What is VRML ?

VRML – **V**irtual **R**eality **M**odelling Language

HTML – **H**yper**T**ext **M**arkup **L**anguage

VRML is more than an extension of HTML

What is VRML ?

- **Purpose:** The Virtual Reality Modeling Language is a file format for describing interactive 3D objects and worlds. VRML is designed to be used on the Internet, intranets, and local client systems. VRML is also intended to be a universal interchange format for integrated 3D graphics and multimedia.
- **Use:** VRML may be used in a variety of application areas such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds.

History

- VRML 1.0 Specification 1995
- VRML 97 / 2.0 Specification 1997
- X3D (and Java3D) Specification (in development)

Designe

- **Composability:** Provide the ability to use and combine dynamic 3D objects within a VRML world and thus allow re−usability (Object orientated approach)
- **Extensibility:** Provide the ability to add new object types not explicitly defined in VRML, e.g. Sound
- **Performance:** Emphasize scalable, interactive performance on a wide variety of computing platforms (Platform independent)

Characteristics of VRML

- VRML is capable of representing static and animated dynamic 3D and multimedia objects with hyperlinks to other media such as text, sounds, movies, and images.
- VRML browsers, as well as authoring tools for the creation of VRML files, are widely available for many different platforms.
- Other formats: OpenGL, Inventor (not designed for WWW use)

Information Sources

- Organisation for the VRML standard: http://www.web3d.org/
- VRML 2 Spec: http://www.web3d.orgtechnicalinfo/specifications/vrml97/
- Books: e.g. The VRML 2 Handbook
- Browsers: e.g. Blaxxun Java−applet, Plug−Ins, Cosmo Player (SGI) http://www.web3d.org/vrml/browpi.htm

VRML − Basics

- A VRML file is essentially a collection of Objects called **Nodes** which can be something physically: Sphere, Cylinder, etc. or non−physically: Viewpoints, Hyperlinks,Transformations, Sound, etc.
- Each Node contains **Fields** which hold the data of the node

