WVHTC FOUNDATION SM

# VRML: Programming Introduction

# Day 2: VRML Lesson 1

**1000 Galliher Dr. | Fairmont, WV 26555-2720**

**Phone 304.368.9300 | www.wvhtf.org**

WVHTC FOUNDATION

## Today's Topics

- Introduction and History of VRML
- VRML Resources
- VRML Tools
- VRML Exploration

WVHTC FOUNDATION

# Introduction and History of VRML

- What is VRML?
  - VRML stands for “Virtual Reality Modeling Language”
  - VRML is a simple text language for describing 3-D shapes and interactive environments
  - VRML worlds are viewed with a Web browser
  - VRML text files use a **.wrl** extension

WVHTC FOUNDATION

# History of VRML

- Late ‘80s HTML
  - “Hypertext Markup Language”
  - Created to instruct the computer how to display information on the Web
  - Is a simple text-based file format with embedded commands (known as tags)

WVHTC FOUNDATION

# History of VRML

- 1994 VRML v1.0
  - Created to instruct the computer how to display 3D content on the Web
  - Like HTML, it uses a simple text-based file format with embedded commands
  - Didn’t have interactivity and animation
- 1996 VRML v2.0 is born
  - Fixes the defects of v1.0
  - This is the version you will be using
- 1997 VRML97
  - International Standard
  - No difference to VRML v2.0

WVHTC FOUNDATION

# Quick HTML vs. VRML Example

- HTML Example
  - http://www.google.com/index.html
  - Open link in browser and click on “View -> Source” from menu to view HTML text
- VRML Example
  - sodablock.wrl
  - Open in text editor for code view or VRML Viewer for 3D representation

WVHTC FOUNDATION

# Examples of VRML Scenes

- Simple Scene
  - Sodablock
- Complex Scene
  - Stone Henge
- Animation
  - Spiral

WVHTC FOUNDATION

# VRML Resources

- I will show you:
  - http://vista.fairmontstate.edu/ for online tutorial
  - CITERA home page
    - http://citerawv.us/
  - Teach Yourself VRML 2 in 21 Days
  - VRML 2.0 Sourcebook
  - Online Tutorials
  - Online Math & Science Sites

WVHTC FOUNDATION

# VRML Programming Tools

- What tools do we need to create a VRML program?
  - Editor to create and edit the .wrl file
    - Any text editor can be used, but Textpad has useful features
  - A Viewer for viewing the 3D world described by the .wrl file
    - Textpad VRML Viewer or Web Browser
- How do we use these tools?

