Realidade Virtual

Programação em RV

Introdução ao VRML 97

Referência

http://www.web3d.org/vrml/vrml.htm

Prefácio
Slides do Curso

Introdução ao VRML 97

Prefácio

Bem vindo aos Slides de *Introdução ao VRML 97*! Estes slides foram escritos para dar a você uma visão rápida e prática de *VRML 97*, a Virtual Reality Modeling Language. Para isso estão incluidos aproximadamente 500 slides. Alguns tem imagens (aproximadamente 200 imagens) e outros com exemplos, sendo mais de 100 exemplos VRML.

Uma palavra sobre as versões VRML

VRML sofreu diversas alterações, o seu início foi no final de 1994. Estes slides são da versão *VRML 97*, a última versão da linguagem. Para inseri-lo no contexto, a tabela a seguir mostra uma visão destas versões VRML e os nomes adotados.

Versão	Liberação	Comentários

VRML 1.0	Maio 1995	Iniciado em 1994, a primeira versão do VRML foi baseada pela maior parte no formato aberto de arquivo desenvolvido pela Silicon Gráphics Inc. A especificação do VRML 1,0 foi terminada em maio 1995 e incluíu suporte para construção de formas luzes e texturas. Os plug-ins de browser VRML 1,0 tornaram-se extensamente disponíveis ao final de 1995, embora poucos suportaram o conjunto das características definidas pela especificação do VRML 1,0.
VRML 1.0c	Janeiro 1996	Enquanto os vendedores começaram a produzir browsers do VRML 1,0, um número de ambigüidades na especificação do VRML 1,0 apareceram. Estes problemas foram corrigidos em uma especificação nova do VRML 1.0c (esclarecido) liberada em janeiro 1996. Nenhuma característica nova foi adicionada à língua em VRML 1.0c
VRML 1.1	cancelado	No final de 1995, a discussão começou em extensões à especificação do VRML 1,0. Estas extensões pretendiam dirigir-se a características de línguagem que na execução feita pelo browser eram difíceis ou ineficientes. A língua estendida foi chamado tentativamente de VRML 1,1. Estes acréscimos foram deixados para mais tarde, no desenvolvimento do VRML 2,0. Não existem browsers VRML 1.1.
Moving Worlds	Janeiro 1996	As características incluídas no VRML 1,0 para a construção estática, não trocando os mundos, apropriados para arquitetura walk-throughs e algumas aplicações científicas de visualização. Para estender a língua a animação e à interação, o grupo da arquitetura VRML fêz um atendimento para propostas para um novo projeto da língua. Silicon Graphics, Netscape, e outros trabalharam juntos para criar a proposta movimentando os mundos, submetida em janeiro 1996. Que a proposta mais tarde foi aceita e se transformou como o ponto de início para desenvolver VRML 2,0. A especificação final da língua do VRML 2,0 é consultada ainda às vezes como à especificação movimentando os mundos, embora difere significativamente da proposta original de movimentar mundos.

VRML 2.0	Agosto 1996	Após sete meses do esforço intenso pela comunidade do VRML, a proposta de movimentação dos mundos evoluiu para transformar-se a especificação final do VRML 2,0, liberada em agosto 1996. A especificação nova reprojetou a sintaxe do VRML e adicionou um conjunto extensivo de características novas para a construção de formas, animações, interações, sons, névoa, os fundos, e as extensões de língua. As versões beta do plug-in VRML 2,0 dos browsers estiveram disponíveis desde o final de 1997. Entretanto, desde esta escrita (maio 1997) não há ainda nenhum browser, completo do VRML 2,0 disponível no mercado.
VRML 97	Setembro 1997	No início de 1997, os esforços começaram para apresentar a especificação do VRML 2,0 à organização de padrões internacional (ISO) que cobre a maioria das principais especificações no uso de línguas na comunidade computacional. A versão ISO do VRML 2,0 foi revista e a especificação foi reescrita significativamente para esclarecer edições. Algumas mudanças menores à língua foram feitas também. O VRML final do ISO era VRML 97. As características da especificação do VRML 97 finalizaram em março 1997, e o texto da especificação finalizou em setembro 1997. Uma versão beta do plug-in do browser do VRML 97 está disponível como se escreve: Silicon Graphics Cosmo Player for SGI platforms. Mais browsers do VRML 97 aguardam-se nos poucos meses seguintes.

O VRML 2,0 difere radicalmente do VRML 1,0 na sintaxe e nas características. Um browser do VRML 1,0 não pode carregar mundos do VRML 2,0. A maioria dos browsers do VRML 2,0, entretanto, podem carregar mundos do VRML 1,0.

O VRML 97 difere bem menos do VRML 2,0. Na maioria de casos, um browser do VRML 2,0 poderá carregar corretamente arquivos do VRML 97. Entretanto, para a exatidão de 100%, você deve ter um browser do VRML 97 para ver os arquivos do VRML contidas dentro destas notas tutoriais.

Introdução ao VRML 97

Sumário

Parte 1 - Forma, Geometria e Aparência

Bem Vindo

Introdução

Construindo um mundo em VRML

Construindo formas primitivas

Transformando formas

Controlando aparências com material

Agrupando nós

Nomeando nós

Exemplos

Parte 2 - Animação, Sensores e Geometria

Introdução à animação

Animando transformações

Sentindo ações do observador

Construindo formas fora de pontos, linhas e faces

Construindo grids elevados

Construindo formas extruded

Controlando propriedades da geometria baseadas em coordenadas

Exemplos

Parte 3 - Texturas, luzes e meio-ambiente

Mapeando texturas

Controlando como as texturas são mapeadas

Iluminando seu mundo

Adicionando fundos

Adicionando nevoeiro

Adicionando som

Controlando o ponto de vista

Controlando a navegação

Sentindo o observador

Exemplos

Parte 4 - Scripts e protótipos

Controlando detalhes

Introduzindo o uso de script

Escrevendo programas scripts com JavaScript

Escrevendo programas scripts com Java

Criando novos tipos de nós

Fornecendo informações sobre seu mundo

Exemplos

Extensões misturadas

Conclusão

Bem Vindo!

Programação para o curso

Escopo do tutorial

Bem-Vindo!

Programação para o curso

Primeiro Dia - Parte 1	Formas, geometria, aparência	100 minutos
Intervalo		
Primeiro Dia - Parte 2	Animação, sensores, geometria	100 minutos
Segundo Dia - Parte 3	Texturas, luzes, meio-ambiente	100 minutos
Intervalo		
Segundo Dia - Parte 4	Scripts, protótipos	100 minutos

Bem-Vindo!

Escopo do tutorial

- **Este tutorial aborda o *VRML 97***
  - **A revisão ISO standard de VRML 2.0**
- **Você aprenderá:**
  - **Estrutura do arquivo em VRML**
  - **Conceitos e terminologias**
  - **Maioria das sintaxes de construção de formas**
  - **Maioria das sintaxes de sensores e animações**
  - **Maioria das sintaxes de programação com scripts**

Sumário | Índice do Capítulo | Slide Anterior | Slide Seguinte | 4

Introdução

O que é VRML?

O que eu preciso para usar VRML?

Exemplo

Como pode-se usar VRML em uma página Web?

O que eu preciso para desenvolver em VRML?

Eu devo usar um editor de texto?

Eu devo usar um construtor de mundos?

Eu devo usar um gerador de formas?

Eu devo usar um modelador e um conversor de formatos?

Como posso obter software de VRML?

Introdução

O que é VRML?

- **VRML é:**
  - **Uma linguagem em forma de texto para descrever formas em 3-D e interagir com o meio-ambiente**
- **Arquivos texto em VRML usam a extensão `.wrl`**

Introdução

O que você precisa para usar VRML?

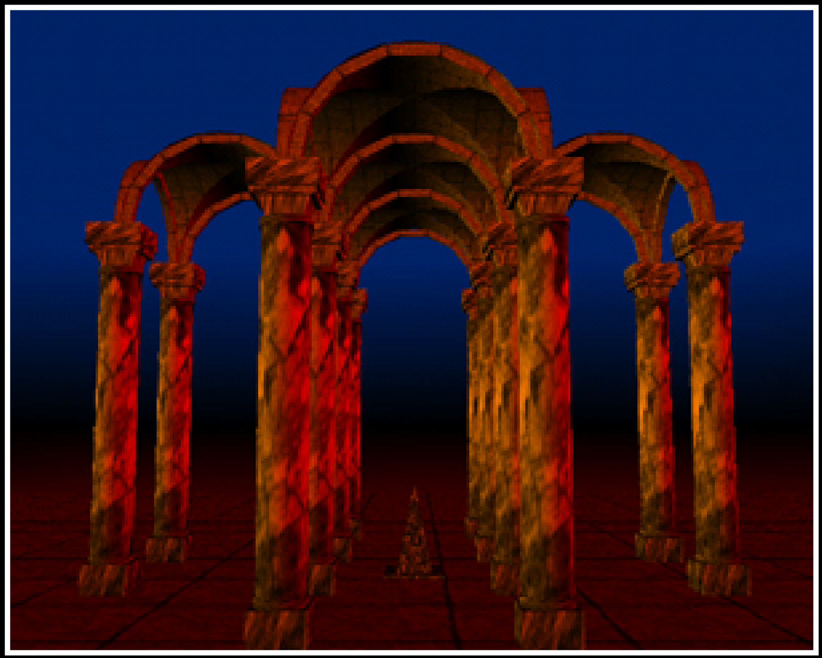
. Você pode ver arquivos em VRML usando *VRML browser*:

- **VRML helper-application**
- **VRML plug-in para um browser em HTML**

. Você pode ver arquivos VRML do disco rígido local ou da Internet

Introdução

Exemplo



[temple.wrl]

Sumário | Índice do Capítulo | Slide Anterior | Slide Seguinte

Introdução

Como VRML pode ser usada em uma página Web?

- Carregando diretamente dentro de um browser Web, preenchendo a página [boxes.wrl]
- Encaixando em uma página, preenchendo um retângulo na página [boxes1.htm]
- Carregando em um frame de página, preenchendo o frame [boxes2.htm]
- Encaixando em um frame de página, preenchendo um frame de retângulo [boxes3.htm]
- Encaixando várias vezes em uma página ou frame [boxes4.htm]

```
#VRML V2.0 utf8
#
#  Temple of the Orb
#  temple.wrl
#        by David R. Nadeau
#
#  This world creates a glowing rotating orb when the pedestal is touched.
#
#  The orb animations include:
#        1.  An OrientationInterpolator rotating an inner emissive textured
#            sphere.
#        2.  An OrientationInterpolator rotating an outer emissive textured
#            semi-transparent sphere.
#        3.  An OrientationInterpolator rotating a pair of rings, angled at
#            a 45 degree angle.
#        4.  An OrientationInterpolator rotating a second pair of rings, angled
#            at a 45 degree angle.
#
#  A ScalarInterpolator with semi-random floating point keyValue that
#  are routed in to a PointLight, thereby causing a flickering
#  effect.  That effect is turned on only after the pedestal has
#  been touched.
#

WorldInfo {
    title "Temple of the Orb"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF EntryView Viewpoint {
    position 0.0 1.6 9.0
    description "Entry View"
}
DEF RightView Viewpoint {
    position 9.0 1.6 0.0
    orientation 0.0 1.0 0.0  1.571
    description "Right View"
}
DEF BackView Viewpoint {
    position 0.0 1.6 -9.0
    orientation 0.0 1.0 0.0  3.14159
    description "Back View"
}
DEF LeftView Viewpoint {
    position -9.0 1.6 0.0
    orientation 0.0 1.0 0.0  4.712
    description "Left View"
}

NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
    speed 2.0
}

Background {
     skyColor [
         0.0 0.1 0.2,
         0.0 0.1 0.3,
         0.0 0.0 0.0,
         0.4 0.0 0.0,
```

```
     ]
     skyAngle [
         1.371,
         1.571,
         1.771,
     ]
}


#
#  Light Sources
#
DEF OrbLight PointLight {
    intensity 0.0
    # animated intensity
    ambientIntensity 0.0
    color 0.3 0.7 1.0
    location 0.0 2.0 0.0
}
DirectionalLight {
    ambientIntensity 0.0
    intensity 0.9
    color 1.0 0.5 0.0
    direction 1.0 0.1 -0.5
}
DirectionalLight {
    ambientIntensity 0.0
    intensity 0.9
    color 1.0 0.0 0.0
    direction -1.0 -0.3 -0.5
}


#
#  Vaulted Ceiling
#
# Center vaulted ceiling
Transform {
    translation 0.0 3.64 0.0
    children [
        DEF VaultedCeiling Group {
            children [
                DEF Vault Inline {
                    url "vault.wrl"
                }
                Transform {
                    rotation 0.0 1.0 0.0 1.57
                    children [ USE Vault ]
                }
                Transform {
                    rotation 0.0 1.0 0.0 3.14
                    children [ USE Vault ]
                }
                Transform {
                    rotation 0.0 1.0 0.0 -1.57
                    children [ USE Vault ]
                }
            ]
        }

        # Left, right, front, and back vaulted ceilings
```

```
        Transform {
            translation -2.0 0.0 0.0
            children [ USE VaultedCeiling ]
        }
        Transform {
            translation  2.0 0.0 0.0
            children [ USE VaultedCeiling ]
        }
        Transform {
            translation 0.0 0.0 -2.0
            children [ USE VaultedCeiling ]
        }
        Transform {
            translation 0.0 0.0 2.0
            children [ USE VaultedCeiling ]
        }
    ]
}

# Columns supporting the vaulted ceilings
Transform {
    translation -3.0 0.0 -1.0
    children [
        DEF Column Transform {
            scale 0.5 0.7 0.5
            children Inline { url "column.wrl" }
        }
    ]
}
Transform {
    translation -1.0 0.0 -1.0
    children [ USE Column ]
}
Transform {
    translation  1.0 0.0 -1.0
    children [ USE Column ]
}
Transform {
    translation  3.0 0.0 -1.0
    children [ USE Column ]
}
Transform {
    translation -3.0 0.0  1.0
    children [ USE Column ]
}
Transform {
    translation -1.0 0.0  1.0
    children [ USE Column ]
}
Transform {
    translation  1.0 0.0  1.0
    children [ USE Column ]
}
Transform {
    translation  3.0 0.0  1.0
    children [ USE Column ]
}
Transform {
    translation -1.0 0.0 -3.0
    children [  USE Column ]
}
```

```
Transform {
    translation -1.0 0.0  3.0
    children [  USE Column ]
}
Transform {
    translation  1.0 0.0 -3.0
    children [  USE Column ]
}
Transform {
    translation  1.0 0.0  3.0
    children [  USE Column ]
}


#
#  Floor
#
Transform {
    translation -11.5 0.0 -11.5
    children Shape {
        appearance Appearance {
            material Material {
                ambientIntensity 0.5
                diffuseColor 1.0 1.0 1.0
            }
            texture ImageTexture { url "stone2.jpg" }
            textureTransform TextureTransform {
                scale 16.0 16.0
            }
        }
        geometry ElevationGrid {
            xDimension 8
            zDimension 8
            xSpacing   3.0
            zSpacing   3.0
            solid      FALSE
            height [
                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
            ]
        }
    }
}


#
#  Orb
#
DEF OrbAll Transform {
    translation 0.0 1.625 0.0
    scale 0.0 0.0 0.0
    # animated scale
    children [

#
```

```
#  Rings
#
        Transform {
            rotation    0.0 0.0 1.0 0.785
            children [
                DEF RingSet1 Transform {
                    rotation 0.0 1.0 0.0 0.0
                    children [
                        DEF Ring1 Transform {
                            translation 0.0 0.0525 0.0
                            rotation    0.0 0.0 1.0 0.35
                            children [
                                DEF Ring Shape {
                                    appearance Appearance {
                                        material Material {
                                            ambientIntensity 0.0
                                            diffuseColor  0.0 0.0 0.0
                                            emissiveColor 0.0 0.3 0.8
                                            specularColor 0.0 0.1 0.1
                                            shininess 2.0
                                        }
                                    }
                                    geometry Cylinder {
                                        height 0.008
                                        radius 0.15
                                        bottom FALSE
                                        top    FALSE
                                    }
                                }
                            ]
                        }
                        DEF Ring2 Transform {
                            translation 0.0 -0.0525 0.0
                            rotation    0.0 0.0 1.0 -0.35
                            children [ USE Ring ]
                        }
                    ]
                }
            ]
        }

        Transform {
            rotation    0.0 0.0 1.0 -0.785
            children [
                DEF RingSet2 Transform {
                    rotation 0.0 1.0 0.0 0.0
                    scale 0.9 0.9 0.9
                    children [ USE Ring1, USE Ring2 ]
                }
            ]
        }


#
#  Glowing orb
#
        DEF Orb Transform {
            rotation    0.0 1.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
```

```
                        material NULL  # force emissive texture
                        texture DEF OrbTexture ImageTexture { url "brtsky.jpg" }
                        textureTransform TextureTransform {
                            scale 2.0 2.0
                        }
                    }
                    geometry Sphere { radius 0.1 }
                }
            ]
        }

        DEF SheerOrb Transform {
            rotation 0.0 1.0 0.0 0.0
            children [
                DirectionalLight { ambientIntensity 1.0 }
                Shape {
                    appearance Appearance {
                        material Material {
                            transparency 0.5
                            emissiveColor 0.0 0.0 0.0
                            diffuseColor  1.0 1.0 1.0
                        }
                        texture USE OrbTexture
                    }
                    geometry Sphere { radius 0.115 }
                }
            ]
        }
    ]
}


#
#  Sound Sources
#
Transform {
    translation 0.0 1.625 0.0
    children [
        DEF OrbSound Sound {
            source DEF OrbClip AudioClip {
                url "willow1.wav"
                loop FALSE
                startTime -1.0
                stopTime  0.0
            }
            maxFront 15.0
            maxBack 15.0
            minFront 1.0
            minBack 1.0
            spatialize TRUE
            intensity 0.0
            # animated intensity
        }
    ]
}


#
#  Pedestal
#
Transform {
```

```
    translation 0.0 0.015 0.0
    scale 3.25 3.25 3.25
    children [
        Shape {
            appearance Appearance {
                material DEF GranitMaterial Material {
                    ambientIntensity 0.0
                    diffuseColor 0.8 0.8 1.0
                }
                texture DEF GranitTexture ImageTexture {
                    url "granit_g.jpg"
                }
            }
            geometry IndexedFaceSet {
                coord Coordinate {
                    point [
                        # Around the base
                        -0.06 0.0  0.06,
                         0.06 0.0  0.06,
                         0.06 0.0 -0.06,
                        -0.06 0.0 -0.06,
                        # Tip
                         0.0  0.3  0.0,
                    ]
                }
                coordIndex [
                    0, 1, 4, -1,  1, 2, 4, -1,
                    2, 3, 4, -1,  3, 0, 4, -1,
                ]
                texCoord TextureCoordinate {
                    point [
                        0.0 0.0, 1.0 0.0, 0.5 1.0,
                    ]
                }
                texCoordIndex [
                    0, 1, 2, -1,  0, 1, 2, -1,
                    0, 1, 2, -1,  0, 1, 2, -1,
                ]
                solid TRUE
            }
        }
        DEF PedestalTouch TouchSensor { }
    ]
}


#
#  Pedestal base
#
Transform {
    translation 0.0 0.0075 0.0
    scale 3.25 3.25 3.25
    children [
        Shape {
            appearance Appearance {
                material USE GranitMaterial
                texture  USE GranitTexture
            }
            geometry Box {
                size 0.2 0.015 0.2
            }
```

```
        }
    ]
}



#
#  Orb Rotation
#
DEF OrbTimer TimeSensor {
    cycleInterval 20.0
    loop TRUE
    startTime 0.0
    stopTime -1.0
}

DEF SheerOrbTimer TimeSensor {
    cycleInterval 21.0
    loop TRUE
    startTime 0.0
    stopTime -1.0
}


DEF OrbRotator OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 1.0 0.0 0.0, 0.0 1.0 0.0 3.14, 0.0 1.0 0.0 6.28 ]
}
ROUTE OrbTimer.fraction_changed TO OrbRotator.set_fraction
ROUTE OrbRotator.value_changed  TO Orb.set_rotation

DEF SheerOrbRotator OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.5 1.0 0.0 0.0, 0.5 1.0 0.0 3.14, 0.5 1.0 0.0 6.28 ]
}
ROUTE SheerOrbTimer.fraction_changed TO SheerOrbRotator.set_fraction
ROUTE SheerOrbRotator.value_changed  TO SheerOrb.set_rotation


#
#  Ring Rotation
#
DEF RingSet1Timer TimeSensor {
    cycleInterval 2.0
    loop TRUE
    startTime 0.0
    stopTime -1.0
}

DEF RingSet2Timer TimeSensor {
    cycleInterval 2.2
    loop TRUE
    startTime 0.0
    stopTime -1.0
}


DEF RingSet1Rotator OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 1.0 0.0 0.0, 0.0 1.0 0.0 -3.14, 0.0 1.0 0.0 -6.28 ]
}
ROUTE RingSet1Timer.fraction_changed TO RingSet1Rotator.set_fraction
ROUTE RingSet1Rotator.value_changed  TO RingSet1.set_rotation
```

```
DEF RingSet2Rotator OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 1.0 0.0 0.0, 0.0 1.0 0.0 3.14, 0.0 1.0 0.0 6.28 ]
}
ROUTE RingSet2Timer.fraction_changed TO RingSet2Rotator.set_fraction
ROUTE RingSet2Rotator.value_changed  TO RingSet2.set_rotation


#
#  Orb Lighting
#
DEF OrbLightTimer TimeSensor {
    cycleInterval 12.0
    loop FALSE
    startTime -1.0
    stopTime 0.0
}

DEF OrbLightIntensity ScalarInterpolator {
    key [ 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 ]
#    keyValue [ 0.2, 1.0, 0.3, 0.4, 0.9, 0.4, 1.0, 0.3, 0.8, 0.5 ]
    keyValue [ 0.0, 0.5, 0.7, 1.0, 0.9, 0.8, 1.0, 0.9, 0.9, 0.5, 0.0 ]
}
ROUTE OrbLightTimer.fraction_changed  TO OrbLightIntensity.set_fraction
ROUTE OrbLightIntensity.value_changed TO OrbLight.set_intensity


#
#  Orb Scaling
#
DEF OrbScaleTimer TimeSensor {
    cycleInterval 12.0
    loop FALSE
    startTime -1.0
    stopTime 0.0
}

DEF OrbScaler PositionInterpolator {
    key [ 0.0, 0.3, 0.9, 1.0 ]
    keyValue [
        0.001 0.001 0.001,
        3.25 3.25 3.25,
        3.25 3.25 3.25,
        0.001 0.001 0.001,
    ]
}

DEF OrbVolume ScalarInterpolator {
    key [ 0.0, 0.3, 0.9, 1.0 ]
    keyValue [
        0.0, 1.0, 1.0, 0.0
    ]
}

ROUTE PedestalTouch.touchTime TO OrbScaleTimer.set_startTime
ROUTE PedestalTouch.touchTime TO OrbLightTimer.set_startTime
ROUTE PedestalTouch.touchTime TO OrbClip.set_startTime
ROUTE OrbScaleTimer.fraction_changed TO OrbScaler.set_fraction
ROUTE OrbScaleTimer.fraction_changed TO OrbVolume.set_fraction
ROUTE OrbScaler.value_changed TO OrbAll.set_scale
ROUTE OrbVolume.value_changed TO OrbSound.set_intensity
```

