Army Research Laboratory

Visualization of a Text Network Structure Using X3D

by Andrew M. Neiderer

ARL-MR-691 May 2008

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer’s or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-MR-691 **May 2008**

Visualization of a Text Network Structure Using X3D

Andrew M. Neiderer
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE	***Form Approved*** ***OMB No. 0704-0188***

1. REPORT DATE *(DD-MM-YYYY)* May 2008	**2. REPORT TYPE** Final	**3. DATES COVERED (From - To)** April 2007–November 2007

4. TITLE AND SUBTITLE Visualization of a Text Network Structure Using X3D	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Andrew M. Neiderer	**5d. PROJECT NUMBER** 8TEDUC
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRD-ARL-CI-CT Aberdeen Proving Ground, MD 21005-5067	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-MR-691
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report describes a dynamic extensible three-dimensional (3-D) (X3D) scene graph for visualizing text documents. It was developed at the U.S. Army Research Laboratory as a tool for social network analysis: specifically, to view a network of nodes where a node is a noun within a news article. Keyword and target nodes are represented as X3D spheres of different radii and color. Keyword node attraction to a specific target node is dynamic by a Java class access to scene content from an X3D script node. The magnitude of the direction vector, or speed, between keyword and target is currently proportional to frequency of the word within the document. This technique can be modified/replaced easily for a more comprehensive examination, such as with an algorithm that considers word selection across multiple documents. X3D code for a particular example is given and viewed using the Xj3D 2.0 browser from Yumetech, Inc.

15. SUBJECT TERMS

network visualization, extensible 3-D, X3D, social networks

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Andrew M. Neiderer
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED	UL	44	**19b. TELEPHONE NUMBER** (*Include area code*) 410-278-3203

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures iv

1. Introduction and Problem Description 1

2. An X3D Scene Graph Description of a Text Network 4

3. Conclusion and Future Considerations 5

Appendix. The X3D and Java SAI Code for Text Network Application 7

List of Symbols, Abbreviations, and Acronyms 35

Distribution List 36

List of Figures

Figure 1. A 2-D simple network consisting of 26 nodes with 43 links. 2

Figure 2. The 3-D navigation and selective viewing within the 26-node network is shown.........2

Figure 3. A more complex 2-D network where nodes (red) are distributed amongst three concentric conics. Note that the number of links is less than the maximum number ${}_{n}C_{2}$ 3

Figure 4. A sample X3D node network. The targets, i.e., selected keywords, are the red spheres having a <TouchSensor> defined for toggling text. The keyword codes are pink with texture animation of text. 3

1. Introduction and Problem Description

An effective display of word usage within a text document could not only assist in reader comprehension but could also provide insight to forecasting future events. This is of particular interest within the Department of Defense in its effort to deter terrorist acts, which often happen without warning but sometimes there are indicators. Identifying and fusing data from a reported news story(s) that answers the who, what, when, where, and why in a timely manner across multiple sources, could make a difference. The U.S. Army Research Laboratory is developing software for display and manipulation of news articles represented as text networks. This report documents an interactive extensible three-dimensional (3-D) (X3D) scene description for a single news article and is designed with a capability of considering multiple sources in future releases.

Note that a large number of documents is more easily and quickly comprehended when graphically displayed than when presented as raw text. Visualization of the presentation is a cognitive process where meaning is given to that data by creation of an ephemeral mental image. Therefore, images generated for computer display must be carefully constructed since they may influence decisions.

The following X3D scene graph description is a directed acyclic graph (DAG) of objects and relationships among those objects. In this case the objects are nodes represented as X3D spheres with relationships between nodes, or links, displayed as line segments.

First, a two-dimensional (2-D) description of a network where spatial relationships are preserved is illustrated (see figure 1). Even for a simple network, critical details may be more apparent when user navigation and viewing are done in 3-D space (see figure 2). If the number of nodes and links increases, identifying a specific link between two nodes can be even more difficult (see figure 3). Perhaps viewing from a particular center of projection for selected node placements would make isolation of a specific link easier and result in a more apparent relationship. To also maximize viewer comprehension, both nodes and links are color coded.

Second, color and transparency animation of the nodes is examined. Words of particular interest are identified and labeled as targets. An X3D <TouchSensor> is placed on each to toggle (i.e., turn on/off) text. Other nouns within the text, called keywords, are animated from opaque to transparent when touched to help off-load some of the cognitive processing to the perceptual system. In this initial version of the software, color animation was not implemented since user color sensitivity can vary greatly, to include color-blindedness. Further research is needed.

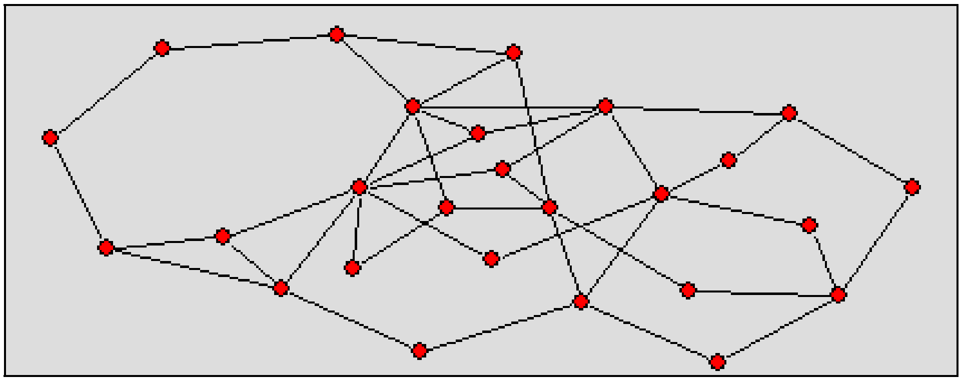

Figure 1. A 2-D simple network consisting of 26 nodes with 43 links.

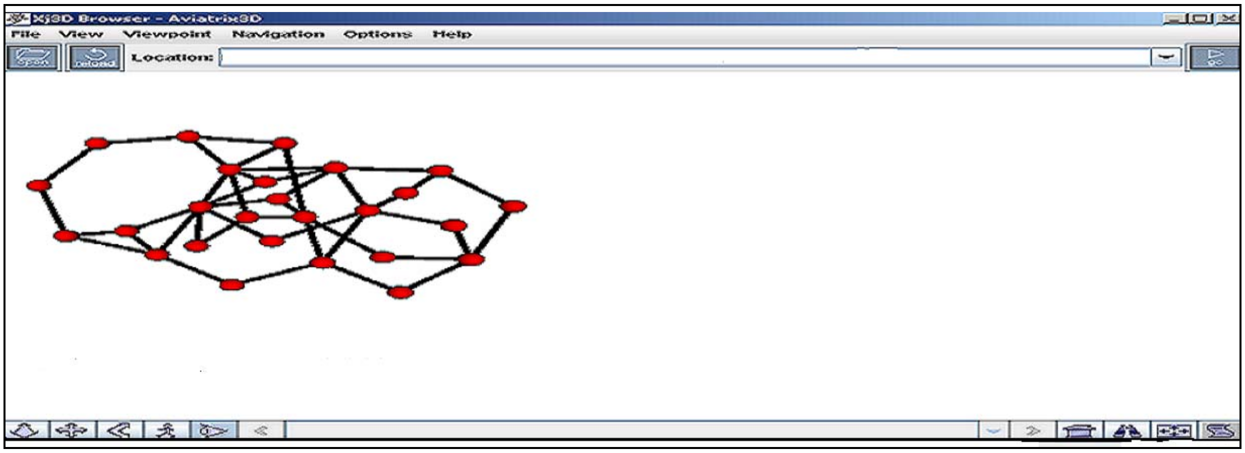

Figure 2. The 3-D navigation and selective viewing within the 26-node network is shown.

The density of links between nodes within a single image can be visually overwhelming, especially for a large network. We allow for all cases: (1) display of no links, (2) all links, and (3) selective display. In the first case, we adapt and implement a basic principle in physics: the Lenard-Jones potential energy function.[1] The internal script is written in standardized JavaScript, which is called the European Computer Manufacturers Association scripting language (ECMAScript), and assumes that target and keyword nodes are point masses so that the force of attraction is directly proportional to velocity, or speed, along the straight line connecting the two centers (see Avoid.x3d in the Appendix). The magnitude of the direction vector is simply proportional to word frequency in the text, but can be replaced by a different algorithm. An example is shown in figure 4, where there are two target nodes and seven keyword nodes. Note that all links between the n nodes is the statistical combination, mathematically expressed as ${}_{n}C_{2} = n! / [(n - 2)!\ 2!]$. For user-specified link display, a <TouchSensor> is defined and attached to each link for one to toggle display; this could assist in identification of indirect links.

[1] Bourg, D. M. *AI for Game Developers;* O'Reilly Media, Inc.: Sebastopol, CA, July 2004.

