- Three types of lighting can be used in a VRML world:
  - **DirectionalLight** node shines a light across the whole world in a certain direction.
  - **PointLight** shines a light from all directions from a certain point in space.
  - **SpotLight** shines a light in a certain direction from a point.
  - **RenderMan**: rendering package created by Pixar.
- The **background** of the VRML world can also be specified using the `Background` node.
- A **Panorama** node can map a texture to the sides of the world. A panorama is mapped onto a large cube surrounding the VRML world.

Animation and Interactions

- The only method of animation in VRML is by tweening — done by slowly changing an object that is specified in aninterpolator node.

- This node will modify an object over time, based on the six types of interpolators: color, coordinate, normal, orientation, position, and scalar.

  (a) All interpolators have two nodes that must be specified: the **key** and **keyValue**.

  (b) The **key** consists of a list of two or more numbers starting with 0 and ending with 1, defines how far along the animation is.

  (c) Each key element must be complemented with a **`keyValue`** element: defines what values should change.

2.3 VRML (Virtual Reality Modelling Language)

Overview

(a) **VRML**: conceived in the first international conference of the World Wide Web as a platform-independent language that would be viewed on the Internet.

(b) **Objective of VRML**: capability to put coloured objects into a 3D environment.

(c) VRML is an interpreted language; however it has been very influential since it was the first method available for displaying a 3D world on the World Wide Web.

History of VRML

- VRML 1.0 was created in May of 1995, with a revision for clarification called VRML 1.0C in January of 1996:
  - VRML is based on a subset of the file inventor format created by Silicon Graphics Inc.
  - VRML 1.0 allowed for the creation of many simple 3D objects such as a cube and sphere as well as user-defined polygons. Materials and textures can be specified for objects to make the objects more realistic.

- The last major revision of VRML was VRML 2.0, standardized by ISO as VRML97:
  - This revision added the ability to create an interactive world. VRML 2.0, also called “Moving Worlds”, allows for animation and sound in an interactive virtual world.
  - New objects were added to make the creation of virtual worlds easier.
  - Java and Javascript have been included in VRML to allow for interactive objects and user-defined actions.
  - VRML 2.0 was a large change from VRML 1.0 and they are not compatible with each other. However, conversion utilities are available to convert VRML 1.0 to VRML 2.0 automatically.

VRML Shapes

- VRML contains basic geometric shapes that can be combined to create more complex objects. Fig. 2.28 displays some of these shapes:

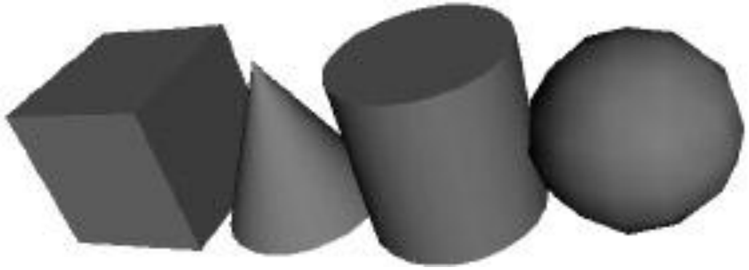


Fig. 2.28: Basic VRML shapes.

  - **Shape node** is a generic node for all objects in VRML.
  - **Material node** specifies the surface properties of an object. It can control what color the object is by specifying the red, green and blue values of the object.

- There are three kinds of texture nodes that can be used to map textures onto any object:

1. **ImageTexture**: The most common one that can take an external JPEG or PNG image file and map it onto the shape.

2. **MovieTexture**: allows the mapping of a movie onto an object; can only use MPEG movies.

3. **PixelTexture**: simply means creating an image to use with ImageTexture within VRML.

