1.0) AIM:

Installation of VRML browser & write a program in VRML to display cylinder.

2.0) SCOPE:

- To understand what VRML is
- To understand VRML file structure and basic syntax.
- To be able to create a new VRML by combining existing ones.
- To be able to install VRML browser & load a VRML file.

3.0) FACILITIES/ APPARATUS:

i) Software: VRML Browser

4.0) THEORY:

Overview of VRML

VRML stands for "**v**irtual **r**eality **m**odeling **l**anguage." Similar to hypertext markup language (HTML), which is composed of predefined tags that define the structure and appearance of documents, VRML has predefined nodes and statements that describe the structure, appearance, and interaction in 3D environments. Although they have a fundamental likeness, VRML goes far beyond a simple markup language.

Technically speaking, VRML is neither a typical virtual reality (VR) language nor a modeling language. Usually virtual reality involves immersive virtual environments rendered on head mounted display (HMD), desktop cathode ray tubes (CTRs), or on a CAVE system. VR systems use three-dimensional (3D) input devices (such as gloves, trackers, or trackballs, or natural input through voice or gesture). The VRML standard only supports mouse interaction and a minimum of geometry-modeling features; there are several extended browsers that support keyboard or other input devices.

Then what is VRML? VRML defines a file format that integrates 3D graphics and multimedia. It is standardized as International Standard ISO/IEC-14772 and can offer an effective way of interchanging 3D data. Many modeling or programming tools, such as 3D Studio Max [Discreet, 2001], WorldToolKit

[Sense8, 1999], and Java 3D [Sowizral et al., 2000], support VRML by importing or exporting VRML files. Sometimes, VRML is referred to as a "3D analog to HTML." This means VRML is a standard way of publishing 3D multimedia data, just as HTML is a standard way of posting multimedia documentation on the Web. VRML supports several multimedia data types such as sound, video files, and animation.

Some VRML Features

VRML has several features that are useful when generating interactive virtual environments on the Internet. First VRML files describe a 3D world based on a scene graph structure. Scene graphs are hierarchical treelike data structures that describe an entire 3D scene, including the geometry representation, the appearance of objects, and the relationships among the objects in the world. Second, VRML supports an event-processing mechanism in the scene. By generating, receiving, or routing events, one can use each node in the graph to transfer information to other models or change the state of the scene. Third, VRML supports user interaction or animation controls through the sensor node. Combining the event mechanism and the sensor nodes, one can easily generate animations or change the environment according to one's actions.

VRML also supports scripts. They enable complex animation and are a powerful extension of VRML. Prototyping mechanisms in VRML allow developers to create a library of reusable custom nodes by combining existing nodes. Furthermore, VRML supports a distributed scene. One can create one VRML world through different files in different locations. This allows multiuser interaction through a single distributed virtual environment. Therefore, the VRML Consortium, which is a community tasked with the creation of an open standard for VRML, defines VRML as "An open standard for 3D multimedia on the Internet." Finally, being an open standard means it is freely available, similar to Java 3D. This was an important criterion when selecting VRML and Java 3D as the languages to be used in this virtual reality technology teaching laboratory.

The VRML Browser

To view VRML files, one needs to use plug-ins for a VRML browser or other application programs that support VRML file loading.

1. Standalone Applications

Standalone applications that can import or export VRML worlds enable us to see the virtual world. Usually they support creating and/or manipulating VRML worlds through C++/Java programming. Horizon (from Open Worlds), Jverge, and Open Inventor (from TGS) are examples of standalone applications for VRML browsing. They support not only standard VRML file loading, but also advanced extensions such as stereoscopic display, network capabilities, or real-time video textures.

2.Plug-ins for Standard Web Browsers

The plug-ins of a VRML browser use the facilities provided by a Web browser to display a VRML world. Plug-ins enable users to see VRML worlds embedded in a Web page. Cortona VRML Client from ParallelGraphics [ParallelGraphics, 2002], Cosmo Player from Computer Associates [CAI, 2002], and WorldView from Intervista are examples of popular VRML plug-ins. The URLs for other web browsers are listed in the Resources section at the end of this manual.

3 VRML Browser Concepts

A VRML browser interprets a VRML file and presents the corresponding 3D virtual world (Fig. 1.3). There are three procedures for showing a VRML world in the browser. First, a VRML parser interprets the VRML file and determines the meaning of its syntax. Parsing generates built-in nodes and user-defined ones. Second, these nodes are combined according to the scene hierarchy in order to construct a scene graph. Other mechanisms such as event processing by routes or script interpretation are processed. The browser also interprets the user's input and changes the viewpoint for navigation of 3D worlds based on sensor nodes in VRML. Finally, it generates audio/visual feedback to the user.

