <Scene>
    <Shape>
      <Box size="1 1 2"/>
      <Appearance>
        <Material diffuseColor="0.3 0.4 0.8"/>
      </Appearance>
    </Shape>
    <Transform translation="1 2 1" rotation="0 1 1 0.5"
scale="0.5 0.5 0.25">
      <Shape>
        <Box size="1 1 2"/>
        <Appearance>
          <Material diffuseColor="1 0 0"/>
        </Appearance>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)



X3D specification - Transform node

Wörmls head and hat

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

Hi,
below you learn how to place my hat on top of my head. For this, we use the Transform node. All the elements we have constructed so far (hat and head) are positioned at the centre of the 3D world (coordinates [0 0 0]). We need to translate the hat so that is placed on top of my head afterwards.

Have a good look at this example. The Transform node is often used and important.

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Full" version="3.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema
xsi:noNamespaceSchemaLocation="..lspect
  <Scene>
    <!-- Wörmls head -->
    <Shape>
      <Sphere radius="1"/>
      <Appearance>
        <ImageTexture url="test.jpg"/>
      </Appearance>
    </Shape>
    <!-- Wörmls hat -->
    <Transform translation="0.0 1.8 0.0">
      <Shape>
        <Cone bottomRadius="0.7" height="2"/>
        <Appearance>
          <Material diffuseColor="0.38 0.12 0.40"/>
        </Appearance>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

We have combined the code for the head and for the hat of Wörml in one X3D file. The code for the hat is included in a Transform node. If the Transform node would include several nodes then all of them would be affected by the transformation. For placing the hat on Wörmls head we have defined a translation with the values 0 1.8 0. This moves the hat from the coordinate origin at 0 0 0 to +1.8 along the y-axis (towards the top of the scene).

Task: Wörmls eyes

Add two black eyes to Wörmls head (with hat).

Hints: Create one eye first, then copy and translate it. The eye colour is: diffuseColor .24 .24 .24 / specularColor .3 .25 .3 / ambientIntensity .06 / shininess .05

Solution

Hint: The solution uses DEF to name the geometric object of the first eye 'Eye'. Thus, the second eye can re-use the geometric object and needs to translate it to the correct position only. This principle will be explained again in the following units.

1.3. X3D advanced geometries

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

In the previous unit you have learned about the basic structure of X3D files, simple 3D objects and their appearance. In this unit, we will go a step further and look at some additional geometry types needed for rebuilding more complex real world objects in the virtual environment. Typical examples are objects built from faces or terrain displays.

Have fun!

1.3.1. Coordinate-based geometries

Coordinate node

The base for modelling user-defined objects (e.g. point clouds, wire frame models or objects built from polygons) are spatial coordinates. The Coordinate node allows to define spatial coordinates in the field point. It cannot be used standalone but serves as coordinate repository of the geometry nodes introduced below.

```
<Coordinate point="-1.0 -0.1 -1.0,
1.0 -0.1 -1.0,
1.0 -0.1 1.0,
-1.0 -0.1 1.0,
-1.0 0.1 -1.0,
1.0 0.1 -1.0,
1.0 0.1 1.0,
-1.0 0.1 1.0,
"/>
```

Example of the Coordinate node



Each point for which the spatial coordinates are stored in the Coordinate node is indexed. Thus, the first point (3 coordinate values) has index 0, the second has index 1, and so on.




[X3D specification - Coordinate node](#)

Geometries

The following geometries are all based on the definition and linking of points or rather spatial coordinate triples.

<p>PointSet</p>  <p>PointSet node</p>	<p>The PointSet node specifies a set of 3D points, in the local coordinate system, with associated colours at each point. The Coordinate node contains the coordinate definitions.</p>	<pre><PointSet> <Coordinate point="1 2 3, 2 3 1"/> </PointSet></pre>
<p>IndexedLineSet</p>  <p>IndexedLineSet node</p>	<p>The IndexedLineSet node represents a 3D geometry formed by constructing polylines from 3D vertices specified in the Coordinate node. IndexedLineSet uses the indices in its coordIndex field to specify the polylines by connecting vertices from the Coordinate node. An index of "-1" indicates that the current polyline has ended and the next one begins.</p>	<pre><IndexedLineSet coordIndex="0 1 -1, 1 2 -1"> <Coordinate point="1 2 3, 2 3 1, 1 1 2"/> </IndexedLineSet></pre>