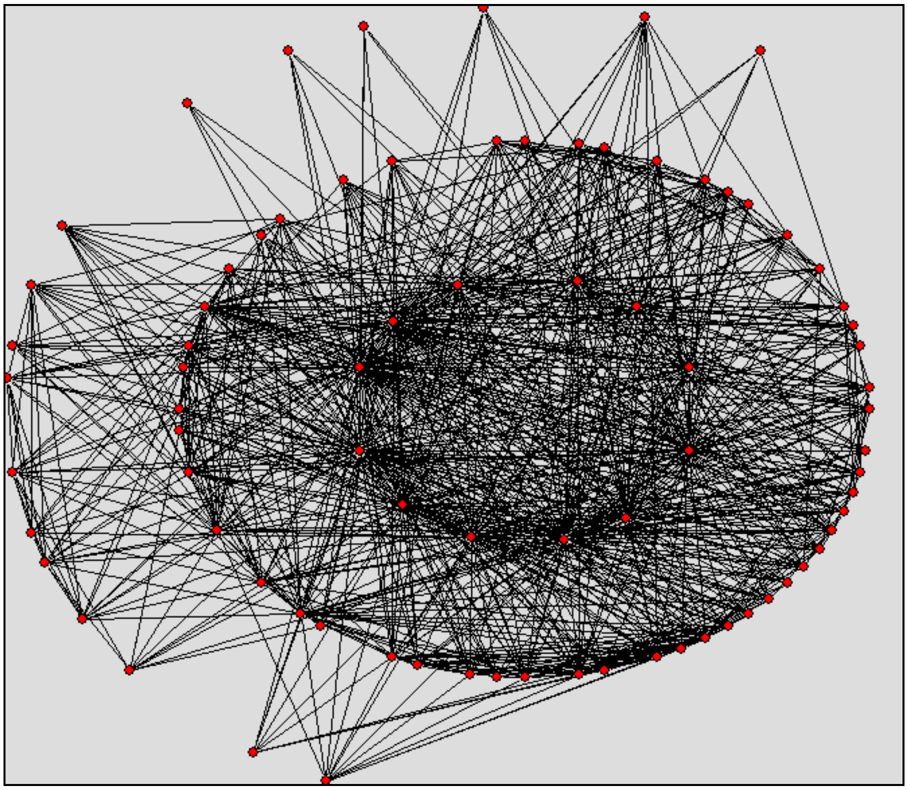

Figure 3. A more complex 2-D network where nodes (red) are distributed amongst three concentric conics. Note that the number of links is less than the maximum number ${}_{n}C_{2}$.

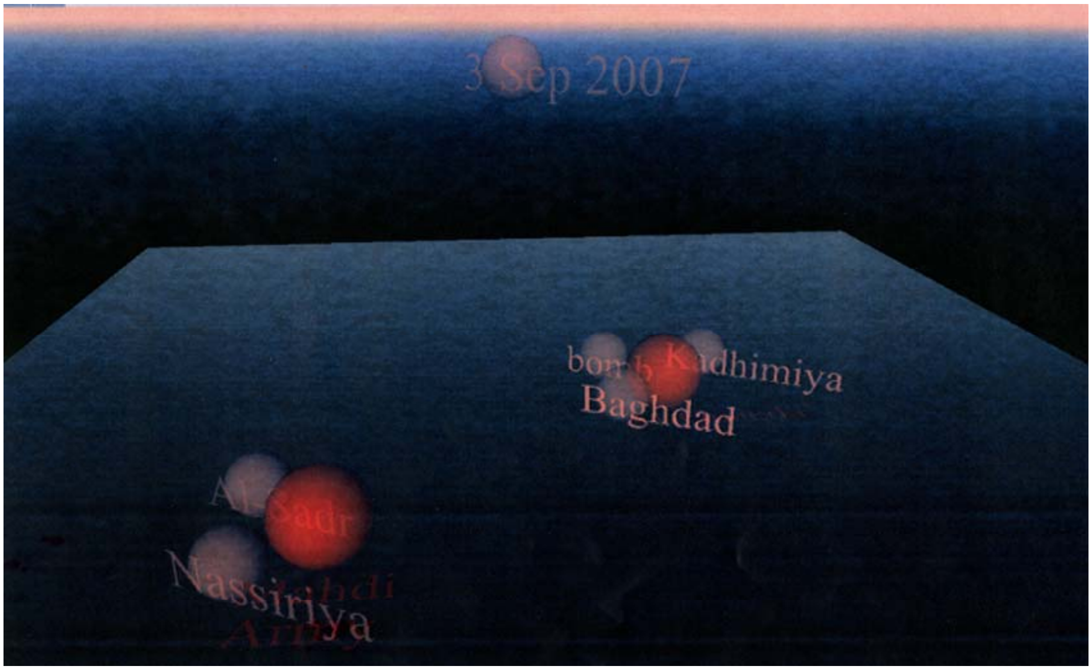

Figure 4. A sample X3D node network. The targets, i.e., selected keywords, are the red spheres having a <TouchSensor> defined for toggling text. The keyword codes are pink with texture animation of text.

Section 2 provides details of the X3D scene graph for text document representation, including the necessary routes. The scene graph is animated by an event cascade across sensors, interpolators, and script nodes in addition to the usual inclusion of <Transform> nodes. A scene access interface (SAI)[*] written in Java allows for dynamic manipulation of the scene graph (e.g., switching the "goal," which is a field of the defined X3D prototype node for a keyword). X3D routes node output fields to another node's input field for animation.

2. An X3D Scene Graph Description of a Text Network

X3D graphics is an international standard for defining scene content across the Web that allows for real-time, interactive communication with that content. It is XML-encoded which makes it simple to expose data for SAI. In our case, manipulation of scene content is accomplished through Java language bindings for the Yumetech, Inc. browser, Xj3D 2.0. This site also provides a comprehensive tutorial of SAI.[2]

The X3D specification defines five profiles for 28 components. A component is a set of functionalities for various X3D objects, and a profile is built from components. An excellent reference for discussion of these concepts, and X3D, in general, can be found in Brutzman and Daly.[3] For our text network structure, we use the inclusive immersive profile and select it on the header (see third line of the X3D in the appendix); i.e., for <X3D>, the attribute value for the attribute name "profile" is "Immersive."

A scene graph is a collection of graphical objects arranged in a hierarchical relationship. The fundamental object is a node. The virtual environment (VE) for viewing and navigation are first defined using the appropriate X3D nodes. Four predefined positions are accessible from the view bar at the bottom of the browser: top, front, back, and front-top view. Navigation is defined for both flying and walking through the scene.

Nodes have properties called fields. A field includes an attribute for storage of the basic data type (e.g., a floating-point value called SFFloat, or a vector of three values called SFVec3f). The routing of an output field of a geometrical child node of an X3D <Shape> to the input field of another makes animation within the scene graph possible.

X3D event animation at author-time is an event cascade across sensor, interpolator, and transform nodes. The <ROUTE> mechanism simply makes the abstract connection between the

[*] Some refer to SAI as a scene "authoring" interface. Since scene content is not authored in our code using a <Script>, and we only "access" content, the latter is used.

[2] Hudson, A. SAI tutorial by Yumetech, Inc. http://www.xj3d.org/tutorials/general_sai.html (accessed 19 September 2007).

[3] Brutzman, D.; Daly, L. *X3D: Extensible 3D Graphics for Web Authors;* Morgan Kaufmann Publishers, Elsevier, Inc., 2007.

node sending the event and the node receiving that event. But when the chain includes a script node, a field value can be modified at run time.

<Script> allows one to embed program code into the X3D scene. In our case, a Java scene access interface (SAI) to X3D scene graph content uses the Browser class defined for the Xj3D viewer

Target nodes can be touched to toggle the text and is made possible by routing its <TouchSensor> to the child <Switch> node of the <Transform>. Geometric translation of the target node in Xj3D is done by dragging the pointer on it. An instance of the Lenard-Jones potential energy is defined for each child keyword node by an internal ECMAScript function.

Keyword nodes are also texture animated, and as a result, the event chain is more complicated. A <TouchSensor> is placed on the keyword geometry. The event is first routed to a <TimeSensor> node where the cycle interval is controlled and currently set to 10 s; this can be easily modified. To complete the fading effect, the “startTime” field of this <TimeSensor> is routed to a <ScalarInterpolator> and then to the <Material> node for the text <Shape>.

Note that target nodes are positioned relative to the root coordinate system of the scene which is a right-handed system (RHS). Keyword nodes for a determined target are also positioned in an RHS but are relative and need to be oriented. This is necessary for use of ECMAScript implementation of potential energy in the Avoid prototype (see Avoid.x3d in the appendix).

The complete code for a sample network of nodes defining a text document is given in the appendix. The Java template for switching a target of a keyword node field “goal” (TargetSelect.java) has been written and a setFields() method within this class communicates with scene content and will be further refined.

