

# O uso do DEF



**#VRML V2.0 #VRML V2.0 utf8**

**#UseDef.wrl**

**#Exemplo do uso de USE e DEF para evitar repetições do Nó  
Cylinder**

**DEF Cilindro Shape**

**{ appearance Appearance**

**{**

**material Material { diffuseColor 1 0 1 }**

**}**

**geometry Cylinder { height 4**

**radius 0.5**

**}**

**}**

**Transform**

**{**

**translation 0.8 1.8 1.8**

**rotation 1.8 3.8 1.8 2.8**

**scale 0.8 1.8 2.8**

**children USE Cilindro**

**}**



# O uso do Point Set



```
geometry PointSet
{ coord Coordinate
{ point [ 1 1 1,
...
1 1 1
]
}
color Color
{ color [ 1 1 1,
...
1 1 1
]
}
```



```

#VRML V2.0 utf8
#PointSet.wrl
#M#VRML V2.0 utf8
#PointSet.wrl
#Mostra uma “chuva de pontos” através da geometria PointSet
Shape
{ appearance Appearance { }
  geometry PointSet
  { coord Coordinate
    { point
      [# Coordenadas dos Pontos cor ciano
      1.0 1.0 1.0, 0.5 1.0 1.0,
      1.5 1.5 1.5, 1.0 0.5 1.0,
      2.0 2.0 2.0, 1.0 1.0 0.5,
      2.5 2.5 2.5, 1.5 2.0 2.0,
      3.0 3.0 3.0, 2.0 1.5 2.0,
      # Coordenadas dos Pontos cor Amarela
      -1.0 -1.0 -1.0, -0.5 -1.0 -1.0,
      -1.5 -1.5 -1.5, -1.0 -0.5 -1.0,
      -2.0 -2.0 -2.0, -1.0 -1.0 -0.5,
      -2.5 -2.5 -2.5, -1.5 -2.0 -2.0,
      -3.0 -3.0 -3.0, -2.0 -1.5 -2.0
    ]
  }
  color Color
  { color
    [# cor ciano
    0.0 1.0 1.0, 0.0 1.0 1.0,
    0.0 1.0 1.0, 0.0 1.0 1.0,
    0.0 1.0 1.0, 0.0 1.0 1.0,
    0.0 1.0 1.0, 0.0 1.0 1.0,
    0.0 1.0 1.0, 0.0 1.0 1.0,
    # cor amarela
    1.0 1.0 0.0, 1.0 1.0 0.0,
    1.0 1.0 0.0, 1.0 1.0 0.0,
    1.0 1.0 0.0, 1.0 1.0 0.0,
    1.0 1.0 0.0, 1.0 1.0 0.0,
    1.0 1.0 0.0, 1.0 1.0 0.0,
  ]
  }
  }
  }
}

```



# O uso do Indexedlinetset



A geometria **IndexedLineSet** especifica **um conjunto de poli-linhas** no sistema de coordenadas local e as cores associadas. Através da ligação das linhas, pode-se criar o contorno de qualquer polígono ou contornos de diferentes objetos, como uma estrela, uma casa, um cubo, uma pirâmide, etc. A geometria contém cinco campos:



**coord:** especifica as coordenadas dos pontos que serão conectados para

formar as linhas; o primeiro ponto é o ponto 0, o segundo ponto 1 e assim por diante.

**coordIndex:** indica como os pontos serão ligados, formando as poli-linhas;

quando uma poli-linha termina, ela é “partida” pelo número **-1**; para formar um

polígono fechado é necessário repetir o primeiro ponto.

**color:** indica as cores que serão usadas para colorir as poli-linhas; a

primeira cor é 0, a segunda 1 e assim por diante.

**colorIndex:** indica uma cor para cada uma das poli-linhas definidas no

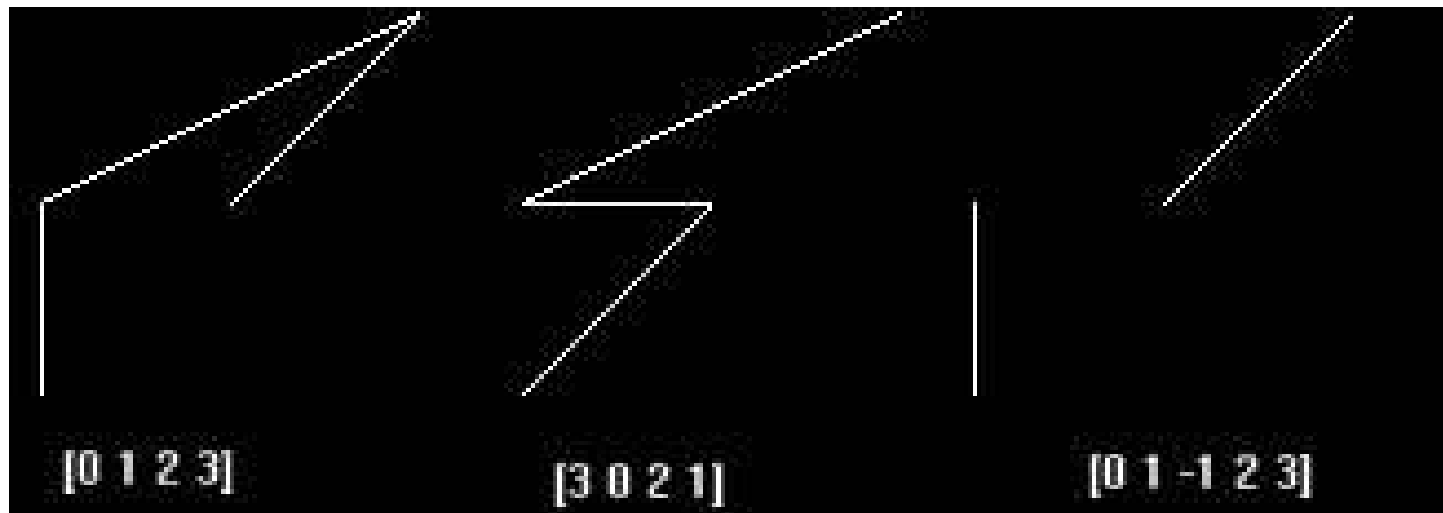
campo coordIndex ou cada um dos pontos definidos em coord.

