

X3D Graphics for Web Authors

Chapter 10

Geometry Nodes part III: Geometry2D

Theorem. For a triangle with a right-angle between sides a and b , with hypotenuse c , then $a^2 + b^2 = c^2$

Pythagoras

Contents

Chapter Overview and Concepts

X3D Nodes and Examples

Additional Resources

Chapter Summary and Suggested Exercises

References

Chapter Overview

Many different geometry nodes

An excellent aspect of X3D is that there are many different ways to create geometry

- Chapter 2, Geometry Primitives
- Chapter 6, Points, Lines and Polygons
- Chapter 10, Geometry2D Nodes
- Chapter 13, Triangles and Quadrilaterals

These are all handled consistently inside a Shape node with corresponding Appearance

Overview: Geometry2D Nodes

These are simple utility (convenience) nodes

Geometry2D nodes

- Arc2D lines
- ArcClose2D polygonal shape
- Circle2D lines
- Disk2D polygonal shape
- Polyline2D lines
- Polypoint2D points
- Rectangle2D polygonal shape
- TriangleSet2D polygonal shapes

Concepts

Geometry2D: convenience nodes

Nodes provide simple forms for basic arcs, line segments, points, lines and rectangles

Each of these geometric shapes might be generated a different way

- using IndexedFaceSet, triangles or another 3D node

Thus the Geometry2D constructs might be considered “convenience” or “utility” nodes, making it easier to build useful 2D shapes

- Not intended for full-fledged interface design
- Can make interesting widgets nevertheless

Precaution: flat 2D nodes are easily missed

Default: 2D nodes defined in vertical X-Y plane

Because rendered 3D polygons have no width,
each of the 2D nodes becomes invisible if
viewed from the side

- Same behavior as other flat 3D geometry
- Moving out of the plane makes them visible again

Take care to keep flat 2D nodes visible to user

- Dedicated Viewpoint, or parent Billboard node
- Useful in screen-aligned Heads-up Display (HUD)

Points and lines remain visible from any angle

Geometry2D nodes are new to X3D

Geometry2D nodes are not automatically provided as part of the Immersive profile

- Initially defined as part of X3D specification

Prerequisite: X3D authors must list Geometry2D component in scene header in order to ensure proper browser support

Not directly supported in VRML97 specification

- Though replacement VRML prototypes are provided by X3dToVrml97.xslt stylesheet conversion
- So backwards compatibility is feasible with VRML

Component support must be declared

Must define this component in X3D scene header

XML (.x3d) syntax

```
<X3D version='3.0' profile='Immersive'>
  <head>
    <component name='Geometry2D' level='2' />
    <!-- meta tags go here -->
  </head>
  <Scene/>
</X3D>
```

Classic VRML (.x3dv), VRML97 (.wrl) syntax

```
#X3D V3.0 utf8
PROFILE Immersive
COMPONENT Geometry2D:2
```

Common geometry fields: few

Because the 2D nodes are quite simple, most common geometry fields aren't necessary

- *ccw, convex, colorPerVertex, creaseAngle*, etc.

But single-sided flat polygons more easily "lost" than double-sided polygons

- Set *solid*='false' for 2-sided polygon geometry

Application of of Appearance, Material, texturing, LineProperties, FillProperties remains the same

Most fields are *accessType*='initializeOnly' so run-time animation of fields not possible

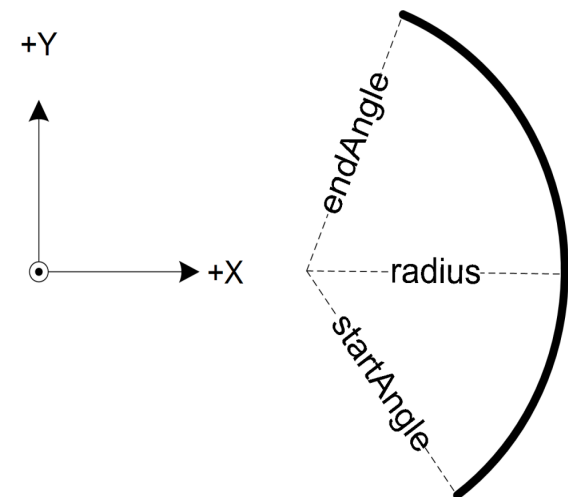
X3D Nodes and Examples

Arc2D node

Arc2D is a line-based node defining a circular arc

- center at local origin (0,0)
- *startAngle*, *endAngle* fields measured in radians, counterclockwise direction positive (right-hand rule)
- Angle measurement starts at positive x-axis, then sweeps toward positive y-axis
- Allowed range $[-2\pi, +2\pi]$

```
<Arc2D  
  startAngle= '1.0'  
  endAngle=' -0.8' />
```





```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
4     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
5   <head>
6     <component level='2' name='Geometry2D'/>
7     <meta content='Arc2D.x3d' name='title'/>
8     <meta content='Example of Arc2D for the X3D Book.' name='description'/>
9     <meta content='Leonard Daly and Don Brutzman' name='creator'/>
10    <meta content='17 April 2006' name='created'/>
11    <meta content='11 March 2008' name='modified'/>
12    <meta content='http://X3dGraphics.com' name='reference'/>
13    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
14    <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights'/>
15    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
16    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Arc2D.x3d' name='identifier'/>
17    <meta content='X3D-Edit, https://savane.nps.edu/X3D-Edit' name='generator'/>
18    <meta content='../license.html' name='license'/>
19  </head>
20  <Scene>
21    <Background skyColor='1 1 1'/>
22    <Viewpoint description='Arc2D example' orientation='0 1 0 0' position='0 0 10'/>
23    <Transform translation='-2.5 1 0'>
24      <Shape>
25        <Arc2D endAngle='0' radius='3' startAngle='4.5'/>
26        <Appearance DEF='MagentaAppearance'>
27          <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 .8 0'/>
28        </Appearance>
29      </Shape>
30    </Transform>
31    <Transform scale='1 .5 0' translation='-1 -1.5 0'>
32      <Shape>
33        <Arc2D endAngle='.78' radius='1.5' startAngle='6'/>
34        <Appearance USE='MagentaAppearance'/>
35      </Shape>
36    </Transform>
37  </Scene>
38 </X3D>

```

Edit Arc2D

containerField
 geometry

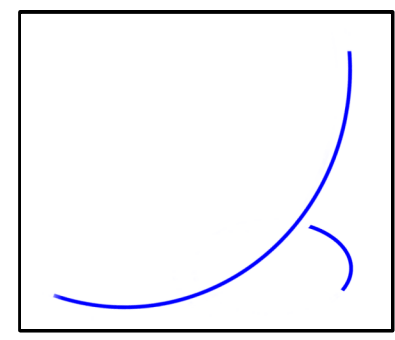
DEF


USE

startAngle 4.5

endAngle 0

radius 3



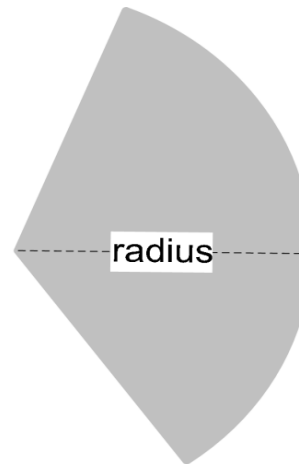
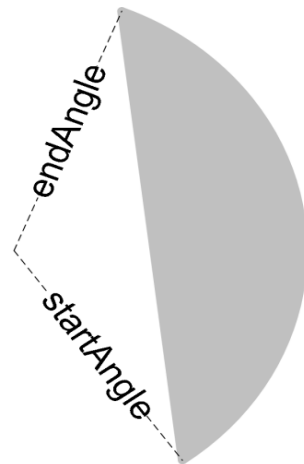
 Arc2D	<p>Arc2D is a geometry node. Arc2D specifies a linear circular arc with center (0,0) with angles measured starting at positive x-axis and sweeping towards positive y-axis.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
radius	<p>[radius: accessType initializeOnly, type SFFloat CDATA "1"]</p> <p>(0..infinity) radius of circle, of which the arc is a portion.</p> <p>Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
startAngle	<p>[startAngle: accessType initializeOnly, type SFFloat CDATA "0"]</p> <p>[0..2pi] Arc extends from startAngle counterclockwise to endAngle, in radians.</p> <p>Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
endAngle	<p>[endAngle: accessType initializeOnly, type SFFloat CDATA "1.570796"]</p> <p>[0..2pi] Arc extends from startAngle counterclockwise to endAngle, in radians.</p> <p>Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
containerField	<p>[containerField: NMTOKEN "geometry"]</p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>

ArcClose2D node

Similar to Arc2D node

ArcClose2D is polygonal, defines enclosed area

- Not just a curved line
- *closureType* field can be CHORD or PIE
- *radius* must be greater than 0
- If *endAngle* – *startAngle* > 2pi, full circle is drawn





```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
4     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
5   <head>
6     <component level='2' name='Geometry2D' />
7     <meta content='ArcClose2D.x3d' name='title' />
8     <meta content='Example of ArcClose2D for the X3D Book.' name='description' />
9     <meta content='Leonard Daly and Don Brutzman' name='creator' />
10    <meta content='17 April 2006' name='created' />
11    <meta content='11 March 2008' name='modified' />
12    <meta content='http://X3dGraphics.com' name='reference' />
13    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
14    <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights' />
15    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject' />
16    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/ArcClose2D.x3d' name='identifier' />
17    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
18    <meta content='../license.html' name='license' />
19  </head>
20  <Scene>
21    <Background skyColor='1 1 1' />
22    <Viewpoint description='ArcClose2D example' orientation='0 1 0 0' position='0 0 10' />
23    <Transform translation='-2.5 1 0'>
24      <Shape>
25        <ArcClose2D closureType='PIE' endAngle='0' radius='3' startAngle='4.5' />
26        <Appearance DEF='MagentaAppearance'>
27          <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 .8 0' />
28        </Appearance>
29      </Shape>
30    </Transform>
31    <Transform scale='1 .5 0' translation='-1 -1.5 0'>
32      <Shape>
33        <ArcClose2D closureType='CHORD' endAngle='.78' radius='1.5' startAngle='6' />
34        <Appearance USE='MagentaAppearance' />
35      </Shape>
36    </Transform>
37  </Scene>
38 </X3D>