<p>IndexedFaceSet</p>  <p>IndexedFaceSet node</p>	<p>The IndexedFaceSet node represents a 3D shape formed by constructing faces (polygons) from vertices listed in the Coordinate node. IndexedFaceSet uses the indices in its coordIndex field to specify the polygonal faces by indexing into the coordinates in the Coordinate node. An index of "#1" indicates that the current face has ended and the next one begins.</p>	<pre><IndexedFaceSet coordIndex="0 1 2 -1"> <Coordinate point="1 2 3, 2 3 1, 1 1 2"/> </IndexedFaceSet></pre>
--	---	---

Wörmls body

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

Hi,

don't be surprised. I've extended my surroundings a little.

Below you learn how to construct my body. This looks quite difficult, but it isn't.

```

<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Full" version="3.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..Ispec/x3d-3.0.xsd">
  <Scene>
    <Transform translation="0 1 0">
      <!-- Wörmis head -->
      <!-- Wörmis hat -->
      <!-- Wörmis eyes -->
    </Transform>

    <!-- Wörmis body -->
    <!-- Red box -->
    <Shape>
      <Appearance>
        <Material diffuseColor="1.0 0.0 0.0"/>
        <!-- Red colour -->
      </Appearance>
      <IndexedFaceSet DEF="Box" coordIndex="0 4 5 1 0 -1
        1, 5, 6, 2, 1, -1,
        2, 6, 7, 3, 2, -1,
        3, 7, 4, 0, 3, -1,
        4, 7, 6, 5, 4, -1,
        0, 1, 2, 3, 0,"/>
      <Coordinate point="-1.0 -0.1 -1.0,
        1.0 -0.1 -1.0,
        1.0 -0.1 1.0,
        -1.0 -0.1 1.0,
        -1.0 0.1 -1.0,
        1.0 0.1 -1.0,
        1.0 0.1 1.0,
        -1.0 0.1 1.0,"/>
      </IndexedFaceSet>
    </Shape>
    <!-- Orange box -->
    <Transform translation="0.0 -0.3 0.0">
      <Shape>
        <Appearance>
          <Material diffuseColor="1.0 0.5 0.0"/>
        </Appearance>
        <IndexedFaceSet USE="Box"/>
      </Shape>
    </Transform>
  </Scene>
</X3D>

```

Firstly, we combine Wörmis head, eyes and hut within a Transform node and translate all these objects 1.0 towards the top of the scene. Then we can build the body of Wörmis below the head.

We define a Material node with diffuseColor red (1 0 0). We use DEF to name the box "Box". Naming a geometric object allows reusing this object later on.

The Coordinate node contains all the indexed spatial coordinates in the attribute point. In the attribute coordIndex of the IndexedFaceSet node all the faces of the box are defined. Each face consists of four spatial coordinates and the first coordinate is listed at the end again. We end a face definition with -1.

For the orange box we reuse and translate the red box. We can use the definition of the red box by calling it - USE="Box"- and assign the colour orange (Material node, diffuseColour 1 0.5 0). We then move this orange box 0.3 towards the bottom of the scene using a Transform node.

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

Task: Wörmls body

Add the other coloured boxes to Wörmls body.

Hints: The colours of Wörmls body are as following:

yellow: diffuseColor 1 1 0

light green: diffuseColor 0.5 1 0

green: diffuseColor 0 1 0

light cyan: diffuseColor 0 1 0.5

cyan: diffuseColor 0 1 1

dark cyan: diffusecolor 0 0.5 1

blue: diffuseColor 0 0 1

Solution

1.3.2. Further 3D objects

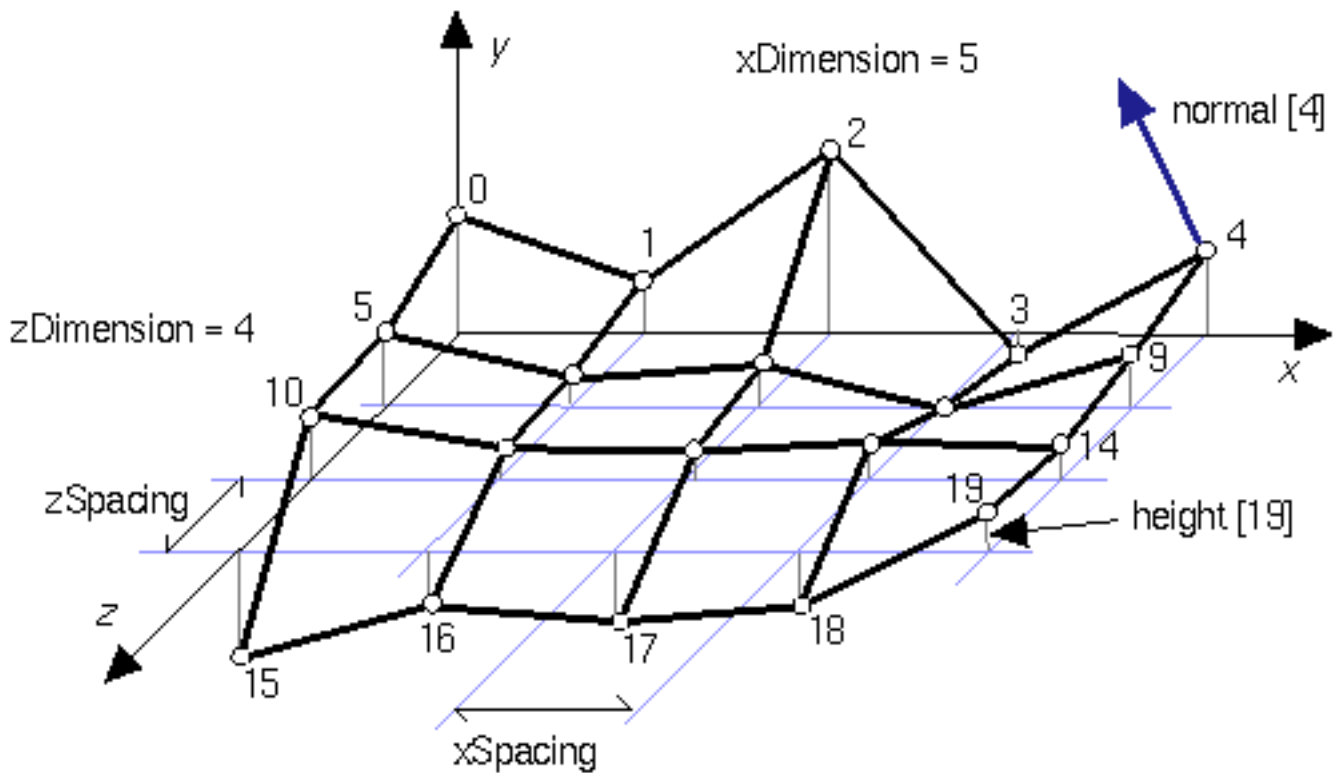
ElevationGrid node

In the previous section we have discussed the IndexedFaceSet node. With this structure we can build any object we want to. We could also build terrain surfaces. However, for this X3D offers a special geometry node, the ElevationGrid node.

The ElevationGrid node specifies a uniform rectangular grid of varying height in the Y=0 plane of the local coordinate system. The geometry is described by a scalar array of height values that specify the height of a surface above each point of the grid.

The xDimension and zDimension attributes indicate the number of elements of the grid height array in the X and Z directions. The vertex locations for the rectangles are defined by the height attribute and the xSpacing and zSpacing attributes.

- The **height** attribute is an xDimension by zDimension array of scalar values representing the height above the grid for each vertex.
- The **xSpacing** and **zSpacing** attributes indicate the distance between vertices in the X and Z directions respectively.