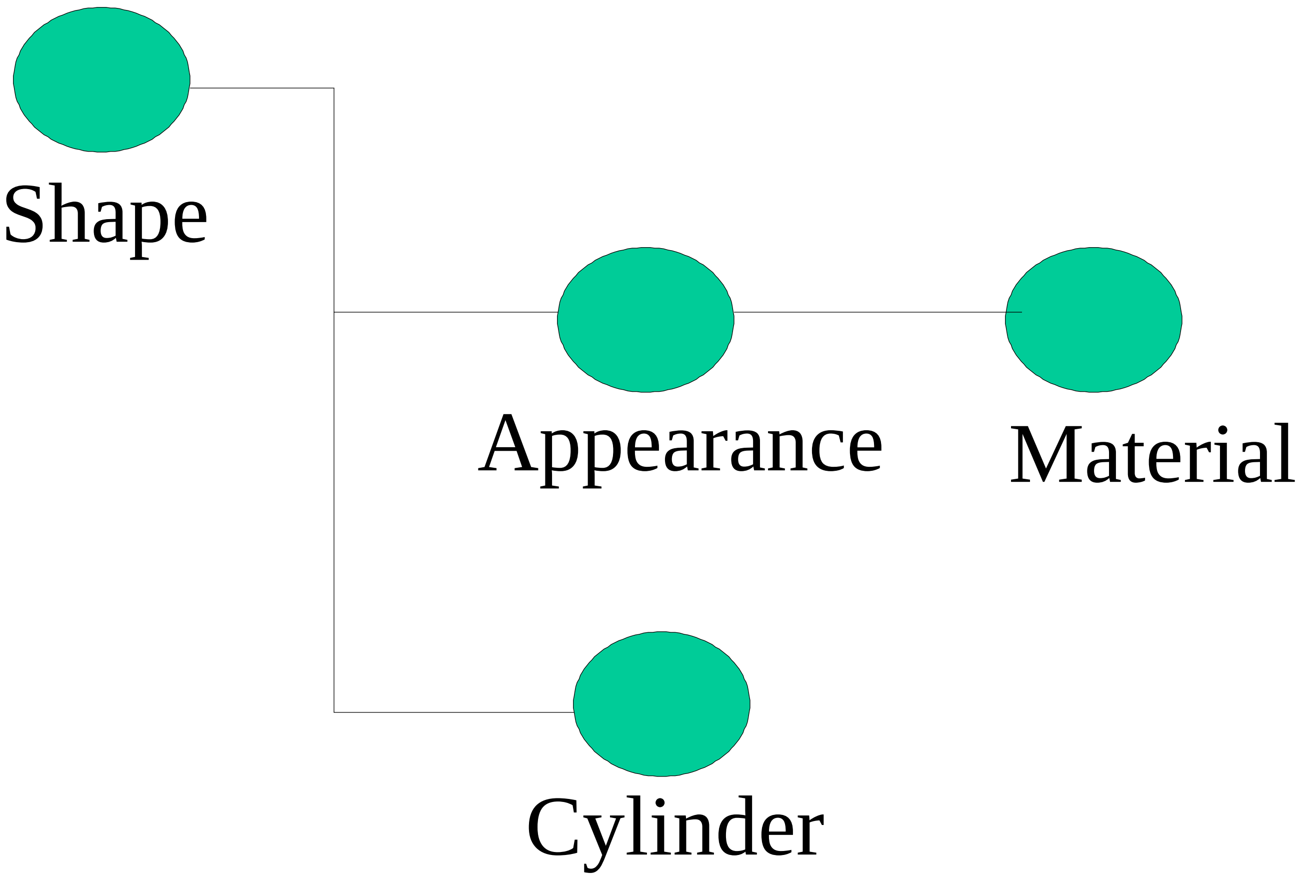
VRML − Basics

- Some nodes are **container nodes** or **grouping nodes**, which contain other nodes
- Nodes are arranged in hierarchical structures called **scene graphs.** Scene graphs are more than just a collection of nodes; the scene graph defines an ordering for the nodes. The scene graph has a notion of **state**, i.e. nodes earlier in the world can affect nodes that appear later in the world.

First VRML − File

```
#VRML V2.0 utf8
   Shape {
                   appearance Appearance  {
                             material Material {
                                       diffuseColor    1 0 0
                             }
                   }
                   geometry Cylinder {
                             radius    3
                             height    6
                   }
         }
```

First File – Scene Graph



Basic Shape Nodes

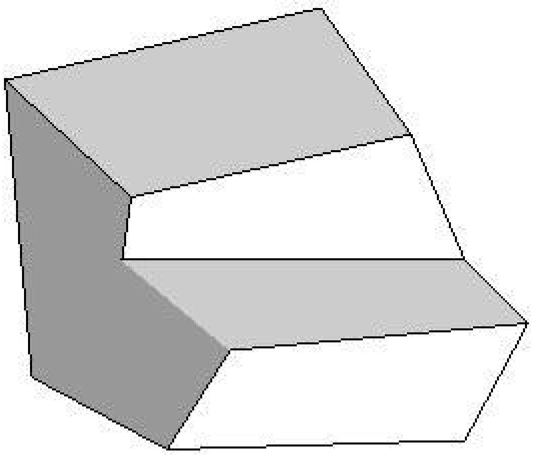
- A lot of simple shapes are given:

  Cylinder, Box, Sphere, etc.