3. Conclusion and Future Considerations

The preceding discussion introduced an X3D text network that will serve as a model for SNA tool development. An initial X3D implementation for the display of nouns within a text document now exists. Keywords and a target word are nouns within a news article that are displayed as X3D spheres in the Xj3D browser.

With the initial version stablized, we plan to research more extensive information visualization. We plan to examine larger, denser networks, including populating the network with more nodes which typically means more links. Strategic node placement and minimizing edge length and crossings should also help reduce clutter. In addition, using X3D <TouchSensor>s for selective display of links should also help.

New additions will be evaluated in an Ajax context as the networks become more complex.

INTENTIONALLY LEFT BLANK.

Appendix. The X3D and Java SAI Code for Text Network Application

The main X3D code (SAIdynamicAttraction.x3d) and the external prototype definition (Avoid.x3d) for attraction of a keyword node to a target now follow. The template for a Java SAI to X3D scene content is also documented.

This appendix appears in its original form, without editorial review.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN"
                     "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile="Immersive" version="3.1">
  <head>
    <!-- an X3D SNA application for "keyword" node attraction to a "target" node,            -->
    <!--  where "speed" represents strength of bond between the two nodes.  Uses a Java      -->
    <!--  scene access interface (SAI) to the X3D scene content from a <Script> node         -->
    <!--  of the <Scene> for a dynamic scene graph.                                          -->
    <!--                                                                                     -->
    <!-- added <TouchSensor> to toggle text on target nodes (10/4/2007).                     -->
    <!-- added transparency animation to text of keyword nodes by touching geometry (10/5/2007).  -->
    <!--                                                                                     -->
    <!-- by Andrew M. Neiderer 15 Oct 2007,                                                  -->
    <!--  using VRML obstacle avoidance technique by Peter Gerstmann (Ohio State Univ).      -->
  </head>

  <Scene>
    <ExternProtoDeclare name="Avoid" url="Avoid.x3d">
      <field name="children"      type="MFNode"   accessType="inputOutput"/>
      <field name="goal"          type="SFVec3f"  accessType="initializeOnly"/>
      <field name="goalPriority"  type="SFFloat"  accessType="initializeOnly"/>
      <field name="obstacles"     type="MFNode"   accessType="initializeOnly"/>
      <field name="position"      type="SFVec3f"  accessType="inputOutput"/>
      <field name="setGoal"       type="SFVec3f"  accessType="inputOnly"/>
      <field name="setState"      type="SFString" accessType="inputOnly"/>
      <field name="setObstacles"  type="MFNode"   accessType="inputOnly"/>
      <field name="speed"         type="SFFloat"  accessType="initializeOnly"/>
      <field name="state"         type="SFString" accessType="initializeOnly"/>
      <field name="updateRequest" type="SFTime"   accessType="inputOnly"/>
    </ExternProtoDeclare>

    <Background groundAngle="0.1, 1.309, 1.571"
                groundColor="0.0 0.0 0.0, 0.0 0.1 0.3, 0.0 0.2 0.5, 0.0 0.3 0.8"
                skyAngle="0.1, 0.15, 1.309, 1.571"
                skyColor="0.4 0.4 0.1, 0.4 0.4 0.1, 0 0.1 0.3, 0 0.2 0.6, 0.8 0.8 0.8"
                frontUrl="horizon_1_front.jpg"
                backUrl="horizon_1_back.jpg"
                leftUrl="horizon_1_left.jpg"
                rightUrl="horizon_1_right.jpg"
                topUrl="horizon_1_top.jpg"
                bottomUrl="horizon_1_bottom.jpg"/>

    <Fog DEF="FOG"
                color="1.0 1.0 1.0" fogType="LINEAR" visibilityRange="75.0"/>

    <Viewpoint  description="bird&apos;s eye view"
                orientation="-1.0 0.0 0.0 1.57"
                position="0.0 33.5 0.027"
                jump="false"/>
    <Viewpoint  description="on the level"
```

```
                orientation="-1.0 0.0 0.0 0.617"
                position="0.0 19.388 27.32"
                jump="false"/>
    <Viewpoint  description="front"
                DEF="VIEWPOINT_zPos"
                position="-1.8288 1.5 19.848103"/>
    <Viewpoint  description="back"
                DEF="VIEWPOINT_zNeg"
                orientation="0.0 1.0 0.0 3.141592"
                position="-1.8288 1.5 -20.457703"/>

    <NavigationInfo type="EXAMINE"/>
    <NavigationInfo type='"EXAMINE","ANY"'/>

    <PositionInterpolator DEF="POSITION_INTERPOLATOR"
                          key="0.0 0.33 0.67 1.0"
                          keyValue="-5.0 0.0 -3.0, 0.0 0.0 -3.0, 5.0 0.0 -3.0, -5.0 0.0 -3.0, -5.0 0.0 -3.0"/>

    <TimeSensor DEF="TIME_SENSOR" loop="true" cycleInterval="10.0"/>

    <!-- backdrop -->

    <Transform DEF="TRANSFORM_backdrop"
               rotation="1.0 0.0 0.0 -1.57"
               translation="0.0 0.0 0.0">
      <Shape DEF="SHAPE_backdrop">
        <Appearance>
          <Material DEF="MATERIAL_backdrop"
                    diffuseColor="0.0 0.05 0.0"
                    specularColor="0.0 0.05 0.0"
                    emissiveColor="0.0 0.05 0.0"/>
        </Appearance>

        <Box size="24.0 24.0 0.25"/>
      </Shape>

      <TouchSensor DEF="TOUCH_SENSOR_backdrop" enabled="false"/>
      <PlaneSensor DEF="PLANE_SENSOR_backdrop" enabled="false"/>
    </Transform>

    <!-- "group" (1) target(s), which is(are) goal(s) -->
    <!--  (see Avoid prototype), and (2) keyword(s).  -->

    <Group>
      <!-- target 1 -->

      <Transform rotation="1.0 0.0 0.0 -1.570796">
        <Transform DEF="TRANSFORM_target1" translation="-7.5 -5.0 1.0">

          <!-- appearance and geometry of target -->
```

```
<Shape DEF="SHAPE_target1">
  <Appearance>
    <Material DEF="MATERIAL_target"
              diffuseColor="1.0 0.0 0.0"
              transparency="0.3"/>
  </Appearance>

  <Sphere DEF="SPHERE_targetRadius"
          radius="0.75"/>
</Shape>

<!-- toggle text of target on the geometric shape -->

<Switch DEF="SWITCH_target1">
  <Transform translation="-1.0 -0.5 -0.5">

    <!-- appearance and text of target -->

    <Shape>
      <Appearance>
        <Material USE="MATERIAL_target"/>
      </Appearance>

      <Text DEF="TEXT_target1"
            containerField="geometry"
            maxExtent="0.0"
            string='"Mahdi" "Army"'>
        <FontStyle DEF="FONT_STYLE_target"
                   containerField="fontStyle"
                   family='"SERIF"'
                   horizontal="true"
                   justify='"BEGIN"'
                   leftToRight="true"
                   size="1.0"
                   spacing="1.0"
                   style="PLAIN"
                   topToBottom="true"/>
      </Text>
    </Shape>
  </Transform>
</Switch>

<!-- sensors and interpolators for animation -->

<TimeSensor  DEF="TIME_SENSOR_target1" loop="true" cycleInterval="10.0"/>

<TouchSensor DEF="TOUCH_SENSOR_target1" enabled="true"/>

<PlaneSensor DEF="PLANE_SENSOR_target1"
             description="click to drag in plane"
             autoOffset="true" offset="-7.5 -5.0 1.0"/>
    <ColorInterpolator DEF="COLOR_INTERPOLATOR_target1"
                       key="0.0 0.33 0.67 1.0"
                       keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>
  </Transform>
</Transform>

<!-- target 2 -->

<Transform rotation="1.0 0.0 0.0 -1.570796">
  <Transform DEF="TRANSFORM_target2" translation="7.5 5.0 1.0">

    <!-- appearance and geometry of target -->

    <Shape DEF="SHAPE_target2">
      <Appearance>
        <Material USE="MATERIAL_target"/>
      </Appearance>

      <Sphere USE="SPHERE_targetRadius"/>
    </Shape>

    <!-- toggle text of target on the geometric shape -->

    <Switch DEF="SWITCH_target2">
      <Transform translation="-1.0 -0.5 -0.5">

        <!-- appearance and text of target -->

        <Shape>
          <Appearance>
            <Material USE="MATERIAL_target"/>
          </Appearance>

          <Text DEF="TEXT_target2"
                containerField="geometry"
                maxExtent="0.0"
                string='Al-Quaeda'>
            <FontStyle USE="FONT_STYLE_target"/>
          </Text>
        </Shape>
      </Transform>
    </Switch>