```

Edit ArcClose2D

containerField
 geometry

DEF

USE

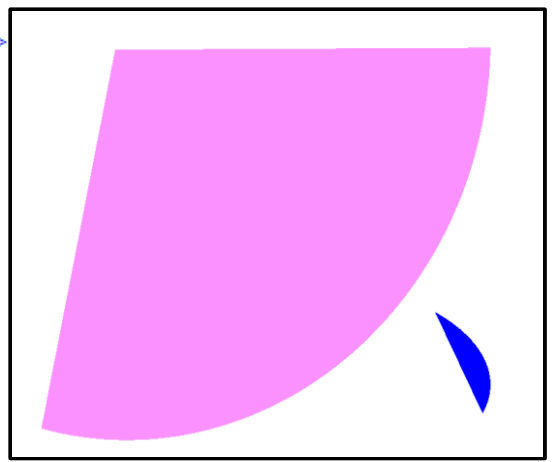
startAngle 4.5


endAngle 0

radius 3

solid

closureType PIE

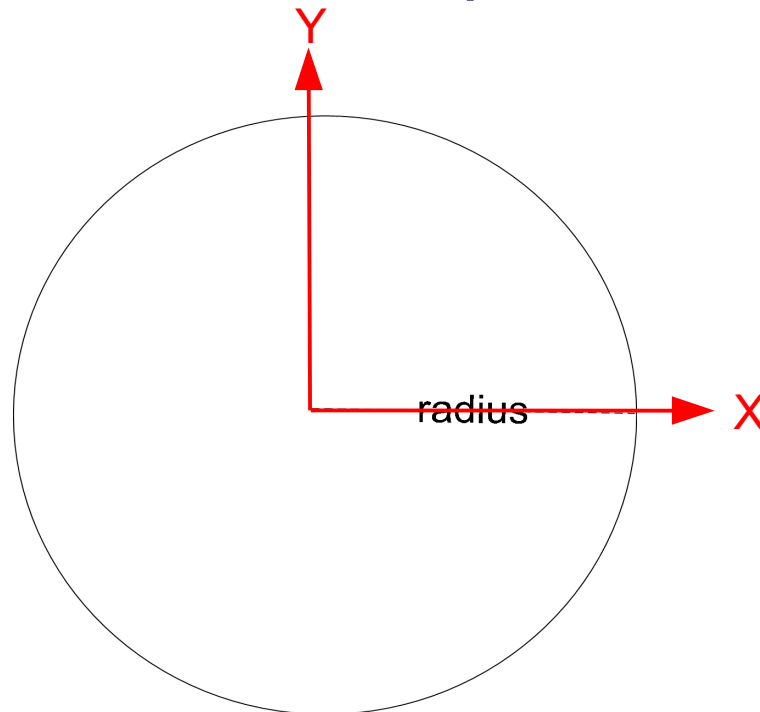


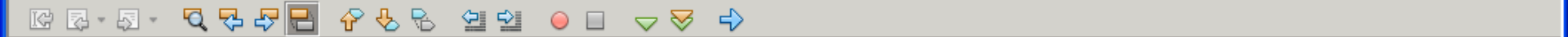
 ArcClose2D	<p>ArcClose2D is a geometry node. Arc2D specifies a linear circular arc with center (0,0) with angles measured starting at positive x-axis and sweeping towards positive y-axis.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
radius	<p>[radius: accessType initializeOnly, type SFFloat CDATA "1"]</p> <p>(0..infinity) radius of circle, of which the arc is a portion.</p> <p>Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
startAngle	<p>[startAngle: accessType initializeOnly, type SFFloat CDATA "0"]</p> <p>[0..2pi] Arc extends from startAngle counterclockwise to endAngle, in radians.</p> <p>Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
endAngle	<p>[endAngle: accessType initializeOnly, type SFFloat CDATA "1.570796"]</p> <p>[0..2pi] Arc extends from startAngle counterclockwise to endAngle, in radians.</p> <p>Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
closureType	<p>[closureType accessType initializeOnly, (PIE CHORD) "PIE"]</p> <p>Whether pair of line segments connect to center (PIE), or single line-segment chord connects arc endpoints (CHORD).</p> <p>Warning: simple-geometry parameters cannot be changed after initial creation.</p>
solid	<p>[solid: accessType initializeOnly, type SFBool (true false) "true"]</p> <p>Setting solid true means draw only one side of polygons (backface culling on), setting solid false means draw both sides of polygons (backface culling off).</p> <p>Warning: default value true can completely hide geometry if viewed from wrong side!</p> <p>Warning: solid false not supported in VRML97.</p>
containerField	<p>[containerField: NMTOKEN "geometry"]</p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>

Circle2D node

Circle2D is line-based, specifying a planar circle with *center* (0,0) and a positive-valued *radius*

Figure 10.5, corrected: example Circle2D node





```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation=
4 <head>
5 <component level='2' name='Geometry2D' />
6 <meta content='Circle2D.x3d' name='title' />
7 <meta content='Example of Circle2D for the X3D Book.' name='description' />
8 <meta content='Leonard Daly and Don Brutzman' name='creator' />
9 <meta content='17 April 2006' name='created' />
10 <meta content='11 March 2008' name='modified' />
11 <meta content='http://X3dGraphics.com' name='reference' />
12 <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
13 <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights' />
14 <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='source' />
15 <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/' name='source' />
16 <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
17 <meta content='../license.html' name='license' />
18 </head>
19 <Scene>
20 <Viewpoint description='Circle2D example' orientation='0 1 0 0' position='0 0 10' />
21 <Transform translation='-2.5 1 0'>
22 <Shape>
23 <Circle2D radius='1' />
24 <Appearance DEF='MagentaAppearance'>
25 <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 .8 0' />
26 </Appearance>
27 </Shape>
28 </Transform>
29 <Transform scale='1 .5 0' translation='-1 -1.5 0'>
30 <Shape>
31 <Circle2D radius='2' />
32 <Appearance USE='MagentaAppearance' />
33 </Shape>
34 </Transform>
35 </Scene>
36 </X3D>

```

Edit Circle2D

containerField

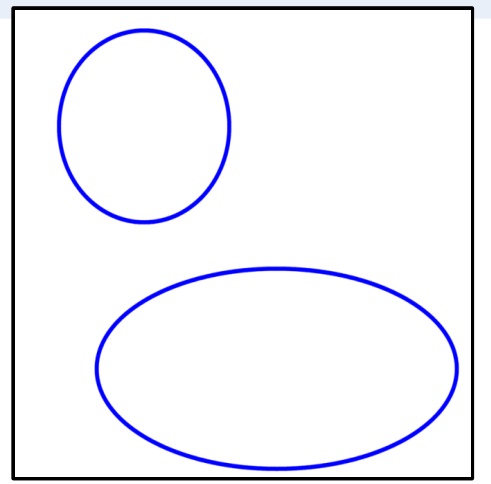
geometry

DEF

USE

radius

OK Cancel Help



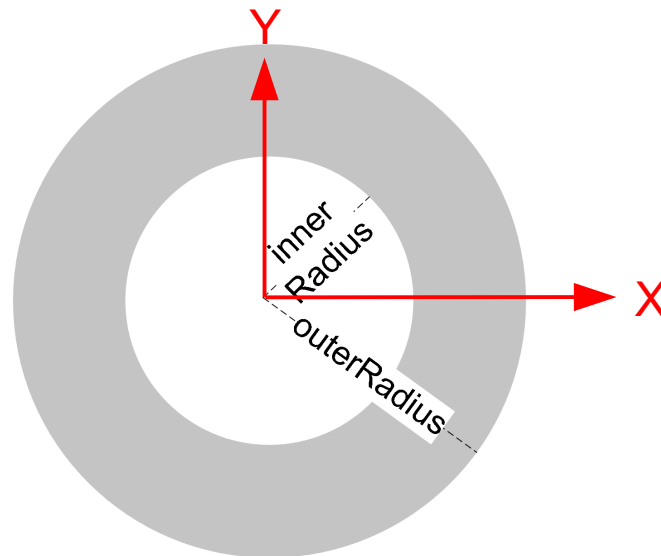
○ Circle2D	<p>Circle2D is a geometry node. Circle2D specifies an X-Y planar circle with center (0,0). Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <i>_all_</i> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
radius	<p>[radius: accessType initializeOnly, type SFFloat CDATA "1"] (0..infinity) radius of circle. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
containerField	<p>[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>

Disk2D node

Disk2D is a polygonal node that defines circular geometry with a circular hole in the center

- Similar to a circular disk washer (for a bolt and nut)
- $0 \leq \text{innerRadius} \leq \text{outerRadius}$

Hint: endcap of Cylinder node can substitute for Disk2D node with *innerRadius*='0'

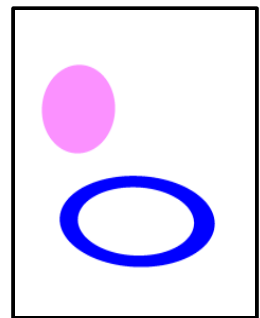




```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www
4 <head>
5   <component level='2' name='Geometry2D' />
6   <meta content='Disk2D.x3d' name='title' />
7   <meta content='Example of Disk2D for the X3D Book.' name='description' />
8   <meta content='Leonard Daly and Don Brutzman' name='creator' />
9   <meta content='17 April 2006' name='created' />
10  <meta content='11 March 2008' name='modified' />
11  <meta content='http://X3dGraphics.com' name='reference' />
12  <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
13  <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights' />
14  <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject' />
15  <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Disk2D.x3d' name='identifier' />
16  <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
17  <meta content='../license.html' name='license' />
18 </head>
19 <Scene>
20   <Viewpoint description='Disk2D example' orientation='0 1 0 0' position='0 0 10' />
21   <Transform translation='-2.5 1 0'>
22     <Shape>
23       <Disk2D outerRadius='1' />
24       <Appearance DEF='MagentaAppearance'>
25         <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 .8 0' />
26       </Appearance>
27     </Shape>
28   </Transform>
29   <Transform scale='1 .5 0' translation='-1 -1.5 0'>
30     <Shape>
31       <Disk2D innerRadius='1.5' outerRadius='2' />
32       <Appearance USE='MagentaAppearance' />
33     </Shape>
34   </Transform>
35 </Scene>
36 </X3D>


```



Edit Disk2D

containerField	DEF <input type="text"/>
<input checked="" type="checkbox"/> geometry	USE <input type="text"/>
innerRadius	<input type="text" value="1.5"/> x Pi / Pi
outerRadius	<input type="text" value="2"/> x Pi / Pi
solid	<input checked="" type="checkbox"/>

OK Cancel Help

 Disk2D	Disk2D is a geometry node. Disk2D specifies a filled planar circle with center (0,0). Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
innerRadius	[innerRadius: accessType initializeOnly, type SFFloat CDATA "0"] (0..infinity) Inner radius of circle, greater than or equal to 0. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
outerRadius	[outerRadius: accessType initializeOnly, type SFFloat CDATA "1"] (0..infinity) Outer radius of circle, greater than 0. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
solid	[solid: accessType initializeOnly, type SFBool (true false) "true"] Setting solid true means draw only one side of polygons (backface culling on), setting solid false means draw both sides of polygons (backface culling off). Warning: default value true can completely hide geometry if viewed from wrong side! Warning: solid false not supported in VRML97.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

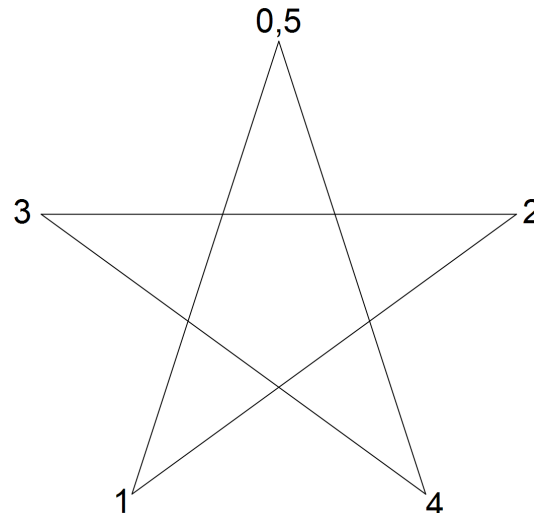
Polyline2D node

Polyline2D defines a connected set of 2D vertices as contiguous, connected set of line segments

Disconnected lines must be constructed using separate Polyline2D nodes

Point values are contained in lineSegments field

- Example: six point values, 0th and 5th are identical

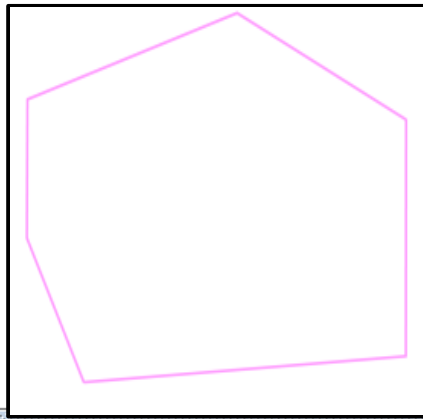




```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://
4 <head>
5   <component level='2' name='Geometry2D' />
6   <meta content='Polyline2D.x3d' name='title' />
7   <meta content='Example of Polyline2D for the X3D Book.' name='description' />
8   <meta content='Leonard Daly and Don Brutzman' name='creator' />
9   <meta content='17 April 2006' name='created' />
10  <meta content='11 March 2008' name='modified' />
11  <meta content='http://X3dGraphics.com' name='reference' />
12  <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
13  <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights' />
14  <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject' />
15  <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Polyline2D.x3d' name='identifier' />
16  <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
17  <meta content='../license.html' name='license' />
18 </head>
19 <Scene>
20   <Background skyColor='1 1 1' />
21   <Viewpoint description='Polyline2D example' orientation='0 1 0 0.159' position='1.21 -0.22 8.04' />
22   <Group>
23     <Shape>
24       <Polyline2D lineSegments='-3 2 -3 0 -2 -2 3 -1.5 3 1.5 .5 3 -3 2' />
25       <Appearance DEF='MagentaAppearance'>
26         <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 .8 0' />
27       </Appearance>
28     </Shape>
29   </Group>
30 </Scene>
31 </X3D>

