# Chapter 10

# Module 8

## Visualization

✔ In visualization, we are concerned with *exploration*

✔ In computer-graphics, we are concerned with *rendering*

## The use of 3D

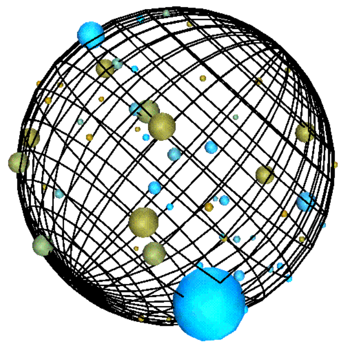
✔ Analog with *real-world* physics.

✔ 10-fold improvement in item density with 3D.

✔ Familiarity with spatial location helps reduce visual clutter.

✔ Need sufficient visual cues.

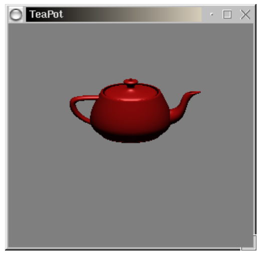
## Use of 3D



## OpenGL

- ✔ SGI in-house graphics system
- ✔ Now a widely accepted graphics standard
- ✔ Standard on UNIX and Windows
- ✔ API supports rendering, buffering, anti-aliasing, shading, colouring, texture-mapping, a display list, Z-buffering...

## OpenGL Application



## OpenGL source

CODE LISTING **teapot.c**

```c
#include <GL/glut.h>

void
Teapot (long grid)
{
    /* ... code to construct drawlist of teapot here. */
}

static void
Init (void)
{
    glEnable (GL_DEPTH_TEST);
    glLightModelfv (GL_LIGHT_MODEL_LOCAL_VIEWER, local_view);
    /* Lighting model, materials... */
}

static void
SpecialKey (int key, int x, int y)
{
    switch (key) {
     case GLUT_KEY_UP:
        rotX -= 20.0;
        glutPostRedisplay ();
        break;
        /* Move in other directions */
    }
}

static void
Draw (void)
{
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPushMatrix ();
    /* ... translations ... */
    glCallList (teaList);
    glPopMatrix ();
    glutSwapBuffers ();
}

int
main (int argc, char **argv)
{
    glutInit (&argc, argv);
    type = GLUT_RGB | GLUT_DEPTH;
    type |= (doubleBuffer) ? GLUT_DOUBLE : GLUT_SINGLE;
    glutInitDisplayMode (type);
    glutInitWindowSize (300, 300);
    glutCreateWindow ("TeaPot");
    Init ();
    glutReshapeFunc (Reshape);
    glutKeyboardFunc (Key);
    glutSpecialFunc (SpecialKey);
    glutDisplayFunc (Draw);
    glutMainLoop ();
}
```

## Java3D & VTK

- ✔ 3D OO toolkits
- ✔ VTK is open source
  - ✔ C++ class library, and
  - ✔ interface layers for Tcl/Tk, Java, and Python.

## Network traffic application

*To help answer questions such as the following:*

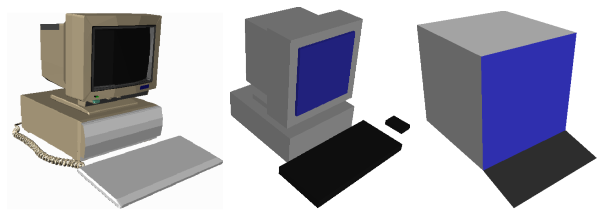
- *Which segments carry the most traffic?*
- *Which sections of the network are down?*
- *At what times, and where do traffic bottlenecks occur?*
- ...

## Application elements

Following elements are represented:

- Background: - to convince the viewer that the display is *three dimensional*...
- Nodes: - a computer, a network device...
- Traffic: - the amount of traffic flow...
- Protocol: - the *type* of traffic...
- ...

## Node representation



## Rendering speed

| Machine | Rendering speed | Computer (a) | Computer (b) | Computer (c) |
|---|---|---|---|---|
| Workstation | 485,000 $\Delta$/sec | 0.485 frames/sec | 11.5 frames/sec | 69 frames/sec |
| PC1 | 30,000 $\Delta$/sec | 0.03 frames/sec | 0.71 frames/sec | 4.3 frames/sec |
| PC2 | 11,000 $\Delta$/sec | 0.011 frames/sec | 0.26 frames/sec | 1.6 frames/sec |

## Levels of Detail

✔ Some representation methods allow different *levels of detail*.