Structure of the ElevationGrid geometry

```
<ElevationGrid xDimension="3" zDimension="4" xSpacing="2" zSpacing="2"
height="1, 0, 0,
0, 0, 2,
0, 1, 1.5,
0, 0, 0,"></ElevationGrid>
```

Code example of the ElevationGrid node



[X3D specification - ElevationGrid node](#)

Texture on ElevationGrid nodes

The texture for a ElevationGrid node (e.g. contained within an ImageTexture node) is laid onto the surface as described in the materials section. The size of the original texture is kept and the origin of the texture coordinate system coincides with the origin of the elevation grid. The s-texture coordinate axis lies along the positive x-axis and t-texture coordinate axis along the positive z-axis of the elevation grid. This results in a mirrored texture (along the x-axis of the elevation grid). To avoid this the texture needs to be mirrored previously.

Extrusion node

The Extrusion node specifies geometric shapes based on a two dimensional cross-section extruded along a three dimensional spine in the local coordinate system. The cross-section can be scaled and rotated at each spine point to produce a wide variety of shapes.

An Extrusion node is defined by:

- a 2D crossSection piecewise linear curve (described as a series of connected vertices);
- a 3D spine piecewise linear curve (also described as a series of connected vertices);
- a list of 2D scale parameters;
- a list of 3D orientation parameters.

```
<Extrusion DEF="Hull" creaseAngle="3.14"
crossSection="0 1, 0.38 0.92, 0.71 0.71, 0.92 0.38, 1 0, 0.92 -0.38, 0.71 -0.71, 0.38 -0.92, 0 -1"
orientation="1 0 0 0, 1 0 0 0, 1 0 0 0, 0 1 0 0, 0 0 1 -0.5"
scale="3 4, 4 4, 4 4, 4 4, 5 .1" solid="false"
spine="-20 0 0, -17 0 0, 0 0 0, 10 0 0, 22 1 0"/>
```

Example code of the Extrusion node

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)



X3D specification - Extrusion node

Wörmls terrain

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Hi,

we'll construct the terrain below my feet now. The ElevationGrid node is a great thing. With a few lines of code we can build quite complex landscapes.

```
<!-- Wörmls terrain -->
<Transform translation="-22.0 -3.5 -22.0">
  <Shape>
    <Appearance>
      <Material diffuseColor="0.0 1.0 0.0"/>
    </Appearance>
    <ElevationGrid xDimension="11"
      xSpacing="4.0" zDimension="11"
      zSpacing="4.0"
      height="0.5, 0.5, 0.6, 0.6, 0.6, 0.6, 0.6,
      0.6, 0.6, 0.6, 0.6,
      0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.7,
      0.7, 0.7, 0.7, 0.9, 0.7, 0.9, 0.8, 0.8, 0.3, 0.7,
      0.7, 0.7, 0.8, 0.6, 0.8, 0.6, 0.8, 0.8, 0.8, 0.8, 0.8,
      0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.9, 0.8, 0.8,
      0.8, 0.6, 0.8, 0.9, 0.8, 0.6, 0.8, 0.8, 0.5, 0.8, 0.8,
      0.6, 0.7, 0.8, 0.6, 0.8, 0.7, 0.8, 0.8, 0.8, 0.8, 0.8,
      0.8, 0.8, 0.6, 0.8, 0.8, 0.8, 0.7, 0.8, 0.8, 0.8, 0.6,
      0.8, 0.7, 0.7, 0.7, 0.7, 0.7, 0.6, 0.7, 0.7, 0.7, 0.7,
      0.3, 0.7, 0.5, 0.7, 0.7, 0.6, 0.5, 0.6, 0.6, 0.5, 0.6,
      0.6, 0.5, 0.6, 0.6, 0.6, 0.6, 0.5, 0.5, 0.6, 0.6, 0.6, "/>
  </Shape>
</Transform>
```

We add the ElevationGrid node to the existing X3D file. To place the terrain beneath Wörml we surround the node by a Transform node. The terrain is moved -22 to the left, -3.5 towards the bottom of the scene and -22 to the back.

Then we define the green colour of the terrain. The ElevationGrid node defines the number of grid vertices in x and z direction and also the space between the vertices.

The attribute height defines for each grid vertices a height above 0.

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Remark: To be able to see the terrain you need to change the viewpoint a little.

Task: Wörmls terrain

Play a little with the ElevationGrid node. You can, for example, build hills or depressions into the underground of Wörml.