Introdução

O que eu preciso para desenvolver em VRML?

. Você pode construir arquivos VRML usando:

- **Um editor de texto**
- **Um construtor de mundos**
- **Um gerador de formas**
- **Um modelador e um conversor de formatos**

```
#VRML V2.0 utf8
#
#  Tumbling Boxes
#  boxes.wrl
#        by David R. Nadeau
#
#  This world builds a batch of perpetually tumbling boxes.
#

WorldInfo {
    title "Tumbling Boxes"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

Viewpoint {
    position 0.0 0.0 10.0
    description "Entry view"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight TRUE
}

Background {
    skyColor [
        0.0 0.0 0.0,
        0.0 0.0 0.0,
        0.0 0.8 0.2,
        0.0 0.0 0.0,
    ]
    skyAngle [
        1.37
        1.57,
        1.77,
    ]
}


#
#  Since we need multiple tumbling shapes, create a generic
#  tumble group to automatically tumble any given group of
#  shapes.
#
PROTO TumbleGroup [
    field MFNode children [ ]
    field SFTime xTumbleInterval 3.0
    field SFTime yTumbleInterval 1.0
    field SFTime zTumbleInterval 5.0
] {
    DEF XRot Transform {
        children DEF YRot Transform {
            children DEF ZRot Transform {
                children IS children
            }
        }
    }
    DEF XClock TimeSensor {
        cycleInterval IS xTumbleInterval
        loop TRUE
        startTime 1.0
        stopTime 0.0
```

```
    }
    DEF YClock TimeSensor {
        cycleInterval IS yTumbleInterval
        loop TRUE
        startTime 1.0
        stopTime 0.0
    }
    DEF ZClock TimeSensor {
        cycleInterval IS zTumbleInterval
        loop TRUE
        startTime 1.0
        stopTime 0.0
    }
    DEF XRotIt OrientationInterpolator {
        key [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
        keyValue [
            1.0 0.0 0.0  0.0,
            1.0 0.0 0.0  1.57,
            1.0 0.0 0.0  3.14,
            1.0 0.0 0.0  4.71,
            1.0 0.0 0.0  6.28,
        ]
    }
    DEF YRotIt OrientationInterpolator {
        key [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
        keyValue [
            0.0 1.0 0.0  0.0,
            0.0 1.0 0.0  1.57,
            0.0 1.0 0.0  3.14,
            0.0 1.0 0.0  4.71,
            0.0 1.0 0.0  6.28,
        ]
    }
    DEF ZRotIt OrientationInterpolator {
        key [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
        keyValue [
            0.0 0.0 1.0  0.0,
            0.0 0.0 1.0  1.57,
            0.0 0.0 1.0  3.14,
            0.0 0.0 1.0  4.71,
            0.0 0.0 1.0  6.28,
        ]
    }
    ROUTE XClock.fraction_changed TO XRotIt.set_fraction
    ROUTE YClock.fraction_changed TO YRotIt.set_fraction
    ROUTE ZClock.fraction_changed TO ZRotIt.set_fraction
    ROUTE XRotIt.value_changed TO XRot.set_rotation
    ROUTE YRotIt.value_changed TO YRot.set_rotation
    ROUTE ZRotIt.value_changed TO ZRot.set_rotation
}


TumbleGroup {
    children DEF Block Shape {
        appearance Appearance {
            material Material { }
            texture ImageTexture { url "brtsky.jpg" }
        }
        geometry Box { size 4.0 4.0 4.0 }
    }
    xTumbleInterval 27.0
```

```
    yTumbleInterval 13.0
    zTumbleInterval 45.0
}

TumbleGroup {
    children USE Block
    xTumbleInterval 29.0
    yTumbleInterval 31.0
    zTumbleInterval 25.0
}

TumbleGroup {
    children USE Block
    xTumbleInterval 38.0
    yTumbleInterval 27.0
    zTumbleInterval 43.0
}
```

Como o VRML pode ser usado em uma Web page?

Uma amostra encaixada em um browser VRML

Uma amostra de texto no lado esquerdo e encaixado no browser VRML

Uma amostra de texto no lado direito e encaixado no browser VRML

Como VRML pode ser usada em uma página Web?

Uma amostra de frame com um browser VRML

```
#VRML V2.0 utf8
#
#  Tumbling Boxes
#  boxes.wrl
#        by David R. Nadeau
#
#  This world builds a batch of perpetually tumbling boxes.
#

WorldInfo {
    title "Tumbling Boxes"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

Viewpoint {
    position 0.0 0.0 10.0
    description "Entry view"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight TRUE
}

Background {
    skyColor [
        0.0 0.0 0.0,
        0.0 0.0 0.0,
        0.0 0.8 0.2,
        0.0 0.0 0.0,
    ]
    skyAngle [
        1.37
        1.57,
        1.77,
    ]
}


#
#  Since we need multiple tumbling shapes, create a generic
#  tumble group to automatically tumble any given group of
#  shapes.
#
PROTO TumbleGroup [
    field MFNode children [ ]
    field SFTime xTumbleInterval 3.0
    field SFTime yTumbleInterval 1.0
    field SFTime zTumbleInterval 5.0
] {
    DEF XRot Transform {
        children DEF YRot Transform {
            children DEF ZRot Transform {
                children IS children
```

```
            }
        }
    }
    DEF XClock TimeSensor {
        cycleInterval IS xTumbleInterval
        loop TRUE
        startTime 1.0
        stopTime 0.0
    }
    DEF YClock TimeSensor {
        cycleInterval IS yTumbleInterval
        loop TRUE
        startTime 1.0
        stopTime 0.0
    }
    DEF ZClock TimeSensor {
        cycleInterval IS zTumbleInterval
        loop TRUE
        startTime 1.0
        stopTime 0.0
    }
    DEF XRotIt OrientationInterpolator {
        key [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
        keyValue [
            1.0 0.0 0.0  0.0,
            1.0 0.0 0.0  1.57,
            1.0 0.0 0.0  3.14,
            1.0 0.0 0.0  4.71,
            1.0 0.0 0.0  6.28,
        ]
    }
    DEF YRotIt OrientationInterpolator {
        key [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
        keyValue [
            0.0 1.0 0.0  0.0,
            0.0 1.0 0.0  1.57,
            0.0 1.0 0.0  3.14,
            0.0 1.0 0.0  4.71,
            0.0 1.0 0.0  6.28,
        ]
    }
    DEF ZRotIt OrientationInterpolator {
        key [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
        keyValue [
            0.0 0.0 1.0  0.0,
            0.0 0.0 1.0  1.57,
            0.0 0.0 1.0  3.14,
            0.0 0.0 1.0  4.71,
            0.0 0.0 1.0  6.28,
        ]
    }
    ROUTE XClock.fraction_changed TO XRotIt.set_fraction
    ROUTE YClock.fraction_changed TO YRotIt.set_fraction
    ROUTE ZClock.fraction_changed TO ZRotIt.set_fraction
    ROUTE XRotIt.value_changed TO XRot.set_rotation
    ROUTE YRotIt.value_changed TO YRot.set_rotation
    ROUTE ZRotIt.value_changed TO ZRot.set_rotation
}

TumbleGroup {
```

```
    children DEF Block Shape {
        appearance Appearance {
            material Material { }
            texture ImageTexture { url "brtsky.jpg" }
        }
        geometry Box { size 4.0 4.0 4.0 }
    }
    xTumbleInterval 27.0
    yTumbleInterval 13.0
    zTumbleInterval 45.0
}

TumbleGroup {
    children USE Block
    xTumbleInterval 29.0
    yTumbleInterval 31.0
    zTumbleInterval 25.0
}

TumbleGroup {
    children USE Block
    xTumbleInterval 38.0
    yTumbleInterval 27.0
    zTumbleInterval 43.0
}
```

Como VRML pode ser usada em uma página Web?

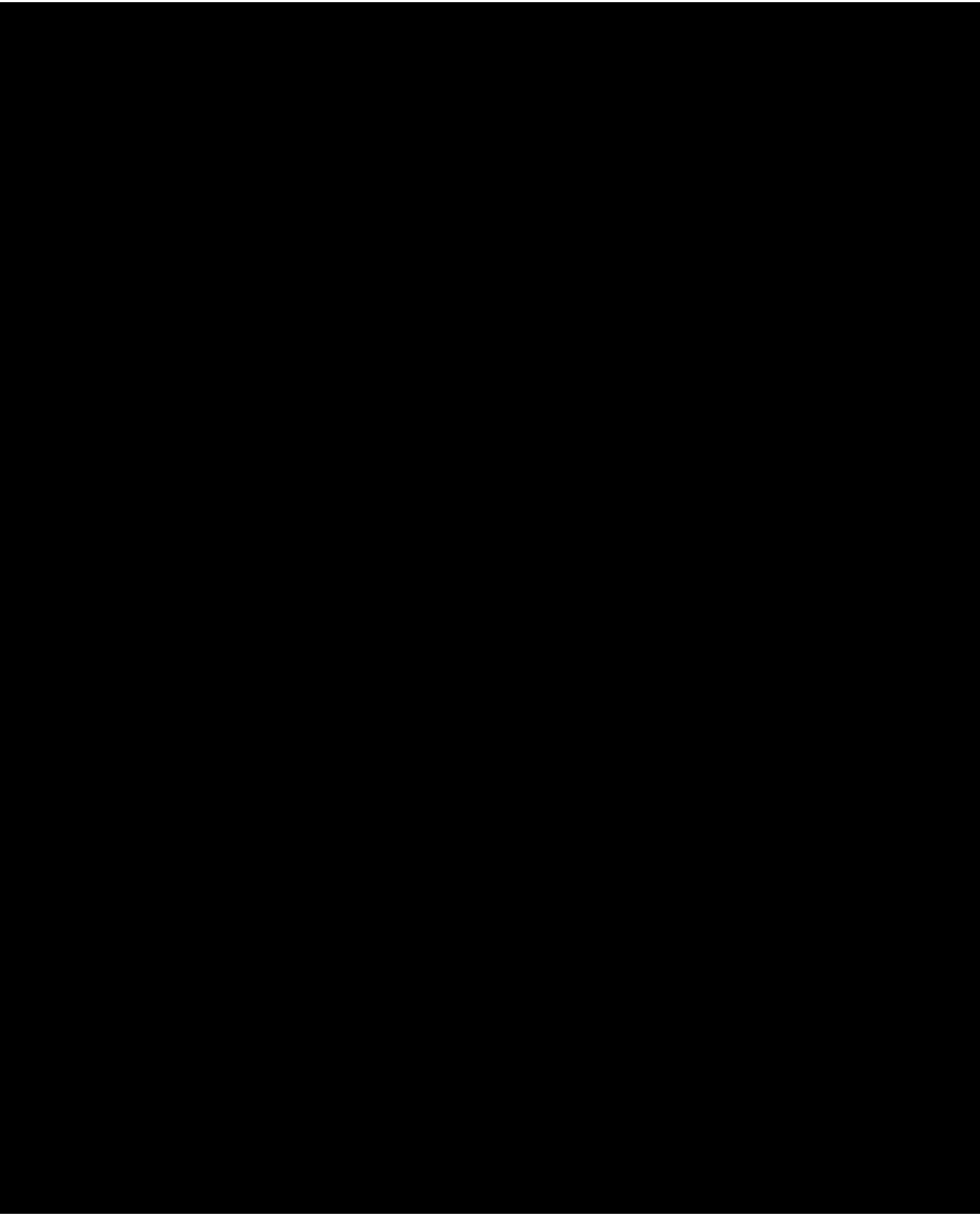
Uma amostra de frame com um browser VRML

Amostra de texto no lado esquerdo e encaixado no browser VRML

Amostra de texto no lado direito e encaixado no browser VRML

Como VRML pode ser usada em uma página Web?

Uma amostra usando browsers VRML multiplos encaixados



Introdução

Devo usar um editor de texto?

- **Pró:**
  - **Não necessita compra de novos softwares**
  - **Acesso a todas as características de VRML**
  - **Controle detalhado do mundo**
- **Contra:**
  - **Difícil de compor formas complexas em 3D**
  - **Requer conhecimento da sintaxe da VRML**

Introdução

Devo usar um construtor de mundos?

- **Pró:**
  - **Interface fácil para desenhar em 3D**
  - **Pouco esforço para aprender a sintaxe VRML**
- **Contra:**
  - **Pode não suportar todas as características da VRML**
  - **Pode não produzir VRML eficientemente**

Introdução

Devo usar um gerador de formas?

- **Pró:**
  - **Fácil de gerar formas complexas**
- **Montanhas de fractal, logos, etc.**
- **Contra:**
  - **Pequeno conjunto de formas**
  - **Melhor quando usado com outros softwares**

Introdução

Devo usar um modelador ou conversor de formatos?

- **Pró:**
  - **Várias característica poderosas disponíveis**
  - **Pode fazer imagens foto-realísticas, também**
- **Contra:**
  - **Pode não suportar todas as característica da VRML**
  - **Não foi projetado para VRML**

Introdução

Como posso obter software de VRML?

. O repositório de VRML mantém links para softwares disponíveis:

http://www.web3d.org/vrml/vrml.htm

Construindo um mundo em VRML

Estrutura de um arquivo em VRML

Um arquivo em VRML

Entendendo o cabeçalho

Entendendo o UTF8

Usando comentários

Usando nós

Usando campos e valores

Usando campos e valores

Resumo

Construindo um mundo VRML

Estrutura de um arquivo VRML

- **Arquivos VRML contém:**
  - **O cabeçalho do arquivo**
  - ***Comentários***
  - ***Nós***
  - ***Campos***
  - ***Valores***
  - **mais. . .**

Construindo um mundo VRML

Um exemplo de arquivo VRML

```
#VRML V2.0 utf8
# A Cylinder
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        height 2.0
        radius 1.5
    }
}
```

Construindo um mundo VRML

Entendendo o cabeçalho

```
#VRML V2.0 utf8
```

. `#VRML`: Arquivo contém texto em VRML

. `V2.0` : Texto conforme a sintaxe da versão 2.0

. `utf8` : Texto usa o conjunto de caracteres UTF8

Construindo um mundo VRML

Entendendo UTF8

- `utf8` é um conjunto internacional de caracteres
- `utf8` significa:
  - UCS (Universal Character Set) Formato de transformação, 8-bit
- Codifica 24.000+ caracteres de muitas línguas
  - ASCII é um subconjunto

Construindo um mundo VRML

Usando comentários

```
# A Cylinder
```

. Comentário inicia com o símbolo # e termina no final da linha

Construindo um mundo VRML

Usando nós

```
Cylinder {
}
```

- **Nós descrevem forma, luzes, sons, etc.**
- **Todo nó possui:**
  - **Um *tipo de nó* (`Shape`, `Cylinder`, etc.)**
  - **Um par de chaves (abre e fecha)**
  - **Zero ou mais campos dentro das chaves**

Construindo um mundo VRML

Usando campos e valores

```
Cylinder {
    height 2.0
    radius 1.5
}
```

. **Campos descrevem atributos do nó**

Construindo um mundo VRML

Usando campos e valores

```
height 2.0
```

. Todo campo possui:

. Um nome do campo

. Um tipo de dado (float, int, etc.)