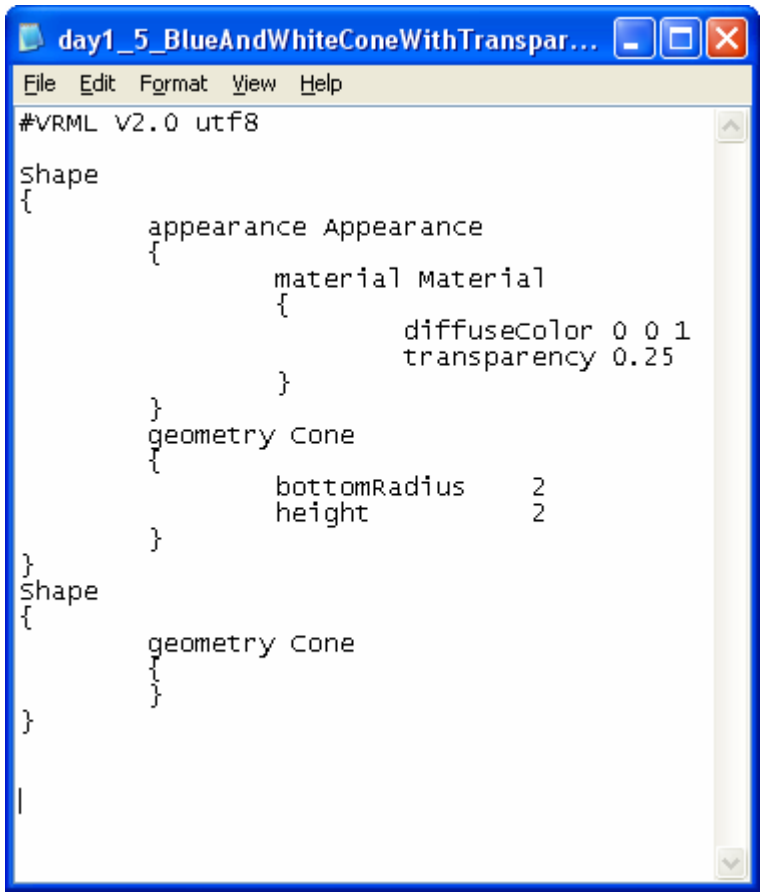
WVHTC FOUNDATION

# CITERA VRML Tools Tutorial

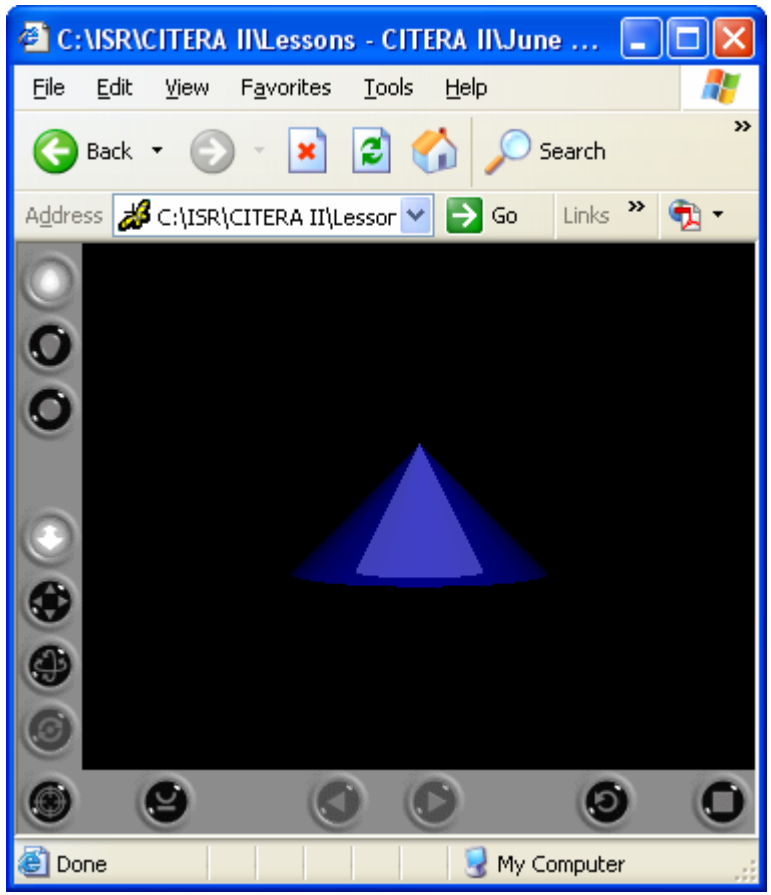
- Download Tools package and Installation Document from Citera website
  - http://citerawv.us/resources/index.htm
- Follow installation document to install VRML Cortona Viewer, Textpad, along with WVHTCF's custom VRML help tools