    <!-- sensors and interpolators for animation -->

    <TimeSensor  DEF="TIME_SENSOR_target2" loop="true" cycleInterval="10.0"/>

    <TouchSensor DEF="TOUCH_SENSOR_target2" enabled="true"/>

    <PlaneSensor DEF="PLANE_SENSOR_target2"
```

```
                 description="click to drag in plane"
                 autoOffset="true" offset="7.5 5.0 1.0"/>

    <ColorInterpolator DEF="COLOR_INTERPOLATOR_target2"
                       key="0.0 0.33 0.67 1.0"
                       keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>
  </Transform>
</Transform>

<!-- keyword 1 -->

<ProtoInstance DEF="PROTO_INSTANCE_keyword1"
               name="Avoid" containerField="children">
  <fieldValue name="children">
    <Transform translation="-1.0 1.0 -1.0">

      <!-- appearance and geometry of keyword -->

      <Shape DEF="SHAPE_keyword1">
        <Appearance>
          <Material DEF="MATERIAL_keyword"
                    diffuseColor="1.0 0.5 0.5"
                    transparency="0.5"/>
        </Appearance>

        <Sphere DEF="SPHERE_keywordRadius"
                radius="0.5"/>
      </Shape>

      <!-- toggle text of keyword on the geometric shape -->

      <Switch DEF="SWITCH_keyword1">
        <Transform translation="-0.75 0.5 0.0"
                   rotation="0.0 1.0 0.0 -0.5">

          <!-- appearance and geometry of text -->

          <Shape>
            <Appearance>
              <!-- Note that for the transparency attribute,     -->
              <!--   1.0 means transparent, and 0.0 means opaque -->
              <Material DEF="MATERIAL_text_keyword1"
                        ambientIntensity="0.2"
                        containerField="material"
                        diffuseColor="1.0 0.5 0.5"
                        emissiveColor="0.0 0.0 0.0"
                        shininess="0.2"
                        specularColor="0.0 0.0 0.0"
                        transparency="1.0"/>
            </Appearance>
```

```
                  <Text DEF="TEXT_keyword1"
                        containerField="geometry"
                        maxExtent="0.0"
                        string="Al-Sadr">
                    <FontStyle DEF="FONT_STYLE_text_keyword"
                               containerField="fontStyle"
                               family='"SERIF"'
                               horizontal="true"
                               justify='"BEGIN"'
                               leftToRight="true"
                               size="1.0"
                               spacing="1.0"
                               style="PLAIN"
                               topToBottom="true"/>
                  </Text>
                </Shape>
              </Transform>
            </Switch>
          </Transform>
        </fieldValue>

        <!-- recall for TRANSFORM_target1, translation="-7.5 -5.0 1.0" -->

        <!-- RHS:                                  x,   y,   z -->
        <fieldValue name="position"      value="-10.0 -5.0  1.0"/>

        <!-- RHS:                              x + 2.5, z - 1.0, -y -->
        <fieldValue name="goal"          value="-7.5    0.0    5.0"/>

        <fieldValue name="goalPriority"  value="1.0"/>

        <fieldValue name="speed"         value="0.25e1"/>

        <fieldValue name="obstacles">
<!--      <Obstacle USE="PROTO_INSTANCE_Obstacle"/> -->
        </fieldValue>

        <!-- sensors and interpolators for animation -->

        <TimeSensor  DEF="TIME_SENSOR_alpha_keyword1"
                     enabled="true"
                     loop="false"
                     startTime="0.0"
                     cycleInterval="10.0"/>

        <TouchSensor DEF="TOUCH_SENSOR_keyword1" enabled="true"/>

        <ColorInterpolator  DEF="COLOR_INTERPOLATOR_keyword1"
                            key="0.0 0.33 0.67 1.0"
                            keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>
        <ScalarInterpolator DEF="SCALAR_INTERPOLATOR_alpha_keyword1"
```

```
                      key="0.0 0.5 1.0"
                      keyValue="1.0 0.0 1.0"/>
</ProtoInstance>

<TimeSensor  DEF="TIME_SENSOR_keyword1" loop="true" cycleInterval="10.0"/>

<!-- keyword 2 -->

<ProtoInstance DEF="PROTO_INSTANCE_keyword2"
               name="Avoid" containerField="children">
  <fieldValue name="children">
    <Transform translation="-1.0 1.0 1.0">

      <!-- appearance and geometry of keyword -->

      <Shape DEF="SHAPE_keyword2">
        <Appearance>
          <Material USE="MATERIAL_keyword"/>
        </Appearance>

        <Sphere USE="SPHERE_keywordRadius"/>
      </Shape>

      <!-- toggle text of keyword on the geometric shape -->

      <Switch DEF="SWITCH_keyword2">
        <Transform translation="-0.75 0.5 0.0"
                   rotation="0.0 1.0 0.0 -0.5">

          <!-- appearance and geometry of text -->

          <Shape>
            <Appearance>
              <!-- Note that for the transparency attribute,     -->
              <!--   1.0 means transparent, and 0.0 means opaque -->
              <Material DEF="MATERIAL_text_keyword2"
                        ambientIntensity="0.2"
                        containerField="material"
                        diffuseColor="1.0 0.5 0.5"
                        emissiveColor="0.0 0.0 0.0"
                        shininess="0.2"
                        specularColor="0.0 0.0 0.0"
                        transparency="1.0"/>
            </Appearance>

            <Text DEF="TEXT_keyword2"
                  containerField="geometry"
                  maxExtent="0.0"
                  string="Nassiriya">
              <FontStyle USE="FONT_STYLE_text_keyword"/>
            </Text>
```

```
              </Shape>
            </Transform>
          </Switch>
        </Transform>
      </fieldValue>

      <!-- recall for TRANSFORM_target1, translation="-7.5 -5.0 1.0" -->

      <!-- RHS:                                  x,   y,   z -->
      <fieldValue name="position"     value="10.0  1.0  1.0"/>

      <!-- RHS:                                               -->
      <fieldValue name="goal"         value="-7.5  0.0  5.0"/>

      <fieldValue name="goalPriority" value="1.0"/>

      <fieldValue name="speed"        value="0.125e1"/>

      <fieldValue name="obstacles">
  <!--      <Obstacle USE="PROTO_INSTANCE_Obstacle"/> -->
      </fieldValue>

      <!-- sensors and interpolators for animation -->

      <TimeSensor  DEF="TIME_SENSOR_alpha_keyword2"
                   enabled="true"
                   loop="false"
                   startTime="0.0"
                   cycleInterval="10.0"/>

      <TouchSensor DEF="TOUCH_SENSOR_keyword2" enabled="true"/>

      <ColorInterpolator  DEF="COLOR_INTERPOLATOR_keyword2"
                          key="0.0 0.33 0.67 1.0"
                          keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>
      <ScalarInterpolator DEF="SCALAR_INTERPOLATOR_alpha_keyword2"
                          key="0.0 0.5 1.0"
                          keyValue="1.0 0.0 1.0"/>
    </ProtoInstance>

    <TimeSensor  DEF="TIME_SENSOR_keyword2" loop="true" cycleInterval="10.0"/>

    <!-- keyword 3 -->

    <ProtoInstance DEF="PROTO_INSTANCE_keyword3"
                   name="Avoid" containerField="children">
      <fieldValue name="children">
        <Transform translation="1.0 1.0 1.0">

          <!-- appearance and geometry of keyword -->
    <Shape DEF="keyword3">
      <Appearance>
        <Material USE="MATERIAL_keyword"/>
      </Appearance>

      <Sphere USE="SPHERE_keywordRadius"/>
    </Shape>

    <!-- toggle text of keyword on the geometric shape -->

    <Switch DEF="SWITCH_keyword3">
      <Transform translation="-0.75 0.5 0.0"
                 rotation="0.0 1.0 0.0 -0.5">

        <!-- appearance and geometry of text -->

        <Shape>
          <Appearance>
            <!-- Note that for the transparency attribute,     -->
            <!--   1.0 means transparent, and 0.0 means opaque -->
            <Material DEF="MATERIAL_text_keyword3"
                      ambientIntensity="0.2"
                      containerField="material"
                      diffuseColor="1.0 0.5 0.5"
                      emissiveColor="0.0 0.0 0.0"
                      shininess="0.2"
                      specularColor="0.0 0.0 0.0"
                      transparency="1.0"/>
          </Appearance>

          <Text DEF="TEXT_keyword3"
                containerField="geometry"
                maxExtent="0.0"
                string="3 Sep 2007">
            <FontStyle USE="FONT_STYLE_text_keyword"/>
          </Text>
        </Shape>
      </Transform>
    </Switch>
  </Transform>
</fieldValue>