1.4. X3D scenes

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

Up to now, we have mainly discussed geometries and their appearances. This unit concentrates on the non-geometrical aspects of a 3D scene. Same as in the real world, we can't see a thing in the virtual environment if there is no lighting defined. So far, we were able to look at the objects we defined because 'wearing' a virtual headlight is a standard setting in X3D scenes. In this unit, you will learn how to define different light sources. Additionally, we can design a background different from black, influence the navigation and add different viewpoints to a virtual environment.




Enjoy!

1.4.1. Lighting

The surface properties of a 3D object are dependent on the material definition (Material node) and the lighting of the scene. To have minimal lighting in the 3D scene the X3D viewer adds a default light source to the scene - the headlight. The headlight is comparable to a far away light source that emits its light in parallel rays. This light source moves together with the viewer or rather the objects are always lit from the user's viewpoint. The headlight can be turned off if other light sources are defined within a 3D scene.

Within a 3D scene we can place as many light sources as needed. Using different light sources in a 3D environment allows simulating different real-world situations and conditions, such as lamp light, twilight, fog or sun rise.

Types of light sources

<p>DirectionalLight</p>  <p>DirectionalLight node</p>	<p>The DirectionalLight node defines a directional light source that illuminates along rays parallel to a given 3-dimensional vector. The direction attribute specifies the direction vector of the illumination emanating from the light source in the local coordinate system. Example: sunlight</p>	<pre><DirectionalLight ambientIntensity="0" color="1 1 1" direction="0 -1 0" intensity="1"/></pre>
<p>PointLight</p>  <p>PointLight node</p>	<p>The PointLight node specifies a point light source at a 3D location in the local coordinate system. A point light source emits light equally in all directions; that is, it is omnidirectional. Example: light bulb</p>	<pre><PointLight ambientIntensity="0" color="1 1 1" intensity="1" location="4 4 4" radius="20"/></pre>
<p>SpotLight</p>  <p>SpotLight node</p>	<p>The SpotLight node defines a light source that emits light from a specific point along a specific direction vector and constrained within a solid angle. Spotlights may illuminate geometry nodes that respond to light sources and intersect the solid angle defined by the SpotLight. Example: torch light</p>	<pre><SpotLight beamWidth="1.5708" direction="0 -1 0" ambientIntensity="0" color="1 1 1" intensity="1" location="4 4 4" radius="20"/></pre>

A light source in the virtual world does not cast shadows. The calculation of light reflections on the surfaces of objects is to expensive to do it in real-time. If you want to have shadows in your virtual world you'll need to program them by yourself.

Lighting for Wörmel

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

Hi,

when my surroundings will be lighted with sunlight (as described below) we might shut off the headlight. You will learn how to do this in the next unit.

<!-- Lights -->

```
<DirectionalLight direction="-0.3 -0.3 -0.3" ambientIntensity="0.2" color="1 1 1"/>
```

We use a DirectionalLight node for the lighting of the 3D scene. This is about the same lighting as the sunlight in the real world.

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

Task: Lighting

Play a little with different light sources. You might want to place a spotlight to light up the hat of Wörmel.

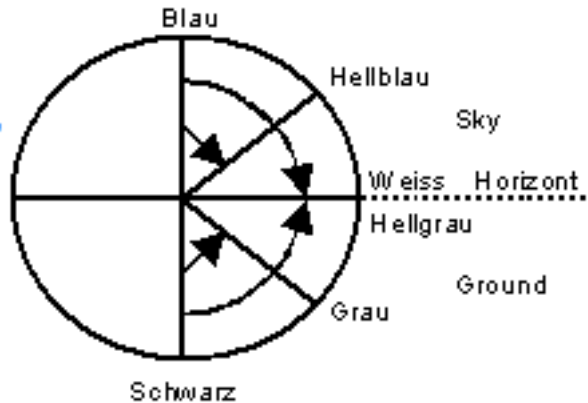
1.4.2. Background

Background node

Defining a background in a virtual environment can make the scene look more realistic. With the Background node we can either define the background as series of continuous colours or as a textured background cube. For the first, you need to define different colours and the angles where they are used. For the latter, you need texture files for each side of the cube.