✔ In VRML an object may be represented in different ways depending on how large it is.

✔ If the object is near you, it could be represented in detail, but if it is a long way away, the representation could be as simple as a coloured square.

## LOD

```
LOD {
    range [20]
    level [
      Shape{                    #full detail 16 sided cone
        appearance Appearance { material Material { ... } }
        geometry Extrusion{  ... }.
      }
      Shape{                    #low detail 4 sided cone
        appearance Appearance { material Material { ... } }
        geometry Extrusion{ ... }
      }
    ]
}
```

## LOD

✔ If the distance from the user to the object is smaller than the first range value specified, then the first version is drawn.

✔ If the distance is greater than the last range specified, the last version is drawn.

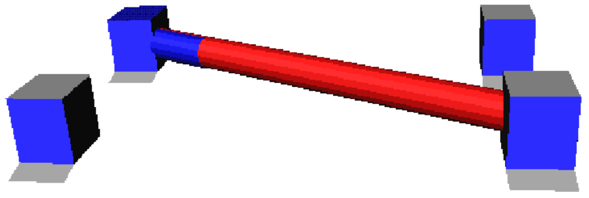
## Traffic and protocols

Draw a line between nodes.

A line indicates source and destination, but not the *amount* of traffic:

1. Colour coding (black through red to white for maximum traffic),
2. Line width, and
3. The length of partial lines, as discussed in Eick's papers.

## Partial lengths



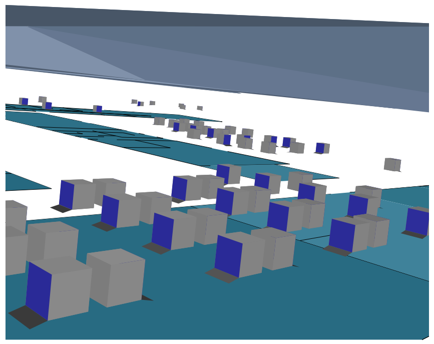
## Trend representation

✔ Graphing

✔ 4D visualization methods

✔ Encode previous *on-top-of* the current - *visual* echoes.

## Display



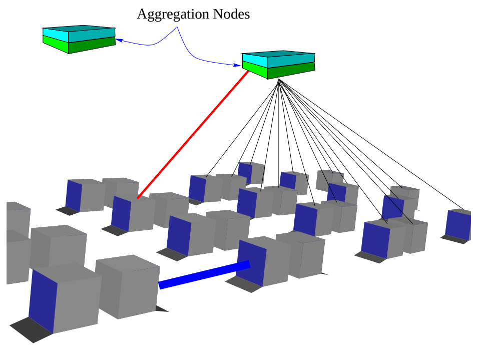
## Systems

✔ CosmoPlayer VRML viewer,

✔ **geomview**.

The visualization is not dependant on the navigation or implementation method.

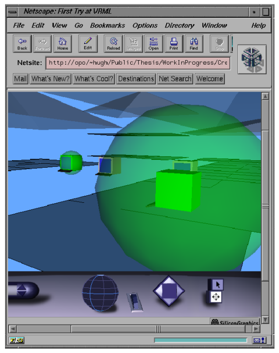
## Aggregation



## Implementation #1

- ✔ A data collector
- ✔ A web page with... a
  - ✔ Java program loaded as an applet, and a
  - ✔ VRML view of the network.

## 3DVNT


Netscape: First Try at VRML
File Edit View Go Bookmarks Options Directory Window Help
Netsite:
Mail What's New? What's Cool? Destinations Net Search Welcome

## Web page

```
<html><head> <title>Sample 3DVNT Page</title> </head>
<center><H1>Sample 3DVNT Page </H1></center>
<center> <embed src="root.wrl" height="600" width="700"> </center>
<center> <ap-
plet code="View1.class" width="100" height="10" mayscript>
<PARAM name="segment" value="MACS">
<PARAM name="port" value="9876">
<PARAM name="host" value="opo.usp.ac.fj"> </applet> </center>
OK?
</html>
```