<!-- recall for TRANSFORM_target1, translation="-7.5 -5.0 1.0" -->

<!-- RHS:                                 x,  y,   z -->
<fieldValue name="position"      value="0.0 10.0  0.0"/>

<!-- RHS:                                            -->
<fieldValue name="goal"          value="-7.5 0.0  5.0"/>

<fieldValue name="goalPriority"  value="1.0"/>
```

```
        <fieldValue name="speed"         value="0.05e0"/>

        <fieldValue name="obstacles">
<!--      <Obstacle USE="PROTO_INSTANCE_Obstacle"/> -->
        </fieldValue>

        <!-- sensors and interpolators for animation -->

        <TimeSensor  DEF="TIME_SENSOR_alpha_keyword3"
                     enabled="true"
                     loop="false"
                     startTime="0.0"
                     cycleInterval="10.0"/>

        <TouchSensor DEF="TOUCH_SENSOR_keyword3" enabled="true"/>

        <ColorInterpolator  DEF="COLOR_INTERPOLATOR_keyword3"
                            key="0.0 0.33 0.67 1.0"
                            keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>
        <ScalarInterpolator DEF="SCALAR_INTERPOLATOR_alpha_keyword3"
                            key="0.0 0.5 1.0"
                            keyValue="1.0 0.0 1.0"/>
      </ProtoInstance>

      <TimeSensor  DEF="TIME_SENSOR_keyword3" loop="true" cycleInterval="10.0"/>

      <!-- keyword 4 -->

      <ProtoInstance DEF="PROTO_INSTANCE_keyword4"
                     name="Avoid" containerField="children">
        <fieldValue name="children">
          <Transform translation="-1.0 1.0 -1.0">

            <!-- appearance and geometry of keyword -->

            <Shape DEF="SHAPE_keyword4">
              <Appearance>
                <Material USE="MATERIAL_keyword"/>
              </Appearance>

              <Sphere USE="SPHERE_keywordRadius"/>
            </Shape>

            <!-- toggle text of keyword on the geometric shape -->

            <Switch DEF="SWITCH_keyword4">
              <Transform translation="-0.75 0.5 0.0"
                         rotation="0.0 1.0 0.0 -0.5">

                <!-- appearance and geometry of text -->
                <Shape>
                  <Appearance>
                    <!-- Note that for the transparency attribute,     -->
                    <!--   1.0 means transparent, and 0.0 means opaque -->
                    <Material DEF="MATERIAL_text_keyword4"
                              ambientIntensity="0.2"
                              containerField="material"
                              diffuseColor="1.0 0.5 0.5"
                              emissiveColor="0.0 0.0 0.0"
                              shininess="0.2"
                              specularColor="0.0 0.0 0.0"
                              transparency="1.0"/>
                  </Appearance>

                  <Text DEF="TEXT_keyword4"
                        containerField="geometry"
                        maxExtent="0.0"
                        string="bomb">
                    <FontStyle USE="FONT_STYLE_text_keyword"/>
                  </Text>
                </Shape>
              </Transform>
            </Switch>
          </Transform>
        </fieldValue>

        <!-- recall for TRANSFORM_target1, translation="-7.5 -5.0 1.0" -->

        <!-- RHS:                                  x,   y,   z -->
        <fieldValue name="position"      value="-10.0  0.0 -5.0"/>

        <!-- RHS:                                              -->
        <fieldValue name="goal"          value="  7.5  0.0 -5.0"/>

        <fieldValue name="goalPriority"  value="1.0"/>

        <fieldValue name="speed"         value="0.25e1"/>

        <fieldValue name="obstacles">
<!--      <Obstacle USE="PROTO_INSTANCE_Obstacle"/> -->
        </fieldValue>

        <!-- sensors and interpolators for animation -->

        <TimeSensor  DEF="TIME_SENSOR_alpha_keyword4"
                     enabled="true"
                     loop="false"
                     startTime="0.0"
                     cycleInterval="10.0"/>
```

```
  <TouchSensor DEF="TOUCH_SENSOR_keyword4" enabled="true"/>

  <ColorInterpolator  DEF="COLOR_INTERPOLATOR_keyword4"
                      key="0.0 0.33 0.67 1.0"
                      keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>
  <ScalarInterpolator DEF="SCALAR_INTERPOLATOR_alpha_keyword4"
                      key="0.0 0.5 1.0"
                      keyValue="1.0 0.0 1.0"/>
</ProtoInstance>

<TimeSensor  DEF="TIME_SENSOR_keyword4" loop="true" cycleInterval="10.0"/>

<!-- keyword 5 -->

<ProtoInstance DEF="PROTO_INSTANCE_keyword5"
               name="Avoid" containerField="children">
  <fieldValue name="children">
    <Transform translation="-1.0 1.0 1.0">

      <!-- appearance and geometry of keyword -->

      <Shape DEF="SHAPE_keyword5">
        <Appearance>
          <Material USE="MATERIAL_keyword"/>
        </Appearance>

        <Sphere USE="SPHERE_keywordRadius"/>
      </Shape>

      <!-- toggle text of keyword on the geometric shape -->

      <Switch DEF="SWITCH_keyword5">
        <Transform translation="-0.75 0.5 0.0"
                   rotation="0.0 1.0 0.0 -0.5">

          <!-- appearance and geometry of text -->

          <Shape>
            <Appearance>
              <!-- Note that for the transparency attribute,     -->
              <!--   1.0 means transparent, and 0.0 means opaque -->
              <Material DEF="MATERIAL_text_keyword5"
                        ambientIntensity="0.2"
                        containerField="material"
                        diffuseColor="1.0 0.5 0.5"
                        emissiveColor="0.0 0.0 0.0"
                        shininess="0.2"
                        specularColor="0.0 0.0 0.0"
                        transparency="1.0"/>
            </Appearance>
```

```
                  <Text DEF="TEXT_keyword5"
                        containerField="geometry"
                        maxExtent="0.0"
                        string="Baghdad">
                    <FontStyle USE="FONT_STYLE_text_keyword"/>
                  </Text>
                </Shape>
              </Transform>
            </Switch>
          </Transform>
        </fieldValue>

        <!-- recall for TRANSFORM_target1, translation="-7.5 -5.0 1.0" -->

        <!-- RHS:                                       x,   y,    z -->
        <fieldValue name="position"      value="10.0  0.0 -5.0"/>

        <!-- RHS:                                                    -->
        <fieldValue name="goal"          value="7.5   0.0 -5.0"/>

        <fieldValue name="goalPriority"  value="1.0"/>

        <fieldValue name="speed"         value="0.125e1"/>

        <fieldValue name="obstacles">
<!--      <Obstacle USE="PROTO_INSTANCE_Obstacle"/> -->
        </fieldValue>

        <!-- sensors and interpolators for animation -->

        <TimeSensor  DEF="TIME_SENSOR_alpha_keyword5"
                     enabled="true"
                     loop="false"
                     startTime="0.0"
                     cycleInterval="10.0"/>

        <TouchSensor DEF="TOUCH_SENSOR_keyword5" enabled="true"/>

        <ColorInterpolator  DEF="COLOR_INTERPOLATOR_keyword5"
                            key="0.0 0.33 0.67 1.0"
                            keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>
        <ScalarInterpolator DEF="SCALAR_INTERPOLATOR_alpha_keyword5"
                            key="0.0 0.5 1.0"
                            keyValue="1.0 0.0 1.0"/>
      </ProtoInstance>

      <TimeSensor  DEF="TIME_SENSOR_keyword5" loop="true" cycleInterval="10.0"/>

      <!-- keyword 6 -->

      <ProtoInstance DEF="PROTO_INSTANCE_keyword6"
```

```
             name="Avoid" containerField="children">
<fieldValue name="children">
  <Transform translation="1.0 1.0 -1.0">

    <!-- appearance and geometry of keyword -->

    <Shape DEF="SHAPE_keyword6">
      <Appearance>
        <Material USE="MATERIAL_keyword"/>
      </Appearance>

      <Sphere USE="SPHERE_keywordRadius"/>
    </Shape>

    <!-- toggle text of keyword on the geometric shape -->

    <Switch DEF="SWITCH_keyword6">
      <Transform translation="-0.75 0.5 0.0"
                 rotation="0.0 1.0 0.0 -0.5">

        <!-- appearance and geometry of text -->

        <Shape>
          <Appearance>
            <!-- Note that for the transparency attribute,    -->
            <!--   1.0 means transparent, and 0.0 means opaque -->
            <Material DEF="MATERIAL_text_keyword6"
                      ambientIntensity="0.2"
                      containerField="material"
                      diffuseColor="1.0 0.5 0.5"
                      emissiveColor="0.0 0.0 0.0"
                      shininess="0.2"
                      specularColor="0.0 0.0 0.0"
                      transparency="1.0"/>
          </Appearance>