A simple white or coloured background can be defined using the attribute skyColor set to the selected colour (e.g. 1 1 1 for white).

```
<Background  
skyColor="0 0.1 0.8, 0 0.5 1, 1 1 1"  
skyAngle="0.785, 1.571"  
groundColor="0 0 0, 0.3 0.3 0.3, 0.5 0.5 0.5"  
groundAngle="0.785, 1.571"/>
```



In this example are the colours for the sky (above the horizon) blue - light blue - white and black - grey - light grey for the ground (below the horizon).



[X3D specification - Background node](#)

Background for Wörmel

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Hi,
- not much to explain to this example. Nice that I don't need to stand in a dark cave any longer.

```
<!-- Background -->  
<Background skyColor="0.0 0.1 0.8,  
0.0 0.5 1.0,  
1.0 1.0 1.0"
```

We define the background for the sky area above the horizon only. The colours used are blue, light blue and white. The blue colour comes a bit closer to the horizon. This is defined in the attribute skyAngle.

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Task: Background

You may want to play a little with different background settings for the Würml scene.

1.4.3. Navigation

NavigationInfo node

The NavigationInfo node contains information describing the physical characteristics of the viewer's avatar and viewing model. The parameters that are defined in this node are used by the X3D viewer to configure the environment. In the NavigationInfo node it is possible to specify the type of movement that is allowed (e.g. FLY, WALK or EXAMINE) or the speed of navigation. The attribute headlight allows switching on and of the headlight. As soon as there are light sources defined in the virtual environment the headlight might be switched off, set to false. In most viewers it is also possible to switch on or off the headlight via the menu.

```
<NavigationInfo headlight="false" speed="1" type="&quot;EXAMINE&quot;"/>
```

Example of the NavigationInfo node



[X3D specification - NavigationInfo node](#)

Wörml: Different scene parameters

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Hi,

in a previous unit we have defined some sunlight for my environment. Thus, we can now switch off the headlight. Additionally, we set navigation mode to "EXAMINE".

Hint: You can change the navigation mode via the menu (e.g. right click into the 3D scene) in most X3D viewers too.

```
<!-- Navigation and scene information -->  
<NavigationInfo headlight="false"  
type="&quot;EXAMINE&quot;"/>
```

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Task: Navigation

You may want to play a little with different navigation and scene settings for the Wörml scene.

1.4.4. Viewpoints

Viewpoint node

The Viewpoint node defines a specific location in the local coordinate system from which the user may view the scene. The different viewpoints within a scene can be named and most often accessed in the viewer. This way, it is possible to define sort of a sightseeing tour through the virtual world or at least serve the user with different pre-defined views of the scene.

The position and orientation attributes of the Viewpoint node specify relative locations in the local coordinate system. Position is relative to the coordinate system's origin (0,0,0), while orientation specifies a rotation relative to the default orientation. In the default position and orientation, the viewer is on the z-axis looking down the #z-axis toward the origin with +x to the right and +y straight up.

```
<Viewpoint description="FromTop"  
position="0 10 0" fieldOfView="0.78"  
orientation="1 0 0 -1.57"/>
```

Example of the Viewpoint node



[X3D specification - Viewpoint node](#)

There are free tools available for the easy definition of viewpoints (calculation of the orientation parameters). Most of them were programmed for VRML but the values can also be used for X3D.

Example: [VRML Viewer](#) ([Download](#))

View Wörmel

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Hi,

below you'll learn how you can define one or several viewpoints to have a better look at me.

```
<!-- Viewpoints -->
<Viewpoint description="DefaultView"
position="5 6 25"
orientation="-0.5089 0.8594 0.05039 0.2294"
fieldOfView="0.8"/>
<Viewpoint description="TopView"
position="0 30 0"
orientation="1 0 0 -1.5708"
fieldOfView="0.8"/>
```

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Using the Flux X3D viewer you can right click into the 3D scene and change between the different viewpoints in the menu 'Views'.

Task: Viewpoints

Add one or two more viewpoints to the Wörml scene.

1.5. Advanced X3D concepts

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [\[link\]](#)

Bye-bye!