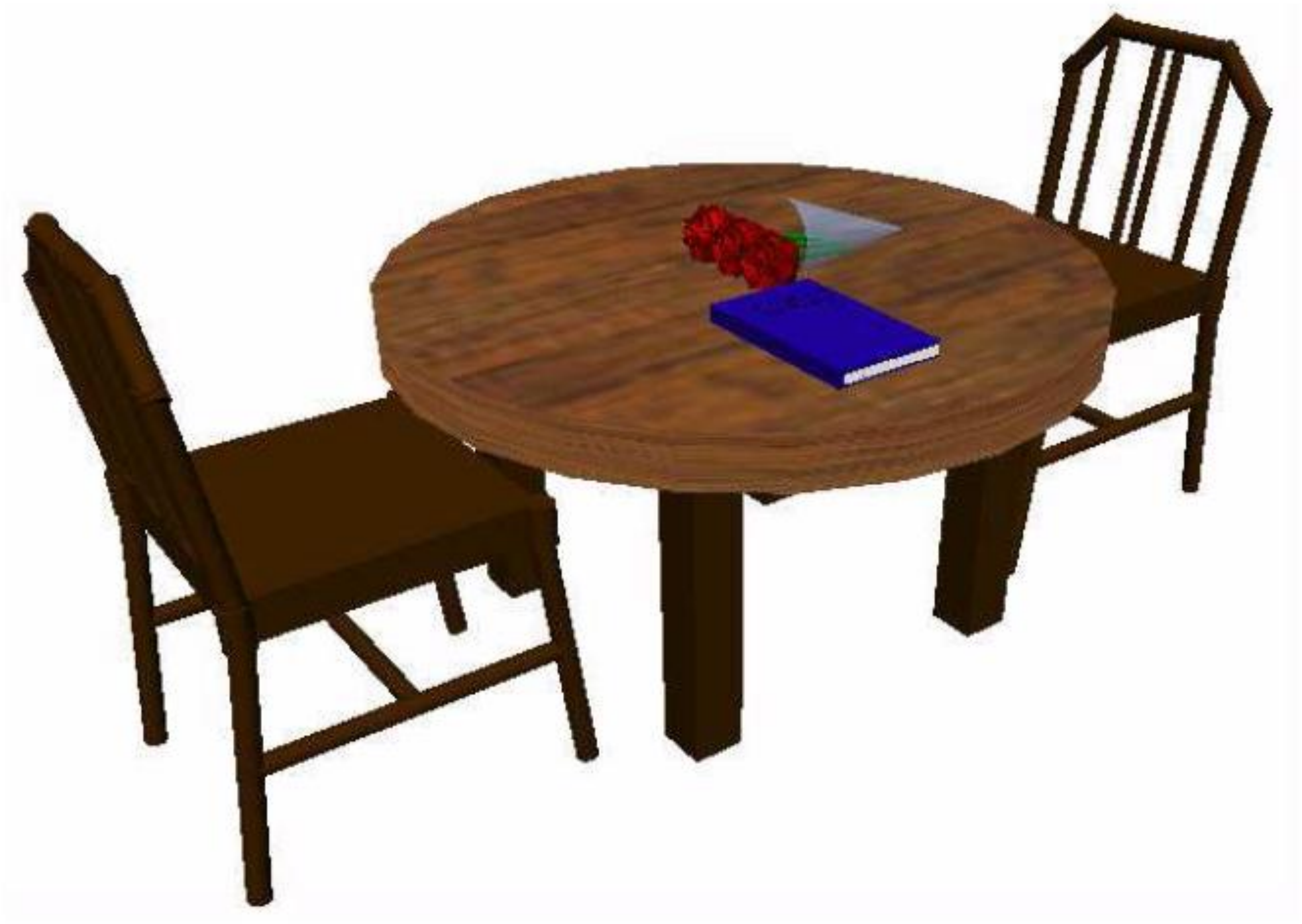
VRML world

- Fig. 2.29 displays a simple VRML scene from one viewpoint:
  → Openable-book VRML simple world!:
  →[Link to mmbook/examples/vrml.html.]

  – The position of a viewpoint can be specified with the position node and it can be rotated from the default view with the `orientation` node.

  – Also the camera's angle for its field of view can be changed from its default 0.78 radians, with the `fieldOfView` node.

  – Changing the field of view can create a telephoto effect.



- Fig. 2.29: A simple VRML scene.

- To time an animation, a **TimeSensor** node should be used:

  (a) **TimeSensor** has no physical form in the VRML world and just keeps time.

  (b) To notify an interpolator of a time change, a **ROUTE** is needed to connect two nodes together.

  (c) Most animation can be accomplished through the method of routing a `TimeSensor` to an interpolator node, and then the interpolator node to the object to be animated.

- Two categories of sensors can be used in VRML to obtain input from a user:

  (a) **Environment sensors**: three kinds of environmental sensor nodes: `VisibilitySensor`, `ProximitySensor`, and `Collision`.

  (b) **Pointing device sensors**: touch sensor and drag sensors.

VRML Specifics

- Some VRML Specifics:
  - (a) A VRML file is simply a text file with a “`.wrl`" extension.
  - (b) VRML97 needs to include the line **`#VRML V2.0 UTF8`** in the first line of the VRML file — tells the VRML client what version of VRML to use.
  - (c) VRML nodes are case sensitive and are usually built in a hierarchical manner.
  - (d) All Nodes begin with “{” and end with “}” and most can contain nodes inside of nodes.
  - (e) Special nodes called group nodes can cluster together multiple nodes and use the keyword “children” followed by “[...]”.

(f) Nodes can be named using `DEF` and be used again later by using the keyword `USE`. This allows for the creation of complex objects using many simple objects.

- A simple VRML example to create a box in VRML: one can accomplish this by typing:

```
Shape {
  Geometry Box{}
  }
```

The Box defaults to a 2-meter long cube in the center of the screen. Putting it into a **`Transform`** node can move this box to a different part of the scene. We can also give the box a different color, such as red.

```
– Transform { translation 0 10 0 children
  [
–  Shape {
–       Geometry Box{}
–       appearance Appearance {
–            material Material {
–                 diffuseColor 1 0 0
–            }
–       }
–  }
– ]}
```