          <Text DEF="TEXT_keyword6"
                containerField="geometry"
                maxExtent="0.0"
                string="Kadhimiya">
            <FontStyle USE="FONT_STYLE_text_keyword"/>
          </Text>
        </Shape>
      </Transform>
    </Switch>
  </Transform>
</fieldValue>

<!-- recall for TRANSFORM_target1, translation="-7.5 -5.0 1.0" -->

<!-- RHS:                                x,    y,    z -->
```

```
        <fieldValue name="position"      value="0.0 -10.0  0.0"/>

        <!-- RHS:                                             -->
        <fieldValue name="goal"          value="7.5   0.0 -5.0"/>

        <fieldValue name="goalPriority"  value="1.0"/>

        <fieldValue name="speed"         value="1.0e0"/>

        <fieldValue name="obstacles">
<!--      <Obstacle USE="PROTO_INSTANCE_Obstacle"/> -->
        </fieldValue>

        <!-- sensors and interpolators for animation -->

        <TimeSensor  DEF="TIME_SENSOR_alpha_keyword6"
                     enabled="true"
                     loop="false"
                     startTime="0.0"
                     cycleInterval="10.0"/>

        <TouchSensor DEF="TOUCH_SENSOR_keyword6" enabled="true"/>

        <ColorInterpolator  DEF="COLOR_INTERPOLATOR_keyword6"
                            key="0.0 0.33 0.67 1.0"
                            keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>
        <ScalarInterpolator DEF="SCALAR_INTERPOLATOR_alpha_keyword6"
                            key="0.0 0.5 1.0"
                            keyValue="1.0 0.0 1.0"/>
      </ProtoInstance>

      <TimeSensor  DEF="TIME_SENSOR_keyword6" loop="true" cycleInterval="10.0"/>
    </Group>

    <Script DEF="SCRIPT_target1">
      <field name="set_id"       type="SFInt32" accessType="inputOnly"/>
      <field name="id"           type="SFInt32" accessType="outputOnly"/>
      <field name="offset"       type="SFVec3f" accessType="initializeOnly"/>
      <field name="setPos"       type="SFVec3f" accessType="inputOnly"/>
      <field name="goalChanged"  type="SFVec3f" accessType="outputOnly"/>

      <field name="g"            type="SFNode"  accessType="initializeOnly">
        <Transform USE="TRANSFORM_target1"/>
      </field>

      <field name="p"            type="SFNode"  accessType="initializeOnly">
        <PlaneSensor USE="PLANE_SENSOR_backdrop"/>
      </field>

      <![CDATA[ecmascript:
                          function set_id()
```

```
                           {
                             id = 1;

                           }

                           function setPos(pos)
                           {
                             var mv = offset.add(pos);

                             goalChanged = new SFVec3f(mv.x,0.0,-mv.y);
                           }

                           function initialize()
                           {
                             p.setOffset = g.translation;

                             setPos(g.translation);
                           }
              ]]>
    </Script>

    <Script DEF="SCRIPT_target2">
      <field name="set_id"       type="SFInt32" accessType="inputOnly"/>
      <field name="id"           type="SFInt32" accessType="outputOnly"/>
      <field name="offset"       type="SFVec3f" accessType="initializeOnly"/>
      <field name="setPos"       type="SFVec3f" accessType="inputOnly"/>
      <field name="goalChanged"  type="SFVec3f" accessType="outputOnly"/>

      <field name="g"            type="SFNode"  accessType="initializeOnly">
        <Transform USE="TRANSFORM_target2"/>
      </field>

      <field name="p"            type="SFNode"  accessType="initializeOnly">
        <PlaneSensor USE="PLANE_SENSOR_backdrop"/>
      </field>

      <![CDATA[ecmascript:
                           function set_id()
                           {
                             id = 1;

                           }

                           function setPos(pos)
                           {
                             var mv = offset.add(pos);

                             goalChanged = new SFVec3f(mv.x,0.0,-mv.y);
                           }

                           function initialize()
```

```
                       {
                         p.setOffset = g.translation;

                         setPos(g.translation);
                       }
          ]]>
</Script>

<Script DEF="SCRIPT_touchSwitch_target1">
  <field name="touch_target" type="SFTime"  accessType="inputOnly"/>
  <field name="onOff_target" type="SFInt32" accessType="outputOnly"/>

  <![CDATA[ecmascript:
                       var active_target = true;

                       function touch_target()
                       {
                         if ( active_target ) {
                           onOff_target = 1;
                           active_target = false;
                         }
                         else {
                           onOff_target = 0;
                           active_target = true;
                         }
                       }
          ]]>
</Script>

<Script DEF="SCRIPT_touchSwitch_target2">
  <field name="touch_target" type="SFTime"  accessType="inputOnly"/>
  <field name="onOff_target" type="SFInt32" accessType="outputOnly"/>

  <![CDATA[ecmascript:
                       var active_target = true;

                       function touch_target()
                       {
                         if ( active_target ) {
                           onOff_target = 1;
                           active_target = false;
                         }
                         else {
                           onOff_target = 0;
                           active_target = true;
                         }
                       }
          ]]>
</Script>

<Script DEF="SCRIPT_touchSwitch_keyword1">
```

```
  <field name="touch_keyword"  type="SFTime"  accessType="inputOnly"/>
  <field name="onOff_keyword"  type="SFInt32" accessType="outputOnly"/>

  <![CDATA[ecmascript:
                      var active_keyword = true;

                      function touch_keyword()
                      {
                        onOff_keyword = 0;
                      }
          ]]>
</Script>

<Script DEF="SCRIPT_touchSwitch_keyword2">
  <field name="touch_keyword"  type="SFTime"  accessType="inputOnly"/>
  <field name="onOff_keyword"  type="SFInt32" accessType="outputOnly"/>

  <![CDATA[ecmascript:
                      var active_keyword = true;

                      function touch_keyword()
                      {
                        onOff_keyword = 0;
                      }
          ]]>
</Script>

<Script DEF="SCRIPT_touchSwitch_keyword3">
  <field name="touch_keyword"  type="SFTime"  accessType="inputOnly"/>
  <field name="onOff_keyword"  type="SFInt32" accessType="outputOnly"/>

  <![CDATA[ecmascript:
                      var active_keyword = true;

                      function touch_keyword()
                      {
                        onOff_keyword = 0;
                      }
          ]]>
</Script>

<Script DEF="SCRIPT_touchSwitch_keyword4">
  <field name="touch_keyword"  type="SFTime"  accessType="inputOnly"/>
  <field name="onOff_keyword"  type="SFInt32" accessType="outputOnly"/>

  <![CDATA[ecmascript:
                      var active_keyword = true;

                      function touch_keyword()
                      {
                        onOff_keyword = 0;
```

```
                       }
          ]]>
    </Script>

    <Script DEF="SCRIPT_touchSwitch_keyword5">
      <field name="touch_keyword"  type="SFTime"  accessType="inputOnly"/>
      <field name="onOff_keyword"  type="SFInt32" accessType="outputOnly"/>

      <![CDATA[ecmascript:
                      var active_keyword = true;

                      function touch_keyword()
                      {
                        onOff_keyword = 0;
                      }
          ]]>
    </Script>

    <Script DEF="SCRIPT_touchSwitch_keyword6">
      <field name="touch_keyword"  type="SFTime"  accessType="inputOnly"/>
      <field name="onOff_keyword"  type="SFInt32" accessType="outputOnly"/>

      <![CDATA[ecmascript:
                      var active_keyword = true;

                      function touch_keyword()
                      {
                        onOff_keyword = 0;
                      }
          ]]>
    </Script>

    <Script DEF="SCRIPT_targetSelect" url="TargetSelect.class">
      <field name="time"     type="SFTime"   accessType="inputOnly"/>
      <field name="location" type="SFVec3f"  accessType="outputOnly"/>
    </Script>

    <!-- animation -->
                                                   <!-- target 1    -->

    <ROUTE fromNode="PLANE_SENSOR_backdrop"        fromField="translation_changed"
           toNode=  "TRANSFORM_target1"            toField="translation"/>

    <ROUTE fromNode="PLANE_SENSOR_target1"         fromField="translation_changed"
           toNode=  "TRANSFORM_target1"            toField="set_translation"/>

    <ROUTE fromNode="PLANE_SENSOR_target1"         fromField="translation_changed"
           toNode=  "SCRIPT_target1"               toField="setPos"/>

    <ROUTE fromNode="TOUCH_SENSOR_target1"         fromField="touchTime"
           toNode=  "SCRIPT_touchSwitch_target1"   toField="touch_target"/>
```

```
<ROUTE fromNode="SCRIPT_touchSwitch_target1"  fromField="onOff_target"
       toNode=  "SWITCH_target1"              toField="whichChoice"/>

                                              <!-- target 2    -->

<ROUTE fromNode="PLANE_SENSOR_backdrop"       fromField="translation_changed"
       toNode=  "TRANSFORM_target2"           toField="translation"/>

