CREATION OF 3D MODELS WITH VIRTUAL REALITY MODELING LANGUAGE

Curilă Mircea [*], Curilă Diana [**], Curilă Corina [***]

[*] University of Oradea, 5 Armatei Române St., Oradea, Romania, e-mail: mirceacurila@yahoo.com
[**] Secondary school "Dacia" from Oradea, 25 Dacia St., Oradea, Romania
[***] National College "Onisifor Ghibu" from Oradea, 3 Onisifor Ghibu St., Oradea, Romania.

Abstract

Static and dynamic descriptions of 3D models, multimedia content and a variety of hyperlinks can be represented with VRML files. The information scenes in the VRML file are organized in a scene graph. It contains nodes that describe objects and their properties, organized as a tree family. Interpretation, execution and presentation of VRML files are performed by the browser, which displays shapes and sounds from the scene graph. The 3D model is constructed using multiple Shape nodes with fields to describe the geometry and appearance of the shape.

Key words: 3D models, virtual reality, VRML, programming language

INTRODUCTION

A professional computer graphics community has expanded the concepts of 3D models to include their interactivity. Static and dynamic descriptions of 3D models, multimedia content and a variety of hyperlinks can be represented with VRML files, with the extension * .wrl, where wrl is the abbreviation for world. VRML initials stand for Virtual Reality Modeling Language. As a text language, VRML allows the rapid construction of the virtual world containing 3D shapes, light sources, fog, animation and even sound effects.

A VRML browser is required for the virtual world described by VRML files to appear. A VRML browser reads files, interprets their syntax, builds a 3D virtual world, and then draws it in a graphical window. Both VRML browsers and competent file creation tools are widely available.

MATERIAL AND METHOD

VRML file structure

A VRML file consists of the following components: header, scene graph, prototypes, and event routing.

The information scenes in the VRML file are organized in a scene graph. It contains nodes that describe objects and their properties, organized as a tree family. Parent scene graphs lead groups of children, which may

include other parents. When groups contain groups and they in turn contain groups, the scene graph grows into a hierarchically grouped tree. Children can be members of more than one group, creating a complex of hierarchical scenes called graphs.

Each of the children and parents in the scene graph are VRML nodes, which create the virtual world. VRML nodes position user observation points, set the background, build 3D shapes, group shapes together, define animation paths, track light sources, add ambient fog, position sounds, and so on.

The coordinate system in which the root nodes are displayed is called the virtual world coordinate system (the default coordinate system). VRML includes the notion of local coordinate systems, defined relative to ancestral coordinate systems, using Transform nodes.

Interpretation, execution and presentation of VRML files are performed by the browser, which displays shapes and sounds from the scene graph. This presentation is the virtual world, which can be navigated through the browser by the user.

Construction of forms

In VRML a shape is constructed using the Shape node with fields to describe the geometry and appearance of the shape. Geometry defines the structure of the shape, and the appearance determines the color and transparency of the shape.

Defining the geometry of the shape

A virtual world is constructed using a lot of Shape nodes, which have the geometry field completed with a geometric node to define the surfaces to be drawn. For this purpose, the following predefined geometric nodes are used, associated with some fundamental geometric shapes: Box, Cylinder, Cone, Sphere.

Defining the appearance of shapes

To determine how the shape appears, the appearance field of the Shape node is completed with an Appearance node. This node has a material field that is completed with a Material node to describe the color and transparency of the shape. Material node fields define the color of the shape, including the diffusion, emissivity, brightness, and transparency of the shape.

The shape can be covered with a texture, completing the texture field of the Appearance node with one of three types of texture nodes: ImageTexture, MovieTexture, or PixelTexture. An ImageTexture node is used to load a textured image from a JPEG, PNG, or GIF file. The

MovieTexture node is used to scroll an MPEG movie file over the 3D form. You can include a small texture directly from the VRML file using a PixelTexture node. In each case, a texture determines the color of the shape, ignoring the diffuseColor field in the Material node of the shape.

Transforming groups of shapes

The shapes corresponding to the root nodes in the scene graph are implicitly constructed in the coordinate system of the virtual world. The Transform node creates a new coordinate system that can be translated, rotated, and / or scaled relative to its parent in the scene graph. The shapes defined in the children field of the Transform node are built in the new coordinate system, allowing the establishment of the position, orientation and resizing of these objects.

Setting the background

The background of the VRML world is by default black. With the help of the Background node you can define particular backgrounds. This node has fields that allow the creation of the color gradient along an infinite radius spherical sky and along an infinite radius hemispherical ground located inside the sky sphere, at the bottom to the equator, which surrounds the virtual world. The skyColor and groundColor fields define the concentric circles of color located on the sky sphere and the ground hemisphere, respectively, at the angles to the vertical defined in the skyAngle and groundAngle fields. Between the portions between these angles there is a linear interpolation of the colors on the color circles that delimit them.