```



Edit Polyline2D

containerField: geometry


DEF:

USE:

lineSegments	
x	y
-3	2
-3	0
-2	-2
3	-1.5
3	1.5
0.5	3
-3	2

+ -

OK Cancel Help

 Polyline2D	<p>Polyline2D is a geometry node. Polyline2D specifies a connected set of vertices in a contiguous set of line segments in X-Y plane.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
lineSegments	<p>[lineSegments: accessType initializeOnly, type MFVec2f CDATA #FIXED ""]</p> <p>Coordinates of vertices connected into contiguous Polyline2D.</p> <p>Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
containerField	<p>[containerField: NMTOKEN "geometry"]</p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>

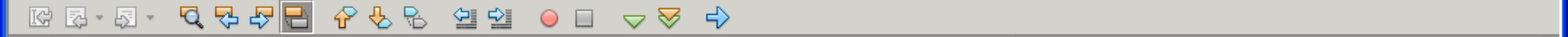
Polypoint2D node

Polypoint2D defines a set of 2D points

- *point* field holds an array of X-Y coordinate pairs

Rendering and limitations match those for PointSet node





```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='h
4 <head>
5   <component level='2' name='Geometry2D' />
6   <meta content='Polypoint2D.x3d' name='title' />
7   <meta content='Example of Polypoint2D for the X3D Book.' name='description' />
8   <meta content='Leonard Daly and Don Brutzman' name='creator' />
9   <meta content='17 April 2006' name='created' />
10  <meta content='11 March 2008' name='modified' />
11  <meta content='http://X3dGraphics.com' name='reference' />
12  <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
13  <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights' />
14  <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject' />
15  <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Polypoint2D.x3d' name='identifier' />
16  <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
17  <meta content='../license.html' name='license' />
18 </head>
19 <Scene>
20   <Viewpoint description='Polypoint2D example' orientation='0 1 0 0.159' position='1.21 -0.22 8.04' />
21   <Group>
22     <Shape>
23       <Polypoint2D point='-3 2 -3 0 -1 0 -3 -1 -2 -2 0 -2 0 -1 -2 -1 3 -1.5 3 1.5 2 -1 .5 3 -.5 1.5 1.5 1.5' />
24       <Appearance DEF='MagentaAppearance'>
25         <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 .8 0' />
26       </Appearance>
27     </Shape>
28   </Group>
29 </Scene>
30 </X3D>

```

Edit Polypoint2D

containerField: geometry

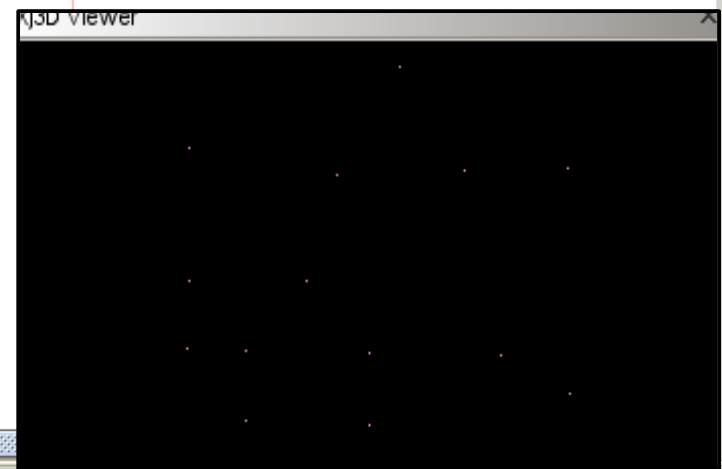
DEF:


USE:

point		
x	y	
-3	2	2
-3	0	0
-1	0	0
-3	-1	-1
-1	-2	-2
-2	-2	-2
0	-2	-2
-2	0	-1
0	-1	-1
-1	-1	-1
-1	3	-1.5
3	1.5	3
1.5	2	1.5
2	-1	.5
3	-.5	1.5
1.5	1.5	1.5
1.5	1.5	1.5

+ -

OK Cancel Help



 Polypoint2D	<p>Polypoint2D is a geometry node. Polypoint2D specifies a set of 2D points. Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
point	<p>[point: accessType initializeOnly, type MFVec2f CDATA #FIXED ""] 2D coordinates of vertices. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
containerField	<p>[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>

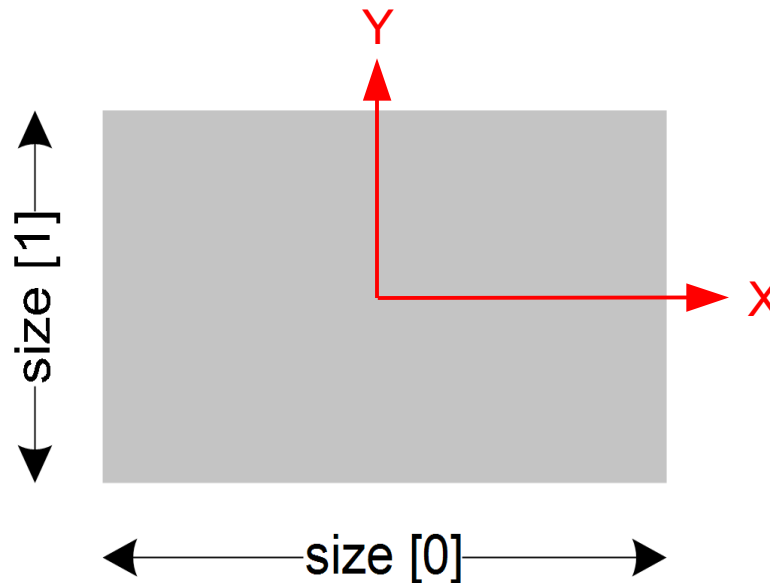
Rectangle2D node

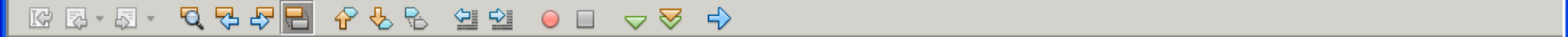
Rectangle2D defines a rectangular quadrangle

- again centered at (0,0) in local coordinate system
- aligned with the local x-y coordinate axes

size is SFVec2f field defining x and y dimensions

- each greater than zero

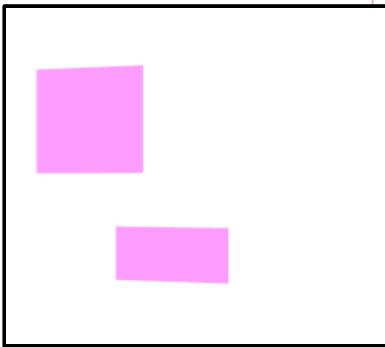




```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='h
4 <head>
5 <component level='2' name='Geometry2D' />
6 <meta content='Rectangle2D.x3d' name='title' />
7 <meta content='Example of Rectangle2D for the X3D Book.' name='description' />
8 <meta content='Leonard Daly and Don Brutzman' name='creator' />
9 <meta content='17 April 2006' name='created' />
10 <meta content='11 March 2008' name='modified' />
11 <meta content='http://X3dGraphics.com' name='reference' />
12 <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
13 <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights' />
14 <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject' />
15 <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Rectangle2D.x3d' name='identifier' />
16 <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
17 <meta content='../license.html' name='license' />
18 </head>
19 <Scene>
20 <Background skyColor='1 1 1' />
21 <Viewpoint description='Rectangle2D example' orientation='0.981 0.178 -0.08 0.01' position='-1.52 -0.17 8.11' />
22 <Transform translation='-2.5 1 0'>
23 <Shape>
24 <Rectangle2D size='2 2' />
25 <Appearance DEF='MagentaAppearance'>
26 <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 .8 0' />
27 </Appearance>
28 </Shape>
29 </Transform>
30 <Transform translation='-1 -1.5 0'>
31 <Shape>
32 <Rectangle2D size='2 1' />
33 <Appearance USE='MagentaAppearance' />
34 </Shape>
35 </Transform>
36 </Scene>
37 </X3D>

```



Edit Rectangle2D

containerField
 geometry

DEF


USE

width x

height y

solid

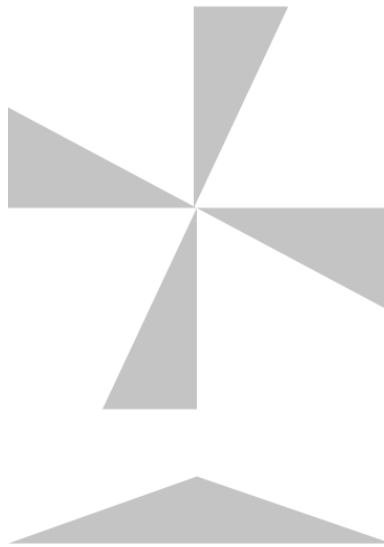
OK Cancel Help

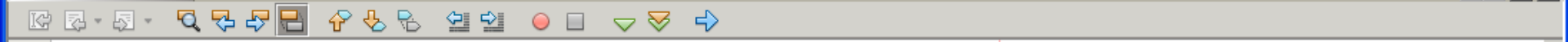
 Rectangle2D	Rectangle2D is a geometry node. Rectangle2D specifies a 2D rectangle. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <i>_all_</i> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
size	[size: accessType initializeOnly, type SFVec2f CDATA "2 2"] 2D dimensions of Rectangle2D. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
solid	[solid: accessType initializeOnly, type SFBool (true false) "true"] Setting solid true means draw only one side of polygons (backface culling on), setting solid false means draw both sides of polygons (backface culling off). Warning: default value true can completely hide geometry if viewed from wrong side! Warning: solid false not supported in VRML97.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

TriangleSet2D node

TriangleSet2D defines a set of triangles, each created by three 2D vertex points

Points are listed individually in the *vertices* field

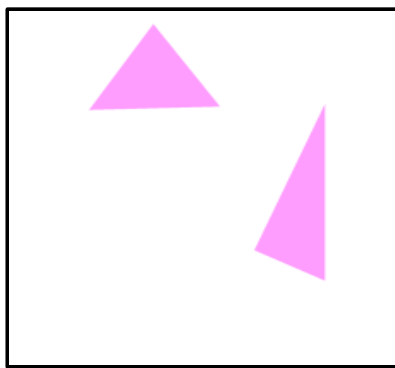




```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.w3.org/2001/XMLSchema-instance'>
4 <head>
5 <component level='2' name='Geometry2D' />
6 <meta content='TriangleSet2D.x3d' name='title' />
7 <meta content='Example of TriangleSet2D for the X3D Book.' name='description' />
8 <meta content='Leonard Daly and Don Brutzman' name='creator' />
9 <meta content='17 April 2006' name='created' />
10 <meta content='11 March 2008' name='modified' />
11 <meta content='http://X3dGraphics.com' name='reference' />
12 <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
13 <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights' />
14 <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject' />
15 <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/TriangleSet2D.x3d' name='identifier' />
16 <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
17 <meta content='../license.html' name='license' />
18 </head>
19 <Scene>
20 <Background skyColor='1 1 1' />
21 <Viewpoint description='TriangleSet2D example' orientation='0 1 0 0.159' position='1.11 0.9 8.06' />
22 <Transform translation='-2.5 1 0'>
23 <Shape>
24 <TriangleSet2D vertices='3 -1.5 3 1.5 2 -1 .5 3 -.5 1.5 1.5 1.5' />
25 <Appearance DEF='MagentaAppearance'>
26 <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 .8 0' />
27 </Appearance>
28 </Shape>
29 </Transform>
30 </Scene>
31 </X3D>

```



Edit TriangleSet2D

containerField: geometry

DEF:

USE:


solid:

vertices:

x0	y0	x1	y1	x2	y2
3	-1.5	3	1.5	2	-1
0.5	3	-0.5	1.5	1.5	1.5

+ -

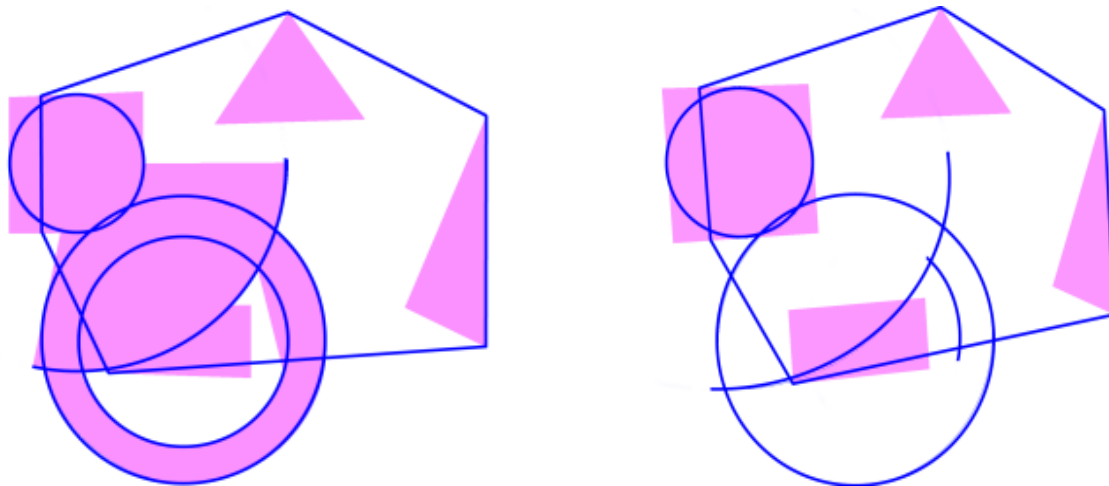
OK Cancel Help

 TriangleSet2D	TriangleSet2D is a geometry node. TriangleSet2D specifies a set of 2D triangles. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
vertices	[vertices: accessType initializeOnly, type MFVec2f CDATA #FIXED ""] 2D coordinates of TriangleSet2D vertices. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
solid	[solid: accessType initializeOnly, type SFBool (true false) "true"] Setting solid true means draw only one side of polygons (backface culling on), setting solid false means draw both sides of polygons (backface culling off). Warning: default value true can completely hide geometry if viewed from wrong side! Warning: solid false not supported in VRML97.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

Example combinations of 3D nodes

Sample scene Summary2D.x3d shows a variety of Geometry2D nodes

- Different values of *solid*='true' and *solid*='false' provides different geometry for front, back views



Additional Resources

X3D prototype implementations

Geometry2dComponentPrototypes.x3d

Geometry2dComponentExamples.x3d

- X3D Geometry2D component nodes, implemented as prototypes, for browsers lacking such support and for backwards compatibility with VRML97
- <http://www.web3d.org/x3d/content/examples/Basic/development/>
- Maps 2D node parameters to 3D node implementations (IndexedFaceSet etc.)

X3dToVrml97.xslt stylesheet automatically inserts external prototype declarations for each of the Geometry2D nodes, as appropriate

Scalable Vector Graphics (SVG) 1

SVG is language for describing two-dimensional graphics and graphical applications in XML

- <http://www.w3.org/Graphics/SVG>
- SVG 1.1 is a W3C Recommendation and forms the core of the current SVG developments
- SVG Tiny 1.2 is the specification currently being developed as the core of the SVG 1.2 language
- SVG Mobile Profiles (Basic and Tiny) are designed for small devices and mobile phones
- SVG Print guidelines help produce final-form documents for archiving and printing

Scalable Vector Graphics (SVG) 2

Many good books are available about SVG
SVG specification and various tutorials are online

- <http://www.w3.org/Graphics/SVG>

Although direct mappings between SVG and X3D are not available, this is an excellent area for further work

- XML stylesheets (.xslt) might convert SVG nodes into X3D nodes, and vice versa
- Perhaps some browsers will implement SVG as an allowed ImageTexture and MovieTexture format

Chapter Summary

Chapter Summary

Common concepts: simple 2D convenience nodes

Geometry2D Nodes

- Arc2D excerpt from a circle's circumference
- ArcClose2D pie or chord subsection of circle shape
- Circle2D line for a complete circle circumference
- Disk2D circular shape with optional center hole
- Polyline2D sequence of line segments
- Polypoint2D one or more points
- Rectangle2D height, width of single polygonal shape
- TriangleSet2D multiple polygonal triangles

Related nodes

Substitutions are available for each Geometry2D node, using the corresponding 3D nodes

Arc2D	IndexedLineSet	LineSet
ArcClosed2D	IndexedFaceSet	Triangle nodes
Circle2D	IndexedLineSet	LineSet
Disk2D	IndexedFaceSet	Triangle nodes
Polyline2D	IndexedLineSet	LineSet
Polypoint2D	PointSet	
Rectangle2D	QuadSet	IndexedFaceSet
TriangleSet2D	IndexedFaceSet	Triangle nodes

Interpolators are not provided for Geometry2D nodes since most fields cannot be animated

- *accessType*='initializeOnly'

Suggested exercises

Build Olympic rings that face the user



- http://commons.wikimedia.org/wiki/Image:Olympic_Rings.svg
- Using Polyline2D
- Using Arc2D or Circle2D

Build 5- or 6-sided star similar to Figure 10.9

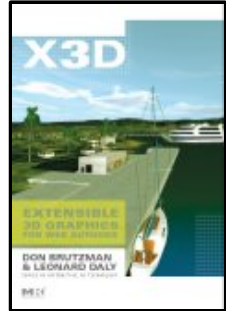
- Using Polyline2D
- Using TriangleSet2D

Build a simple clickable user interface to control animation in a scene: pause, play, stop

References

References 1

X3D: Extensible 3D Graphics for Web Authors
by Don Brutzman and Leonard Daly, Morgan
Kaufmann Publishers, April 2007, 468 pages.



- Chapter 10, Geometry2D Nodes
- <http://x3dGraphics.com>
- <http://x3dgraphics.com/examples/X3dForWebAuthors>

X3D Resources

- <http://www.web3d.org/x3d/content/examples/X3dResources.html>

References 2

X3D-Edit Authoring Tool

- <https://savage.nps.edu/X3D-Edit>

X3D Scene Authoring Hints

- <http://x3dgraphics.com/examples/X3dSceneAuthoringHints.html>

X3D Graphics Specification

- <http://www.web3d.org/x3d/specifications>
- Geometry2D component
- Also available as help pages within X3D-Edit

Contact

Don Brutzman

brutzman@nps.edu

<http://faculty.nps.edu/brutzman>

Code USW/Br, Naval Postgraduate School

Monterey California 93943-5000 USA

1.831.656.2149 voice

CGEMS, SIGGRAPH, Eurographics

The Computer Graphics Educational Materials Source(CGEMS) site is designed for educators

- to provide a source of refereed high-quality content
- as a service to the Computer Graphics community
- freely available, directly prepared for classroom use
- <http://cgems.inesc.pt>

X3D for Web Authors recognized by CGEMS! 😊


- Book materials: X3D-Edit tool, examples, slidesets
- Received jury award for Best Submission 2008

CGEMS supported by SIGGRAPH, Eurographics





Creative Commons open-source license

<http://creativecommons.org/licenses/by-nc-sa/3.0>






Attribution-Noncommercial-Share Alike 3.0 Unported

You are free:

-  **to Share** — to copy, distribute and transmit the work
-  **to Remix** — to adapt the work

Under the following conditions:

-  **Attribution.** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
-  **Noncommercial.** You may not use this work for commercial purposes.
-  **Share Alike.** If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

- ♦ For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.
- ♦ Any of the above conditions can be waived if you get permission from the copyright holder.
- ♦ Nothing in this license impairs or restricts the author's moral rights.

Disclaimer

Your fair dealing and other rights are in no way affected by the above.

Open-source license for X3D-Edit software and X3D example scenes

<http://www.web3d.org/x3d/content/examples/license.html>

Copyright (c) 1995-2013 held by the author(s). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the Naval Postgraduate School (NPS) Modeling Virtual Environments and Simulation (MOVES) Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

X3D Graphics for Web Authors

Chapter 10

Geometry Nodes part III: Geometry2D

Theorem. For a triangle with a right-angle between sides a and b , with hypotenuse c , then $a^2 + b^2 = c^2$

Pythagoras



1

The Pythagorean Theorem is widely known.

Reference: Wikipedia http://en.wikipedia.org/wiki/Pythagorean_theorem

Contents

Chapter Overview and Concepts

X3D Nodes and Examples

Additional Resources

Chapter Summary and Suggested Exercises

References



Chapter Overview



Many different geometry nodes

An excellent aspect of X3D is that there are many different ways to create geometry

- Chapter 2, Geometry Primitives
- Chapter 6, Points, Lines and Polygons
- Chapter 10, Geometry2D Nodes
- Chapter 13, Triangles and Quadrilaterals

These are all handled consistently inside a Shape node with corresponding Appearance



The 2D geometry nodes are *tessellated* (turned into triangles) by X3D browsers.

The 2D nodes are provided for X3D authoring convenience. They can also be identically represented using IndexedFaceSet or any of the Triangle nodes, if the vertex definitions and connectivity ordering are properly re-expressed. In each conversion, the z coordinates will be set to 0.

Also of interest is the X3D Non-Uniform Rational B-Spline (NURBS) component.

Overview: Geometry2D Nodes

These are simple utility (convenience) nodes

Geometry2D nodes

- Arc2D lines
- ArcClose2D polygonal shape
- Circle2D lines
- Disk2D polygonal shape
- Polyline2D lines
- Polypoint2D points
- Rectangle2D polygonal shape
- TriangleSet2D polygonal shapes



When you have more than one way for a language to express something, meaning more than one way for the syntax to express the semantics of something, then we sometimes call that extra (maybe simpler) way of saying something as

“syntactic sugar”

http://en.wikipedia.org/wiki/Syntactic_sugar

It is interesting to note that for many years authors using VRML97 (i.e. X3D version 2.0) were able to create 2D shapes and lines just fine (thank you) without these Geometry2D convenience nodes. However the utility value of the nodes was helpful enough that the X3D working group decided to add them to X3D version 3.0. Since there is often pushback against the opposite motivation of keeping a language terse and expressive, i.e. avoiding

“language bloat” (similar to “code bloat”)

then it is interesting that these Geometry2D nodes eventually were accepted. Probably the most compelling argument that decided this language choice was that addition of flat 2D nodes to an author's palette can make it easier to construct 2D user interfaces.

[back to Table of Contents](#)

Concepts



Geometry2D: convenience nodes

Nodes provide simple forms for basic arcs, line segments, points, lines and rectangles

Each of these geometric shapes might be generated a different way

- using IndexedFaceSet, triangles or another 3D node

Thus the Geometry2D constructs might be considered “convenience” or “utility” nodes, making it easier to build useful 2D shapes

- Not intended for full-fledged interface design
- Can make interesting widgets nevertheless



Convenience is good, especially since it can reduce errors.

Precaution: flat 2D nodes are easily missed

Default: 2D nodes defined in vertical X-Y plane

Because rendered 3D polygons have no width,
each of the 2D nodes becomes invisible if
viewed from the side

- Same behavior as other flat 3D geometry
- Moving out of the plane makes them visible again

Take care to keep flat 2D nodes visible to user

- Dedicated Viewpoint, or parent Billboard node
- Useful in screen-aligned Heads-up Display (HUD)

Points and lines remain visible from any angle

web|3D
CONSORTIUM



8

p. 280, Section 2.1, insert new third paragraph:

Because Geometry2D nodes are defined by default in the vertical X-Y plane, this means that no Z values need to be defined. Node definitions are thus somewhat terser than their counterpart 3D nodes.

So these nodes are a shorthand, easier than building IFS or ILS nodes with all the z values equal to zero.

We are not suddenly working in Flatland. These 2D nodes are drawn in regular 3D space.

X3D does not (yet) provide a simple built-in HUD mechanism. An example HUD implementation is considered in Chapter 12, Environment Sensor and Sound Nodes.