<ROUTE fromNode="PLANE_SENSOR_target2"        fromField="translation_changed"
       toNode=  "TRANSFORM_target2"           toField="set_translation"/>

<ROUTE fromNode="PLANE_SENSOR_target2"        fromField="translation_changed"
       toNode=  "SCRIPT_target2"              toField="setPos"/>

<ROUTE fromNode="TOUCH_SENSOR_target2"        fromField="touchTime"
       toNode=  "SCRIPT_touchSwitch_target2"  toField="touch_target"/>
<ROUTE fromNode="SCRIPT_touchSwitch_target2"  fromField="onOff_target"
       toNode=  "SWITCH_target2"              toField="whichChoice"/>

                                              <!-- keyword 1 -->

<ROUTE fromNode="SCRIPT_target1"              fromField="goalChanged"
       toNode=  "PROTO_INSTANCE_keyword1"     toField="setGoal"/>

<ROUTE fromNode="TIME_SENSOR_keyword1"        fromField="time"
       toNode=  "PROTO_INSTANCE_keyword1"     toField="updateRequest"/>

<ROUTE fromNode="TOUCH_SENSOR_keyword1"       fromField="touchTime"
       toNode=  "SCRIPT_touchSwitch_keyword1" toField="touch_keyword"/>
<ROUTE fromNode="SCRIPT_touchSwitch_keyword1" fromField="onOff_keyword"
       toNode=  "SWITCH_keyword1"             toField="whichChoice"/>

<ROUTE fromNode="TOUCH_SENSOR_keyword1"       fromField="touchTime"
       toNode=  "TIME_SENSOR_alpha_keyword1"  toField="startTime"/>
<ROUTE fromNode="TIME_SENSOR_alpha_keyword1"  fromField="fraction_changed"
       toNode=  "SCALAR_INTERPOLATOR_alpha_keyword1"
                                              toField="set_fraction"/>
<ROUTE fromNode="SCALAR_INTERPOLATOR_alpha_keyword1"
                                              fromField="value_changed"
       toNode=  "MATERIAL_text_keyword1"      toField="transparency"/>

                                              <!-- keyword 2 -->

<ROUTE fromNode="SCRIPT_target1"              fromField="goalChanged"
       toNode=  "PROTO_INSTANCE_keyword2"     toField="setGoal"/>

<ROUTE fromNode="TIME_SENSOR_keyword2"        fromField="time"
       toNode=  "PROTO_INSTANCE_keyword2"     toField="updateRequest"/>

<ROUTE fromNode="TOUCH_SENSOR_keyword2"       fromField="touchTime"
       toNode=  "SCRIPT_touchSwitch_keyword2" toField="touch_keyword"/>
```

```
<ROUTE fromNode="SCRIPT_touchSwitch_keyword2"  fromField="onOff_keyword"
       toNode=  "SWITCH_keyword2"              toField="whichChoice"/>

<ROUTE fromNode="TOUCH_SENSOR_keyword2"        fromField="touchTime"
       toNode=  "TIME_SENSOR_alpha_keyword2"   toField="startTime"/>
<ROUTE fromNode="TIME_SENSOR_alpha_keyword2"   fromField="fraction_changed"
       toNode=  "SCALAR_INTERPOLATOR_alpha_keyword2"
                                               toField="set_fraction"/>
<ROUTE fromNode="SCALAR_INTERPOLATOR_alpha_keyword2"
                                               fromField="value_changed"
       toNode=  "MATERIAL_text_keyword2"       toField="transparency"/>

                                               <!-- keyword 3 -->

<ROUTE fromNode="SCRIPT_target1"               fromField="goalChanged"
       toNode=  "PROTO_INSTANCE_keyword3"      toField="setGoal"/>

<ROUTE fromNode="TIME_SENSOR_keyword3"         fromField="time"
       toNode=  "PROTO_INSTANCE_keyword3"      toField="updateRequest"/>

<ROUTE fromNode="TOUCH_SENSOR_keyword3"        fromField="touchTime"
       toNode=  "SCRIPT_touchSwitch_keyword3"  toField="touch_keyword"/>
<ROUTE fromNode="SCRIPT_touchSwitch_keyword3"  fromField="onOff_keyword"
       toNode=  "SWITCH_keyword3"              toField="whichChoice"/>

<ROUTE fromNode="TOUCH_SENSOR_keyword3"        fromField="touchTime"
       toNode=  "TIME_SENSOR_alpha_keyword3"   toField="startTime"/>
<ROUTE fromNode="TIME_SENSOR_alpha_keyword3"   fromField="fraction_changed"
       toNode=  "SCALAR_INTERPOLATOR_alpha_keyword3"
                                               toField="set_fraction"/>
<ROUTE fromNode="SCALAR_INTERPOLATOR_alpha_keyword3"
                                               fromField="value_changed"
       toNode=  "MATERIAL_text_keyword3"       toField="transparency"/>

                                               <!-- keyword 4 -->

<ROUTE fromNode="SCRIPT_target2"               fromField="goalChanged"
       toNode=  "PROTO_INSTANCE_keyword4"      toField="setGoal"/>

<ROUTE fromNode="TIME_SENSOR_keyword4"         fromField="time"
       toNode=  "PROTO_INSTANCE_keyword4"      toField="updateRequest"/>

<ROUTE fromNode="TOUCH_SENSOR_keyword4"        fromField="touchTime"
       toNode=  "SCRIPT_touchSwitch_keyword4"  toField="touch_keyword"/>
<ROUTE fromNode="SCRIPT_touchSwitch_keyword4"  fromField="onOff_keyword"
       toNode=  "SWITCH_keyword4"              toField="whichChoice"/>

<ROUTE fromNode="TOUCH_SENSOR_keyword4"        fromField="touchTime"
       toNode=  "TIME_SENSOR_alpha_keyword4"   toField="startTime"/>
<ROUTE fromNode="TIME_SENSOR_alpha_keyword4"   fromField="fraction_changed"
       toNode=  "SCALAR_INTERPOLATOR_alpha_keyword4"
```

```
                                                         toField="set_fraction"/>
      <ROUTE fromNode="SCALAR_INTERPOLATOR_alpha_keyword4"
                                                         fromField="value_changed"
             toNode=  "MATERIAL_text_keyword4"           toField="transparency"/>

                                                         <!-- keyword 5 -->

      <ROUTE fromNode="SCRIPT_target2"                   fromField="goalChanged"
             toNode=  "PROTO_INSTANCE_keyword5"          toField="setGoal"/>

      <ROUTE fromNode="TIME_SENSOR_keyword5"             fromField="time"
             toNode=  "PROTO_INSTANCE_keyword5"          toField="updateRequest"/>

      <ROUTE fromNode="TOUCH_SENSOR_keyword5"            fromField="touchTime"
             toNode=  "SCRIPT_touchSwitch_keyword5"      toField="touch_keyword"/>
      <ROUTE fromNode="SCRIPT_touchSwitch_keyword5"      fromField="onOff_keyword"
             toNode=  "SWITCH_keyword5"                  toField="whichChoice"/>

      <ROUTE fromNode="TOUCH_SENSOR_keyword5"            fromField="touchTime"
             toNode=  "TIME_SENSOR_alpha_keyword5"       toField="startTime"/>
      <ROUTE fromNode="TIME_SENSOR_alpha_keyword5"       fromField="fraction_changed"
             toNode=  "SCALAR_INTERPOLATOR_alpha_keyword5"
                                                         toField="set_fraction"/>
      <ROUTE fromNode="SCALAR_INTERPOLATOR_alpha_keyword5"
                                                         fromField="value_changed"
             toNode=  "MATERIAL_text_keyword5"           toField="transparency"/>

                                                         <!-- keyword 6 -->

      <ROUTE fromNode="SCRIPT_target2"                   fromField="goalChanged"
             toNode=  "PROTO_INSTANCE_keyword6"          toField="setGoal"/>

      <ROUTE fromNode="TIME_SENSOR_keyword6"             fromField="time"
             toNode=  "PROTO_INSTANCE_keyword6"          toField="updateRequest"/>

      <ROUTE fromNode="TOUCH_SENSOR_keyword6"            fromField="touchTime"
             toNode=  "SCRIPT_touchSwitch_keyword6"      toField="touch_keyword"/>
      <ROUTE fromNode="SCRIPT_touchSwitch_keyword6"      fromField="onOff_keyword"
             toNode=  "SWITCH_keyword6"                  toField="whichChoice"/>