3D model created with VRML

According to the notions described about the Virtual Reality Modeling Language, we built a 3D model, which represents the airplane shown in figure 1, using this programming language. For this purpose, Cylinder-type nodes are mainly used to obtain a better definition of the shapes of the component elements of which it is composed.

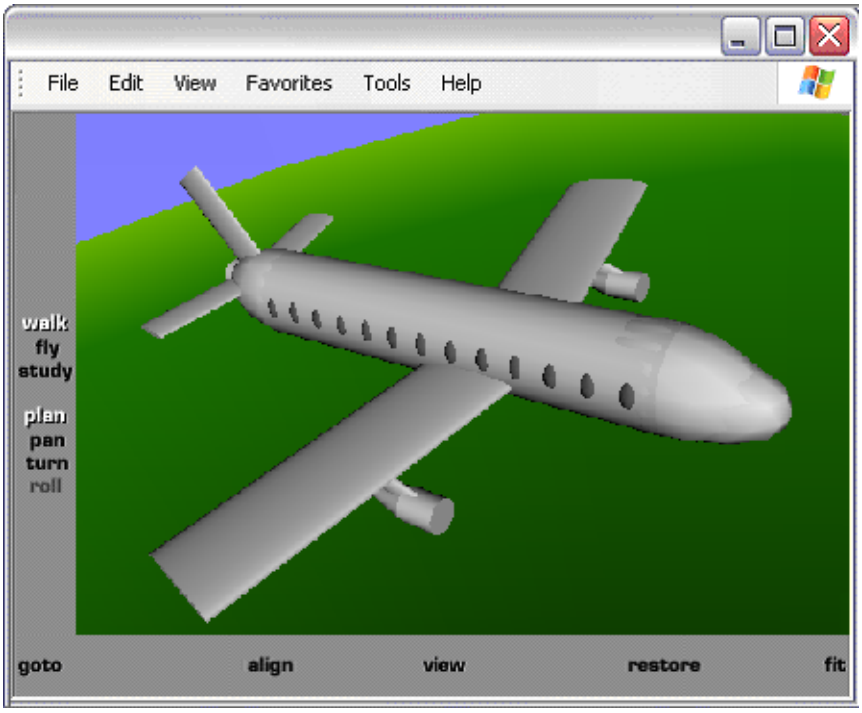

Figure 1. Airplane

The tube body is made of a cylinder without bases, turned horizontally, as in Figure 2.

Fig. 2. The body of the airplane

The cockpit, front, and rear of the airplane are made of ellipsoids placed in front and rear, respectively, so as to cover the ends of the tube. The ellipsoids from which the cockpit is made and the rear of the airplane are inclined, and the one from which the front is obtained is smaller, not inclined and protrudes slightly outwards, as seen in figure 3.

Fig. 3. The rear, cockpit and front

The tail and wings of the airplane, shown in Figure 4, are obtained from cylinders strongly compressed in the direction perpendicular to their plane, inclined by corresponding rotations and overturned horizontally in the case of the wings. The exhaust pipe is made of a cylinder overturned horizontally and placed behind the body of the airplane.

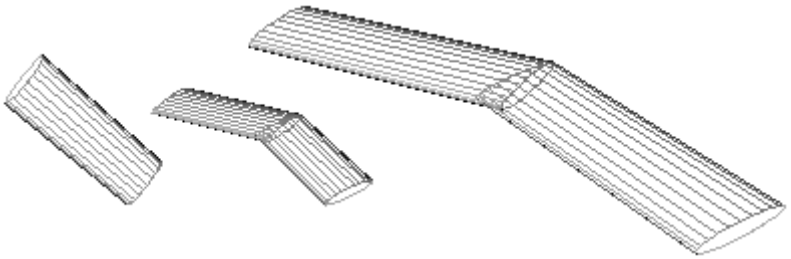

Fig. 4. The tail and wings

The engine is constructed of a cone and a cylinder turned horizontally and placed in the rear or front, respectively, according to figure 5. It is fixed under the wing by a support obtained from a sphere flattened by compression and inclined by rotation.

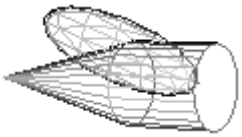

Fig. 5. The engine

The elliptical windows, shown in Figure 6, are made of flattened spheres by strong compression in the direction perpendicular to their plane.