. Um valor default

. Campos são adicionais e são colocados em qualquer ordem

. O valor default é usado quando o campo não é fornecido

Construindo um mundo VRML

Resumo

- **O cabeçalho do arquivo fornece a versão e a codificação de caracteres**
- **Nós descrevem o conteúdo de uma cena**
- **Campos e valores especificam os atributos dos nós**

Construindo formas primitivas

Motivação

Exemplo

Sintaxe: Forma

Especificando a geometria

Sintaxe: Box (Caixa)

Sintaxe: Cone

Sintaxe: Cylinder (Cilindro)

Sintaxe: Sphere (Esfera)

Sintaxe: Text (Texto)

Um exemplo de forma primitiva

Um exemplo de forma primitiva

Construindo múltiplas formas

Um exemplo de arquivo com múltiplas formas

Um exemplo de arquivo com múltiplas formas

Sintaxe: FontStyle (Estilo de fonte)

Sintaxe: FontStyle (Estilo de fonte)

Resumo

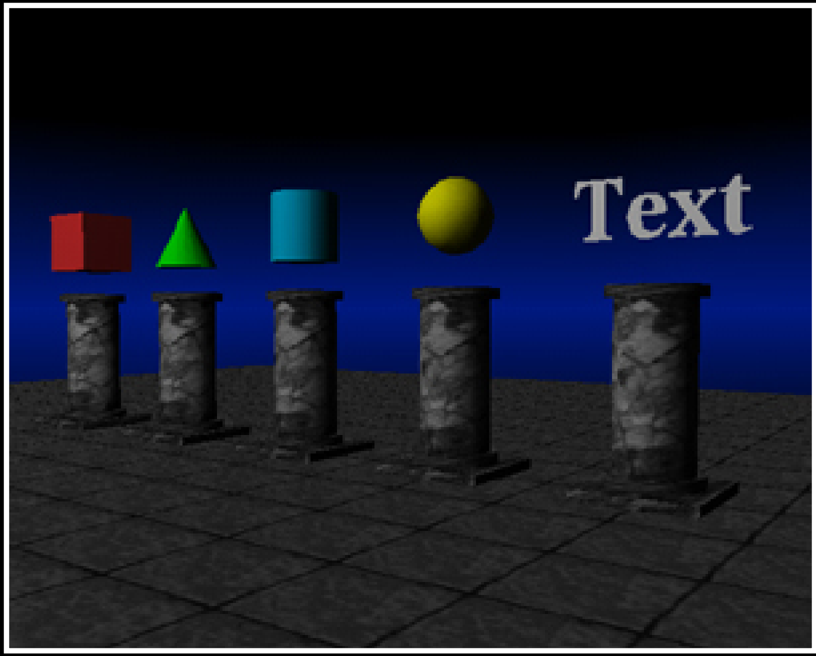
Construindo formas primitivas

Motivação

- ***Formas* são os blocos construtores de um mundo VRML**
- ***Formas Primitivas* são blocos construtores padrões:**
  - **Box**
  - **Cone**
  - **Cylinder**
  - **Sphere**
  - **Text**

Construindo formas primitivas

Exemplo



[prim.wrl]

Construindo formas primitivas

Sintaxe: Shape

- **Um nó `Shape` constrói uma forma**
  - ***appearance* - cor e textura**
  - ***geometry* - forma ou estrutura**

```
Shape {
    appearance . . .
    geometry   . . .
}
```

Construindo formas primitivas

Especificando a geometria

. Formas geométricas são construidas com *nós geométricos*:

```
Box      { . . . }
Cone     { . . . }
Cylinder { . . . }
Sphere   { . . . }
Text     { . . . }
```

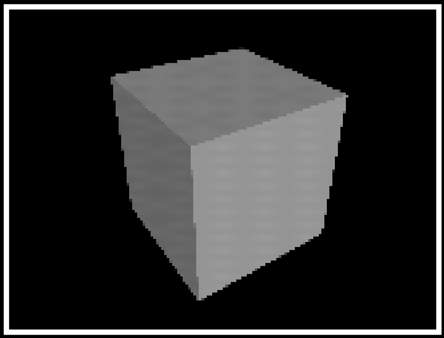
. Os campos dos nós geométricos controlam as dimensões

◦ As dimensões são consideradas em metros, mas podem ser qualquer medida desejada

Construindo formas primitivas

Sintaxe: Box

. Um nó geométrico **Box** constrói uma caixa



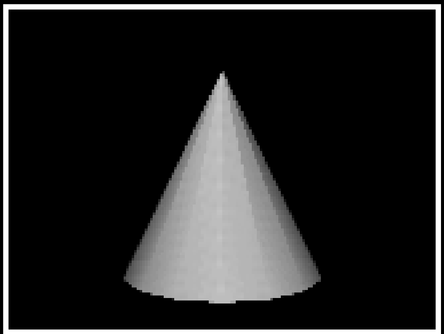
[box.wrl]

```
Box {
    size 2.0 2.0 2.0
}
```

Construindo formas primitivas

Sintaxe: Cone

. Um nó geométrico `Cone` constrói um cone com base para baixo



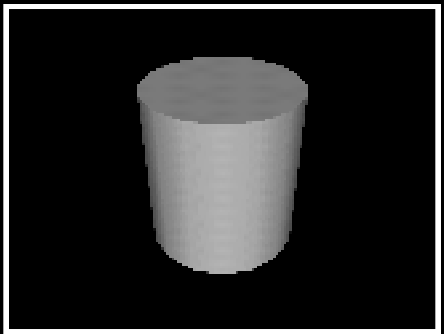
[cone.wrl]

```
Cone {
    height 2.0
    bottomRadius 1.0
}
```

Construindo formas primitivas

Sintaxe: Cylinder

. **Um nó geométrico `Cylinder` constrói um cilindro na vertical**



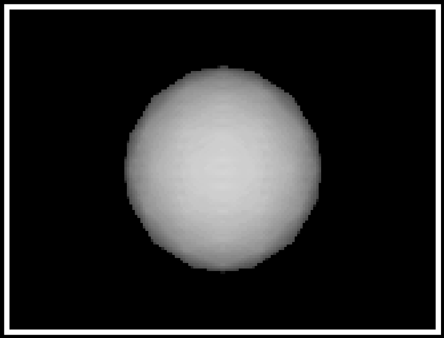
[cyl.wrl]

```
Cylinder {
    height 2.0
    radius 1.0
}
```

Construindo formas primitivas

Sintaxe: Sphere

. Um nó geométrico `Sphere` constrói uma esfera



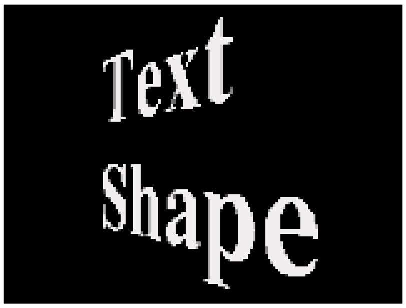
```
Sphere {
    radius 1.0
}
```

[sphere.wrl]

Construindo formas primitivas

Syntax: Text

. Um nó geométrico **Text** constrói um texto



[text.wrl]

```
Text {
    string [ "Text",
             "Shape" ]
    fontStyle FontStyle {
        style "BOLD"
    }
}
```

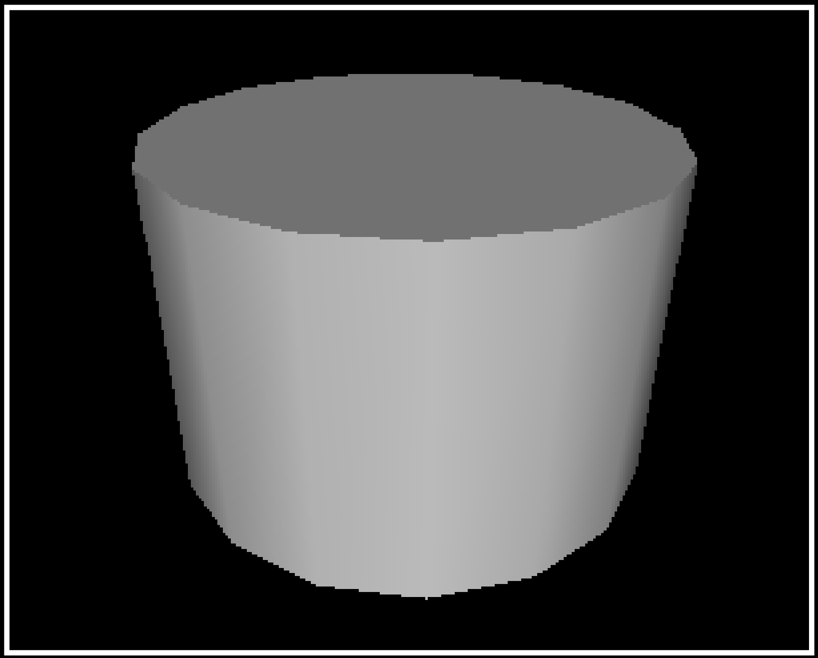
Construindo formas primitivas

Um exemplo de forma primitiva

```
#VRML V2.0 utf8
# A cylinder
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        height 2.0
        radius 1.5
    }
}
```

Construindo formas primitivas

Um exemplo de forma primitiva



[cylinder.wrl]

Construindo formas primitivas

Construindo múltiplas formas

- **Formas são construídas no centro do mundo**
- **Um arquivo VRML pode conter várias formas**
- **Formas são sobrepostas quando construídas na mesma posição**

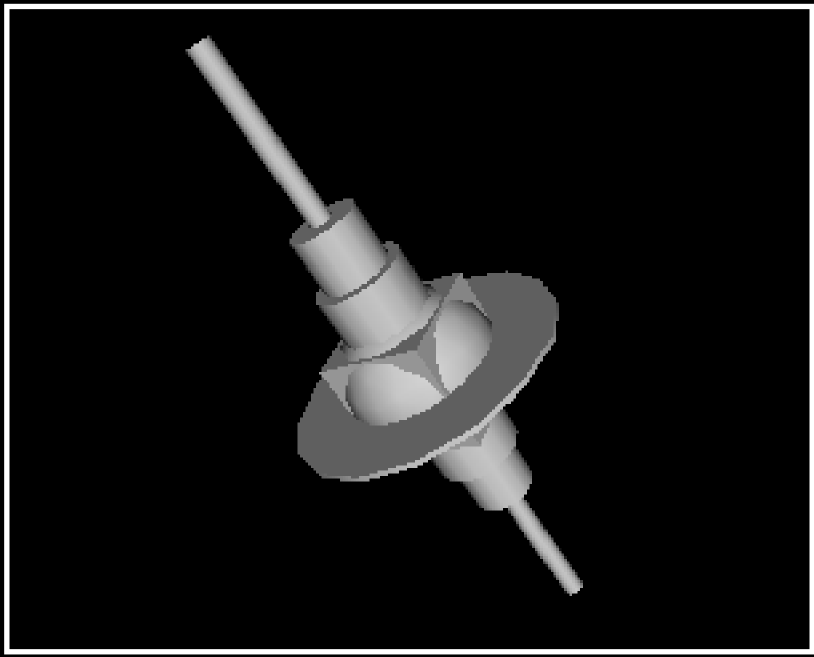
Construindo formas primitivas

Um exemplo de arquivo com múltiplas formas

```
#VRML V2.0 utf8
Shape { . . . }
Shape { . . . }
. . .
Shape { . . . }
```

Construindo formas primitivas

Um exemplo de arquivo com múltiplas formas



[space.wrl]

Construindo formas primitivas

Sintaxe: FontStyle

- **Um nó `FontStyle` descreve uma fonte**
  - ***family* - `SERIF`, `SANS`, or `TYPEWRITER`**
  - ***style* - `BOLD`, `ITALIC`, `BOLDITALIC`, or `PLAIN`,**
  - **mais . . .**

```
Text {
    string . . .
    fontStyle FontStyle {
        family  "SERIF"
        style   "BOLD"
    }
}
```

Construindo formas primitivas

Sintaxe: FontStyle

- **Um nó `FontStyle` descreve uma fonte**
  - ***size* - altura do caractere**
  - ***spacing* - espaçamento de linhas/colunas**
  - **mais . . .**

```
Text {
    string . . .
    fontStyle FontStyle {
        size    1.0
        spacing 1.0
    }
}
```

Construindo formas primitivas

Resumo

- **Formas são construídas usando nós `Shape`**
- **Geometria de forma é construída usando nós geométricos, tais como `Box`, `Cone`, `Cylinder`, `Sphere`, e `Text`**
- **Fontes de texto são controladas usando-se o nó `FontStyle`**

```
#VRML V2.0 utf8
#
#  VRML Primitives
#  prim.wrl
#        by David R. Nadeau
#
#  This file highlights the VRML primitives, placing each one above
#  a gothic pedestal.  Of note is that the pedestal itself is built using
#  only the primitives.
#

WorldInfo {
    title "VRML Primitives"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 5.0 1.6 8.0
    orientation 0.0 1.0 0.0 0.52
    description "Entry View"
}
NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
}

Background {
    skyColor [
        0.0 0.0 0.0,
        0.0 0.0 0.0,
        0.0 0.1 0.5,
        0.0 0.0 0.0,
    ]
    skyAngle [
        1.37,
        1.57,
        1.77,
    ]
}


#
#  Light Sources
#
PointLight {
    ambientIntensity 0.2
    color 0.7 0.7 0.7
    location 0.0 8.0 10.0
}



#
#  Primitives on pedestals
#
# Box
Transform {
    translation -3.0 0.0 0.0
    children [
        # Pedestal
        DEF Pedestal Inline {
            url "pedestal.wrl"
```

```
        }
        # Box Primitive
        Transform {
            translation 0.0 2.5 0.0
            scale 0.375 0.375 0.375
            children [
                Shape {
                    appearance Appearance {
                        material Material { diffuseColor 1.0 0.2 0.2 }
                    }
                    geometry Box { }
                }
            ]
        }
    ]
}

# Cone
Transform {
    translation -1.0 0.0 0.0
    children [
        # Pedestal
        USE Pedestal,
        # Cone Primitive
        Transform {
            translation 0.0 2.5 0.0
            scale 0.375 0.375 0.375
            children [
                Shape {
                    appearance Appearance {
                        material Material { diffuseColor 0.0 1.0 0.0 }
                    }
                    geometry Cone { }
                }
            ]
        }
    ]
}

# Cylinder
Transform {
    translation 1.0 0.0 0.0
    children [
        # Pedestal
        USE Pedestal,
        # Cylinder Primitive
        Transform {
            translation 0.0 2.5 0.0
            scale 0.375 0.375 0.375
            children [
                Shape {
                    appearance Appearance {
                        material Material { diffuseColor 0.0 0.8 1.0 }
                    }
                    geometry Cylinder { }
                }
            ]
        }
    ]
}
```

```
# Sphere
Transform {
    translation 3.0 0.0 0.0
    children [
        # Pedestal
        USE Pedestal,
        # Sphere Primitive
        Transform {
            translation 0.0 2.5 0.0
            scale 0.375 0.375 0.375
            children [
                Shape {
                    appearance Appearance {
                        material Material { diffuseColor 1.0 1.0 0.0 }
                    }
                    geometry Sphere { }
                }
            ]
        }
    ]
}

# Text
Transform {
    translation 5.0 0.0 0.0
    children [
        # Pedestal
        USE Pedestal,
        # Sphere Primitive
        Transform {
            translation 0.0 2.2 0.0
            scale 0.375 0.375 0.375
            children [
                Shape {
                    appearance Appearance {
                        material Material { diffuseColor 1.0 1.0 1.0 }
                    }
                    geometry Text {
                        string "Text"
                        fontStyle FontStyle {
                            size 2.0
                            style "BOLD"
                            family "SERIF"
                            justify "MIDDLE"
                        }
                    }
                }
            ]
        }
    ]
}


#
#  Floor
#
Shape {
    appearance Appearance {
        material Material { ambientIntensity 0.1 }
        texture ImageTexture { url "stone2.jpg" }
        textureTransform TextureTransform { scale 20.0 20.0 }
```

```
    }
    geometry IndexedFaceSet {
        coord Coordinate {
            point [
                -10.0 0.0  10.0,
                 10.0 0.0  10.0,
                 10.0 0.0 -10.0,
                -10.0 0.0 -10.0,
            ]
        }
        coordIndex [ 0, 1, 2, 3, -1, ]
        solid FALSE
    }
}
```

```
#VRML V2.0 utf8
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Box {
        size 2.0 2.0 2.0
    }
}
```

```
#VRML V2.0 utf8
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cone {
        height 2.0
        bottomRadius 1.0
    }
}
```

```
#VRML V2.0 utf8
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        height 2.0
        radius 1.0
    }
}
```

```
#VRML V2.0 utf8
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Sphere {
        radius 1.0
    }
}
```

```
#VRML V2.0 utf8
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Text {
        string [ "Text",
                 "Shape" ]
        fontStyle FontStyle {
            style "BOLD"
        }
    }
}
```

```
#VRML V2.0 utf8
# A cylinder
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        height 2.0
        radius 1.5
    }
}
```

```
#VRML V2.0 utf8
#
#  Space Station
#  space.wrl
#        by David R. Nadeau
#
#  This file illustrates the use of multiple Shape nodes in the same file.
#  Notice that all of the shapes are built overlapping each other!
#

Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Box {
        size 1.0 1.0 1.0
    }
}
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Sphere {
        radius 0.7
    }
}
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        radius 1.25
        height 0.05
    }
}
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        radius 0.4
        height 2.0
    }
}
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        radius 0.3
        height 3.0
    }
}
Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        radius 0.1
        height 6.0
    }
```

```
}
```

Transformando formas

Motivação

Exemplo

Usando o sistema de coordenadas

Visualizando o sistema de coordenadas

Transformando o sistema de coordenadas

Sintaxe: Transform

Incluindo children

Transladando

Transladando

Rotacionando

Especificando o eixo de rotação

Usando a regra da mão direita

Usando a regra da mão direita

Rotacionando

[Escalonando]