# Traditional Tools: Notepad and Internet Explorer

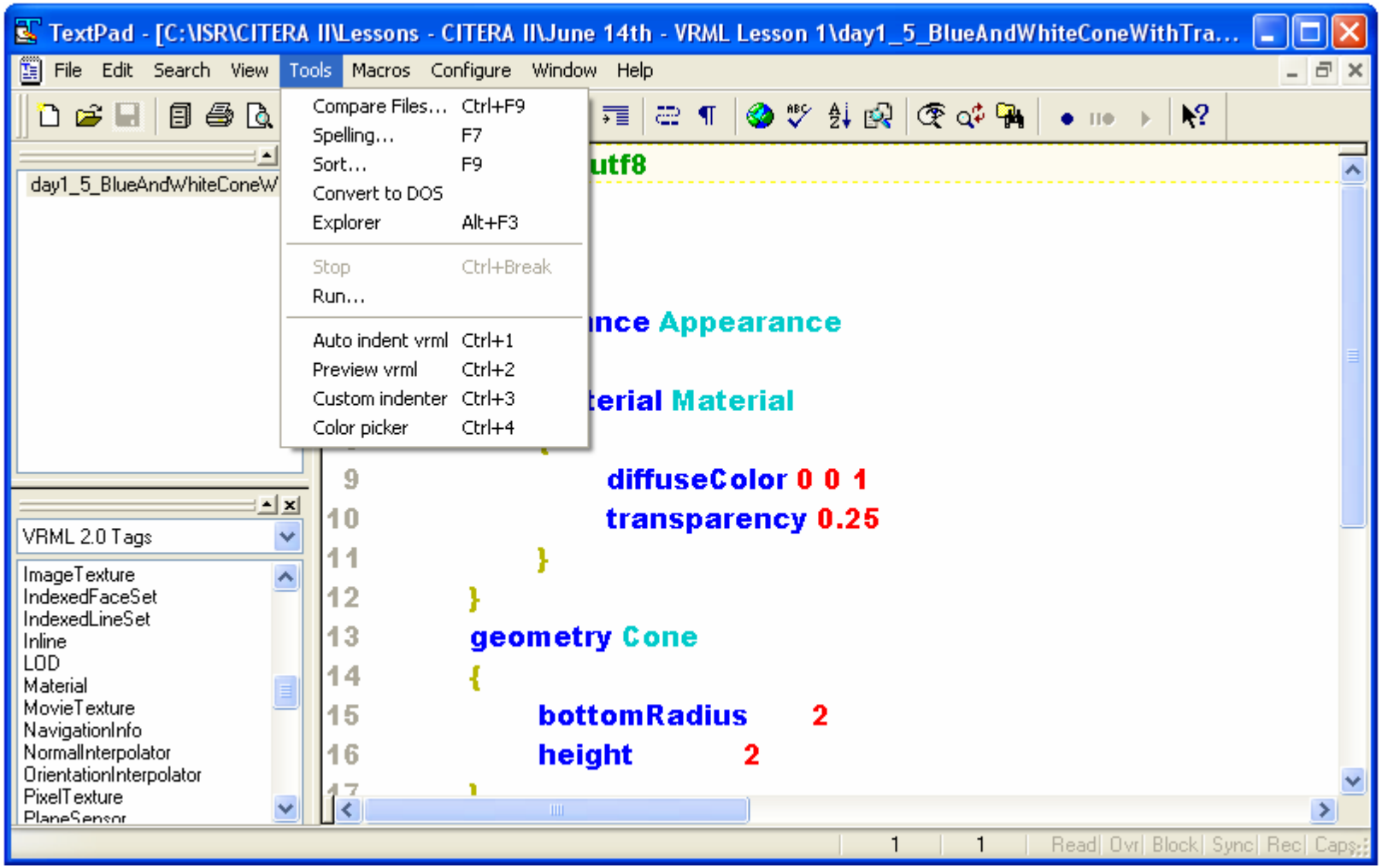
## Edit VRML Code




## View VRML Code



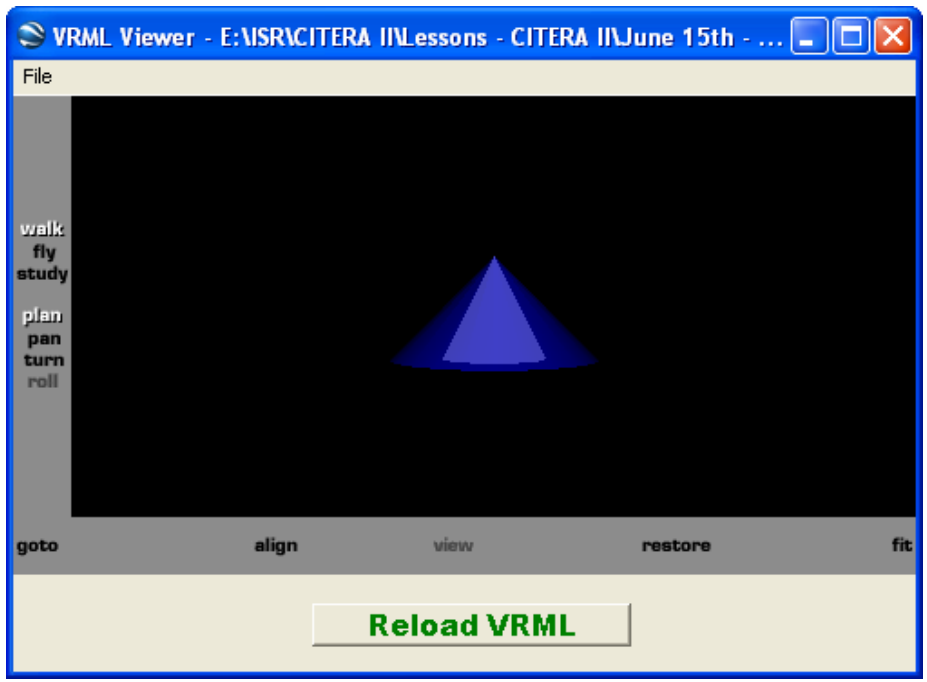
# Textpad Tools



WVHTC FOUNDATION SM

# Using Preview Tool

- Run From Tools menu in Textpad, or press Ctrl+2
  - Saves VRML and launches previewer



WVHTC FOUNDATION

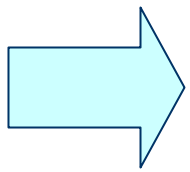
# Using Indent Tool

- Auto-Indent:
  - Run From Tools menu in Textpad, or press Ctrl+1
  - Indents code using Tabs automatically
- Custom Indent
  - Run From Tools menu in Textpad, or press Ctrl+3
  - Launches Indenter tool and gives your option of indenting code with tabs or spaces