Geometry2D nodes are new to X3D

Geometry2D nodes are not automatically provided as part of the Immersive profile

- Initially defined as part of X3D specification

Prerequisite: X3D authors must list Geometry2D component in scene header in order to ensure proper browser support

Not directly supported in VRML97 specification

- Though replacement VRML prototypes are provided by X3dToVrml97.xslt stylesheet conversion
- So backwards compatibility is feasible with VRML

Component support must be declared

Must define this component in X3D scene header

XML (.x3d) syntax

```
<X3D version='3.0' profile='Immersive'>
  <head>
    <component name='Geometry2D' level='2' />
    <!-- meta tags go here -->
  </head>
  <Scene />
</X3D>
```

Classic VRML (.x3dv), VRML97 (.wrl) syntax

```
#X3D V3.0 utf8
PROFILE Immersive
COMPONENT Geometry2D:2
```

web **3D**
CONSORTIUM



Profiles and components are described in detail in Chapter 1, Technical Overview. This is one of the primary extensibility mechanisms in X3D, allowing the specification to evolve carefully over time.

Common geometry fields: few

Because the 2D nodes are quite simple, most common geometry fields aren't necessary

- *ccw*, *convex*, *colorPerVertex*, *creaseAngle*, etc.

But single-sided flat polygons more easily "lost" than double-sided polygons

- Set *solid*='false' for 2-sided polygon geometry

Application of of Appearance, Material, texturing, LineProperties, FillProperties remains the same

Most fields are *accessType*='initializeOnly' so run-time animation of fields not possible

p. 280, Section 2.2, append paragraph:

Most fields in Geometry2D nodes have *accessType*='initializeOnly', meaning that they cannot be modified directly after creation. Some animation effects can be easily applied to Geometry2D nodes by modifying the *translation*, *rotation* or *scale* fields in a parent Transform node.

Warning: remember that line and point nodes depend on Material emissiveColor, not diffuseColor.

LineProperties and FillProperties are covered in Chapter 5, Appearance, Material, and Textures.

[back to Table of Contents](#)

X3D Nodes and Examples



12

Arc2D node

Arc2D is a line-based node defining a circular arc

- center at local origin (0,0)
- *startAngle*, *endAngle* fields measured in radians, counterclockwise direction positive (right-hand rule)
- Angle measurement starts at positive x-axis, then sweeps toward positive y-axis
- Allowed range $[-2\pi, +2\pi]$

```
<Arc2D  
  startAngle= '1.0'  
  endAngle= '-0.8' />
```

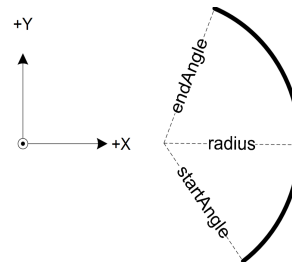


Figure 10.1. Example Arc2D node. The *startAngle* and *endAngle* fields are measured counterclockwise from the positive x-axis to the positive y-axis.

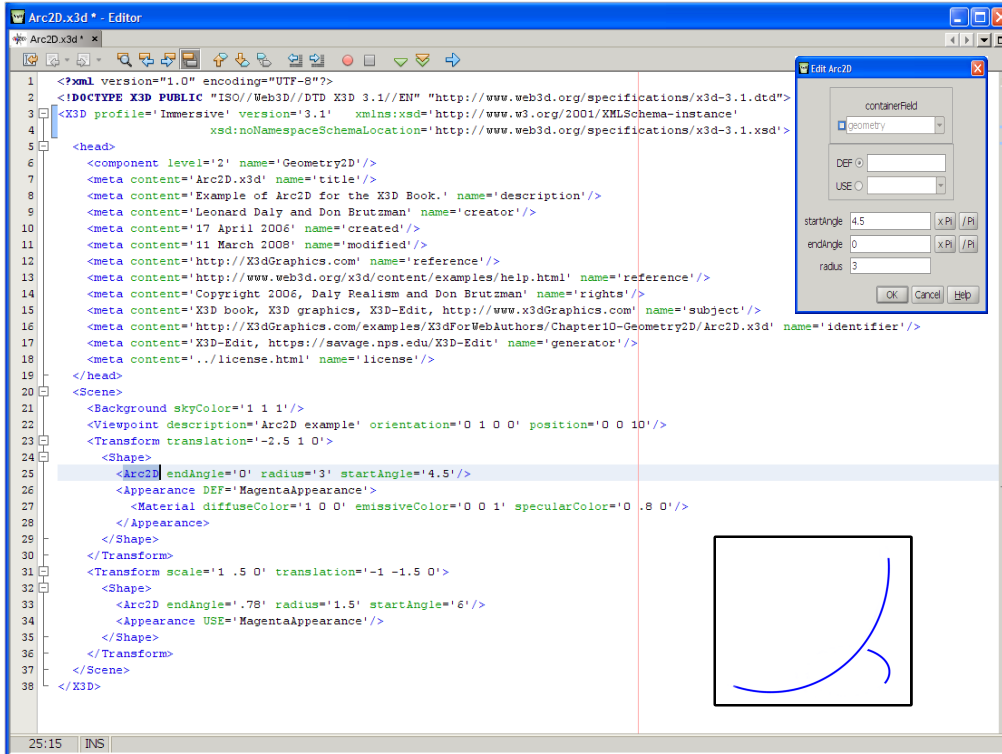



Figure 10.2. Arc2D example display. The lines have been thickened to make them more visible.

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Arc2D.x3d>

Warning: only Material *emissiveColor* can be used to light lines

TODO: Xj3D browser fixed for Geometry2D nodes

 Arc2D	Arc2D is a geometry node. Arc2D specifies a linear circular arc with center (0,0) with angles measured starting at positive x-axis and sweeping towards positive y-axis. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
radius	[radius: accessType initializeOnly, type SFFloat CDATA "1"] (0..infinity) radius of circle, of which the arc is a portion. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
startAngle	[startAngle: accessType initializeOnly, type SFFloat CDATA "0"] [0..2pi] Arc extends from startAngle counterclockwise to endAngle, in radians. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
endAngle	[endAngle: accessType initializeOnly, type SFFloat CDATA "1.570796"] [0..2pi] Arc extends from startAngle counterclockwise to endAngle, in radians. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

<http://www.web3d.org/x3d/content/X3dTooltips.html#Arc2D>

ArcClose2D node

Similar to Arc2D node

ArcClose2D is polygonal, defines enclosed area

- Not just a curved line
- *closureType* field can be CHORD or PIE
- *radius* must be greater than 0
- If $endAngle - startAngle > 2\pi$, full circle is drawn

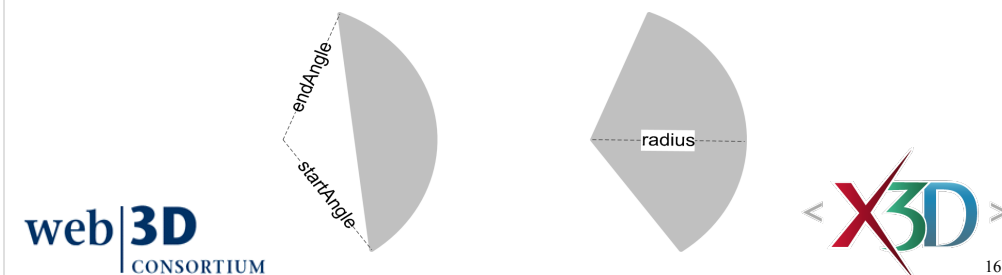


Figure 10.3. Example ArcClose2D nodes, first with *closureType*="CHORD" and then with *closureType*="PIE".

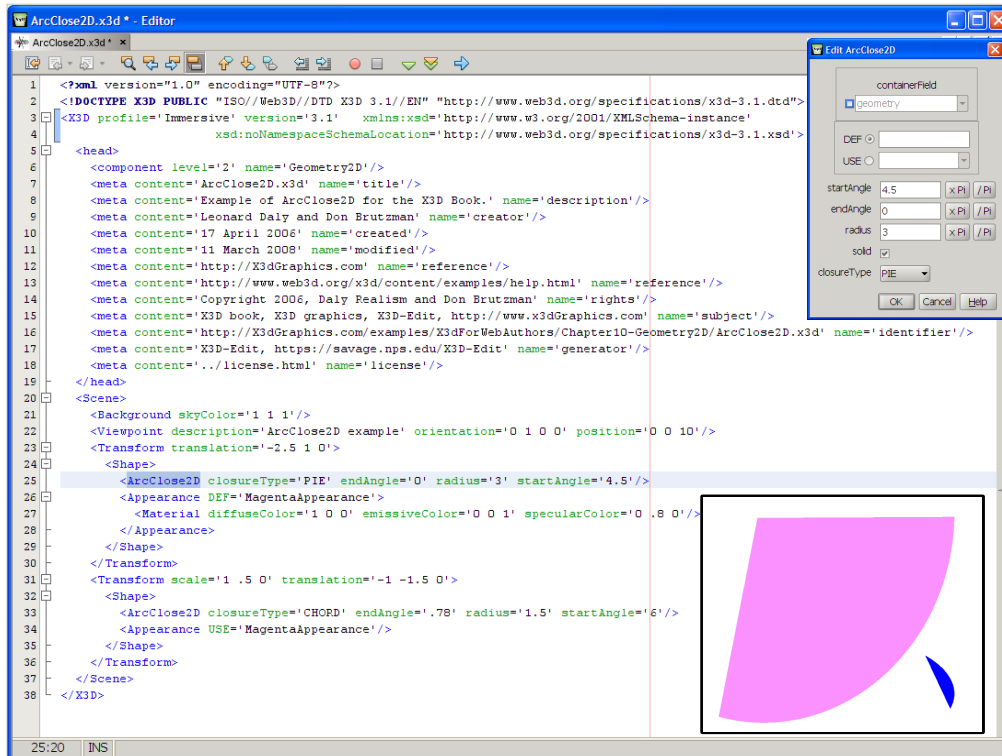



Figure 10.4. An ArcClose2D example display. The blue shape is closed with *closureType*="CHORD", and the pink shape uses *closureType*="PIE".

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/ArcClose2D.x3d>

 ArcClose2D	ArcClose2D is a geometry node. Arc2D specifies a linear circular arc with center (0,0) with angles measured starting at positive x-axis and sweeping towards positive y-axis. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring all other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
radius	[radius: accessType initializeOnly, type SFFloat CDATA "1"] (0..infinity) radius of circle, of which the arc is a portion. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
startAngle	[startAngle: accessType initializeOnly, type SFFloat CDATA "0"] [0..2pi] Arc extends from startAngle counterclockwise to endAngle, in radians. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
endAngle	[endAngle: accessType initializeOnly, type SFFloat CDATA "1.570796"] [0..2pi] Arc extends from startAngle counterclockwise to endAngle, in radians. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
closureType	[closureType accessType initializeOnly, (PIE CHORD) "PIE"] Whether pair of line segments connect to center (PIE), or single line-segment chord connects arc endpoints (CHORD). Warning: simple-geometry parameters cannot be changed after initial creation.
solid	[solid: accessType initializeOnly, type SFBool (true/false) "true"] Setting solid true means draw only one side of polygons (backface culling on), setting solid false means draw both sides of polygons (backface culling off). Warning: default value true can completely hide geometry if viewed from wrong side! Warning: solid false not supported in VRML97.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

<http://www.web3d.org/x3d/content/X3dTooltips.html#ArcClose2D>

Circle2D node

Circle2D is line-based, specifying a planar circle with *center* (0,0) and a positive-valued *radius*

Figure 10.5, corrected: example Circle2D node

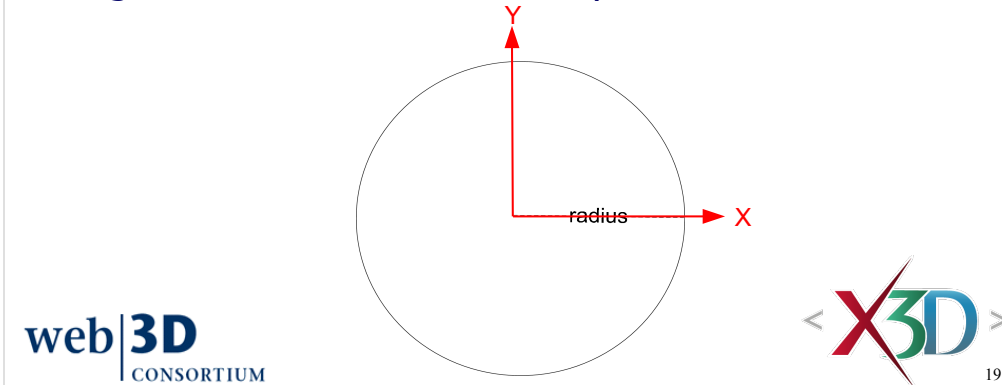


Figure 10.5. An example Circle2D node.