[Escalonando]

[Escalonando, rotacionando e transladando]

[Escalonando, rotacionando e transladando]

[Um exemplo de um grupo de transformações]

[Um exemplo de um grupo de transformações]

[Resumo]

Transformando formas

Motivação

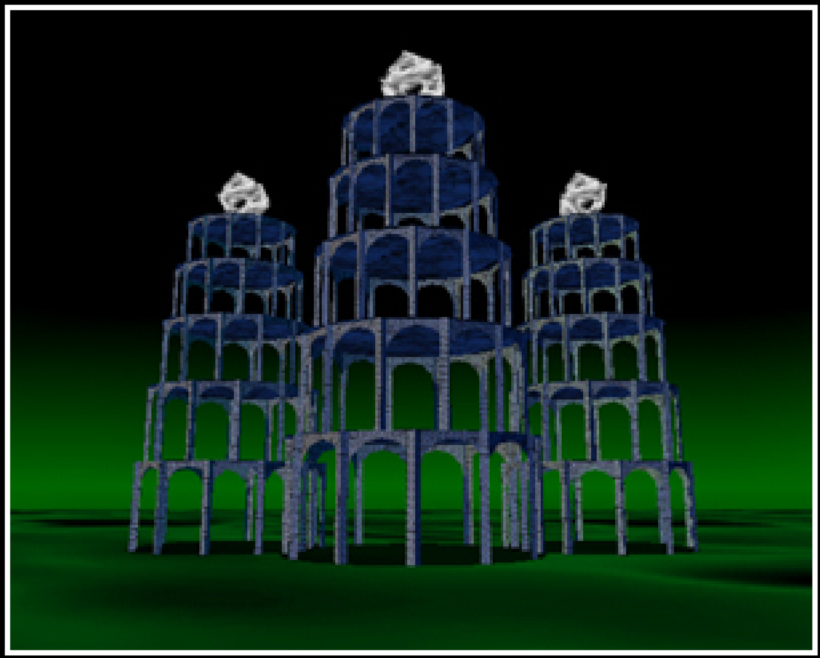
. Inicialmente, todas as formas são construídas no centro do mundo

. Um transformação permite que você

- Posicione formas
- Rotacione formas
- Escalone formas

Transformando formas

Exemplo



[towers.wrl]

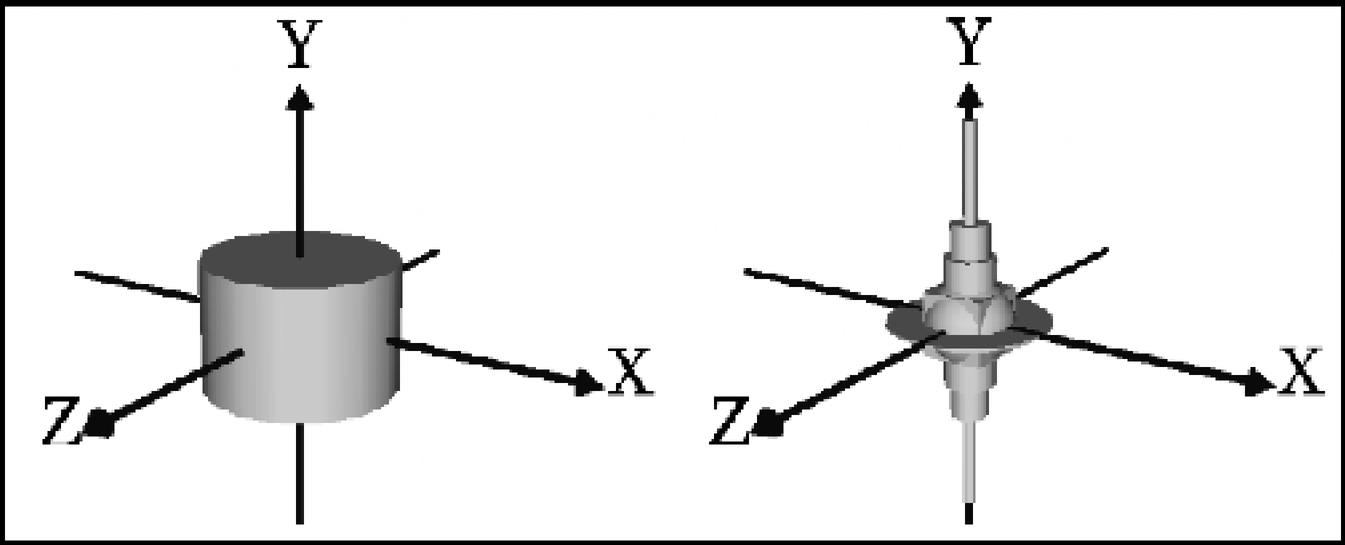
Transformando formas

Usando o sistema de coordenadas

- **Um arquivo VRML constrói componentes para o mundo**
- **Os componentes do mundo são construídos no sistema de coordenada do mundo**
- **Inicialmente, todas as formas são construídas na origem do sistema de coordenada do mundo**

Transformando formas

Visualizando o sistema de coordenadas



Transformando formas

Transformando o sistema de coordenadas

- Uma *transformação* cria um sistema de coordenada que é
- Posicionada
- Rotacionada
- Escalonada

  relativa ao sistema de coordenada inicial
- Formas construídas no novo sistema de coordenadas são posicionadas, rotacionadas e escalonadas de acordo com este sistema

Transformando formas

Sintaxe: Transform

- **O grupo de nó `Transform` cria um grupo com o seu próprio sistema de coordenadas**
  - ***children* - formas para construir**
  - ***translation* - posicionamento**
  - ***rotation* - orientação**
  - ***scale* - tamanho**

```
Transform {
    translation . . .
    rotation    . . .
    scale       . . .
    children  [ . . . ]
}
```

Transformando formas

Incluindo children

. O campo `children` incluí a lista de um ou mais nós

```
Transform {
    . . .
    children [
        Shape { . . . }
        Transform { . . . }
        . . .
    ]
}
```

Transformando formas

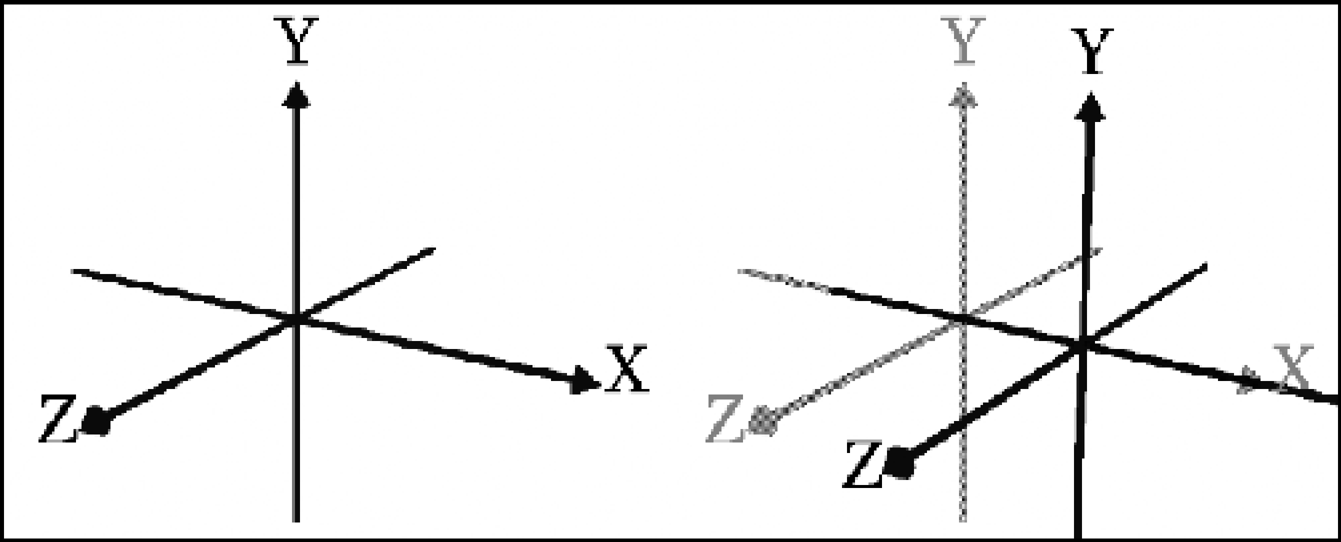
Transladando

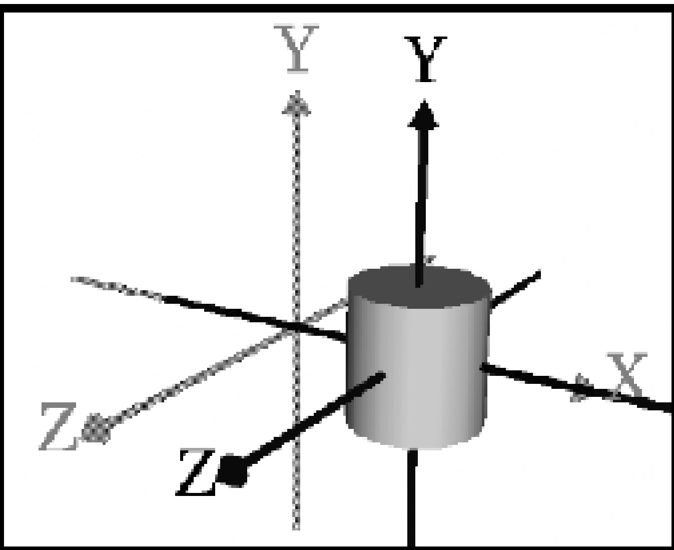
- *Translação* **posiciona um sistema de coordenadas em X, Y, and Z**

```
Transform {
    #               X   Y   Z
    translation 2.0 0.0 0.0
    children [ . . . ]
}
```

Transformando formas

Transladando





Transformando formas

Rotacionando

- *Rotação* **orienta um sistema de coordenadas em torno de um eixo de um determinado ângulo de rotação**
- **Ângulos são medidos em** ***radianos***

```
Transform {
    #          X   Y   Z    Angle
    rotation 0.0 0.0 1.0  0.52
    children [ . . . ]
}
```

Transformando formas

Especificando o eixo de rotação

- **Um eixo de rotação define um pólo para rotacionar**
- **Como os pólos Norte-Sul da Terra**
- **Rotações típicas são realizadas em torno dos eixos X, Y, or Z:**

Rotaciona sobre	**Eixo**
Eixo - X	`1.0 0.0 0.0`
Eixo - Y	`0.0 1.0 0.0`
Eixo - Z	`0.0 0.0 1.0`

Transformando formas

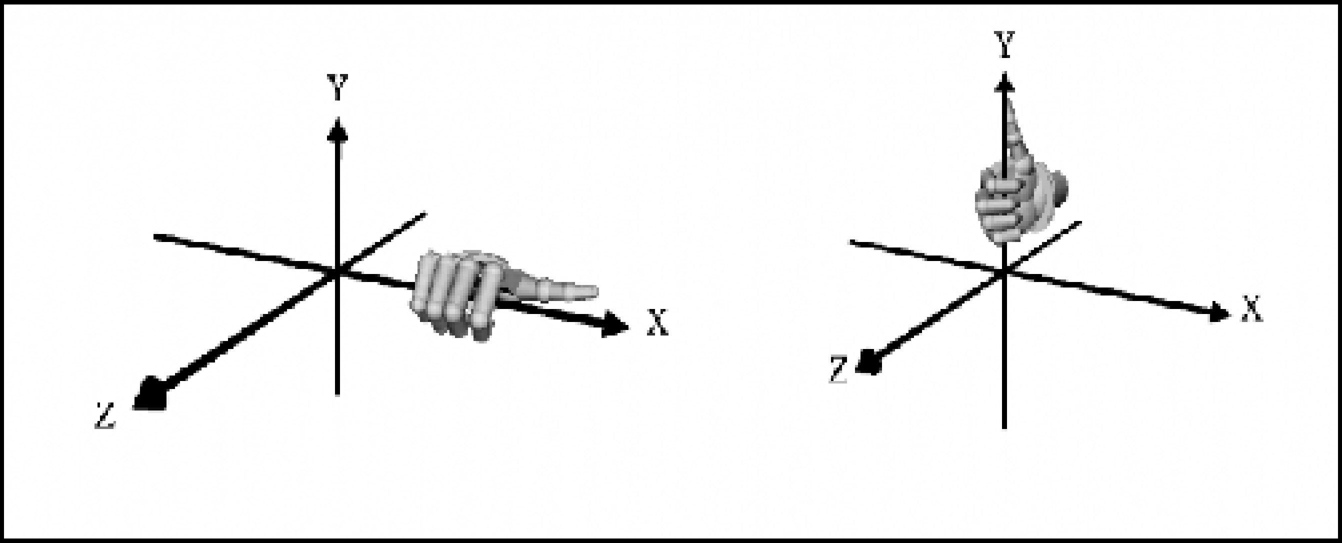
Usando a regra da mão direita

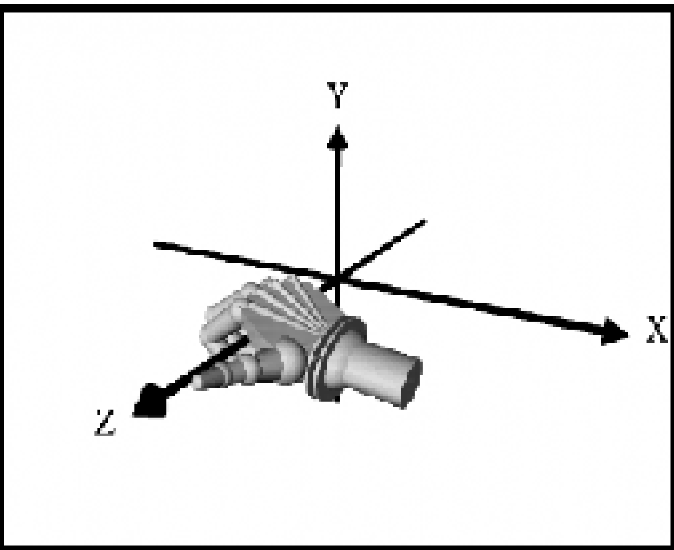
. Para ajudar a lembrar as direções de rotação positivas e negativas:

- **Abra sua mão**
- **Estique seu dedo polegar**
- **Dirija seu polegar na direção positiva do eixo**
- **Curve seus dedos sobre o eixo**
- **A direção da curva dos dedos é o sentido de rotação *positivo***