WVHTC FOUNDATION

# Why Indent? Code is easier to read/debug.

```
Shape
{
appearance Appearance
{
    material Material
{
            diffuseColor 0,0,0
            }
    }
            geometry Box{}
}
```

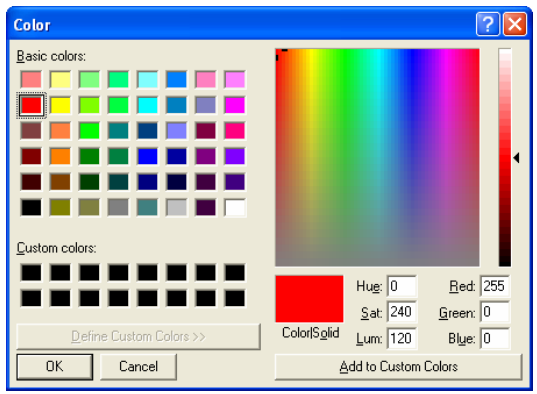


```
Shape
{
    appearance Appearance
    {
        material Material
        {
            diffuseColor 0,0,0
        }
    }
    geometry Box
    {
    }
}
```

WVHTC FOUNDATION SM

# Using Color Picker

- Run From Tools menu in Textpad, or press Ctrl+4
- Choose a color you like and press OK.  Now just paste the RGB value into your VRML code




1 0 0

WVHTC FOUNDATION

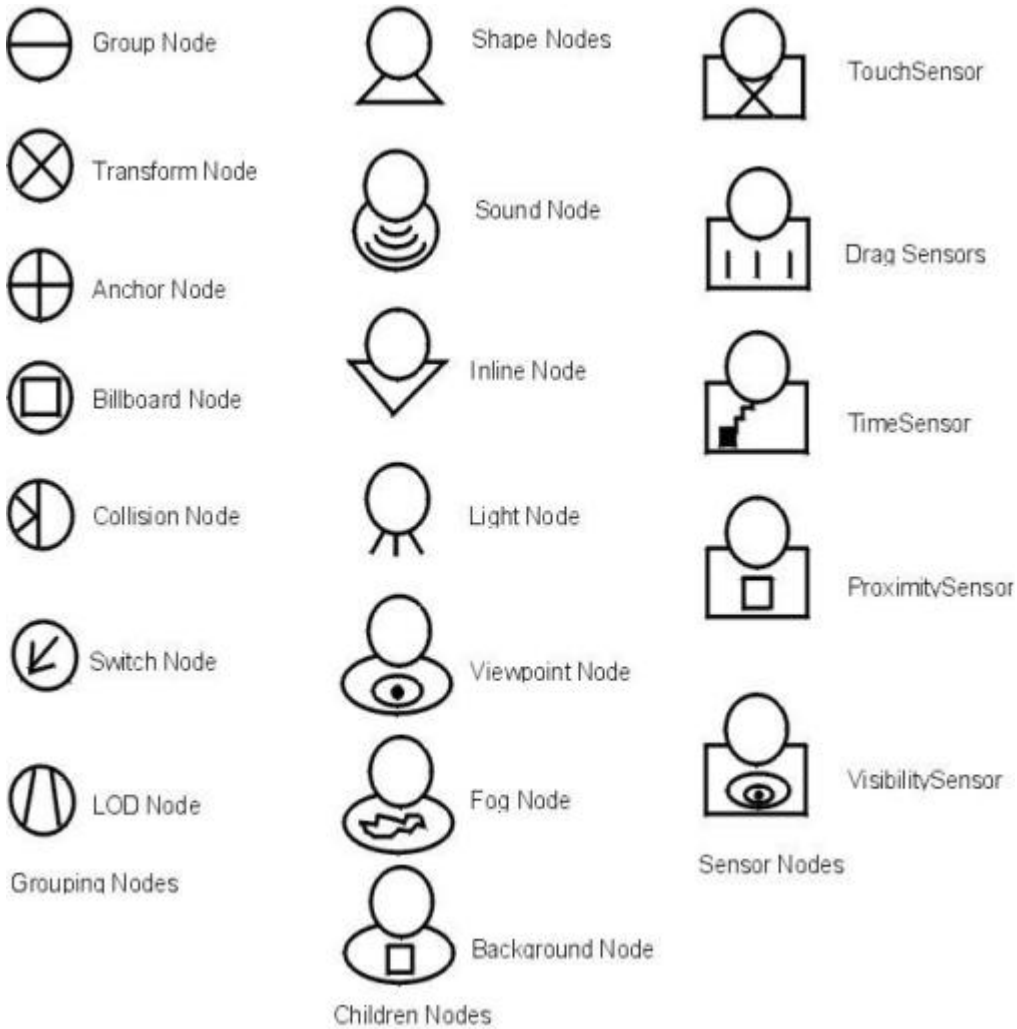
# Basic VRML Concepts

- Scene Graph and Nodes
- Nodes and Fields
- Shapes and Geometry
- Materials and Appearances