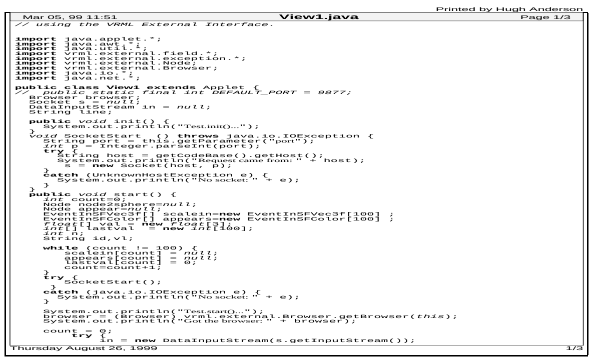
## VRML

```
PROTO CLUSTER  [] { ... }     # Cluster defi nition
PROTO KEYBOARD [] { ... }     # Keyboard defi nition
PROTO SCREEN   [] { ... }     # Screen defi nition
PROTO GLOBE    [] { ... }     # Traffi c sphere defi nition
# Some setting up declarations
Background { skyColor .4 .66 1 }
NavigationInfo { type [ "EXAMINE", "ANY" ] speed 400 }
Viewpoint { position 0 400 0 orientation 0 1 0 4 description "Camera 1" }
# Lines, floors and roofs
DEF LINES  Transform { ... }
DEF FLOORS Transform { ... }
DEF ROOFS  Transform { ... }
# and then the nodes
DEF node1  Transform { ... }
DEF node2  Transform { ... }
# ... and so on ...
```

## VRML nodes

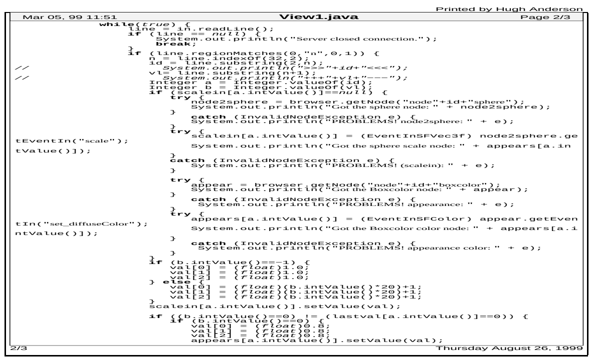
```
DEF node1 Transform {
    translation 4350 150 4365
    rotation 0 1 0 4.71238
    children [
      KEYBOARD {}
      SCREEN {}
      DEF node1box Transform {
        children [
          Shape {  ... }
      ] }
      DEF node1sphere Transform {
        scale 1 1 1
        children [
          Shape { ... }
      ] } ] }
```

## Java 1



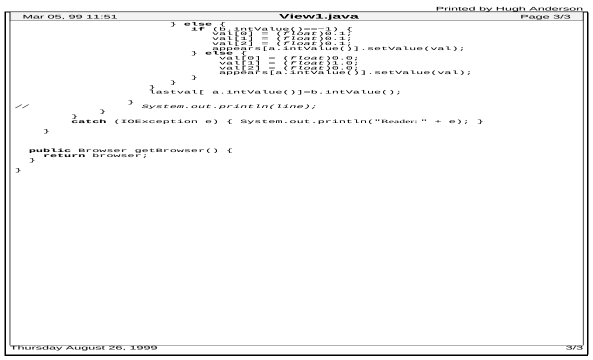
```
                                                            Printed by Hugh Anderson
Mar 05, 99 11:51                      View1.java                          Page 1/3
// using the VRML External Interface.

import java.applet.*;
import java.awt.*;
import java.util.*;
import vrml.external.field.*;
import vrml.external.exception.*;
import vrml.external.Node;
import vrml.external.Browser;
import java.io.*;
import java.net.*;

public class View1 extends Applet {
//   public static final int DEFAULT_PORT = 9877;
  Browser browser;
  Socket s = null;
  DataInputStream in = null;
  String line;

  public void init() {
    System.out.println("Test.init()...");
  }
  void SocketStart  () throws java.io.IOException {
    String port = this.getParameter("port");
    int p = Integer.parseInt(port);
    try {
      String host = getCodeBase().getHost();
      System.out.println("Request came from: " + host);
       s = new Socket(host, p);
    }
    catch (UnknownHostException e) {
      System.out.println("No socket: " + e);
    }
  }
  public void start() {
    int count=0;
    Node node2sphere=null;
    Node appear=null;
    EventInSFVec3f[] scalein=new EventInSFVec3f[100] ;
    EventInSFColor[] appears=new EventInSFColor[100] ;
    float[] val = new float[3];
    int[] lastval  = new int[100];
    int n;
    String id,vl;

    while (count != 100) {
        scalein[count] = null;
        appears[count] = null;
        lastval[count] = 0;
        count=count+1;
    }
    try {
        SocketStart();
      }
    catch (java.io.IOException e) {
      System.out.println("No socket: " + e);
    }

    System.out.println("Test.start()...");
    browser = (Browser) vrml.external.Browser.getBrowser(this);
    System.out.println("Got the browser: " + browser);

    count = 0;
          try {
              in = new DataInputStream(s.getInputStream());
Thursday August 26, 1999                                                     1/3
```