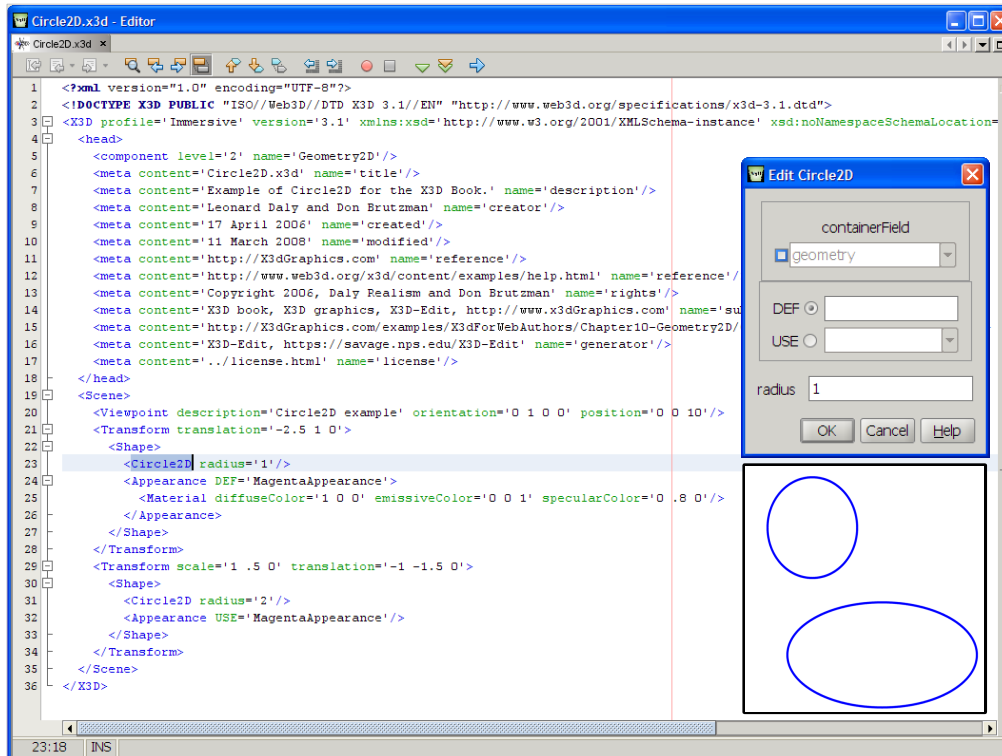


Figure 10.6. A Circle2D example display. The lines have been thickened to make them more visible.

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Circle2D.x3d>

Warning: only Material *emissiveColor* can be used to light lines

○ Circle2D	Circle2D is a geometry node. Circle2D specifies an X-Y planar circle with center (0,0). Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
radius	[radius: accessType initializeOnly, type SFFloat CDATA "1"] (0..infinity) radius of circle. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

<http://www.web3d.org/x3d/content/X3dTooltips.html#Circle2D>

Disk2D node

Disk2D is a polygonal node that defines circular geometry with a circular hole in the center

- Similar to a circular disk washer (for a bolt and nut)
- $0 \leq \text{innerRadius} \leq \text{outerRadius}$

Hint: endcap of Cylinder node can substitute for Disk2D node with *innerRadius='0'*

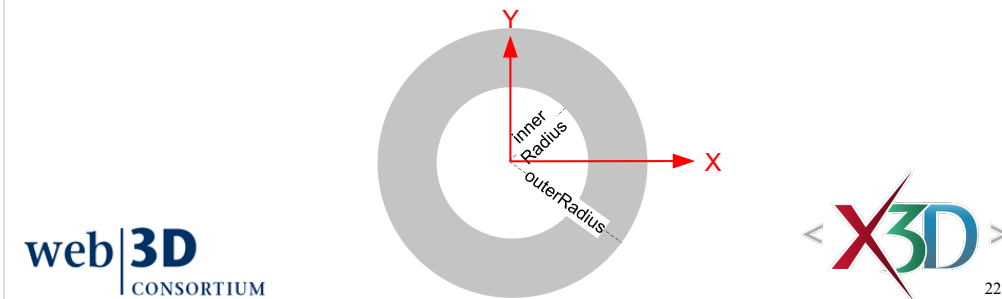


Figure 10.7. An example of a Disk2D node.

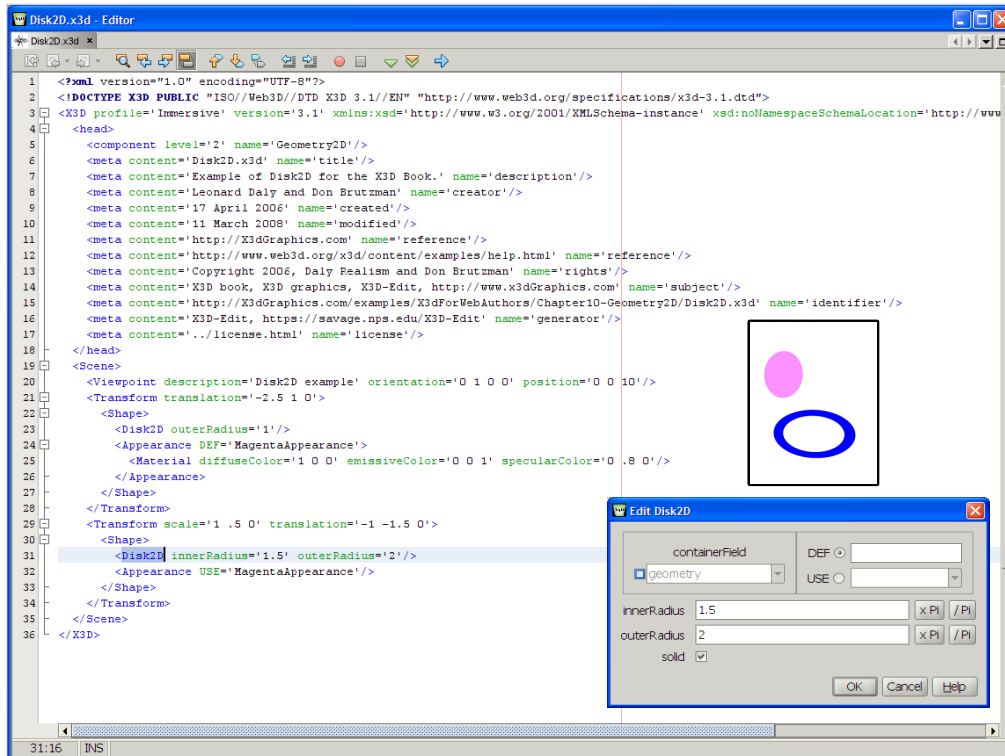



Figure 10.8. A Disk2D example display. The blue disk has a nonzero innerRadius.

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Disk2D.x3d>

 Disk2D	Disk2D is a geometry node. Disk2D specifies a filled planar circle with center (0,0). Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring all other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
innerRadius	[innerRadius: accessType initializeOnly, type SFFloat CDATA "0"] [0..infinity) Inner radius of circle, greater than or equal to 0. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
outerRadius	[outerRadius: accessType initializeOnly, type SFFloat CDATA "1"] (0..infinity) Outer radius of circle, greater than 0. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
solid	[solid: accessType initializeOnly, type SFBool (true/false) "true"] Setting solid true means draw only one side of polygons (backface culling on), setting solid false means draw both sides of polygons (backface culling off). Warning: default value true can completely hide geometry if viewed from wrong side! Warning: solid false not supported in VRML97.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

<http://www.web3d.org/x3d/content/X3dTooltips.html#Disk2D>

Polyline2D node

Polyline2D defines a connected set of 2D vertices as contiguous, connected set of line segments

Disconnected lines must be constructed using separate Polyline2D nodes

Point values are contained in lineSegments field

- Example: six point values, 0th and 5th are identical

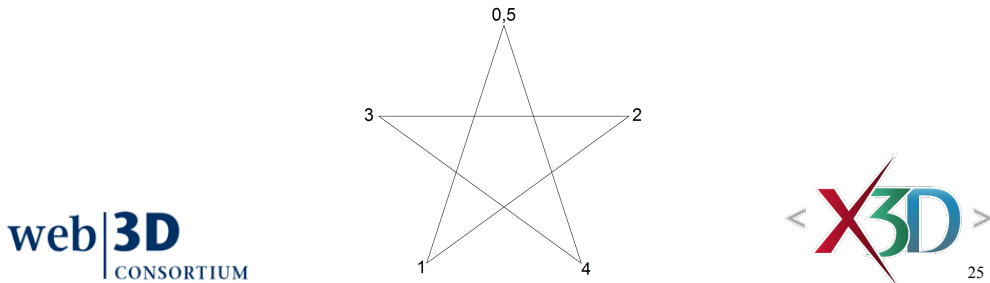


Figure 10.9. An example Polyline2D node; a set of six 2D coordinates defining five contiguous line segments.

Warning: only Material *emissiveColor* can be used to light line segments

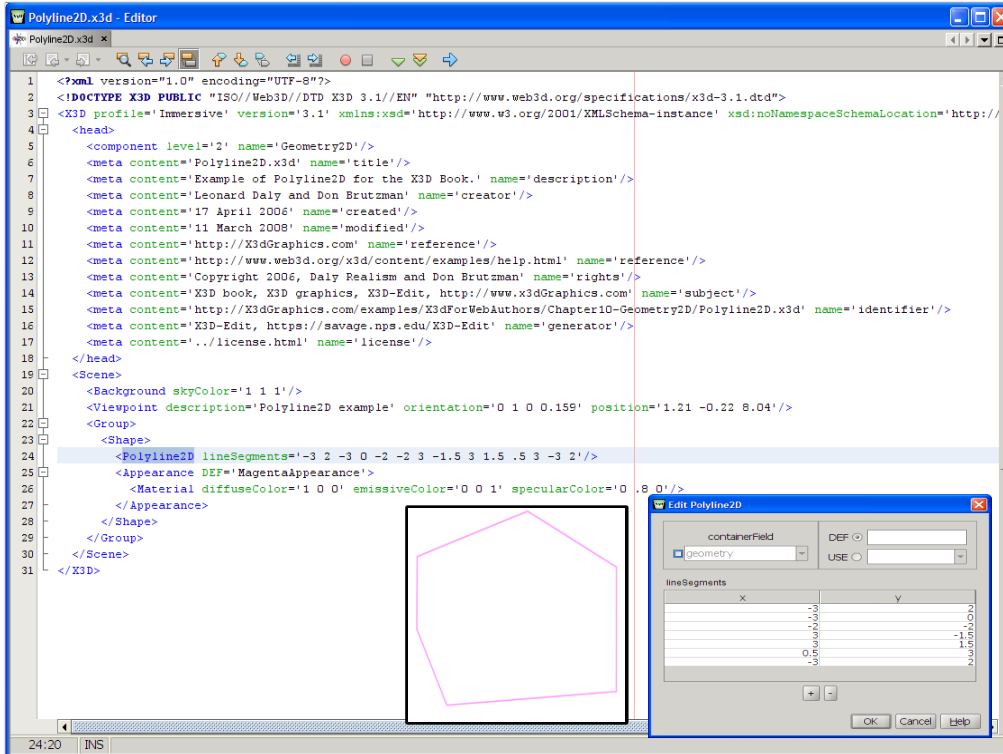


Figure 10.10. A Polyline2D example. The lines have been thickened to make them more visible.