WVHTC FOUNDATION

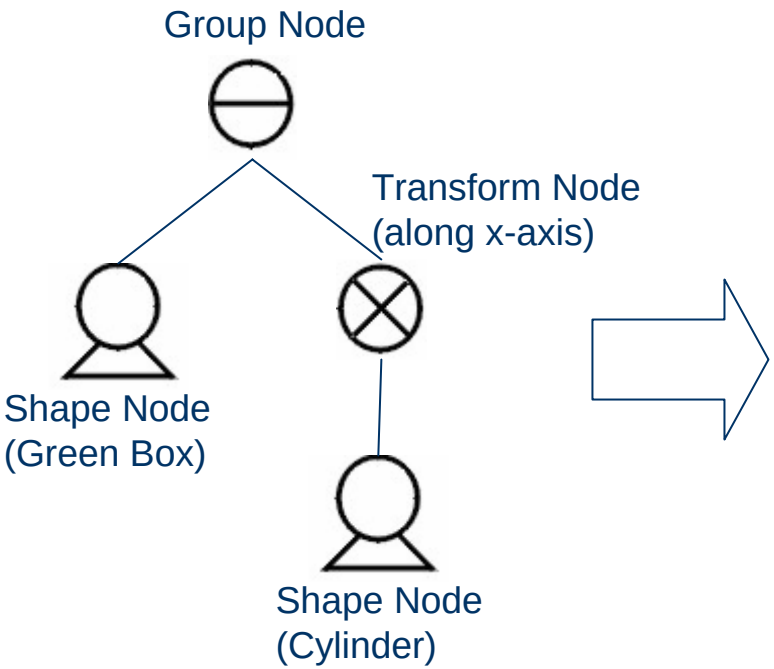
# Scene Graph and Nodes

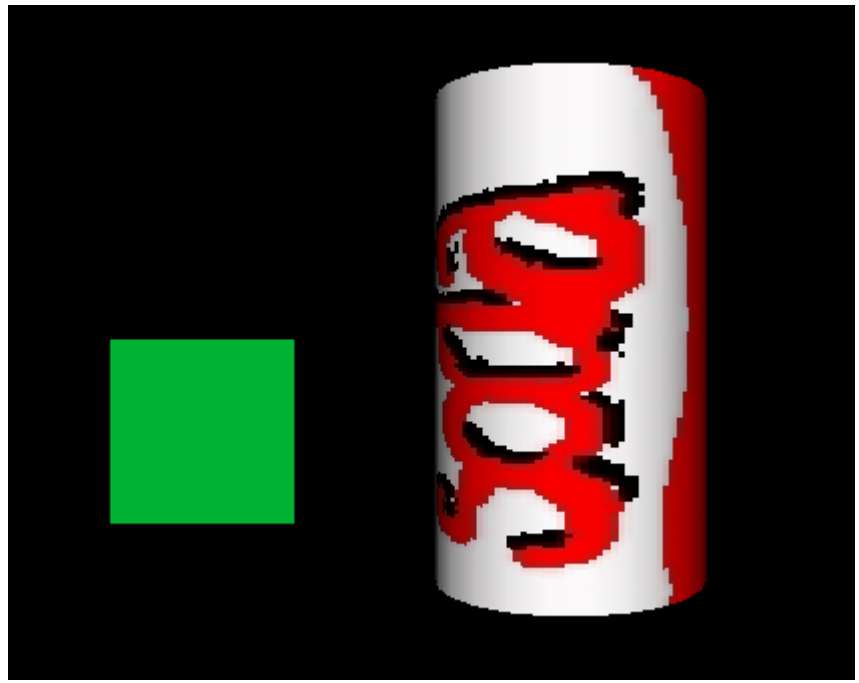
- Every VRML scene can be expressed as a scene graph consisting of nodes



WVHTC FOUNDATION

# Scene Graph Example



WVHTC FOUNDATION

# Nodes and Fields

- Every Node in VRML has associated fields
  - See Appendix A for detailed description of the fields of all defined VRML nodes along with their default values
- Example: Default Values of a Cone Node (pg 375)

```
Cone {
            bottomRadius                1
            height                      2
            side                        TRUE
            bottom                      TRUE
  }

  OR using implicit default values

Cone { }
```

- NOTE: By default all fields that specify size have dimensions in meters
  - For this example, by default a Cone shape has a height of 1 meter and a bottom radius of 2 meters

WVHTC FOUNDATION

# Shapes and Geometry

- A node in VRML used to add appearance to any geometry node
  - For example you can add color to a Cone
- Definition of a Shape Node (pg. 408)

```
Shape {
  appearance            NULL
  geometry              NULL
}
```