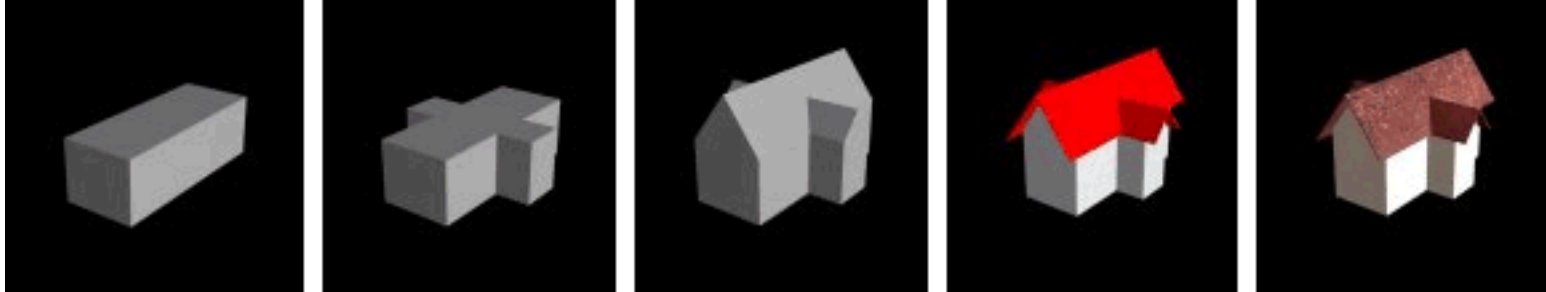
Hope you've enjoyed me leading you through the previous units. From now on you have to find your way by yourself. But you are prepared. Wish you good luck for your X3D future!

This unit introduces some further concepts of X3D, the levels of detail (LOD) and 'real world' geo aspects. The LOD concept is introduced briefly here. There exists a whole lesson about levels of details. If you are interested ask your tutor if you can get access to that lesson.

1.5.1. Level of Detail (LOD)

LOD node

The LOD node specifies various levels of detail or complexity for a given object, and provides hints allowing browsers to automatically choose the appropriate version of the object based on the distance from the user.



Example of different level of detail (LOD) for a building model

The subnodes of the LOD node are a number of nodes that represent the same object or objects at varying levels of detail, ordered from highest level of detail to the lowest level of detail. The range attribute specifies the ideal distances at which to switch between the levels.

The center attribute is a translation offset in the local coordinate system that specifies the centre of the LOD node for distance calculations. The number of subnodes in the LOD node needs to exceed the number of values in the range field by one.

```
LOD center="0 0 0" range="10, 20">  
  Shape>  
    <Box size="1 1 2"/>  
  <Appearance><Material diffuseColor="0.3 0.4 0.8"/></Appearance>  
/Shape>  
  Shape>  
    <Box size="1.5 1.5 3"/>  
  <Appearance><Material diffuseColor="0.3 0.4 0.8"/></Appearance>  
/Shape>  
  Shape>  
    <Box size="2 2 4"/>  
  <Appearance><Material diffuseColor="0.3 0.4 0.8"/></Appearance>  
/Shape>  
/LOD>
```

In this example, the box is replaced twice (at a distance of 10 and 20 from the coordinate origin) by a slightly bigger box.

LOD node code example

Hint: The LOD node can also be used to hide objects completely when the viewer is a specific distance away.



[X3D specification - LOD node](#)

LOD example 'Church'

Only pictures can be viewed in the PDF version! For Flash etc. see online version. Only screenshots of animations will be displayed. [link]

The example of the church on the left hand side uses the LOD principle to show three different detailed models of the church dependent on the distance of the viewer from the church. The most detailed model is shown at a distance < 20 , then the church model without texture is displayed and after a distance of 30, the generalised church model without roof shapes is shown.

Navigate backwards away from the church to see the three different models of the church.

Download the [code of this example](#) (right mouse click, save file as...).

Task: LOD for Wörmls body

Use the LOD node to implement three different models of Wörmls body.

Hint: You might want to model Wörmls body as IndexedLineSet, then as simple Box (perhaps with a colour gradient) and as detailed body with differently coloured boxes.