Transformando formas

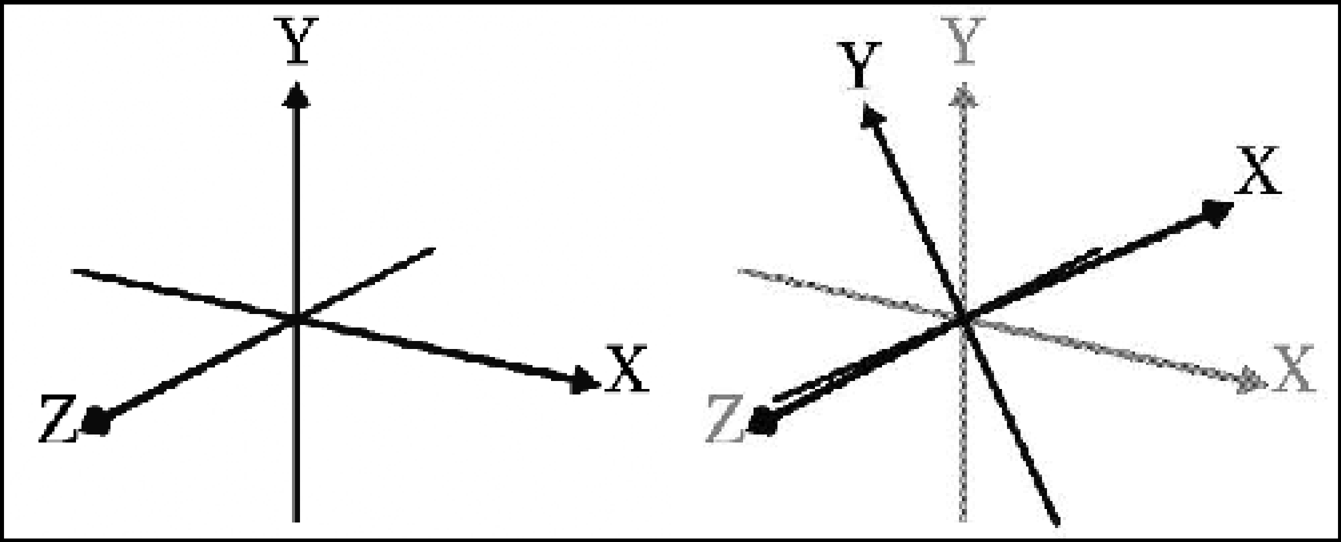
Usando a regra da mão direita

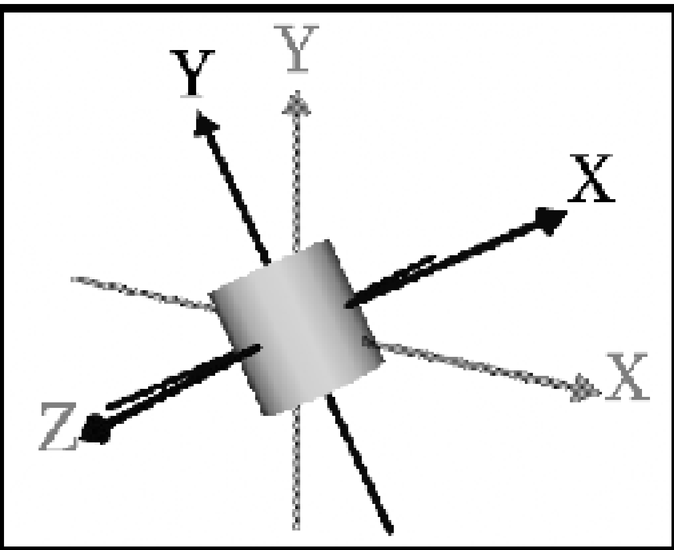




Transformando formas

Rotacionando





Transformando formas

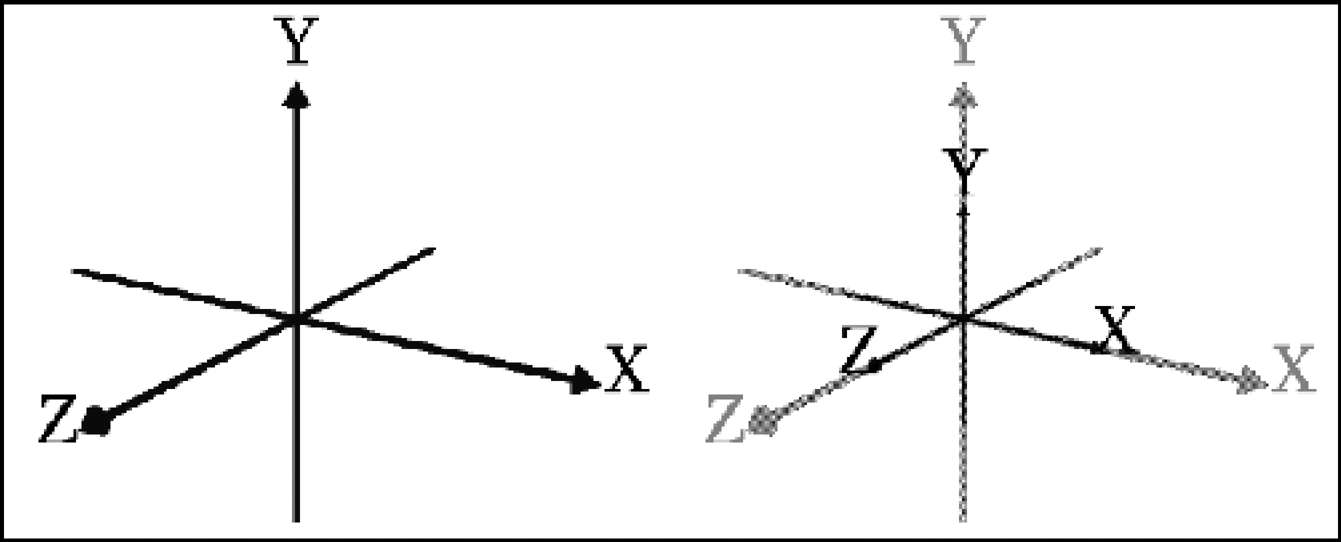
Escalonando

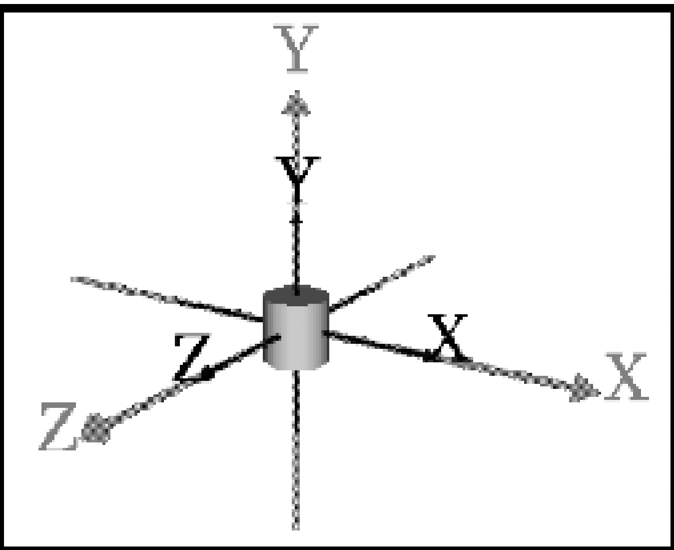
. *Escalar* aumentando ou diminuindo um sistema de coordenadas através do fator de escala em X, Y, e Z

```
Transform {
    #      X    Y    Z
    scale 0.5 0.5 0.5
    children [ . . . ]
}
```

Transformando formas

Escalonando





Transformando formas

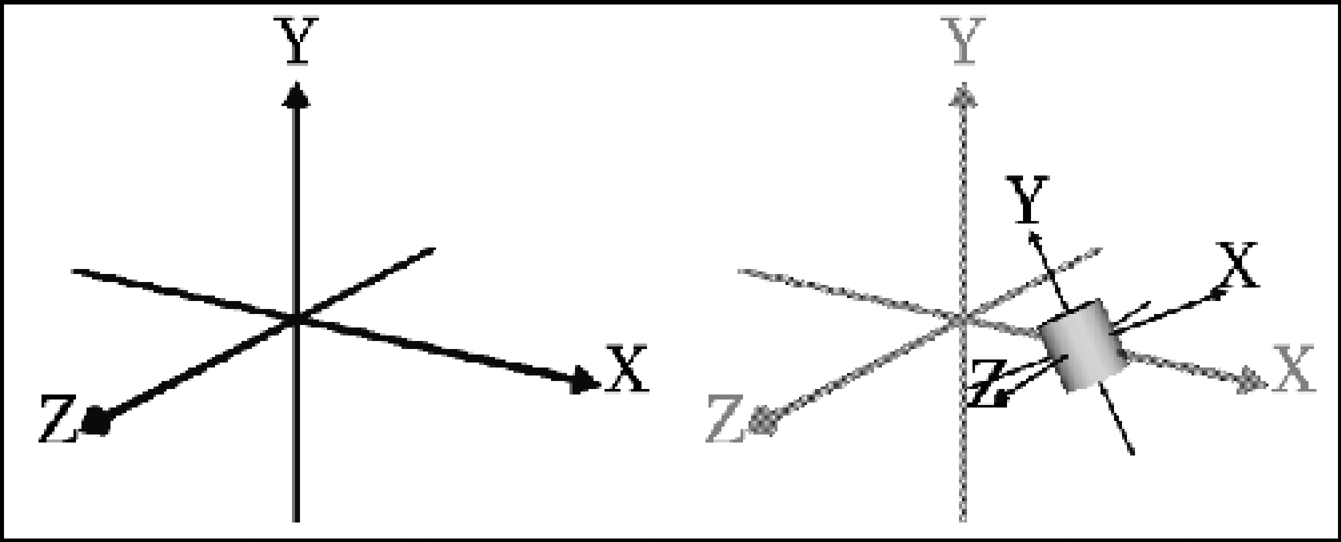
Escalonando, rotacionando e transladando

- *Escalonar, rotacionar e transladar* **um sistema de coordenadas, um após o outro**

```
Transform {
    translation 2.0 0.0 0.0
    rotation 0.0 0.0 1.0  0.52
    scale 0.5 0.5 0.5
    children [ . . . ]
}
```

Transformando formas

Escalonando, rotacionando e transladando



Transformando formas

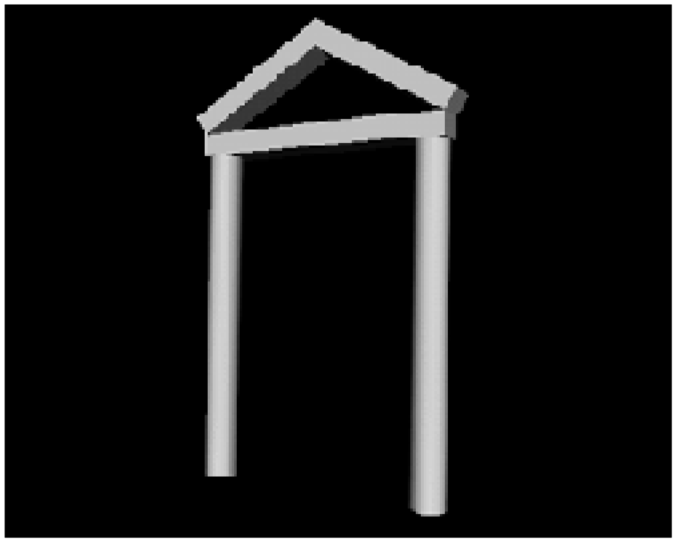
Um exemplo de um grupo de transformações

```
Transform {
    translation 4.0 0.0 0.0
    rotation    0.0 1.0 0.0  0.785
    scale       0.5 0.5 0.5
    children [ . . .  ]
}
```

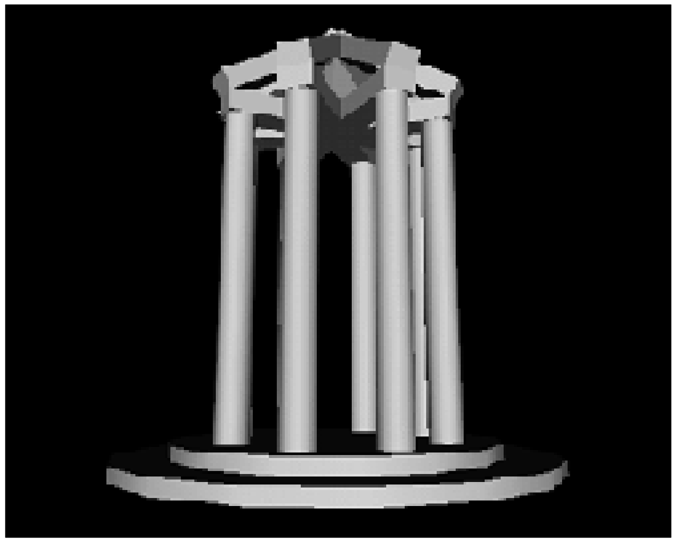
Sumário Índice do Capítulo Slide Anterior Slide Seguinte 63

Transformando formas

Um exemplo de um grupo de transformações



[arch.wrl]



[arches.wrl]

Sumário Índice do Capítulo Slide Anterior Slide Seguinte 64

Transformando formas

Resumo

- **Todas as formas são construídas em um sistema de coordenadas**
- **Um nó `Transform` cria um novo sistema de coordenada relativo ao sistema de coordenada anterior**
- **Os campos do nó `Transform` são**
  - **`translation`**
  - **`rotation`**
  - **`scale`**

```
#VRML V2.0 utf8
#
#  Towers Three
#  towers.wrl
#        by David R. Nadeau
#
#  This world illustrates the use of translation, rotation, and scaling to
#  build three towers.
#
#  A first tower is built starting with a single pole and archway inlined
#  from "towerprt.wrl".  This part is instanced 12 times around the perimeter
#  of the tower by doing successive rotations.  Add a cylinder for the
#  ceiling, and a level is complete.
#
#  A level is then instanced four more times to build the next four upper
#  levels of the tower.  Each instance moves up to the next level, then
#  scales the generic level down a bit and puts it there.  This repeated
#  scaling is what gives the tower it's diminishing size for the upper
#  levels.
#
#  A glowing, rotating cube is added to the top of the generic tower.  An
#  OrientationInterpolator keeps the cube rotating.
#
#  Finally, the generic tower is instanced twice more, scaled down a bit,
#  and positioned to the left and right of the main tower.
#
#  Things to experiment with:
#        Add sound!
#
#        Add a spiraling ramp that leads to the upper levels.  Perhaps put
#        something interesting on each level.  Perhaps make the rotating
#        cube on top do something... such as be an Anchor leading to
#        another world!
#

WorldInfo {
    title "Towers Three"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF EntryView Viewpoint {
    position 0.0 1.6 20.0
    orientation 1.0 0.0 0.0 0.2
    description "Entry View"
}
NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
    speed 2.0
}

Background {
    skyColor [
        0.0 0.0 0.0,
        0.0 0.0 0.0,
        0.0 0.4 0.0,
        0.0 0.0 0.0,
    ]
    skyAngle [
        1.37,
        1.57,
```

```
        1.77,
    ]
}


#
#  Generic lighting
#
DirectionalLight {
    direction 1.0 -0.5 -1.0
    color 1.0 1.0 1.0
    ambientIntensity 0.2
    intensity 0.8
}
DirectionalLight {
    direction -1.0 1.0 -1.0
    color 0.0 0.3 1.0
    ambientIntensity 0.2
    intensity 0.5
}
PointLight {
    location 0.0 31.0 0.0
    color 0.0 1.0 0.0
    ambientIntensity 0.0
    intensity 0.6
}


#
#  Tower
#
DEF Tower Transform {
    scale 0.4 0.4 0.4
    children [
#
#  Bottom tower level
#
        DEF OneLevel Transform {
            translation 0.0 3.5 0.0
            children [
        #
        #  Half a level using 6 pole-and-arch parts in a half
        #  circle
        #
                DEF HalfLevel Group {
                    children [
                        DEF Part Transform {
                            translation 0.0 0.0 7.73
                            rotation 0.0 1.0 0.0 0.26
                            children [
                                Inline {
                                    url "towerprt.wrl"
                                }
                            ]
                        }
                        Transform {
                            rotation 0.0 1.0 0.0 0.52
                            children [ USE Part ]
                        }
                        Transform {
                            rotation 0.0 1.0 0.0 1.05
```

```
                        children [ USE Part ]
                    }
                    Transform {
                        rotation 0.0 1.0 0.0 1.57
                        children [ USE Part ]
                    }
                    Transform {
                        rotation 0.0 1.0 0.0 2.09
                        children [ USE Part ]
                    }
                    Transform {
                        rotation 0.0 1.0 0.0 2.62
                        children [ USE Part ]
                    }
                ]
            }

        #
        #  Another half a level in order to complete a full circle
        #
            Transform {
                rotation 0.0 1.0 0.0 3.14
                children [ USE HalfLevel ]
            }

        #
        #  A ceiling for the level
        #
            Transform {
                translation 0.0 3.5 0.0
                children [
                    Shape {
                        appearance Appearance {
                            material Material {
                                ambientIntensity 0.4
                            }
                            texture ImageTexture {
                                url "marble_g.jpg"
                            }
                            textureTransform TextureTransform {
                                scale 1.0 1.0
                            }
                        }
                        geometry Cylinder {
                            height 0.05
                            radius 8.2
                            side FALSE
                        }
                    }
                ]
            }

          ]
        }

#
#  A fake shadow under the first level
#
        Shape {
            appearance Appearance {
                material Material {
```

```
                    diffuseColor 0.0 0.0 0.0
                    transparency 0.25
                }
            }
            geometry Cylinder {
                height 0.05
                radius 8.2
                side FALSE
                bottom FALSE
            }
        }


#
#  Second tower level
#
        Transform {
            translation 0.0 7.0 0.0
            rotation 0.0 1.0 0.0 0.26
            scale 0.9 0.9 0.9
            children [ USE OneLevel ]
        }


#
#  Third tower level
#
        Transform {
            translation 0.0 13.3 0.0
            scale 0.8 0.8 0.8
            children [ USE OneLevel ]
        }


#
#  Fourth tower level
#
        Transform {
            translation 0.0 18.9 0.0
            rotation 0.0 1.0 0.0 0.26
            scale 0.7 0.7 0.7
            children [ USE OneLevel ]
        }


#
#  Fifth tower level
#
        Transform {
            translation 0.0 23.8 0.0
            scale 0.6 0.6 0.6
            children [ USE OneLevel ]
        }


#
#  Glowing thing
#
        DEF Thing Transform {
            translation 0.0 31.0 0.0
            rotation 0.0 1.0 0.0 0.0
            children [
```

```
                Transform {
                    rotation -1.0 1.0 1.0 0.785
                    children [
                        Shape {
                            appearance Appearance {
                                material NULL  # emissive texturing
                                texture DEF BlurImage ImageTexture {
                                    url "fire_g.jpg"
                                }
                            }
                            geometry Box {
                                size 3.0 3.0 3.0
                            }
                        }
                    ]
                }
            ]
        }
    ]
}

#
#  Second tower to the left and back
#
Transform {
    translation -5.0 0.0 -2.0
    scale 0.8 0.8 0.8
    children [ USE Tower ]
}

#
#  Third tower to the right and back
#
Transform {
    translation 5.0 0.0 -2.0
    scale 0.8 0.8 0.8
    children [ USE Tower ]
}


#
#  Floor
#
Shape {
    appearance Appearance {
        material Material {
            ambientIntensity 0.5
            diffuseColor 0.0 0.5 0.1
        }
        texture USE BlurImage
        textureTransform TextureTransform {
            scale 5.0 5.0
        }
    }
    geometry IndexedFaceSet {
        coord Coordinate {
            point [
                -40.0 0.0  80.0,
                 40.0 0.0  80.0,
                 40.0 0.0 -40.0,
                -40.0 0.0 -40.0,
```

```
            ]
        }
        coordIndex [ 0, 1, 2, 3, -1, ]
        solid TRUE
    }
}


#
#  Animation control
#
DEF ThingTimer TimeSensor {
    cycleInterval 4.0
    loop TRUE
    startTime 1.0
    stopTime 0.0
}

DEF ThingSpinner OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 1.0 0.0 0.0, 0.0 1.0 0.0 3.14, 0.0 1.0 0.0 6.28 ]
}

ROUTE ThingTimer.fraction_changed TO ThingSpinner.set_fraction
ROUTE ThingSpinner.value_changed  TO Thing.set_rotation
```

```
#VRML V2.0 utf8

# Left and right columns
Transform {
    translation -2.0 3.0 0.0
    children [
        DEF Column Shape {
            appearance DEF White Appearance {
                material Material { }
            }
            geometry Cylinder {
                radius 0.3
                height 6.0
                top FALSE
            }
        }
    ]
}
Transform {
    translation 2.0 3.0 0.0
    children [ USE Column ]
}

# Cross-piece
Transform {
    translation 0.0 6.05 0.0
    children [
        Shape {
            appearance USE White
            geometry Box { size 4.6 0.4 0.6 }
        }
    ]
}

# Roof pieces
Transform {
    translation -1.15 7.12 0.0
    rotation 0.0 0.0 1.0  0.524
    children [
        DEF Roof Shape {
            appearance USE White
            geometry Box { size 2.86 0.4 0.6 }
        }
    ]
}
Transform {
    translation 1.15 7.12 0.0
    rotation 0.0 0.0 1.0  -0.524
    children [ USE Roof ]
}
```

```
#VRML V2.0 utf8
#
#  arches.wrl
#  Gazebo made of arches
#        by David R. Nadeau
#
#  This example illustrates the use of Transform grouping nodes and their
#  translation, rotation, and scale features.
#
#  First, an Arch is created in a group.  The Arch includes of
#  left and right columns, each positioned using a Transform group and
#  its translation field.
#
#  Next, each Arch group includes a horizontal box spanning the two
#  columns.  The box is translated up to the top of the columns using a
#  Transform group node.
#
#  The angled top of each Arch is created using two boxes, each rotated
#  about the Z axis to tilt them.  A translation then positions the tilted
#  boxes at the top of the archway.
#
#  Next, the entire Arch group is instanced multiple times, rotating
#  each instance to create a circle of archs forming a gazeebo-like
#  structure.
#
#  Finally a base for the structure is added.
#

DEF Arch Transform {
    children [
    # Left and right columns
        Transform {
            translation -2.0 3.0 0.0
            children [
                DEF Column Shape {
                    appearance DEF White Appearance {
                        material Material { }
                    }
                    geometry Cylinder {
                        radius 0.3
                        height 6.0
                        top FALSE
                        bottom FALSE
                    }
                }
            ]
        }
        Transform {
            translation 2.0 3.0 0.0
            children [ USE Column ]
        }

    # Cross-piece
        Transform {
            translation 0.0 6.05 0.0
            children [
                Shape {
                    appearance USE White
                    geometry Box { size 4.6 0.4 0.6 }
                }
            ]
```

```
        }

    # Roof pieces
        Transform {
            translation -1.15 7.12 0.0
            rotation 0.0 0.0 1.0  0.524
            children [
                DEF Roof Shape {
                    appearance USE White
                    geometry Box { size 2.86 0.4 0.6 }
                }
            ]
        }
        Transform {
            translation 1.15 7.12 0.0
            rotation 0.0 0.0 1.0  -0.524
            children [ USE Roof ]
        }
    ]
}
Transform {
    rotation 0.0 1.0 0.0 0.785
    children [ USE Arch ]
}
Transform {
    rotation 0.0 1.0 0.0 -0.785
    children [ USE Arch ]
}
Transform {
    rotation 0.0 1.0 0.0 1.571
    children [ USE Arch ]
}

# Floor
Transform {
    translation 0.0 -0.125 0.0
    children [
        Shape {
            appearance USE White
            geometry Cylinder {
                radius 3.0
                height 0.25
                bottom FALSE
            }
        }
    ]
}
Transform {
    translation 0.0 -0.375 0.0
    children [
        Shape {
            appearance USE White
            geometry Cylinder {
                radius 4.0
                height 0.25
            }
        }
    ]
}
```