- By default all fields in Shape are NULL (empty)
  - An example of a Shape node with Cone geometry is

```
Shape {
  geometry Cone { }
}
```

WVHTC FOUNDATION

# Materials and Appearances

- Definition of an Appearance Node (pg. 366)

```
Appearance {
        material            NULL
        texture             NULL
        textureTransform    NULL
}
```

- Definition of a Material Node (pg. 394)

```
Material {
        ambientIntensity    0.2
        diffuseColor        0.8 0.8 0.8
        emissiveColor       0 0 0
        shininess           0.2
        specularColor       0 0 0
        transparency        0
}
```

WVHTC FOUNDATION℠

# Material Field Definitions

- ambientIntensity
  - The base intensity of the object. This value is multiplied by the diffuseColor to set the minimum color over the entire surface of the object.
- diffuseColor
  - The object color affected by light striking the object, but not by the angle of incidence of that light.
- emissiveColor
  - The object color that has the appearance of being emitted by the object.
- shininess
  - The coefficient that affects how narrow the angle of incidence of the light striking the object must be to create a shiny spot on the object. Lower shininess values distribute the specular color over a larger area of the object, while higher values restrict the specular color over a small area.
- SpecularColor
  - The object color affected by the shininess factor.
- transparency
  - The amount of transparency in this object. Lower values are less transparent. A value of 1 is completely transparent.

NOTE: diffuseColor is the field you will use most to change color of objects

WVHTC FOUNDATION

# Appearance Example

- Example of a Cone with a Diffuse Red Color
  - This value can be thought of as the base color being reflected from the object in all directions when light hits it
  - Color is in the form of RGB (Red, Green, Blue) values, each ranging from 0.0 – 1.0

```
Shape {
        appearance Appearance {
                material Material {
                        diffuseColor 1 0 0
                }
        }
        geometry Cone { }
}
```