      <ROUTE fromNode="TOUCH_SENSOR_keyword6"            fromField="touchTime"
             toNode=  "TIME_SENSOR_alpha_keyword6"       toField="startTime"/>
      <ROUTE fromNode="TIME_SENSOR_alpha_keyword6"       fromField="fraction_changed"
             toNode=  "SCALAR_INTERPOLATOR_alpha_keyword6"
                                                         toField="set_fraction"/>
      <ROUTE fromNode="SCALAR_INTERPOLATOR_alpha_keyword6"
                                                         fromField="value_changed"
             toNode=  "MATERIAL_text_keyword6"           toField="transparency"/>
   </Scene>
</X3D>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN"
                     "http://www.web3d.org/specifications/x3d-3.1.dtd">

<X3D profile="Immersive" version="3.0">
  <head>
    <!-- at each update request, the children node(s) move closer (if chasing)  -->
    <!--  or farther (if not chasing) to goal avoiding obstacles.               -->
    <!--                                                                          -->
    <!-- original by Peter Gerstmann, Ohio State University,                     -->
    <!--  acknowledging Matt Lewis also at OSU.                                  -->
    <!--                                                                          -->
    <!-- This X3D translation by Andrew M. Neiderer, 16 July 2007.               -->
  </head>

  <Scene>
    <ProtoDeclare name="Avoid">
      <ProtoInterface>
        <field name="children"      type="MFNode"        accessType="inputOutput"/>
        <field name="goal"          type="SFVec3f"
                                    value="0.0 0.0 0.0" accessType="initializeOnly"/>
        <field name="goalPriority"  type="SFFloat"
                                    value="2.0"          accessType="initializeOnly"/>
        <field name="obstacles"     type="MFNode"        accessType="initializeOnly"/>
        <field name="position"      type="SFVec3f"
                                    value="0.0 0.0 0.0" accessType="inputOutput"/>
        <field name="setGoal"       type="SFVec3f"       accessType="inputOnly"/>
        <field name="setState"      type="SFString"      accessType="inputOnly"/>
        <field name="setObstacles"  type="MFNode"        accessType="inputOnly"/>
        <field name="speed"         type="SFFloat"
                                    value="1.0"          accessType="initializeOnly"/>
        <field name="state"         type="SFString"
                                    value="CHASING"      accessType="initializeOnly"/>
        <field name="updateRequest" type="SFTime"        accessType="inputOnly"/>
      </ProtoInterface>

      <ProtoBody>
        <Group>
          <Transform DEF="TRANSFORM_mover">
            <IS>
              <connect nodeField="children"    protoField="children"/>
              <connect nodeField="translation" protoField="position"/>
            </IS>
          </Transform>

          <Script DEF="SCRIPT" directOutput="true">
            <field name="direction"    type="SFVec3f"  accessType="initializeOnly"/>
            <field name="first"        type="SFBool"   accessType="initializeOnly"/>
            <field name="goal"         type="SFVec3f"  accessType="initializeOnly"/>
            <field name="goalPriority" type="SFFloat"  accessType="initializeOnly"/>
            <field name="lastBeat"     type="SFTime"   accessType="initializeOnly"/>
```

```
<field name="mv"           type="SFNode"   accessType="initializeOnly">
   <Transform USE="TRANSFORM_mover"/>
</field>
<field name="obstacles"    type="MFNode"   accessType="initializeOnly"/>
<field name="power"        type="SFFloat"  accessType="initializeOnly"/>
<field name="setGoal"      type="SFVec3f"  accessType="inputOnly"/>
<field name="setObstacles" type="MFNode"   accessType="inputOnly"/>
<field name="setState"     type="SFString" accessType="inputOnly"/>
<field name="speed"        type="SFFloat"  accessType="initializeOnly"/>
<field name="state"        type="SFString" accessType="initializeOnly"/>
<field name="update"       type="SFTime"   accessType="inputOnly"/>
<field name="wanderInc"    type="SFFloat"  accessType="initializeOnly"/>

<IS>
  <connect nodeField="goal"          protoField="goal"/>
  <connect nodeField="goalPriority"  protoField="goalPriority"/>
  <connect nodeField="obstacles"     protoField="obstacles"/>
  <connect nodeField="setGoal"       protoField="setGoal"/>
  <connect nodeField="setObstacles"  protoField="setObstacles"/>
  <connect nodeField="speed"         protoField="speed"/>
  <connect nodeField="state"         protoField="state"/>
  <connect nodeField="update"        protoField="updateRequest"/>
</IS>

<![CDATA[ecmascript:
                    function repulseForce(obstacle)
                    {
                      var vec = mv.translation.subtract(obstacle.position);
                      var ods = obstacle.repulsion * 1.0 / Math.pow(vec.length(),power);

                      return (vec.normalize()).multiply(ods);
                    }

                    function calcDirection()
                    {
                      var newDir;

                      if ( state == ’WANDER’ ) {
                        var r = Math.floor(Math.random() * 3.0);

                        // use internal var instead of goal,
                        //  and set it to goal in other states

                        var g = goal;
                        g[r] += wanderInc;
                        setGoal(g);
                      }

                      // compute force of attraction towards goal
                      var gForce = ((goal.subtract(mv.translation)).normalize()).multiply(goalPriority);
```

```
                                  if ( state == 'FLEE' ) {
                                    gForce = gForce.inverse();
                                  }

                                  // compute force of repulsion from obstacles

                                  var rForce = new SFVec3f(0.0,0.0,0.0);

                                  for ( i = 0; i < obstacles.length; i++ ) {
                                    rForce = rForce.add(repulseForce(obstacles[i]));
                                  }

                                  direction = (gForce.add(rForce)).normalize();
                                }

                                function setGoal(val)
                                {
                                  goal = val;
                                }

                                function setObstacles(val)
                                {
                                  obstacles = val;
                                }

                                function setState(str)
                                {
                                  state = str;
                                }

                                function update(val)
                                {
                                  if ( first ) {
                                    first = FALSE;
                                  }
                                  else {
                                    var timeElapsed = val - lastBeat;

                                    calcDirection();
                                    mv.translation = mv.translation.add(direction.multiply(speed * timeElapsed));
                                  }

                                  lastBeat = val;
                                }
                    ]]>
          </Script>
        </Group>
      </ProtoBody>
    </ProtoDeclare>
  </Scene>
</X3D>
```

```
/* ====================================================================== */
/*                                                                        */
/*   File name:  TargetSelect.java                                        */
/*                                                                        */
/*   Purpose:    a Java scene access interface (SAI) to an X3D scene graph */
/*                from a <Script> node of an X3D <Scene> for switching the */
/*                "target" of a "keyword" node field "goal".  At the      */
/*                "elapsedTime" field of TIME_SENSOR_switch the ROUTE of   */
/*                a "keyword" changes its "target".                       */
/*                                                                        */
/*   By:         Andrew M. Neiderer, 25-September-2007.                   */
/* ====================================================================== */

import java.util.Map;

import org.web3d.x3d.sai.*;

public class TargetSelect
 implements X3DScriptImplementation {

  // member data

  private Browser   browser;
  private Map       fields;
  private MFNode    children;
  private X3DScene  scene;
  private X3DNode[] nodes;

  // member functions

  public void setBrowser(Browser browser)
  {
    System.out.println("setBrowser()");

    this.browser = browser;
  }

  public void setFields(X3DScriptNode externalView, Map fields)
  {
    System.out.println("setFields()");

    this.fields = fields;
  }

  public void initialize()
  {
    System.out.println("initialize()");
  }

  public void shutdown()
  {
    System.out.println("shutdown()");
  }

  public void eventsProcessed()
  {
    System.out.println("eventsProcessed()");

    System.out.println("getCurrentFrameRate() = " + browser.getCurrentFrameRate());

    System.out.println("getName() = " + browser.getName());

    browser.println("end eventsProcessed()");
  }
}
```

List of Symbols, Abbreviations, and Acronyms

Ajax	asynchronous JavaScript and XML
API	application programming interface
DAG	directed acyclic graph
DOM	document object model
HTML	hypertext markup language
RHS	right-handed system
SAI	scene access interface
SNA	social network analysis
VE	virtual environment
X3D	extensible three-dimensional (3-D) graphics
XHTML	extensible hypertext markup language
XML	extensible markup language

NO. OF COPIES	ORGANIZATION
1 (PDF ONLY)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FORT BELVOIR VA 22060-6218
1	US ARMY RSRCH DEV & ENGRG CMD SYSTEMS OF SYSTEMS INTEGRATION AMSRD SS T 6000 6TH ST STE 100 FORT BELVOIR VA 22060-5608
1	DIRECTOR US ARMY RESEARCH LAB IMNE ALC IMS 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB AMSRD ARL CI OK TL 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB AMSRD ARL CI OK T 2800 POWDER MILL RD ADELPHI MD 20783-1197
	ABERDEEN PROVING GROUND
1	DIR USARL AMSRD ARL CI OK TP (BLDG 4600)

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
7	DIR USARL AMSRD ARL CI CT B BROOME J O’MAY A NEIDERER (4 CPS) M THOMAS

INTENTIONALLY LEFT BLANK.