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Polyline2D.x3d>

^ Polyline2D	<p>Polyline2D is a geometry node. Polyline2D specifies a connected set of vertices in a contiguous set of line segments in X-Y plane. Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
lineSegments	<p>[lineSegments: accessType initializeOnly, type MFVec2f CDATA #FIXED ""] Coordinates of vertices connected into contiguous Polyline2D. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.</p>
containerField	<p>[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>

<http://www.web3d.org/x3d/content/X3dTooltips.html#Polyline2D>

Polypoint2D node

Polypoint2D defines a set of 2D points

- *point* field holds an array of X-Y coordinate pairs

Rendering and limitations match those for
PointSet node

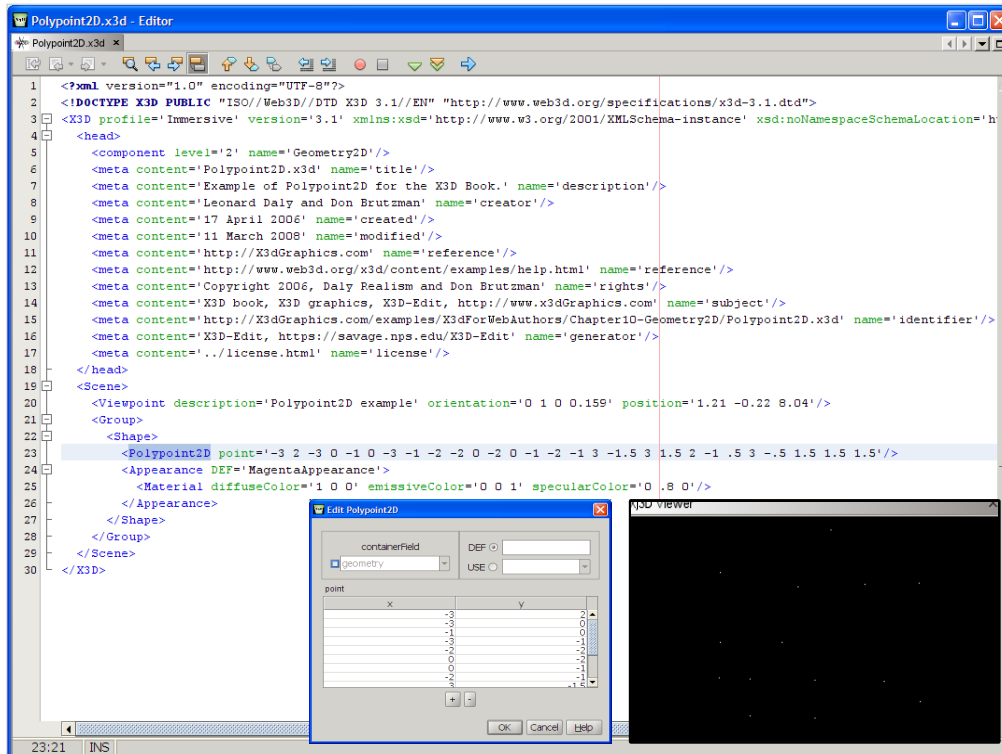


web|3D
CONSORTIUM




Figure 10.11. An example Polypoint2D node containing 22 2D point values. Point size is exaggerated for visibility, because each point is actually drawn as a single pixel.

Warning: only Material *emissiveColor* can be used to light points



Points are often hard to see, especially when projected since projectors typically do not have the same resolution quality or color fidelity as computer monitors.

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Polypoint2D.x3d>

 Polypoint2D	Polypoint2D is a geometry node. Polypoint2D specifies a set of 2D points. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
point	[point: accessType initializeOnly, type MFVec2f CDATA #FIXED ""] 2D coordinates of vertices. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

<http://www.web3d.org/x3d/content/X3dTooltips.html#Polypoint2D>

Rectangle2D node

Rectangle2D defines a rectangular quadrangle

- again centered at (0,0) in local coordinate system
- aligned with the local x-y coordinate axes

size is SFVec2f field defining x and y dimensions

- each greater than zero

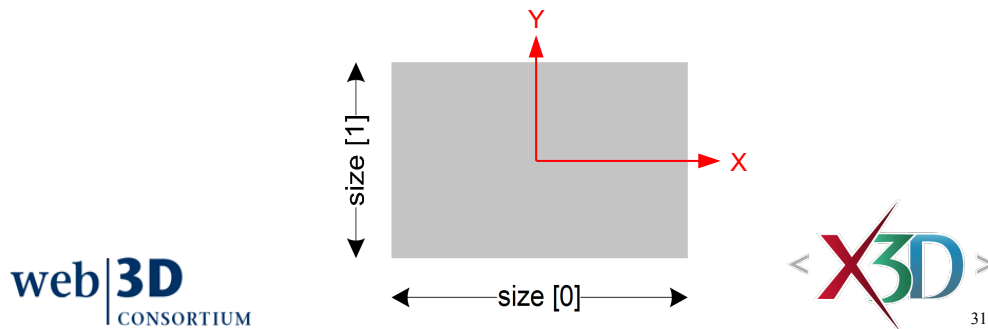


Figure 10.13. An example Rectangle2D node showing 2D size components in x and y directions respectively. Local origin is at the center.

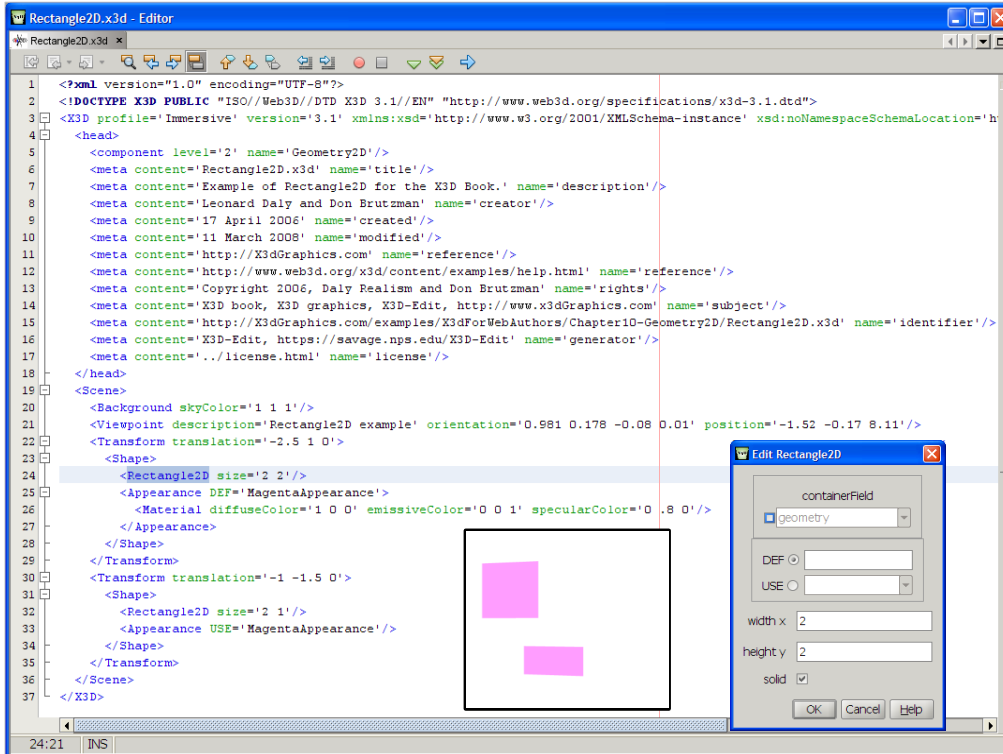



Figure 10.14. A Rectangle2D example display.

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Rectangle2D.x3d>

 Rectangle2D	Rectangle2D is a geometry node. Rectangle2D specifies a 2D rectangle. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
size	[size: accessType initializeOnly, type SFVec2f CDATA "2 2"] 2D dimensions of Rectangle2D. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
solid	[solid: accessType initializeOnly, type SFBool (true/false) "true"] Setting solid true means draw only one side of polygons (backface culling on), setting solid false means draw both sides of polygons (backface culling off). Warning: default value true can completely hide geometry if viewed from wrong side! Warning: solid false not supported in VRML97.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

<http://www.web3d.org/x3d/content/X3dTooltips.html#Rectangle2D>

TriangleSet2D node

TriangleSet2D defines a set of triangles, each created by three 2D vertex points

Points are listed individually in the *vertices* field

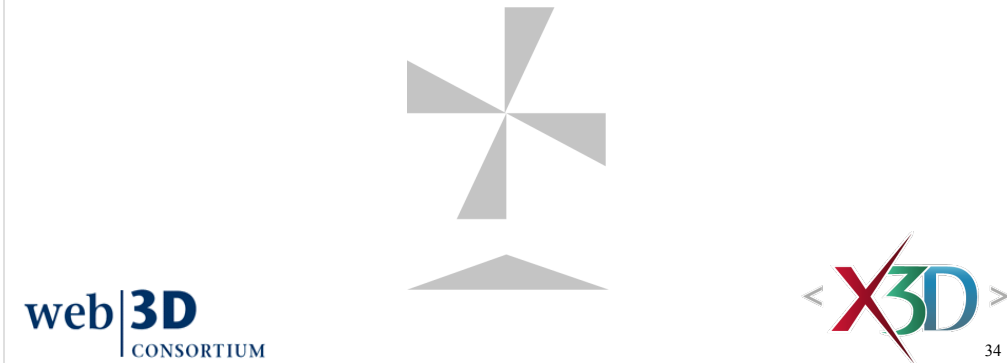


Figure 10.15. An example TriangleSet2D node, containing a set of fifteen 2D vertices that create five 2D triangles. Coincident coordinates are listed multiple times.

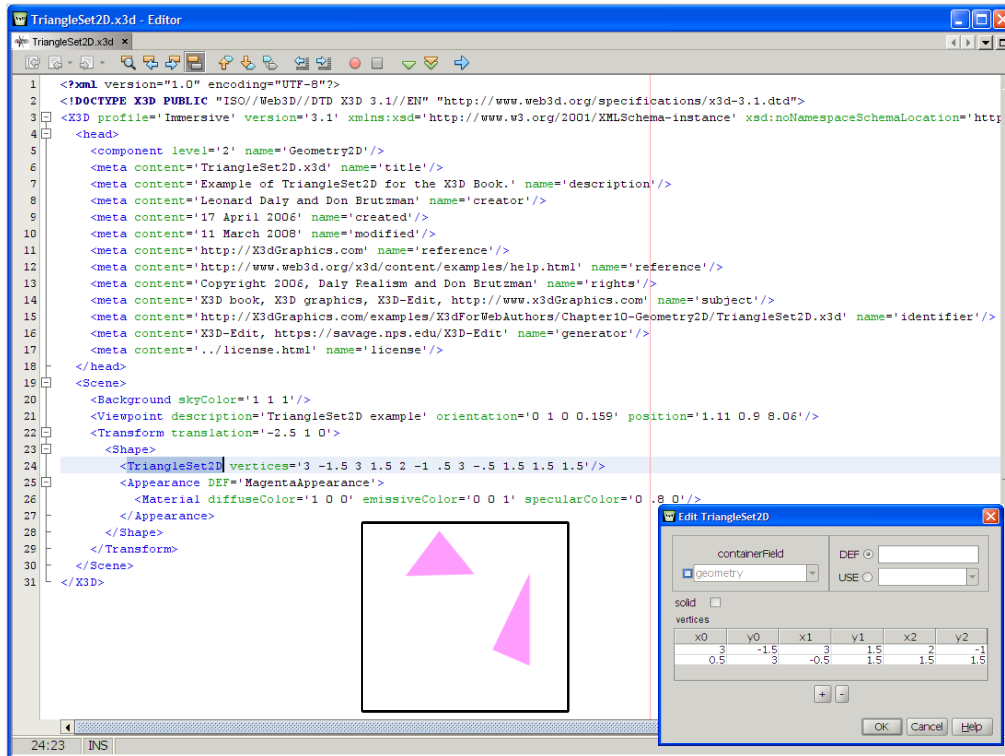



Figure 10.16. A TriangleSet2D example display. A single TriangleSet2D can generate multiple triangles.