Controlando aparências com materiais

Motivação

Exemplo

Sintaxe: Shape

Sintaxe: Appearance

Sintaxe: Material

Especificando cores

Sintaxe: Material

Experimentando materiais brilhantes

Exemplo

Um exemplo de mundo usando appearance

Um exemplo de mundo usando appearance

Resumo

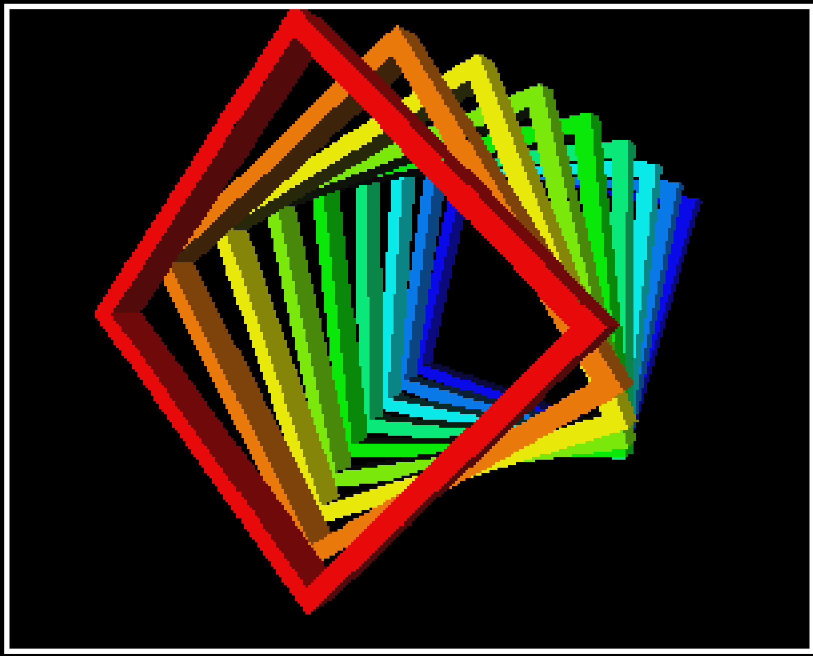
Controlando aparências com materiais

Motivação

- As formas primitivas tem uma aparência branca, inicialmente
- Você pode controlar o seguinte de uma forma:
- Tom da cor
- Brilho da cor
- Transparência
- Claridade
- Intensidade do ambiente

Controlando aparências com materiais

Exemplo



[colors.wrl]

Controlando aparências com materiais

Sintaxe: Shape

- **Lembrar que o nó `Shape` descreve:**
  - ***appearance* - cor e textura**
  - ***geometry* - forma ou estrutura**

```
Shape {
    appearance . . .
    geometry   . . .
}
```

Controlando aparências com materiais

Sintaxe: Appearance

- **Um nó `Appearance` descreve toda a aparência da forma**
  - **propriedades do *material* - cor, transparência, etc.**
  - **mais . . .**

```
Shape {
    appearance Appearance {
        material . . .
    }
    geometry . . .
}
```

Controlando aparências com materiais

Sintaxe: Material

- **Um nó `Material` controla os atributos do material da forma**
- ***diffuse color* - principal tom de cor**
- ***emissive color* - cor brilhante**
- ***transparency* - opaca ou não**
- **mais . . .**

```
Material {
    diffuseColor  . . .
    emissiveColor . . .
    transparency  . . .
}
```

Controlando aparências com materiais

Especificando cores

- **Cores especificam:**
  - **Uma mistura de red (vermelho), green (verde), e blue light (azul brilhante)**
  - **Valores entre 0.0 (nenhuma) and 1.0 (toda)**

Cor	Red	Green	Blue	Resultado
Branco	1.0	1.0	1.0	
Vermelho	0.0	0.0	0.0	
Amarelo	1.0	1.0	0.0	
Violeta	1.0	0.0	1.0	
Marrom	0.5	0.2	0.0	

Controlando aparências com materiais

Sintaxe: Material

. Um nó `Material` também controla a claridade da forma

- ***specular color* - cor principal**
- ***shininess* - proporção principal**
- ***ambient intensity* - efeitos de luz no ambiente**

```
Material {
    . . .
    specularColor 0.71 0.70 0.56
    shininess 0.16
    ambientIntensity 0.4
}
```

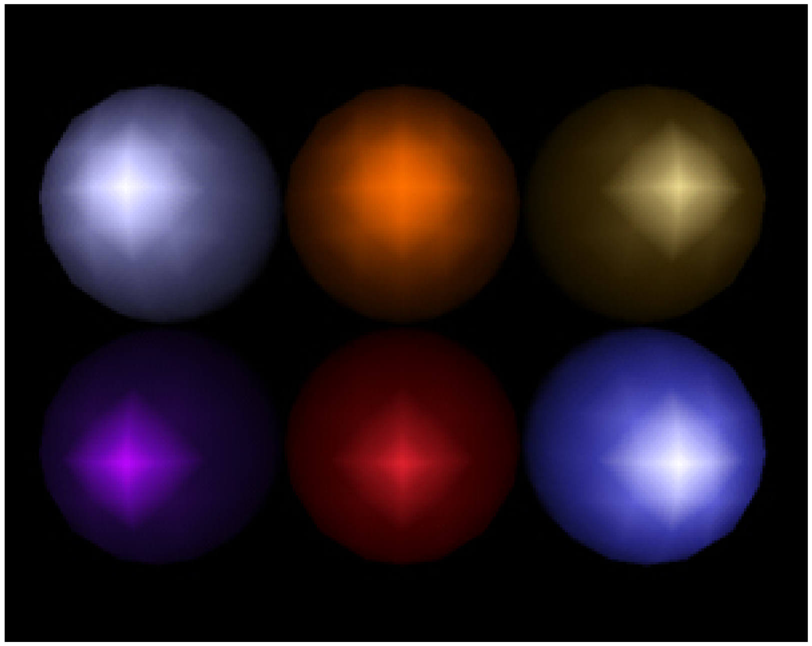
Controlando aparências com materiais

Experimentando materiais brilhantes

Descrição	**ambientIntensity**	**diffuseColor**	**specularColor**	**shininess**
Alumínio	0.3	0.30 0.30 0.50	0.70 0.70 0.80	0.10
Cobre	0.26	0.30 0.11 0.00	0.75 0.33 0.00	0.08
Ouro	0.4	0.22 0.15 0.00	0.71 0.70 0.56	0.16
Púrpura Metálico	0.17	0.10 0.03 0.22	0.64 0.00 0.98	0.20
Vermelho Metálico	0.15	0.27 0.00 0.00	0.61 0.13 0.18	0.20
Azul plástico	0.10	0.20 0.20 0.71	0.83 0.83 0.83	0.12

Controlando aparências com materiais

Exemplo



[shiny.wrl]

Controlando aparências com materiais

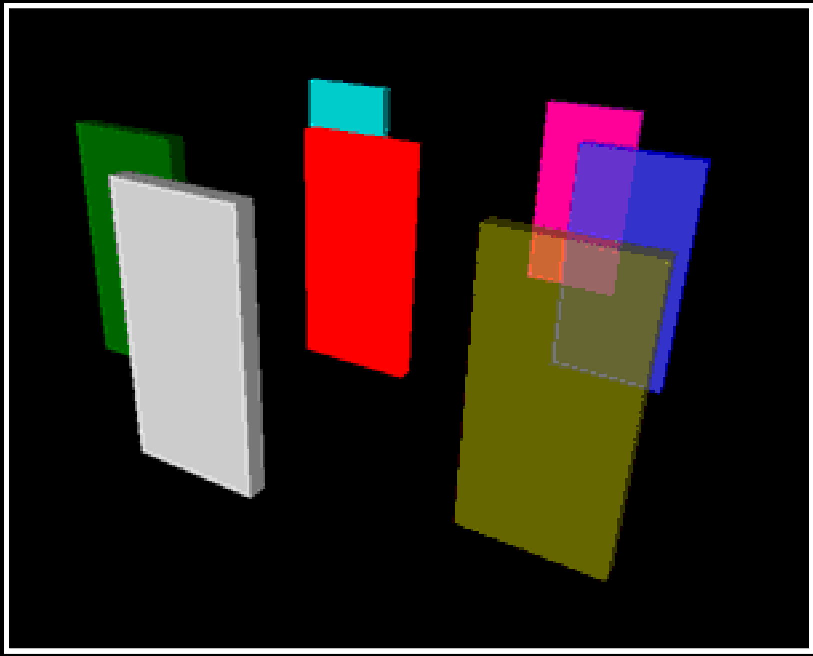
Um exemplo de mundo usando appearance

```
Shape {
    appearance Appearance {
        material Material {
            diffuseColor 1.0 1.0 1.0
        }
    }
    geometry . . .
}
```

Sumário Índice do Capítulo Slide Anterior Slide Seguinte 76

Controlando aparências com materiais

Um exemplo de mundo usando appearance



[slabs.wrl]

Controlando aparências com materiais

Resumo

- O nó `Appearance` controla toda a aparência da forma
- O nó `Material` controla todas as propriedades do material, incluindo:
- Tom de cor
- Brilho da cor
- Transparência
- Claridade
- Intensidade ambiente

```
#VRML V2.0 utf8
#
#  Colors
#  colors.wrl
#        by David R. Nadeau
#
#  This world is simply used as something colorful.  Orientation interpolators
#  slowly spin 9 frames, each with a slightly different speed.  At first
#  there is a clear structure, but as the animation evolves the structure
#  dissolves in to chaos.  If you wait long enough, they'll all get back in
#  sync again... after about 560 years!
#
#  Things to experiment with:
#        Try using other cycleIntervals for the rotation speed of the 9 squares.
#        If the intervals are multiples of a common factor, the squares will
#        get in sync sooner than if they don't have a common factor.
#


WorldInfo {
    title "Colors"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 0.0 3.0
    description "Entry View"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight TRUE
}

Background {
    skyColor [ 0.0 0.0 0.0 ]
}


#
#  Red frame
#
DEF Frame1 Transform {
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 1.0 0.0 0.0
                }
            }
            geometry DEF Frame IndexedFaceSet {
                coord Coordinate {
                    point [
                        # Outer front
                        -1.0 -1.0 0.1,
                        -0.9 -1.0 0.1,
                         0.9 -1.0 0.1,
                         1.0 -1.0 0.1,
                         1.0  1.0 0.1,
                         0.9  1.0 0.1,
                        -0.9  1.0 0.1,
                        -1.0  1.0 0.1,
```

```
                        # Outer back
                        -1.0 -1.0 -0.1,
                        -0.9 -1.0 -0.1,
                         0.9 -1.0 -0.1,
                         1.0 -1.0 -0.1,
                         1.0  1.0 -0.1,
                         0.9  1.0 -0.1,
                        -0.9  1.0 -0.1,
                        -1.0  1.0 -0.1,
                        # Inner front
                        -0.9 -0.9 0.1,
                         0.9 -0.9 0.1,
                         0.9  0.9 0.1,
                        -0.9  0.9 0.1,
                        # Inner front
                        -0.9 -0.9 -0.1,
                         0.9 -0.9 -0.1,
                         0.9  0.9 -0.1,
                        -0.9  0.9 -0.1,
                    ]
                }
                coordIndex [
                    0, 1, 6, 7, -1,
                    1, 2, 17, 16, -1,
                    2, 3, 4, 5, -1,
                    19, 18, 5, 6, -1,
                    11, 10, 13, 12, -1,
                    10, 9, 20, 21, -1,
                    9, 8, 15, 14, -1,
                    22, 23, 14, 13, -1,
                    3, 11, 12, 4, -1,
                    4, 12, 15, 7, -1,
                    8, 0, 7, 15, -1,
                    0, 8, 11, 3, -1,
                    16, 17, 21, 20, -1,
                    21, 17, 18, 22, -1,
                    19, 23, 22, 18, -1,
                    16, 20, 23, 19, -1,
                ]
            }
        }
    ]
}

#
#  Orange frame
#
DEF Frame2 Transform {
    translation 0.0 0.0 -0.4
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 1.0 0.5 0.0
                }
            }
            geometry USE Frame
        }
    ]
}
```

```
#
#  Yellow frame
#
DEF Frame3 Transform {
    translation 0.0 0.0 -0.8
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 1.0 1.0 0.0
                }
            }
            geometry USE Frame
        }
    ]
}

#
#  Light green frame
#
DEF Frame4 Transform {
    translation 0.0 0.0 -1.2
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.5 1.0 0.0
                }
            }
            geometry USE Frame
        }
    ]
}

#
#  Green frame
#
DEF Frame5 Transform {
    translation 0.0 0.0 -1.6
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 1.0 0.0
                }
            }
            geometry USE Frame
        }
    ]
}

#
#  Light Cyan frame
#
DEF Frame6 Transform {
    translation 0.0 0.0 -2.0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 1.0 0.5
```

```
                }
            }
            geometry USE Frame
        }
    ]
}

#
#  Cyan frame
#
DEF Frame7 Transform {
    translation 0.0 0.0 -2.4
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 1.0 1.0
                }
            }
            geometry USE Frame
        }
    ]
}

#
#  Dark Cyan frame
#
DEF Frame8 Transform {
    translation 0.0 0.0 -2.8
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 0.5 1.0
                }
            }
            geometry USE Frame
        }
    ]
}

#
#  Blue frame
#
DEF Frame9 Transform {
    translation 0.0 0.0 -3.2
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 0.0 1.0
                }
            }
            geometry USE Frame
        }
    ]
}


#
#  Trigger things on touch
```

```
#
DEF Touch TouchSensor { }


#
#  Animation control
#

DEF Timer1 TimeSensor { cycleInterval 10.0 loop TRUE startTime -1 }
DEF Timer2 TimeSensor { cycleInterval 11.0 loop TRUE startTime -1 }
DEF Timer3 TimeSensor { cycleInterval 12.0 loop TRUE startTime -1 }
DEF Timer4 TimeSensor { cycleInterval 13.0 loop TRUE startTime -1 }
DEF Timer5 TimeSensor { cycleInterval 14.0 loop TRUE startTime -1 }
DEF Timer6 TimeSensor { cycleInterval 15.0 loop TRUE startTime -1 }
DEF Timer7 TimeSensor { cycleInterval 16.0 loop TRUE startTime -1 }
DEF Timer8 TimeSensor { cycleInterval 17.0 loop TRUE startTime -1 }
DEF Timer9 TimeSensor { cycleInterval 18.0 loop TRUE startTime -1 }

DEF Rot1 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}
DEF Rot2 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}
DEF Rot3 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}
DEF Rot4 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}
DEF Rot5 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}
DEF Rot6 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}
DEF Rot7 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}
DEF Rot8 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}
DEF Rot9 OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0, 0.0 0.0 1.0 3.14, 0.0 0.0 1.0 6.28 ]
}

ROUTE Touch.touchTime TO Timer1.set_startTime
ROUTE Touch.touchTime TO Timer2.set_startTime
ROUTE Touch.touchTime TO Timer3.set_startTime
ROUTE Touch.touchTime TO Timer4.set_startTime
ROUTE Touch.touchTime TO Timer5.set_startTime
ROUTE Touch.touchTime TO Timer6.set_startTime
```

```
ROUTE Touch.touchTime TO Timer7.set_startTime
ROUTE Touch.touchTime TO Timer8.set_startTime
ROUTE Touch.touchTime TO Timer9.set_startTime

ROUTE Timer1.fraction_changed TO Rot1.set_fraction
ROUTE Timer2.fraction_changed TO Rot2.set_fraction
ROUTE Timer3.fraction_changed TO Rot3.set_fraction
ROUTE Timer4.fraction_changed TO Rot4.set_fraction
ROUTE Timer5.fraction_changed TO Rot5.set_fraction
ROUTE Timer6.fraction_changed TO Rot6.set_fraction
ROUTE Timer7.fraction_changed TO Rot7.set_fraction
ROUTE Timer8.fraction_changed TO Rot8.set_fraction
ROUTE Timer9.fraction_changed TO Rot9.set_fraction

ROUTE Rot1.value_changed TO Frame1.set_rotation
ROUTE Rot2.value_changed TO Frame2.set_rotation
ROUTE Rot3.value_changed TO Frame3.set_rotation
ROUTE Rot4.value_changed TO Frame4.set_rotation
ROUTE Rot5.value_changed TO Frame5.set_rotation
ROUTE Rot6.value_changed TO Frame6.set_rotation
ROUTE Rot7.value_changed TO Frame7.set_rotation
ROUTE Rot8.value_changed TO Frame8.set_rotation
ROUTE Rot9.value_changed TO Frame9.set_rotation
```

```
#VRML V2.0 utf8
#
#  shiny.wrl
#  Shiny spheres
#       by David R. Nadeau
#
#  This world creates a set of spheres, each given a different set of
#  material attributes.  The attributes illustrate creation of metalic
#  surfaces by using the specularColor, shininess, and ambientIntensity
#  fields along with the standard diffuseColor field.
#

# Aluminum
Transform {
        translation -2.0 1.0 0.0
        children [
                Shape {
                        appearance Appearance {
                                material Material {
                                        ambientIntensity 0.3
                                        diffuseColor 0.3 0.3 0.5
                                        specularColor 0.7 0.7 0.8
                                        shininess 0.1
                                }
                        }
                        geometry Sphere { }
                }
        ]
}

# Copper
Transform {
        translation 0.0 1.0 0.0
        children [
                Shape {
                        appearance Appearance {
                                material Material {
                                        ambientIntensity 0.26
                                        diffuseColor 0.3 0.11 0.0
                                        specularColor 0.75 0.33 0.0
                                        shininess 0.08
                                }
                        }
                        geometry Sphere { }
                }
        ]
}

# Gold
Transform {
        translation 2.0 1.0 0.0
        children [
                Shape {
                        appearance Appearance {
                                material Material {
                                        ambientIntensity 0.4
                                        diffuseColor 0.22 0.15 0.0
                                        specularColor 0.71 0.7 0.56
                                        shininess 0.16
                                }
                        }
```

```
                        geometry Sphere { }
                }
        ]
}



# Metalic Purple
Transform {
        translation -2.0 -1.0 0.0
        children [
                Shape {
                        appearance Appearance {
                                material Material {
                                        ambientIntensity 0.17
                                        diffuseColor 0.1 0.03 0.22
                                        specularColor 0.64 0.0 0.98
                                        shininess 0.2
                                }
                        }
                        geometry Sphere { }
                }
        ]
}

# Metalic Red
Transform {
        translation 0.0 -1.0 0.0
        children [
                Shape {
                        appearance Appearance {
                                material Material {
                                        ambientIntensity 0.15
                                        diffuseColor 0.27 0.0 0.0
                                        specularColor 0.61 0.13 0.18
                                        shininess 0.2
                                }
                        }
                        geometry Sphere { }
                }
        ]
}

# Plastic Blue
Transform {
        translation 2.0 -1.0 0.0
        children [
                Shape {
                        appearance Appearance {
                                material Material {
                                        ambientIntensity 0.1
                                        diffuseColor 0.20 0.2 0.71
                                        specularColor 0.83 0.83 0.83
                                        shininess 0.12
                                }
                        }
                        geometry Sphere { }
                }
        ]
}
```

```
#VRML V2.0 utf8
#
#  Slab World
#  slabs.wrl
#        by David R. Nadeau
#
#  This example illustrates use of Appearance and Material nodes to set
#  the main shading color (diffuseColor), glow color (emissiveColor), and
#  transparency level of a shape.
#
#  Seven slabs are created, each using a Box node.  Each slab is shaded
#  a different material:
#
#        Slab       Diffuse     Emissive    Transparency
#        White      white       black       0.0
#        Red        black       red         0.0
#        Green      green       black       0.5
#        Blue       blue        white       0.25
#        Yellow     yellow      black       0.5
#        Cyan       cyan        black       0.0
#        Magenta    red         magenta     0.0
#
#  Some PC browsers may not support transparency.  Others may not support
#  shading using both diffuse and emissive colors.  So, your results may
#  vary.
#
#  Things to experiment with:
#        Try different transparency levels.
#
#        Turn off the headlight on your VRML browser and observe which
#        slabs glow.
#


WorldInfo {
    title "Slab World"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 1.6 20.0
    description "Entry View"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight TRUE
}

Background {
    skyColor [ 0.0 0.0 0.0 ]
}

#
#  Slabs
#
# White slab (shaded)
Transform {
    translation 0.0 2.0 4.0
    children [
        Shape {
            appearance Appearance {
```

```
                material Material {
                    diffuseColor 1.0 1.0 1.0
                }
            }
            geometry DEF Slab Box {
                size 2.0 4.0 0.3
            }
        }
    ]
}

# Red slab (emissive)
Transform {
    translation 0.0 2.0 0.0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    ambientIntensity 0.0
                    diffuseColor  0.0 0.0 0.0
                    emissiveColor 1.0 0.0 0.0
                }
            }
            geometry DEF Slab Box {
                size 2.0 4.0 0.3
            }
        }
    ]
}

# Green slab (shaded + transparent)
Transform {
    translation -3.0 2.0 2.0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 1.0 0.0
                    transparency 0.5
                }
            }
            geometry USE Slab
        }
    ]
}

# Blue slab (shaded + emissive + transparent)
Transform {
    translation 4.0 2.0 -2.0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.2 0.2 0.2
                    emissiveColor 0.0 0.0 0.8
                    transparency 0.25
                }
            }
            geometry USE Slab
        }
    ]
```

```
}

# Yellow slab (shaded + transparent)
Transform {
    translation 4.5 2.0 3.0
    children [
        # Outer
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 1.0 1.0 0.0
                    transparency 0.5
                }
            }
            geometry USE Slab
        }
    ]
}

# Cyan slab (shaded)
Transform {
    translation -3.5 2.0 -5.0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 1.0 1.0
                }
            }
            geometry USE Slab
        }
    ]
}

# Magenta slab (shaded + emissive)
Transform {
    translation 2.0 2.0 -6.0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.5 0.0 0.0
                    emissiveColor 0.5 0.0 0.5
                }
            }
            geometry USE Slab
        }
    ]
}
```