- More ”general” shapes are needed for realistic worlds − IndexedFaceSet

IndexFaceSet Node

- Define 3D Polyhedrons from a collection of 2D Polygons, e.g. Triangules

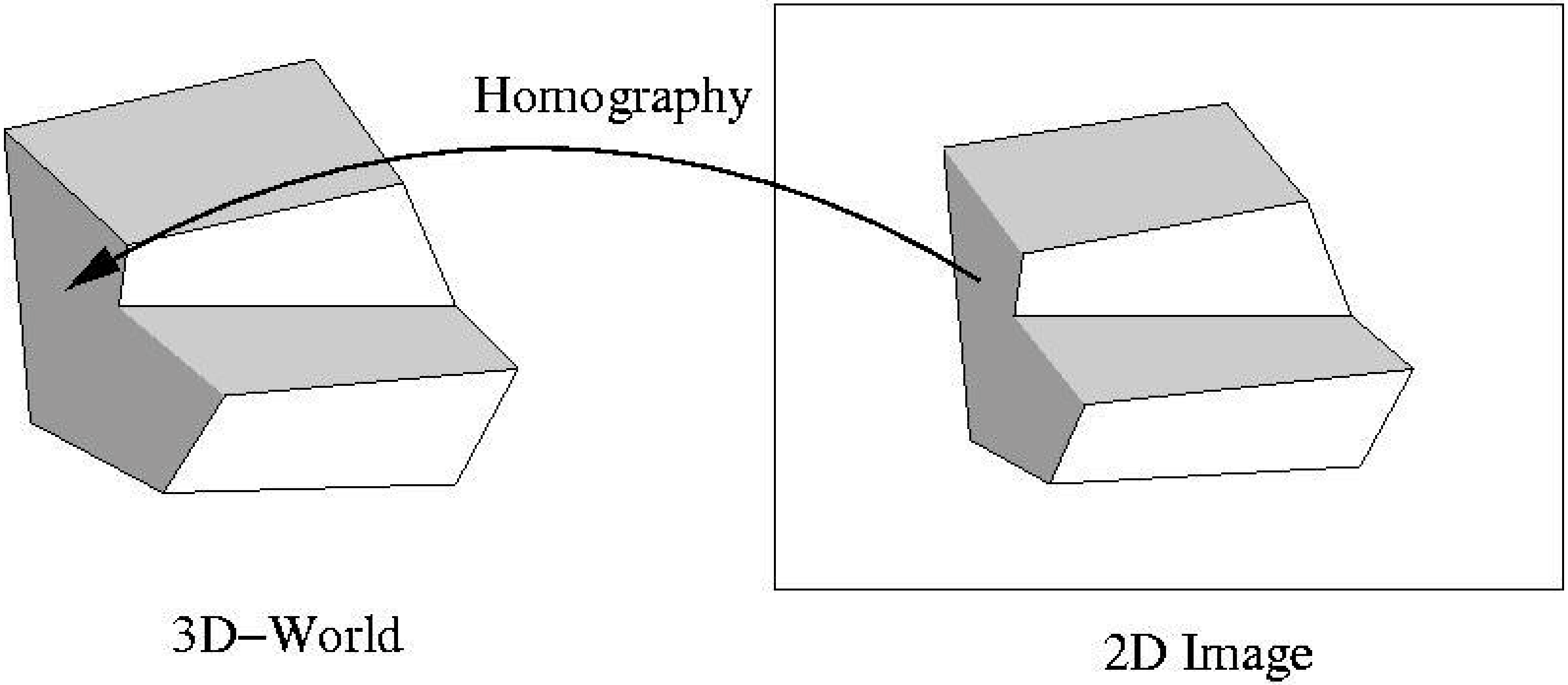


IndexFaceSet Node

```
IndexedFaceSet {
        coord Coordinate {
                point [ x1 y1 z1, x2 y2 z2, ... ] }
        coordIndex [        3 0 5 1 –1,
                            2 0 1 4 5 –1,
                            3 1 5 –1 ]
```