**colorPervertex:** se **FALSE** indica que as cores serão aplicadas às poli-linhas;





Se considerarmos as coordenadas dos vértices como sendo **ponto 0 (0 0 0)**, **ponto 1 (1 1 0)**, **ponto 2 (-1 0 0)** e **ponto 3 (-1 -1 0)**, a **figura 11** mostra o resultado de 3 ligações diferentes destes pontos que seriam definidos no campo coordIndex.



```

#VRML V2.0 utf8
#IndexedLineSet.wrl
#Desenho das arestas de um cubo
com#VRML V2.0 utf8
#IndexedLineSet.wrl
#Desenho das arestas de um cubo
com 3 cores diferentes
Shape
{ appearance Appearance { }
  geometry IndexedLineSet
  { coord Coordinate
  { point
  [ # Cubo
  1.0 1.0 1.0, # ponto 0
  1.0 -1.0 1.0, # ponto 1
  -1.0 -1.0 1.0, # ponto 2
  -1.0 1.0 1.0, # ponto 3
  1.0 1.0 -1.0, # ponto 4
  1.0 -1.0 -1.0, # ponto 5
  -1.0 -1.0 -1.0, # ponto 6
  -1.0 1.0 -1.0 # ponto 7
  ]
  }
}

```

```

color Color
{ color
[1.0 0.0 0.0, 0.0 1.0 0.0, 0.0 0.0 1.0
# apenas 3 cores 0=Vermelho, 1=Verde e
2=Azul
]
}
coordIndex
[ # Linha vermelha (paralela ao eixo
x)
0, 3, -1, 1, 2, -1, 4, 7, -1, 5, 6, -1,
# Linha verde (paralela ao eixo y)
0, 1, -1, 2, 3, -1, 4, 5, -1, 6, 7, -1,
# Linha azul (paralela ao eixo z)
0, 4, -1, 1, 5, -1, 2, 6, -1, 3, 7, -1
]
colorIndex
[ 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2 ]
colorPerVertex FALSE
}
}

```



# O uso do geometry Text



**string:** contém o texto a ser mostrado; pode ter uma ou mais linhas.

**fontStyle:** especifica um node à parte que permite que sejam definidos aspectos de como o texto será apresentado. É composto dos campos:

**family:** especifica o fonte; valores neste campo podem ser:

"SERIF",

"SANS", "TYPEWRITER".

**style:** especifica o estilo do fonte; valores neste campo podem ser:

"PLAIN", "BOLD", "ITALIC", "BOLDITALIC".

**horizontal:** TRUE indica que o texto deve ser mostrado na horizontal,

FALSE, na vertical.

**leftToRight:** TRUE indica que o texto deve ser escrito da esquerda para a

direita; FALSE, da direita para a esquerda (modo árabe)

**topToBottom:** TRUE indica que o texto deve ser escrito de cima para



**justify:** se **horizontal** é **TRUE**, a justificação principal é **horizontal** e a

secundária é vertical; se **horizontal** for **FALSE** é o contrário. Há quatro

valores para este campo: **"FIRST"**, **"BEGIN"** (justificado à esquerda), **"MIDDLE"** (centralizado) e **"END"** (justificado à direita).

**language:** especifica o alfabeto: **"en"** para **English**, **"en\_US"** para **US**

**english**, **"zh"** para chinês, etc. Consulte o RFC 1766 para a lista completa.

**size:** especifica a altura dos caracteres em unidades VRML.

**spacing:** especifica o espaço entre linhas.

**length:** especifica o comprimento de cada string em unidades VRML, não em

caracteres. Se o string for muito curto, é escalado, se for muito longo, é

comprimido. Um valor zero indica que o string não deve ser nem escalado nem

comprimido. Zero é o valor default.



```
Transform
{translation 0.0 3.0 0.0
children
[ Shape
{ appearance Appearance
{ material Material { diffuseColor 0.0
1.0 0.0 }
}
geometry Text
{ string ["Hello", "sans-serif"]
fontStyle FontStyle
{ size 1.4
justify "MIDDLE"
family "SANS"
style "BOLD"
topToBottom TRUE
horizontal TRUE
}
}
}
]
```