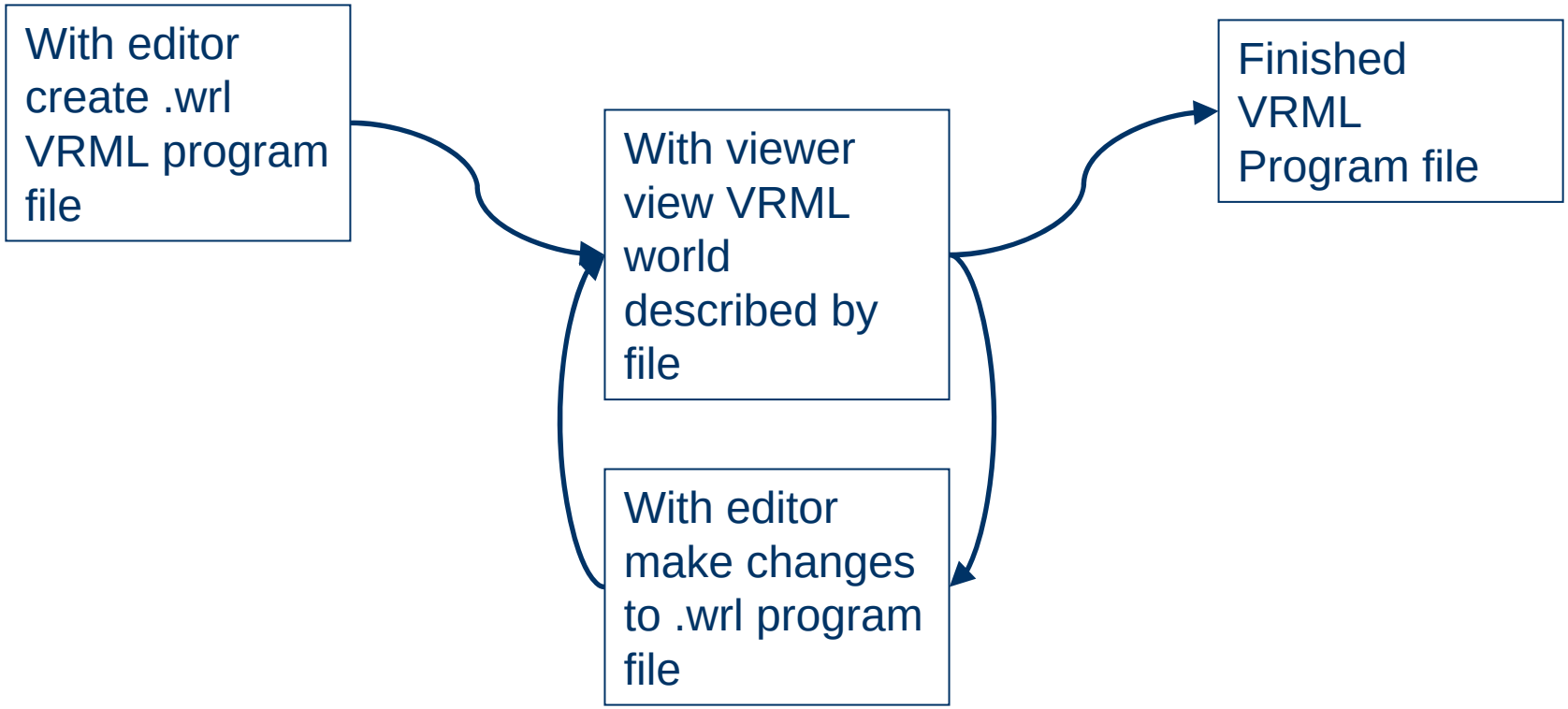
WVHTC FOUNDATION

# VRML Exploration

- VRML Development Process
- Basic VRML World Example
- Examples using Basic VRML Concepts
  - Scene Graph and Nodes
  - Nodes and Fields
  - Shapes and Geometry
  - Materials and Appearances
- Programming Assignment

WVHTC FOUNDATION

# Software Development Process – Spiral Model



# Simple VRML Scene: White Cone

- Create a Cone Shape

Scene Graph

Shape Node
(Basic Cone)

Simple VRML program

```
#VRML V2.0 utf8

Shape {
        geometry Cone { }
}
```

**IMPORTANT: Must have matching open and close braces for a valid VRML file!**

*Program day2_1_Cone.wrl*

# Simple VRML Scene: White Cone modified

- Alter the Shape of the Cone

Scene Graph

Shape Node
(Thick Cone)

VRML program

```
#VRML V2.0 utf8

Shape {
        geometry Cone {
                bottomRadius    2
                height          2
                }
}
```

**IMPORTANT: Everything is Case-Sensitive in VRML!!**

*Program day2_2_WideCone.wrl*

# Using Appearance and Material Nodes to Change Color

- Add Blue Color to the Cone

Scene Graph

Shape Node
(Thick Blue Cone)

**NOTE: Order of Nodes and Fields is not important**

```
#VRML V2.0 utf8

Shape {
        appearance Appearance {
          material Material {
                diffuseColor 0 0 1
                }
         }
        geometry Cone {
                bottomRadius    2
                height          2
                }
}
```

*Program day2_3_BlueCone.wrl*

WVHTC FOUNDATION

# Shininess, Etc.

| Effect | ambientIntensity | diffuseColor | specularColor | shininess |
|---|---|---|---|---|
| Aluminum | 0.3 | 0.3 0.3 0.5 | 0.7 0.7 0.8 | 0.1 |
| Copper | 0.26 | 0.3 0.11 0 | 0.75 0.33 0.00 | 0.08 |
| Gold | 0.4 | 0.22 0.15 0 | 0.71 0.7 0.56 | 0.16 |
| Metallic Purple | 0.17 | 0.1 0.03 0.22 | 0.64 0 0.98 | 0.20 |
| Metallic Red | 0.15 | 0.27 0 0 | 0.61 0.13 0.18 | 0.20 |
| Plastic Blue | 0.1 | 0.2 0.2 0.71 | 0.83 0.83 0.83 | 0.12 |

WVHTC FOUNDATION

# Shininess Example

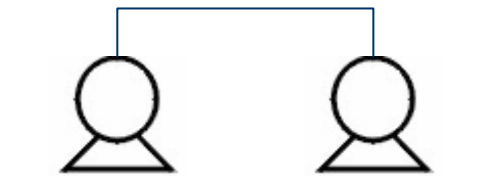
```
#VRML V2.0 utf8

#Copper Cone
Shape {
        appearance Appearance {
          material Material {
                diffuseColor 0.3 0.11 0
                specularColor 0.75 0.33 0.00
                shininess 0.08
                ambientIntensity 0.26
                }
         }
        geometry Cone {
                bottomRadius        2
                height              2
                }
}
```

*Program day2_4_CopperCone.wrl*

# Multiple Objects in VRML Scene

- Add a plain white cone to the scene

Scene Graph



Shape Node (Thick Blue Cone)

Shape Node (White Cone)

NOTE: Nodes are Stacked on top of each other at origin.