Agrupando nós

Motivação

Sintaxe: Group

Sintaxe: Switch

Sintaxe: Transform

Sintaxe: Billboard

Eixos de rotação do Billboard

Um exemplo de grupo do billboard

Um exemplo de grupo do billboard

Sintaxe: Anchor

Um exemplo de Anchor

Sintaxe: Inline

Um exemplo de arquivo inlined

Um exemplo de arquivo inlined

Resumo

Resumo

Agrupando nós

Motivação

. Você pode agrupar formas para compor formas complexas
. VRML possui vários agrupamento de nós, incluindo:

```
Group      { . . . }
Switch     { . . . }
Transform  { . . . }
Billboard  { . . . }
Anchor     { . . . }
Inline     { . . . }
```

Agrupando nós

Sintaxe: Group

- O nó grupo `Group` cria um grupo básico
  - Todo nó dentro do grupo é mostrado

```
Group {
    children [ . . . ]
}
```

Sumário Índice do Capítulo Slide Anterior Slide Seguinte 81

Agrupando nós

Sintaxe: Switch

- O grupo `Switch` cria um grupo de escolha
  - Somente um nó no grupo é mostrado
  - Você seleciona qual o nó

```
Switch {
    whichChoice 0
    choice [ . . . ]
}
```

Agrupando nós

Sintaxe: Transform

- **O grupo `Transform` cria um grupo com o seu próprio sistema de coordenada**
- **. Todos os nós do grupo são mostrados**

```
Transform {
    translation . . .
    rotation    . . .
    scale       . . .
    children [ . . . ]
}
```

Agrupando nós

Sintaxe: Billboard

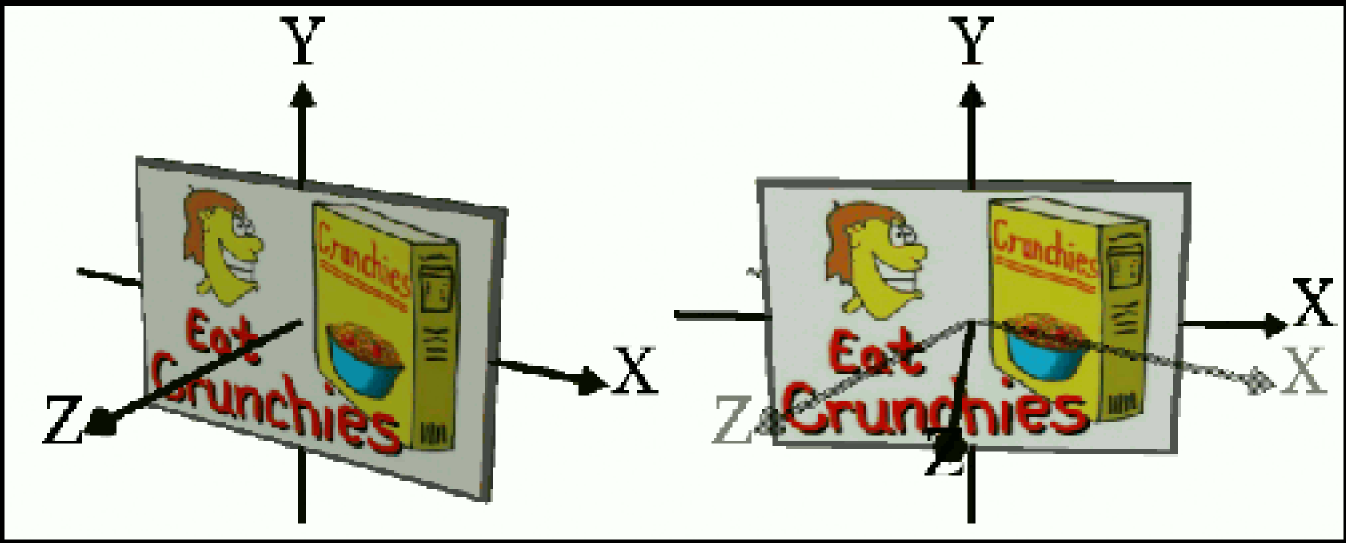
- O grupo `Billboard` cria um grupo com um sistema de coordenadas especial
  - Todos os nós do grupo são mostrados
  - O sistema de coordenadas é voltado para a face do observador

```
Billboard {
    axisOfRotation . . .
    children [ . . . ]
}
```

Agrupando nós

Eixos de rotação do Billboard

- Um eixo de rotação define um pólo para rotacionar em torno deste
  - Similar ao nó `Transform` com o campo `rotation`, mas nenhum ângulo é fornecido (auto computed)



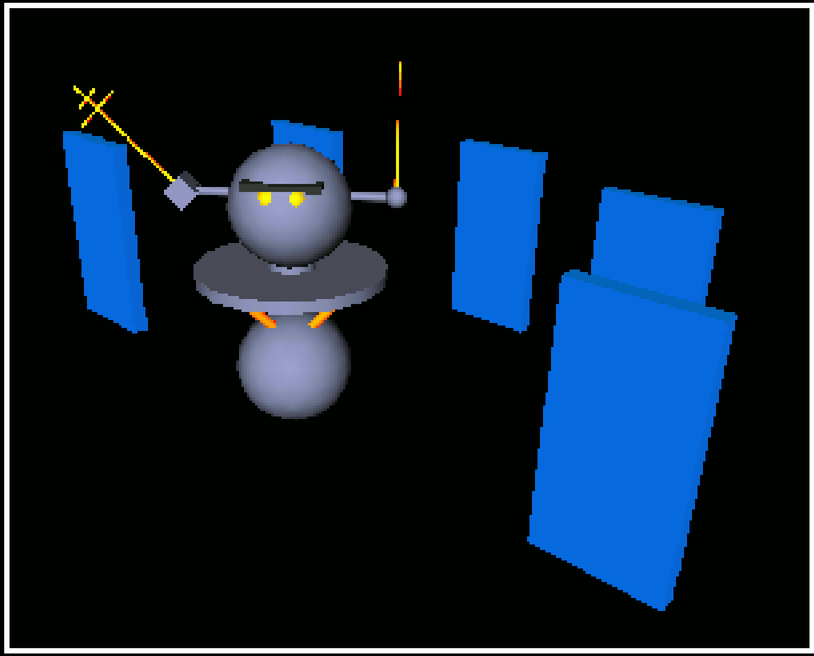
Agrupando nós

Um exemplo de grupo do billboard

```
Group {
    children [
        Billboard {
            axisOfRotation 0.0 1.0 0.0
            children [ ... ]
        }
        Transform { . . . }
    ]
}
```

Agrupando nós

Um exemplo de grupo do billboard



[robobill.wrl]

Agrupando nós

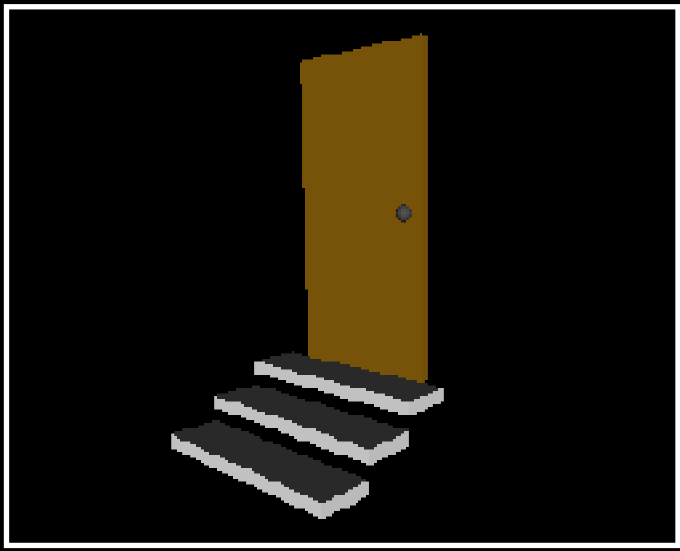
Sintaxe: Anchor

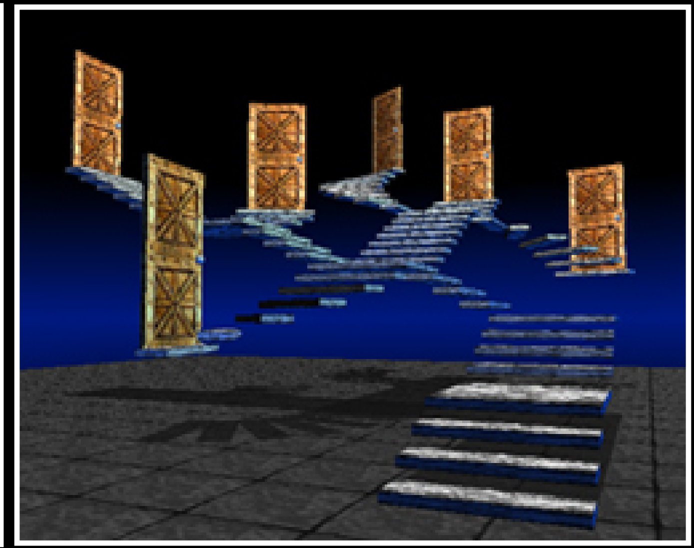
- **Um nó `Anchor` cria um grupo que aciona uma âncora clicável**
- **. Todos os nós do grupo são mostrados**
- **. Clicando em qualquer nó dentro do grupo segue-se para uma URL**
- **. O campo *description* nomeia a âncora**

```
Anchor {
    url "stairwy.wrl"
    description "Twisty Stairs"
    children [ . . . ]
}
```

Agrupando nós

Um exemplo de Anchor





[**anchor.wrl**] [**stairwy.wrl**]

Agrupando nós

Sintaxe: Inline

- **Um nó `Inline` cria um grupo especial a partir do conteúdo de outro arquivo VRML**
- **. Lê-se do arquivo selecionado pela URL**
- **. Todos os nós do grupo são mostrados**

```
Inline {
    url "table.wrl"
}
```

Agrupando nós

Um exemplo de arquivo inlined

```
Inline { url "table.wrl" }
. . .
Transform {
    translation . . .
    children [
        Inline { url "chair.wrl" }
    ]
}
```

Agrupando nós

Um exemplo de arquivo inlined

[table.wrl, chair.wrl, dinette.wrl]

Agrupando nós

Resumo

- **O nó `Group` cria um grupo básico**
- **O nó `Switch` cria um nó com uma escolha**
- **O nó `Transform` cria um grupo com um novo sistema de coordenadas**