[Possible solution](#)

1.5.2. The Geospatial component of X3D

So far, we have always constructed virtual environments with local coordinate systems. The Geospatial component of X3D describes how to associate real world locations to elements in the X3D world as well as specifying nodes particularly tuned for geospatial applications. With the Geospatial component it is, for example, possible to embed geospatial coordinates in certain X3D nodes, to support high-precision geospatial modeling, and to handle large multi-resolution terrain databases.

Nine nodes comprise the Geospatial component:

- GeoCoordinate
-  X3D specification - GeoCoordinate node
- GeoElevationGrid
-  X3D specification - GeoElevationGrid node
- GeoLocation
-  X3D specification - GeoLocation node
- GeoLOD
-  X3D specification - GeoLOD node
- GeoMetadata
-  X3D specification - GeoMetadata node
- GeoOrigin
-  X3D specification - GeoOrigin node
- GeoPositionInterpolator
-  X3D specification - GeoPositionInterpolator node
- GeoTouchSensor
-  X3D specification - GeoTouchSensor node
- GeoViewpoint
-  X3D specification - GeoViewpoint node

Some of these node names should sound familiar as we already know the corresponding 'normal' X3D nodes.

Notes on 'Geo' X3D

Some interesting aspects of the Geospatial component of X3D are:

- **Coordinate systems :**
It is possible to specify a geodetic or projective spatial reference frame and thus to deviate from the implicit Cartesian, right-handed three-dimensional coordinate system defined by X3D (see also [X3D coordinate system](#) in this lesson). Scenes defined in different coordinate systems (e.g. different UTM zones) can geometrically correct be combined within a X3D viewer (if the viewer supports it).

However, so far only geodetic (GD: latitude, longitude, elevation), geocentric (GC: X, Y, Z) or Universal Transverse Mercator (UTM: northing, easting, elevation) are supported as spatial reference frames.

- **High-precision coordinates :**

Most computer graphics systems, including X3D, use single-precision floating point values to model and render all geometry. However, single-precision is insufficient to model data over the range of the earth at accurate ground resolutions. For visualizing the earth (georeferenced data) it is not possible to achieve resolutions better than around 0.8 metres using single-precision floating point numbers. This geo-referencing problem is one avoided by establishing a geo-referenced local coordinate system (LCS). An absolute geo-referenced location is defined as the origin of the LCS (the GeoOrigin node). This becomes the reference point that correlates to the X3D world's (0,0,0) origin.

- **Navigation :**

Browsing large geographic areas needs different navigation concepts. The elevation scaled velocity makes the velocity at which users can navigate around a world dependent upon their height above the terrain. This behavior is addressed by the GeoViewpoint node.

- **Streaming :**

The GeoLOD node provides a terrain specialized form of the LOD node. It is a grouping node that specifies two different levels of detail for an object using a tree structure, where 0 to 4 children can be specified, and also efficiently manages the loading and unloading of these levels of detail. Thus, not the whole scene needs to be loaded from the beginning.

- **Metadaten :**

The GeoMetadata node supports the specification of metadata describing any number of geospatial nodes. This is similar to a WorldInfo node, but specifically for describing geospatial information. There are a number of standards and representations for geospatial metadata. Rather than adopt any particular standard, the purpose of the GeoMetadata node is to provide links to any of these complete metadata descriptions, with the option to also supply a short, human-readable summary.

1.6. Bibliography

- **ISO/IEC 19775** (2004). *Information technology — Computer graphics and image processing — Extensible 3D (X3D)* [online]. Available from: <http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/> [Accessed 22 August 2007].
Download: [../spec/ISO-IEC-19775-X3DAbstractSpecification/X3D.html](http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/X3D.html)
- **theMightyAtom** (2007). *Scene Details - Reflections* [online]. Available from: <http://www.mediamachines.com/show.php?modelName=Reflections&username=theMightyAtom&uid=233&version=1> [Accessed 20 September 2007].
- **TurricaN** (2007). *Scene Details - Earth* [online]. Available from: <http://www.mediamachines.com/show.php?modelName=Earth&username=TurricaN&uid=472&version=2> [Accessed 20 September 2007].