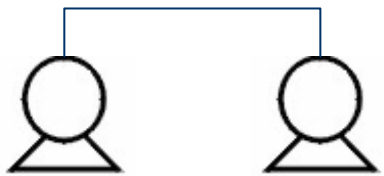
```
#VRML V2.0 utf8

Shape {
        appearance Appearance {
          material Material {
                diffuseColor 0 0 1
                }
         }
        geometry Cone {
                bottomRadius        2
                height              2
                }
}
Shape {
        geometry Cone {}
    }
```

*Program day2_5_BlueAndWhiteCone.wrl*

# Use Transparency to See Through Object

- Make the Blue Cone Transparent

Scene Graph



Shape Node (Transparent Thick Blue Cone)

Shape Node (White Cone)

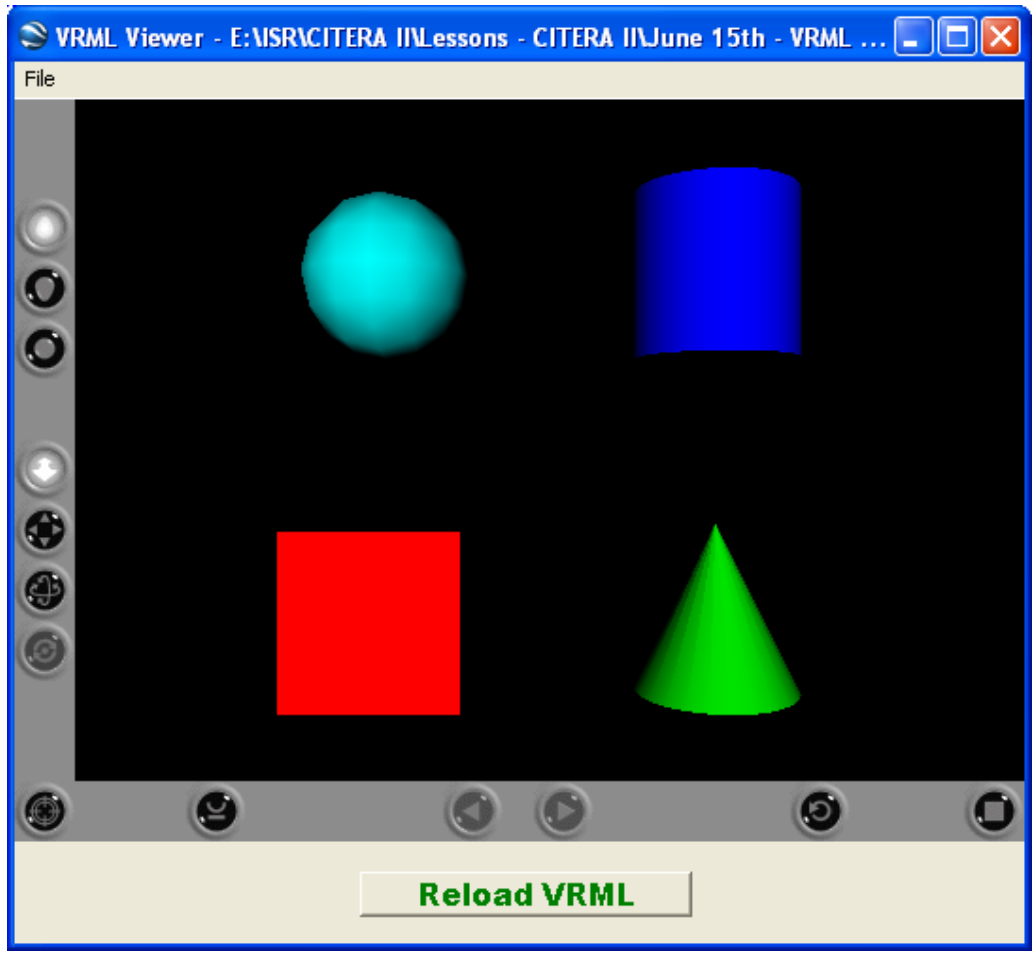
```
#VRML V2.0 utf8

Shape {
        appearance Appearance {
          material Material {
                diffuseColor 0 0 1
                transparency 0.25
                }
         }
        geometry Cone {
                bottomRadius    2
                height          2
                }
}
Shape {
        geometry Cone {}
    }
```

*Program day2_6_BlueAndWhiteConeWithTransparency.wrl*

WVHTC FOUNDATION

# Lessons: Basic Geometric Primitives

- Primitives
  - Cone
  - Box
  - Cylinder
  - Sphere
- Use to make more complex objects



WVHTC FOUNDATION℠

# VRML Assignment

- Install CITERA VRML Tools, gather resources, try sample programs, and look for examples.

- Create a VRML scene script and a scene graph with three cones:  one purple and shiny, another aquamarine and short, and the last tan and transparent.  Make the cones different heights and widths so they can all be seen, even though they are sitting on top of one another.

# Install CITERA Tools

1. Run the “Install CITERA VRML Tools” from Desktop
2. Install all tools in order of menu. Install wizards will pop up for the menu options 1 and 2. Simply accept all defaults by clicking next until install has finished.
3. For menu option 3, keep pressing enter until menu shows up again