Texture Node

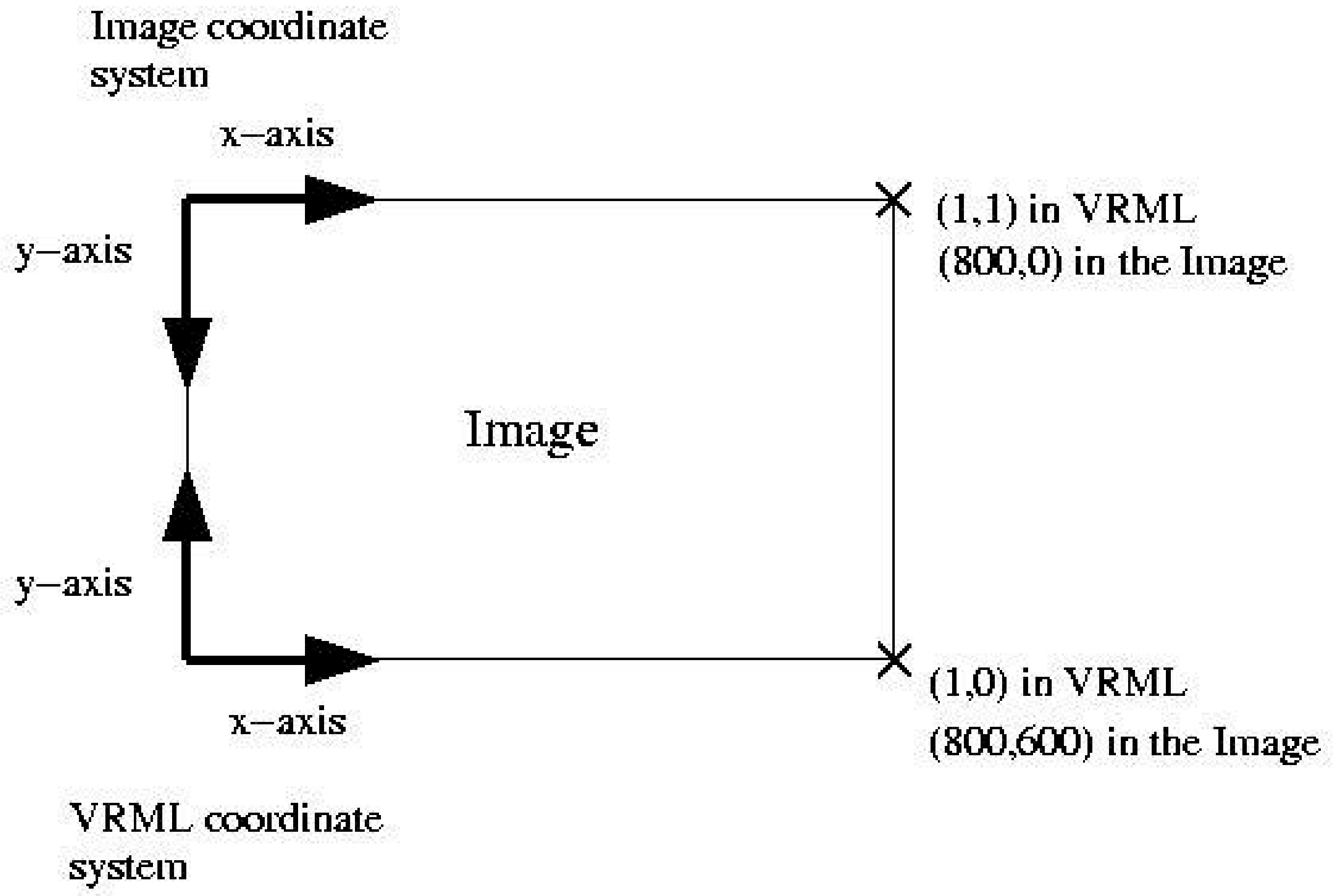
- Map the 2D image texture onto the 2D Polygons in the scene



Texture Node

```
Shape {
        appearnce Appearance {
                texture ImageTexture {url ”image.jpg”}}
        geometry IndexedFaceSet {
                coord Coordinate {
                        point [ x1 y1 z1, x2 y2 z2, ... ] }
                coordIndex [1 0 3 –1, 2 3 4 –1, ...]
                texCoord TextureCoordinate {
                        point [ x1 y1, x2 y2, ... ] }
}
```

Texture Node



Re−use with DEF

- You can give a certain node a name with DEF and re−use it later on:

```
DEF COLUMN Shape {
              appearance Appearance {...}
     geometry Cylinder { radius 3, height 6}

# later in the file
Transformation {
      transform 1 1 1
     USE COLUMN }
```

Transformation Node

- Is a **grouping node** of the form:

```
Transform {
                    rotation x y z angle
                    translation x y z
                    scale x y z
          children [ Shape {} , Shape {}, .... ] }
```

- Different Coordinate Systems for groups of Objects

Viewpoint Node

- Specify the position of the camera; default:

```
Viewpoint {
        position 0  0  10
        orientation 0 0 1 0
        fieldofView 0.78  # 45deg is normal
}
```

- Multiple Viewpoints possible

Further Topics

- **Lightning:** Directional Light, Spot Light, Point Light
- **Colors:** e.g.vary in a 2D polygon (Normals)
- **Texture:** e.g. Movie Texture
- **Sound:** Audio Clips e.g. sound.wav
- **Complex Shapes:** Extrusions (2½D Objects)
- **Background:** Sky and Ground

Further Topics

- **Scripting:** Include a Script from a URL.You should use commen Script Languages: JavaScript, Java which most broswer understand.
- **Animation:** From static to dynamic; Event Handling; Different Sensor nodes, e.g. Touchsensor; Timers
- **LOD:** Level of Detail

Further Topics

- **Inline Nodes**: Inline { url ”vrml.wrl” }

  Loads a file in the current VRML−file
- **Anchor Nodes**: Links to HTML pages, other worlds, sound, etc. E.g. Anchor { url ”discribe_vrml.html”}