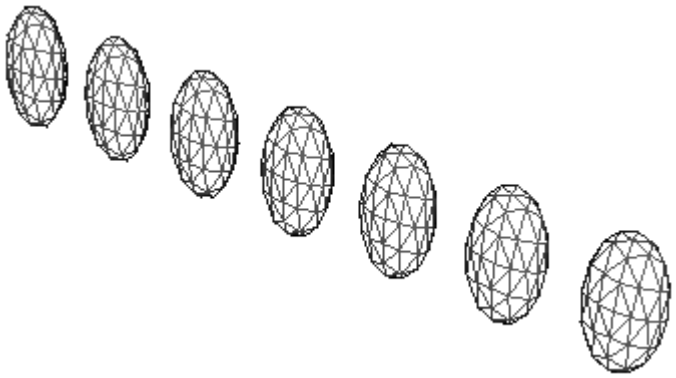

Fig. 6. The windows

The source code that creates this 3D model is:

```
#VRML V2.0 utf8

Viewpoint
 {
 position 40 35 40
 orientation -1 1 0 .95
 }
Background
 {
 skyColor
  [
  0 0 0,
  0 0 .9,
  .5 .5 1
  ]
 skyAngle
  [
  1.2, 1.45
  ]
 groundColor
  [
  0 0 0,
  .1 .45 0,
  .4 .7 0
  ]
 groundAngle  [
  1.15, 1.26
  ]
 }

Transform
 {
 rotation 0 0 1 1.571
 children
  [
  Shape
   {
   appearance DEF color Appearance
            {
            material Material
                {
                diffuseColor .9 .9 .9
                }
            }
   geometry Cylinder
                {
                bottom FALSE
                height 40
                radius 4
                top FALSE
                }
   }
  ]
 }

Transform
 {
 translation 19 0 0
 children
  [
  DEF cockpit Transform
   {
   rotation 0 0 1 -.13
   scale 1.5 .81 .81
   children
    [
     DEF sphere Shape
      {
      appearance USE color
      geometry Sphere
                {
                radius 5
                }
      }
    ]
   }
  ]
 }
Transform
 {
 translation -19.9 0 0
 children
  [
   USE cockpit
  ]
 }
Transform
 {
 translation 21.5 -.75 0
 rotation 0 0 1 .13
 scale .8 .8 .8
 children
  [
   USE cockpit
  ]
 }

DEF wings Transform
 {
 children
  [
   Transform
    {
    translation 2 -1 14
    rotation 1 0 0 1.571
    children
     [
      DEF wing Transform
       {
       rotation 0 0 1 .2
       scale 1 1 .17
       children
        [
         DEF cylinder Shape
          {
          appearance USE color
          geometry Cylinder
                    {
                    height 30
                    radius 4.3
                    }
          }
        ]
       }
     ]
    }
   Transform
    {
    translation 2 -1 -14
    rotation 1 0 0 -1.571
    children
     [
      USE wing
     ]
    }
  ]
 }
Transform
 {
 translation -26.1 2 0
 scale .5 .5 .4
 children
  [
   USE wings
  ]
 }

Transform
 {
 translation -25.2 5 0
 rotation 0 0 1 .5
 scale .5 .5 .1
 children
  [
   USE cylinder
  ]
 }

Transform
 {
 translation -25.85 .8 0
 rotation 0 0 1 1.571
 scale .4 .1 .4
 children
  [
   USE cylinder
  ]
 }

Transform
 {
 translation 6 -2.4 15
 children
  [
   DEF engine Transform
    {
    children
     [
      Transform
       {
       rotation 0 0 1 -.3
       scale .6 .2 .08
       children
        [
         USE sphere
        ]
       }
      Transform
       {
       translation -2 -1 0
       rotation 0 0 1 1.571
       children
        [
         Shape
          {
          appearance USE color
          geometry Cone
                    {
                    bottomRadius 1.28
                    height 5
                    }
          }
        ]
       }
      Transform
       {
       translation 0 -4 0
       scale .3 .1 .3
       children
        [
         USE cylinder
        ]
       }
     ]
    }
  ]
 }
Transform
 {
 translation 6 -2.4 -15
 children
  [
   USE engine
  ]
 }

Transform
 {
 translation 0 1 3.7
 rotation 1 0 0 -.26
 children
  [
   DEF windows Transform
    {
    children
     [
      DEF window Transform
       {
       scale .6 1 .3
       children
        [
         Shape
          {
          appearance Appearance
           {
            material Material
                {
                diffuseColor .4 .4 .4
                }
           }
          geometry Sphere
                    {
                    radius 1
                    }
          }
        ]
       }
     ]
    }
   Transform
    {
    translation -3 0 0
    children
     [
      USE window
      Transform
       {
       translation -3 0 0
       children
        [
         USE window
         Transform
          {
          translation -3 0 0
          children
           [
            USE window
            Transform
             {
             translation -3 0 0
             children
              [
               USE window
               Transform
                {
                translation -3 0 0
                children
                 [
                  USE window
                 ]
                }
              ]
             }
           ]
          }
        ]
       }
     ]
    }
  ]
 }
Transform
 {
 translation 3 0 0
 children
  [
   USE window
   Transform
    {
    translation 3 0 0
    children
     [
      USE window
      Transform
       {
       translation 3 0 0
       children
        [
         USE window
         Transform
          {
          translation 3 0 0
          children
           [
            USE window
            Transform
             {
             translation 3 0 0
             children
              [
               USE window
               Transform
                {
                translation 3 0 0
                children
                 [
                  USE window
                 ]
                }
              ]
             }
           ]
          }
        ]
       }
     ]
    }
  ]
 }
Transform
 {
 translation 0 1 -3.7
 rotation 1 0 0 .26
 children
  [
   USE windows
  ]
 }
```