## Java 2



```
                                                            Printed by Hugh Anderson
Mar 05, 99 11:51                      View1.java                          Page 2/3
                    while(true) {
                          line = in.readLine();
                          if (line == null) {
                               System.out.println("Server closed connection.");
                               break;
                          }
                          if (line.regionMatches(0,"n",0,1)) {
                              n = line.indexOf(32,2);
                              id = line.substring(2,n);
//                             System.out.println(">>>"+id+"<<<");
                              vl= line.substring(n+1);
//                             System.out.println("+++"+vl+"---");
                              Integer a = Integer.valueOf(id);
                              Integer b = Integer.valueOf(vl);
                              if (scalein[a.intValue()]==null) {
                                  try {
                                      node2sphere = browser.getNode("node"+id+"sphere");
                                      System.out.println("Got the sphere node: " + node2sphere);
                                  }
                                      catch (InvalidNodeException e) {
                                      System.out.println("PROBLEMS! node2sphere: " + e);
                                  }
                                  try {
                                      scalein[a.intValue()] = (EventInSFVec3f) node2sphere.ge
tEventIn("scale");
                                      System.out.println("Got the sphere scale node: " + appears[a.in
tValue()]);
                                  }
                                  catch (InvalidNodeException e) {
                                      System.out.println("PROBLEMS! (scalein): " + e);
                                  }

                                  try {
                                      appear = browser.getNode("node"+id+"boxcolor");
                                      System.out.println("Got the Boxcolor node: " + appear);
                                  }
                                      catch (InvalidNodeException e) {
                                       System.out.println("PROBLEMS! appearance: " + e);
                                  }
                                  try {
                                      appears[a.intValue()] = (EventInSFColor) appear.getEven
tIn("set_diffuseColor");
                                      System.out.println("Got the Boxcolor color node: " + appears[a.i
ntValue()]);
                                  }
                                      catch (InvalidNodeException e) {
                                       System.out.println("PROBLEMS! appearance color: " + e);
                                  }
                              }
                              if (b.intValue()==-1) {
                                  val[0] = (float)1.0;
                                  val[1] = (float)1.0;
                                  val[2] = (float)1.0;
                              } else {
                                  val[0] = (float)(b.intValue()*20)+1;
                                  val[1] = (float)(b.intValue()*20)+1;
                                  val[2] = (float)(b.intValue()*20)+1;
                              }
                              scalein[a.intValue()].setValue(val);

                              if ((b.intValue()==0) != (lastval[a.intValue()]==0)) {
                                  if (b.intValue()==0) {
                                      val[0] = (float)0.8;
                                      val[1] = (float)0.8;
                                      val[2] = (float)0.8;
                                      appears[a.intValue()].setValue(val);
2/3                                                     Thursday August 26, 1999
```

## Java 3



Printed by Hugh Anderson

Mar 05, 99 11:51 **View1.java** Page 3/3

```
                        } else {
                           if (b.intValue()==-1) {
                              val[0] = (float)0.1;
                              val[1] = (float)0.1;
                              val[2] = (float)0.1;
                              appears[a.intValue()].setValue(val);
                           } else {
                               val[0] = (float)0.0;
                               val[1] = (float)1.0;
                               val[2] = (float)0.0;
                               appears[a.intValue()].setValue(val);
                           }
                        }
                     }
                     lastval[ a.intValue()]=b.intValue();

                  }
//                   System.out.println(line);
              }
          }
          catch (IOException e) { System.out.println("Reader: " + e); }
      }


   public Browser getBrowser() {
      return browser;
   }
}
```

Thursday August 26, 1999 3/3

## Summary of topics

In this module, we introduced the following topics:

- Visualization versus computer-graphics
- OpenGL
- (Briefly) Java3D, VTK
- VRML/Java/EAI