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/TriangleSet2D.x3d>

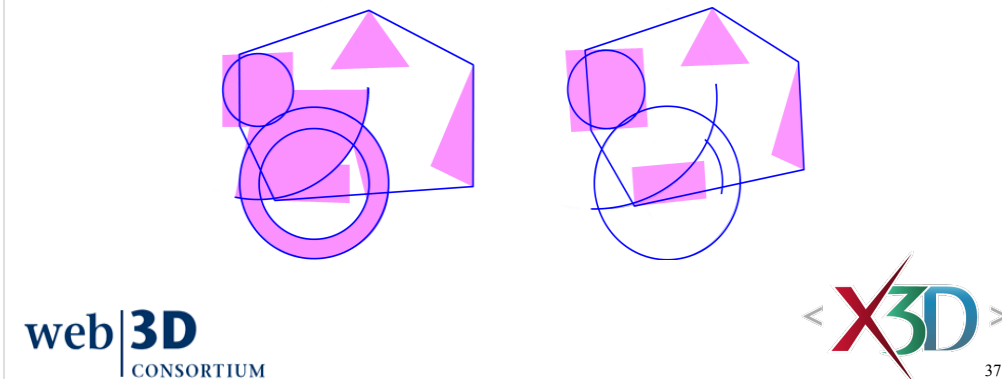
 TriangleSet2D	TriangleSet2D is a geometry node. TriangleSet2D specifies a set of 2D triangles. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
vertices	[vertices: accessType initializeOnly, type MFVec2f CDATA #FIXED ""] 2D coordinates of TriangleSet2D vertices. Warning: simple-geometry dimensions cannot be changed after initial creation, use Transform scale instead.
solid	[solid: accessType initializeOnly, type SFBool (true/false) "true"] Setting solid true means draw only one side of polygons (backface culling on), setting solid false means draw both sides of polygons (backface culling off). Warning: default value true can completely hide geometry if viewed from wrong side! Warning: solid false not supported in VRML97.
containerField	[containerField: NMTOKEN "geometry"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

<http://www.web3d.org/x3d/content/X3dTooltips.html#TriangleSet2D>

Example combinations of 3D nodes

Sample scene Summary2D.x3d shows a variety of Geometry2D nodes

- Different values of *solid*='true' and *solid*='false' provides different geometry for front, back views



<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter10-Geometry2D/Summary2D.x3d>

Figure 10.17. The front view of the collection of Geometry2D nodes.

Figure 10.18. The underside view of the same collection of Geometry2D nodes, which is different because some geometry has *solid*="true" which results in backface culling when rendered.

[back to Table of Contents](#)

Additional Resources



38

X3D prototype implementations

Geometry2dComponentPrototypes.x3d

Geometry2dComponentExamples.x3d

- X3D Geometry2D component nodes, implemented as prototypes, for browsers lacking such support and for backwards compatibility with VRML97
- <http://www.web3d.org/x3d/content/examples/Basic/development/>
- Maps 2D node parameters to 3D node implementations (IndexedFaceSet etc.)

X3dToVrml97.xslt stylesheet automatically inserts external prototype declarations for each of the Geometry2D nodes, as appropriate



39

Scalable Vector Graphics (SVG) 1

SVG is language for describing two-dimensional graphics and graphical applications in XML

- <http://www.w3.org/Graphics/SVG>
- SVG 1.1 is a W3C Recommendation and forms the core of the current SVG developments
- SVG Tiny 1.2 is the specification currently being developed as the core of the SVG 1.2 language
- SVG Mobile Profiles (Basic and Tiny) are designed for small devices and mobile phones
- SVG Print guidelines help produce final-form documents for archiving and printing



More information about SVG is available in the notes for Chapter 5, Appearance Materials and Textures.

Scalable Vector Graphics (SVG) 2

Many good books are available about SVG
SVG specification and various tutorials are online

- <http://www.w3.org/Graphics/SVG>

Although direct mappings between SVG and X3D are not available, this is an excellent area for further work

- XML stylesheets (.xslt) might convert SVG nodes into X3D nodes, and vice versa
- Perhaps some browsers will implement SVG as an allowed ImageTexture and MovieTexture format



More information about SVG is available in the notes for Chapter 5, Appearance Materials and Textures.

[back to Table of Contents](#)

Chapter Summary



42

Chapter Summary

Common concepts: simple 2D convenience nodes

Geometry2D Nodes

- Arc2D excerpt from a circle's circumference
- ArcClose2D pie or chord subsection of circle shape
- Circle2D line for a complete circle circumference
- Disk2D circular shape with optional center hole
- Polyline2D sequence of line segments
- Polypoint2D one or more points
- Rectangle2D height, width of single polygonal shape
- TriangleSet2D multiple polygonal triangles

Related nodes

Substitutions are available for each Geometry2D node, using the corresponding 3D nodes

Arc2D	IndexedLineSet	LineSet
ArcClosed2D	IndexedFaceSet	Triangle nodes
Circle2D	IndexedLineSet	LineSet
Disk2D	IndexedFaceSet	Triangle nodes
Polyline2D	IndexedLineSet	LineSet
Polypoint2D	PointSet	
Rectangle2D	QuadSet	IndexedFaceSet
TriangleSet2D	IndexedFaceSet	Triangle nodes

Interpolators are not provided for Geometry2D nodes since most fields cannot be animated

- *accessType*='initializeOnly'

Suggested exercises

Build Olympic rings that face the user



- http://commons.wikimedia.org/wiki/Image:Olympic_Rings.svg
- Using Polyline2D
- Using Arc2D or Circle2D

Build 5- or 6-sided star similar to Figure 10.9

- Using Polyline2D
- Using TriangleSet2D

Build a simple clickable user interface to control animation in a scene: pause, play, stop

[back to Table of Contents](#)

References

References 1

X3D: Extensible 3D Graphics for Web Authors
by Don Brutzman and Leonard Daly, Morgan
Kaufmann Publishers, April 2007, 468 pages.



- Chapter 10, Geometry2D Nodes
- <http://x3dGraphics.com>
- <http://x3dgraphics.com/examples/X3dForWebAuthors>

X3D Resources

- <http://www.web3d.org/x3d/content/examples/X3dResources.html>



47

References 2

X3D-Edit Authoring Tool

- <https://savage.nps.edu/X3D-Edit>

X3D Scene Authoring Hints

- <http://x3dgraphics.com/examples/X3dSceneAuthoringHints.html>

X3D Graphics Specification

- <http://www.web3d.org/x3d/specifications>
- Geometry2D component
- Also available as help pages within X3D-Edit



48

Contact

Don Brutzman

brutzman@nps.edu

<http://faculty.nps.edu/brutzman>

Code USW/Br, Naval Postgraduate School
Monterey California 93943-5000 USA
1.831.656.2149 voice

web|**3D**
CONSORTIUM



49

CGEMS, SIGGRAPH, Eurographics

The Computer Graphics Educational Materials Source(CGEMS) site is designed for educators

- to provide a source of refereed high-quality content
- as a service to the Computer Graphics community
- freely available, directly prepared for classroom use
- <http://cgems.inesc.pt>

X3D for Web Authors recognized by CGEMS! ☺

- Book materials: X3D-Edit tool, examples, slidesets
- Received jury award for Best Submission 2008

CGEMS supported by SIGGRAPH, Eurographics



From the CGEMS home page:

- <http://cgems.inesc.pt>

Welcome to CGEMS - Computer Graphics Educational Materials Source. The CGEMS site is designed for educators to provide a source of refereed high-quality content as a service to the Computer Graphics community as a whole. Materials herein are freely available and directly prepared for your classroom.

List of all published modules:

- <http://cgems.inesc.pt/authors/ListModules.aspx>

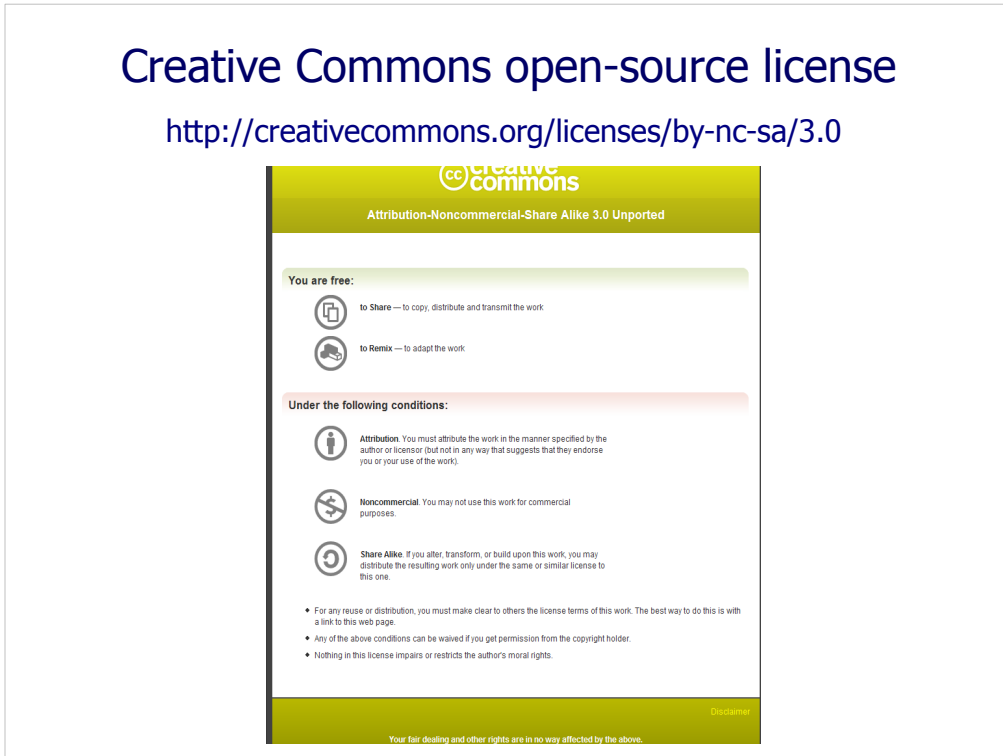


CGEMS Editorial Policy:

- <http://cgems.inesc.pt/EditorialPolicy.htm>

Creative Commons open-source license

<http://creativecommons.org/licenses/by-nc-sa/3.0>



Attribution-Noncommercial-Share Alike 3.0 Unported

You are free:

- * to Share — to copy, distribute and transmit the work
- * to Remix — to adapt the work

Under the following conditions:

* Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Attribute this work: What does "Attribute this work" mean?

The page you came from contained embedded licensing metadata, including how the creator wishes to be attributed for re-use. You can use the HTML here to cite the work. Doing so will also include metadata on your page so that others can find the original work as well.

- * Noncommercial. You may not use this work for commercial purposes.
- * Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.
- * For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.
- * Any of the above conditions can be waived if you get permission from the copyright holder.
- * Nothing in this license impairs or restricts the author's moral rights.

Open-source license for X3D-Edit software and X3D example scenes

<http://www.web3d.org/x3d/content/examples/license.html>

Copyright (c) 1995-2013 held by the author(s). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the Naval Postgraduate School (NPS) Modeling Virtual Environments and Simulation (MOVES) Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

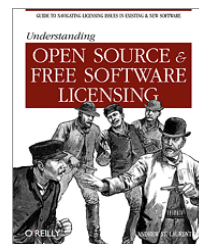
License available at

<http://www.web3d.org/x3d/content/examples/license.txt>

<http://www.web3d.org/x3d/content/examples/license.html>

Good references on open source:

Andrew M. St. Laurent, *Understanding Open Source and Free Software Licensing*, O'Reilly Publishing, Sebastopol California, August 2004. <http://oreilly.com/catalog/9780596005818/index.html>



Herz, J. C., Mark Lucas, John Scott, *Open Technology Development: Roadmap Plan*, Deputy Under Secretary of Defense for Advanced Systems and Concepts, Washington DC, April 2006. <http://handle.dtic.mil/100.2/ADA450769>