Visualization

VRML is a popular way of visualizing 3D data. If 3D data are posted on a Web page in the VRML format, they can be accessed in any place and can be visualized on any system. This requires, however, that a VRML browser be installed on the local system. For the execution of VRML program download

Cortona. Write the program on notepad or any text editor and save the program with extension .vrml or .wrl

The Basic VRML Syntax

I. The VRML File Structure

A VRML file begins with a **header**. The header is used for easy identification of a VRML file in the browser; if the file lacks the header, the browser cannot identify the file format and cannot read it properly. The header is followed by a number of nodes, fields and field values, and prototyping, scripting and route statements. Each node can be given a name by the ***DEF*** statement and be reused by the ***USE*** statement: Thus the programmer does not need to write the same object several times. The following is the header for VRML version 2.0

#VRML V2.0 utf8

II. The Node Statement

A node is the basic element of a VRML scene graph, as discussed before. A VRML scene is composed of a number of nodes. A node can be compared to an object, or class, in object-oriented programming (OOP) languages like C++ or Java. Objects that have their own properties are generated when a browser parses the VRML file if a user built nodes to construct a VRML scene. A node is composed of any number of field statements, ***IS*** statements, ***ROUTE*** statements, ***PROTO*** statements, or ***EXTERNPROTO*** statements in any order. Each node can have its own name for further reference for reuse or event processing.

[**DEF** <name>] <nodeType> **{**<body>}

III. Data Types

The elementary data type in VRML is the **field**. Nodes are composed of fields. There are about 20 field types in VRML to support various kinds of data. There are different data types for image, color, time, and so on. They can have single or multiple values. The date types are differentiated by a prefix: **SF** - for single valued fields and **MF** - for multiple-valued fields. Each node can also be a field for another node. Therefore, one has **SFNode** and **MFNode** in the field type.

<fieldName> <fieldValue> # Single-valued Node

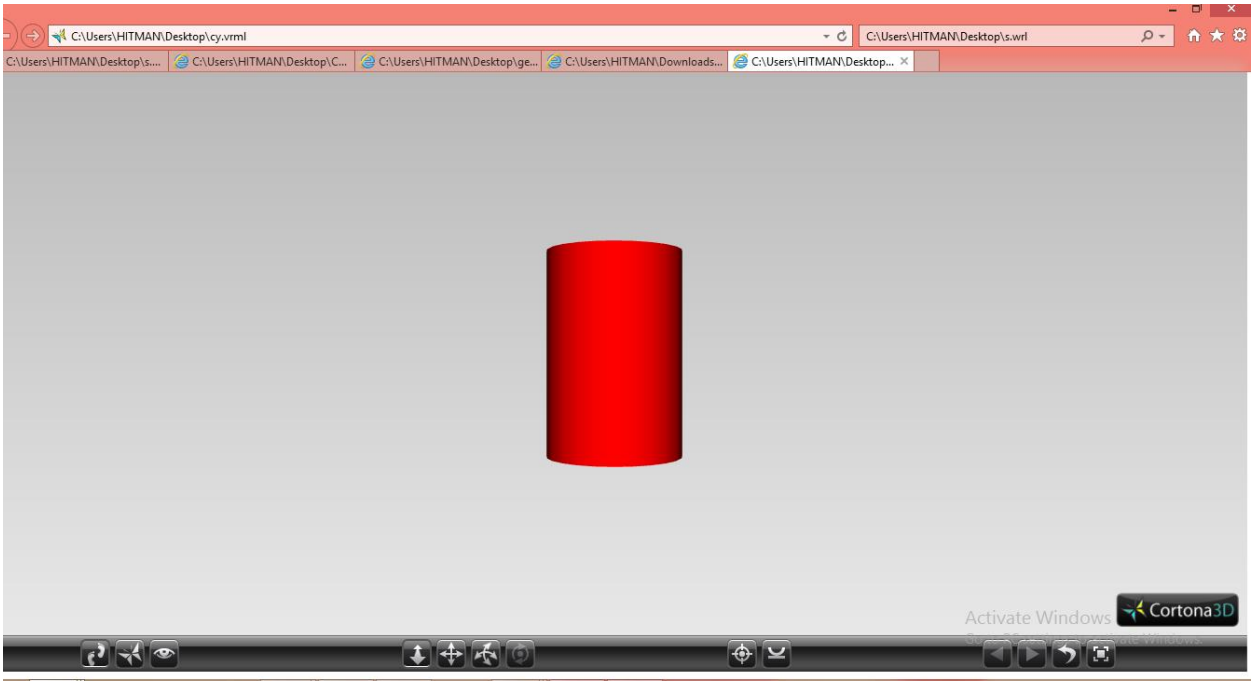
<fieldName> **[** <fieldValues> **]** # Multiple-valued Node

The Primitive Geometry Nodes

VRML provides several primitive geometry nodes for convenience: **Box**, **Cone**, **Cylinder**, and **Sphere**. Each node has fields for specifying parameters and determining whether the surface is visible or not. For example, the **Cylinder** node has five fields. The ***height*** and ***radius*** fields indicate the size of the cylinder and the other three fields indicate whether each face of the shape will be displayed or not. If the ***top*** field is FALSE, the top surface of the cylinder will not appear in the image:

Simple Cylinder in VRML

```
#VRML V2.0 utf8
# This is an example of a very simple
# Shape node that uses the primitive
# Cylinder object for geometry.
Shape {
appearance Appearance {
material Material {
diffuseColor 1 0 0 # R G B (red)
}
}
geometry Cylinder {
      height 3
}
}
```



5.0) Conclusion:

Installed Cortona & executed first VRML program on the browser.