RESULTS AND DISCUSSION

At the beginning, it is established how the browser presents the initial image and the background color through the Viewpoint node, respectively Background.

The body of the airplane is created from a cylinder whose bases are not considered and is rotated 90° around the Z axis in order to position it along the X axis. The object "color" defined here, determines the silver color, which will be applied components of the airplane. The definition of the "cockpit" object determines the realization of the cockpit in inclined ellipsoidal shape, obtained by scaling the "sphere" object (defined within it) by 150% on the X axis and by 81% on the Y and Z axes and its rotation around the Z axis. Instantiation of this object, translated to the rear position, creates the rear of the airplane. These two components are constructed so as

to seal the ends of the tube that forms the body of the airplane. The front of the airplane is made by instantiating the object "cockpit", rotated in the opposite direction of rotation within its definition in order to cancel the inclination of the ellipsoid, scaled by 80% on the X, Y and Z axes to reduce its size and translated to the front position so as to it comes out a little.

Defining the "wings" object constructs the right and left wing by the corresponding translation and rotation of the "wing" object definition by 90°, respectively of its instantiation by –90° with respect to the X axis. The "wing" object creates a wing by scaling the object definition "cylinder" by 17% on the Z axis in order to flatten it in the plane of the wing and rotate it around the Z axis to obtain its inclination. The wings of the tail are made by instantiating the object "wings", translated into the rear position and scaled by 50% on the X and Y axes and by 40% on the Z axis to reduce its size. The tail of the airplane is obtained by scaling the instantiation of the "cylinder" object by 50% on the X and Y axes and by 10% on the Z axis to flatten the cylinder in its plane, the rotation around the Z axis to obtain its inclination and its translation at the rear. The exhaust pipe located behind the airplane is created by instantiating the "cylinder" object, translated into the appropriate position, rotated 90° around the Z axis and scaled by 40% on the X and Z axes and by 10% on the Y axis.

The airplane engines are made by defining, respectively instantiating the "engine" object. It rotates the cone by 90° around the Z axis and instantiates the "cylinder" object, translated relative to the cone and scaled by 30% on the X and Z axes and by 10% on the Y axis, in order to build the engine. The support that fixes the engine under the wing of the airplane is obtained by scaling the instantiation of the object "sphere" by 60% on the X axis, 20% on the Y axis and 8% on the Z axis and its rotation around the Z axis to obtain its inclination.

Defining and instantiating the "windows" object determines the creation of the windows on the right side, respectively on the left one. Their rotation around the X-axis allows their location on the surface of the airplane body. The "window" object, defined within the "windows" object, achieves the central window by scaling the sphere by 60% on the X axis and by 30% on the Z axis. The six windows located towards the back, respectively towards the front, are obtained by instantiating the "window" object, translated into the corresponding positions relative to the previous positions.

CONCLUSIONS

The abundance and importance of 3D models, the permission of interactive 3D display, the simplicity of this programming language, give a wide utility to VRML. It quickly became a standard file format for 3D virtual worlds. 3D models remain the main representation used in industrial design, architecture, geographic information systems and entertainment industries.

REFERENCES

1. Curilă M., 2004, Programarea Realităţii Virtuale, Editura Universităţii din Oradea
2. Jamsa K., P. Schmauder, N. Yee, 1998, Biblioteca programatorului VRML, Editura All Educational
3. ”The Virtual Reality Modeling Language”. International Standard ISO/IEC 14772-1: 1997.
4. Zorin D., P. Schroder, W. Sweldens, 1995, Interactive multiresolution mesh editing, In *Siggraph’97 Conference Proceedings*, pages 259-268, Los Angeles, August, 1995.
5. Apple Computer, Inc. “3D graphics programming with QuickDraw 3D”, Addison Wesley, 1995.