Agrupando nós

Resumo

- **O nó `Billboard` cria um grupo com um sistema de coordenada que é rotacionado para a face do observador**
- **O nó `Anchor` cria um grupo que pode ser clicado**
  - **Clicando qualquer nó no grupo carrega-se uma URL**
- **O nó `Inline` cria um grupo especial carregado de outro arquivo VRML**

```
#VRML V2.0 utf8
#
#  Robot Billboard
#  robobill.wrl
#        by David R. Nadeau
#
#  This example illustrates the use of Group, Transform, and Billboard groups.
#
#  A robot is built with its head in a Billboard group, but its torso
#  left out of it.  As the viewer walks about the robot, the head always
#  turns (about the Y axis) to face the viewer.
#
#  To make it clear what is, and is not moving to track the viewer, the
#  torso of the robot remains fixed, as does a background of blue slabs.
#
#  Things to experiment with:
#        Place the entire robot in the Billboard group so that the whole
#        robot tracks the viewer, not just the robot's head.
#
#        Add textures to the environment.
#
#        Add light sources, such as one placed in the lower ball of the
#        robot.  Make that ball emissive as well so that it looks like the
#        ball is the source of light.  Remember to put the light source
#        inside the robot's transform group so that as the robot moves,
#        the light source moves with it.
#
#        Add behavior to the robot.  Make it move on its own!
#
#        Place the robot in to a separate VRML file and inline it in to
#        this, and other environments.
#

WorldInfo {
    title "Robot Billboard"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 1.6 10.0
    description "Entry View"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight TRUE
}

Background {
    skyColor [ 0.0 0.0 0.0 ]
}


#
#  Robot
#
Transform {
    translation 0.0 1.0 0.0
    scale 2.0 2.0 2.0
    children [

#
```

```
#  Stationary body parts for the robot (the torso)
#
        # Torso sphere
        Shape {
            appearance DEF RobotColor Appearance {
                material Material {
                    diffuseColor 0.6 0.6 0.8
                }
            }
            geometry Sphere {
                radius 0.5
            }
        }

        # Support rods
        DEF Rod Transform {
            translation 0.4 0.6 0.0
            rotation 0.0 0.0 1.0 -0.785
            children [
                Shape {
                    appearance DEF RobotGlow Appearance {
                        material Material {
                            diffuseColor 0.0 1.0 0.0
                            emissiveColor 1.0 0.0 0.0
                        }
                    }
                    geometry Cylinder {
                        height 0.65
                        radius 0.04
                    }
                }
            ]
        }
        Transform {
            rotation 0.0 1.0 0.0 1.57
            children [ USE Rod ]
        }
        Transform {
            rotation 0.0 1.0 0.0 3.14
            children [ USE Rod ]
        }
        Transform {
            rotation 0.0 1.0 0.0 -1.57
            children [ USE Rod ]
        }

        # Shoulders
        Transform {
            translation 0.0 0.85 0.0
            children [
                Shape {
                    appearance USE RobotColor
                    geometry Cylinder {
                        height 0.1
                        radius 0.8
                    }
                }
            ]
        }

        # Neck
```

```
        Transform {
            translation 0.0 0.95 0.0
            children [
                Shape {
                    appearance USE RobotColor
                    geometry Cylinder {
                        height 0.1
                        radius 0.2
                    }
                }
            ]
        }

#
#  Parts that track the viewer using a Billboard group
#
        Transform {
            translation 0.0 1.4 0.0
            children [
                Billboard {
                    axisOfRotation 0.0 1.0 0.0
                    children [

                        # Head
                        Shape {
                            appearance USE RobotColor
                            geometry Sphere {
                                radius 0.5
                            }
                        }

                        # Eyebrow
                        Transform {
                            translation 0.0 0.25 0.35
                            children [
                                Shape {
                                    appearance Appearance {
                                        material Material {
                                            diffuseColor 0.2 0.2 0.2
                                        }
                                    }
                                    geometry Box {
                                        size 0.6 0.05 0.2
                                    }
                                }
                            ]
                        }

                        # Right Eye
                        Transform {
                            translation 0.1 0.15 0.38
                            children [
                                DEF Eye Shape {
                                    appearance Appearance {
                                        material Material {
                                            diffuseColor 0.4 1.0 0.0
                                            emissiveColor 0.6 0.0 0.0
                                        }
                                    }
                                    geometry Sphere {
                                        radius 0.09
```

```
                }
            }
        ]
    }

    # Left Eye
    Transform {
        translation -0.1 0.15 0.38
        children [ USE Eye ]
    }

    # Right Antenae
    Transform {
        translation 0.85 0.1 0.0
        children [
            Transform {
                translation -0.2 0.0 0.0
                rotation 0.0 0.0 1.0 1.57
                children [
                    DEF AntenaeBar Shape {
                        appearance USE RobotColor
                        geometry Cylinder {
                            height 0.4
                            radius 0.04
                        }
                    }
                ]
            }
            Shape {
                appearance USE RobotColor
                geometry Sphere {
                    radius 0.09
                }
            }
            Transform {
                translation 0.0 0.58 0.0
                children [
                    Shape {
                        appearance USE RobotGlow
                        geometry Cone {
                            height 1.0
                            bottomRadius 0.02
                        }
                    }
                ]
            }
        ]
    }

    # Left Antenae
    Transform {
        translation -0.85 0.1 0.0
        children [
            Transform {
                translation 0.2 0.0 0.0
                rotation 0.0 0.0 1.0 -1.57
                children [ USE AntenaeBar ]
            }
            Transform {
                rotation 0.0 0.0 1.0 0.785
                children [
```

```
                                        Shape {
                                            appearance USE RobotColor
                                            geometry Box {
                                                size 0.2 0.2 0.2
                                            }
                                        }
                                        Transform {
                                            translation 0.0 0.58 0.0
                                            children [
                                                Shape {
                                                    appearance USE RobotGlow
                                                    geometry Box {
                                                        size 0.02 1.0 0.02
                                                    }
                                                }
                                            ]
                                        }
                                        Transform {
                                            translation 0.0 0.95 0.0
                                            children [
                                                Shape {
                                                    appearance USE RobotGlow
                                                    geometry Box {
                                                        size 0.2 0.02 0.02
                                                    }
                                                }
                                            ]
                                        }
                                        Transform {
                                            translation 0.0 0.85 0.0
                                            children [
                                                Shape {
                                                    appearance USE RobotGlow
                                                    geometry Box {
                                                        size 0.4 0.02 0.02
                                                    }
                                                }
                                            ]
                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}


#
#  Environment
#
Transform {
    translation 0.0 2.0 -6.0
    children [
        DEF Slab Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 0.1 0.5
```

```
                    emissiveColor 0.0 0.3 0.5
                }
            }
            geometry Box {
                size 2.0 4.0 0.3
            }
        }
    ]
}

Transform {
    translation -6.0 2.0 0.0
    children [ USE Slab ]
}

Transform {
    translation 6.0 2.0 0.0
    children [ USE Slab ]
}

Transform {
    translation -4.3 2.0 -4.3
    children [ USE Slab ]
}

Transform {
    translation 4.3 2.0 -4.3
    children [ USE Slab ]
}
```

```
#VRML V2.0 utf8
#
#  Simple anchor
#  anchor.wrl
#        by David R. Nadeau
#
#  This example illustrates the use of an Anchor group around the
#  shapes used to build a door.  Clicking on the door jumps you to
#  a new world ("stairwy.wrl").
#
#  Things to experiment with:
#        Change the anchor's URL to point to one of your own worlds.
#
#        Add more doors and stairs.  Add walls, floor, ceiling, etc.
#

Transform {
    translation 0.0 -2.5 0.0
    children [
        #
        #  Three stairs
        #
        DEF Stair Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 1.0 1.0 1.0
                }
            }
            geometry Box {
                size 4.0 0.25 1.0
            }
        }
        Transform {
            translation 0.0 0.5 -1.0
            children USE Stair
        }
        Transform {
            translation 0.0 1.0 -2.0
            children USE Stair
        }


        #
        #  Door
        #
        Transform {
            translation 0.0 4.125 -2.6
            children [
                Anchor {
                    url "stairwy.wrl"
                    description "Twisty Stairs"
                    children [
                        # Door
                        Shape {
                            appearance Appearance {
                                material Material {
                                    diffuseColor 0.6 0.4 0.0
                                }
                            }
                            geometry Box {
                                size 3.0 6.0 0.2
```

```
                        }
                    }
                    # Door knob
                    Transform {
                        translation 1.18 -0.0 0.2
                        children [
                            Shape {
                                appearance Appearance {
                                    material Material {
                                        diffuseColor 0.3 0.3 0.3
                                    }
                                }
                                geometry Sphere {
                                    radius 0.15
                                }
                            }
                        ]
                    }
                ]
            }
        ]
    }
  ]
}
```

```
#VRML V2.0 utf8
#
#  The Stairway
#  stairwy.wrl
#        by David R. Nadeau
#
#  This world uses a twisting, branching, floating stairway to lead the
#  visitor to six doorways.  Each doorway is an anchor leading to another
#  world.
#
#  This world was *hand authored*, which is way tough to do given the
#  number of stairs to be positioned and oriented.  This accounts for
#  the heavy use of DEF and USE and rather convoluted nested transforms.
#  By using a world builder application (when they become available),
#  you can position each stair or group of stairs interactively, instead
#  of typing in translations and rotations and hoping they work out.
#
#  Things to experiment with:
#        Change the anchor urls to point to your own worlds.
#
#        Add a flock of birds following a path animated by a
#        PositionInterpolator and an OrientationInterpolator.
#
#        Drop the sky background wall and replace it with a Background node.
#
#        Add sound.  Perhaps a gentle wind with some occasional wind chimes.
#

WorldInfo {
    title "The Stairway"
    info [ "copyright (c) 1997, David R. Nadeau" ]
}

Background {
    skyColor [
        0.0 0.0 0.0,
        0.0 0.0 0.0,
        0.0 0.1 0.5,
        0.0 0.0 0.0,
    ]
    skyAngle [
        1.37,
        1.57,
        1.77,
    ]
}

Transform {
    scale 0.4 0.4 0.4
    children [
        DEF EntryView Viewpoint {
            position 9.0 -5.0 25.0
            description "Entry View"
        }
        NavigationInfo {
            type [ "WALK", "ANY" ]
            headlight FALSE
            avatarSize [ 0.5, 4.0, 1.5 ]
            speed 4.0
        }
```

```
#
#  Lighting
#
        DirectionalLight {
            ambientIntensity 0.0
            color 0.0 0.3 1.0
            direction -3.0 2.0 -1.0
        }
        PointLight {
            ambientIntensity 0.3
            color 1.0 1.0 1.0
            location 14.0 10.0 14.0
        }



#
#  (Main) platform
#
        Transform {
            translation 5.0 -3.0 5.0
            children [
                DEF Platform Shape {
                    appearance DEF Marble Appearance {
                        material Material {
                            diffuseColor 1.0 1.0 1.0
                        }
                        texture ImageTexture {
                            url "marble_g.jpg"
                        }
                    }
                    geometry Box {
                        size 4.0 0.25 4.0
                    }
                }
            ]
        }

#
#  (Main, +Y -Z) Stairs
#
        Transform {
            translation 5.0 -2.5 2.5
            children [
                DEF ThreeUpStraightStairs Group {
                    children [
                        DEF Stair Shape {
                            appearance USE Marble
                            geometry Box {
                                size 4.0 0.25 1.0
                            }
                        }
                        Transform {
                            translation 0.0 0.5 -1.0
                            children [ USE Stair ]
                        }
                        Transform {
                            translation 0.0 1.0 -2.0
                            children [ USE Stair ]
                        }
```

```
                    ]
                }
            ]
        }
        Transform {
            translation 5.0 -1.0 -0.5
            children [ USE ThreeUpStraightStairs ]
        }

#
#  (Main, +Y -Z) Platform
#
        Transform {
            translation 5.0 0.5 -5.0
            children [ USE Platform ]
        }

#
#  (Main, +Y -Z, +Y -Z) Stairs
#
        Transform {
            translation 5.0 1.0 -7.5
            children [ USE ThreeUpStraightStairs ]
        }

#
#  (Main, +Y -Z, +Y -Z) Door
#
        Transform {
            translation 5.0 5.0 -10.1
            children [
                Anchor {
                    url "temple.wrl"
                    description "Temple of the Orb"
                    children [
                        DEF Door Group {
                            children [
                                Shape {
                                    appearance Appearance {
                                        material Material {
                                            diffuseColor 1.0 1.0 1.0
                                        }
                                        texture ImageTexture {
                                            url "panel.jpg"
                                        }
                                        textureTransform TextureTransform {
                                            scale 1.0 2.0
                                        }
                                    }
                                    geometry Box {
                                        size 3.0 6.0 0.2
                                    }
                                }
                                Transform {
                                    translation 1.18 -0.0 0.2
                                    children [
                                        Shape {
                                            appearance Appearance {
                                                material Material {
                                                    diffuseColor 1.0 1.0 1.0
                                                }
```

```
                                                  texture ImageTexture {
                                                      url "blur.jpg"
                                                  }
                                              }
                                              geometry Sphere {
                                                  radius 0.15
                                              }
                                          }
                                      ]
                                  }
                              ]
                          }
                      ]
                  }
              ]
          }

#
#  (Main, +Y -Z, -X +Y -Z) Stairs
#
        Transform {
            translation 2.5 1.0 -5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                DEF ThreeUpRightCurvingStairs Transform {
                    translation 5.0 0.0 0.0
                    children [
                        DEF OffToLeftStair Transform {
                            translation -5.0 0.0 0.0
                            children [ USE Stair ]
                        }
                        Transform {
                            translation 0.0 0.5 0.0
                            rotation 0.0 1.0 0.0 -0.26
                            children [ USE OffToLeftStair ]
                        }
                        Transform {
                            translation 0.0 1.0 0.0
                            rotation 0.0 1.0 0.0 -0.52
                            children [ USE OffToLeftStair ]
                        }
                    ]
                }
            ]
        }
        Transform {
            translation 2.5 2.5 -5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                Transform {
                    rotation 0.0 1.0 0.0 -0.785
                    center 5.0 0.0 0.0
                    children [ USE ThreeUpRightCurvingStairs ]
                }
            ]
        }
        Transform {
            translation 2.5 4.0 -5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                Transform {
```

```
                    rotation 0.0 1.0 0.0 -1.57
                    center 5.0 0.0 0.0
                    children [ USE ThreeUpRightCurvingStairs ]
                }
            ]
        }

#
#  (Main, +Y -Z, -X +Y -Z) Door
#
        Transform {
            translation 2.5 5.0 -5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                Transform {
                    rotation 0.0 1.0 0.0 -2.09
                    center 5.0 0.0 0.0
                    children [
                        Transform {
                            translation 0.0 3.0 -0.6
                            children [
                                Anchor {
                                    url "prim.wrl"
                                    description "Primitives on Pedestals"
                                    children [ USE Door ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }

#
#  (Main, +Y -Z, +X -Y -Z) Stairs
#
        Transform {
            translation 7.5 0.0 -5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                DEF ThreeDownLeftCurvingStairs Transform {
                    translation 5.0 0.0 0.0
                    children [
                        USE OffToLeftStair,
                        Transform {
                            translation 0.0 -0.5 0.0
                            rotation 0.0 1.0 0.0 0.26
                            children [ USE OffToLeftStair ]
                        }
                        Transform {
                            translation 0.0 -1.0 0.0
                            rotation 0.0 1.0 0.0 0.52
                            children [ USE OffToLeftStair ]
                        }
                    ]
                }
            ]
        }
        Transform {
            translation 7.5 -1.5 -5.0
            rotation 0.0 1.0 0.0 1.57
```

```
            children [
                Transform {
                    rotation 0.0 1.0 0.0 0.785
                    center 5.0 0.0 0.0
                    children [ USE ThreeDownLeftCurvingStairs ]
                }
            ]
        }

#
#  (Main, +Y -Z, +X -Y -Z) Door
#
        Transform {
            translation 7.5 -2.5 -5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                Transform {
                    rotation 0.0 1.0 0.0 1.31
                    center 5.0 0.0 0.0
                    children [
                        Transform {
                            translation 0.0 3.0 0.6
                            rotation 0.0 1.0 0.0 3.14
                            children [
                                Anchor {
                                    url "doorway.wrl"
                                    description "The Doorway"
                                    children [ USE Door ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }

#
#  (Main, -X +Y) Stairs
#
        Transform {
            translation 2.5 -2.5 5.0
            rotation 0.0 1.0 0.0 1.57
            children [ USE ThreeUpStraightStairs ]
        }


#
#  (Main, -X +Y) Platform
#
        Transform {
            translation -2.0 -1.0 5.0
            children [ USE Platform ]
        }


#
#  (Main, -X +Y, +Y -Z) Stairs
#
        Transform {
            translation -2.0 -0.5 2.5
            children [
                DEF ThreeUpLeftCurvingStairs Transform {
```

```
                    translation -5.0 0.0 0.0
                    children [
                        DEF OffToRightStair Transform {
                            translation 5.0 0.0 0.0
                            children [ USE Stair ]
                        }
                        Transform {
                            translation 0.0 0.5 0.0
                            rotation 0.0 1.0 0.0 0.26
                            children [ USE OffToRightStair ]
                        }
                        Transform {
                            translation 0.0 1.0 0.0
                            rotation 0.0 1.0 0.0 0.52
                            children [ USE OffToRightStair ]
                        }
                    ]
                }
            ]
        }

#
#  (Main, -X +Y, +Y -Z) Door
#
        Transform {
            translation -2.0 0.5 2.5
            children [
                Transform {
                    rotation 0.0 1.0 0.0 0.52
                    center -5.0 0.0 0.0
                    children [
                        Transform {
                            translation 0.0 3.0 -0.6
                            children [
                                Anchor {
                                    url "monolith.wrl"
                                    description "Monolith World"
                                    children [ USE Door ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }

#
#  (Main, -X +Y, -X +Y) Stairs
#
        Transform {
            translation -4.5 -0.5 5.0
            rotation 0.0 1.0 0.0 1.57
            children [ USE ThreeUpStraightStairs ]
        }
        Transform {
            translation -7.5 1.0 5.0
            rotation 0.0 1.0 0.0 1.57
            children [ USE ThreeUpStraightStairs ]
        }

#
```

```
#  (Main, -X +Y, -X +Y) Door
#
        Transform {
            translation -10.1 5.0 5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                Anchor {
                    url "towers.wrl"
                    description "Towers Three"
                    children [ USE Door ]
                }
            ]
        }

#
#  (Main, +X -Y) Stairs
#
        Transform {
            translation 9.5 -4.5 5.0
            rotation 0.0 1.0 0.0 1.57
            children [ USE ThreeUpStraightStairs ]
        }


#
#  (Main, +X -Y) Platform
#
        Transform {
            translation 12.0 -5.0 5.0
            children [ USE Platform ]
        }

#
#  (Main, +X -Y, -Y +Z) Stairs
#
        Transform {
            translation 12.0 -6.5 9.5
            children [ USE ThreeUpStraightStairs ]
        }

#
#  (Main, +X -Y, -Y +Z) Platform
#
        Transform {
            translation 12.0 -7.0 12.0
            children [ USE Platform ]
        }

#
#  (Main, +X -Y, -Y +Z, -Y +Z) Stairs
#
        Transform {
            translation 12.0 -8.5 16.5
            children [ USE ThreeUpStraightStairs ]
        }

#
#  (Main, -X -Y +Z) Stairs
#
        Transform {
            translation 5.0 -3.5 7.5
            rotation 0.0 1.0 0.0 3.14
```

```
            children [
                DEF ThreeDownRightCurvingStairs Transform {
                    translation 5.0 0.0 0.0
                    children [
                        USE OffToLeftStair,
                        Transform {
                            translation 0.0 -0.5 0.0
                            rotation 0.0 1.0 0.0 -0.26
                            children [ USE OffToLeftStair ]
                        }
                        Transform {
                            translation 0.0 -1.0 0.0
                            rotation 0.0 1.0 0.0 -0.52
                            children [ USE OffToLeftStair ]
                        }
                    ]
                }
            ]
        }
        Transform {
            translation 5.0 -5.0 7.5
            rotation 0.0 1.0 0.0 3.14
            children [
                Transform {
                    rotation 0.0 1.0 0.0 -0.785
                    center 5.0 0.0 0.0
                    children [ USE ThreeDownRightCurvingStairs ]
                }
            ]
        }

#
#  (Main, -X -Y +Z) Door
#
        Transform {
            translation 5.0 -6.0 7.5
            children [
                Transform {
                    rotation 0.0 1.0 0.0 -1.48
                    center -5.0 0.0 0.0
                    children [
                        Transform {
                            translation 0.0 3.0 -0.6
                            rotation 0.0 1.0 0.0 3.14
                            children [
                                Anchor {
                                    url "robobill.wrl"
                                    description "Robot Bill"
                                    children [ USE Door ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }

#
#  Floor
#
        Shape {
```

```
            appearance Appearance {
                material Material {
                    ambientIntensity 0.3
                    diffuseColor 0.8 0.8 0.8
                }
                texture ImageTexture {
                    url "stone2.jpg"
                }
                textureTransform TextureTransform {
                    scale 20.0 20.0
                }
            }
            geometry IndexedFaceSet {
                coord Coordinate {
                    point [
                        -30.0 -9.0  30.0,
                         30.0 -9.0  30.0,
                         30.0 -9.0 -30.0,
                        -30.0 -9.0 -30.0,
                    ]
                }
                coordIndex [ 0, 1, 2, 3, -1, ]
                solid FALSE
            }
        }

#
#  Fake shadows on the floor
#
        Shape {
            appearance DEF ShadowColor Appearance {
                material Material {
                    ambientIntensity 0.0
                    diffuseColor 0.1 0.1 0.1
                    transparency 0.25
                }
            }
            geometry IndexedFaceSet {
                coord Coordinate {
                    point [
                        10.0 -8.5 17.0,
                        14.0 -8.5 17.0,
                        14.0 -8.5 3.0,
                        10.0 -8.5 3.0,

                        -10.0 -8.5 7.0,
                         10.0 -8.5 7.0,
                         10.0 -8.5 3.0,
                        -10.0 -8.5 3.0,

                         3.0 -8.5  7.0,
                         7.0 -8.5  7.0,
                         7.0 -8.5 -10.0,
                         3.0 -8.5 -10.0,
                    ]
                }
                coordIndex [
                    0, 1, 2, 3, -1,
                    4, 5, 6, 7, -1,
                    8, 9, 10, 11, -1,
                ]
```

```
        }
    }
    Transform {
        translation 2.5 -8.5 -5.0
        rotation 0.0 1.0 0.0 1.57
        children [
            DEF ThreeUpRightCurvingShadows Transform {
                translation 5.0 0.0 0.0
                children [
                    DEF OffToLeftShadow Transform {
                        translation -5.0 0.0 0.0
                        children [
                            DEF Shadow Shape {
                                appearance USE ShadowColor
                                geometry IndexedFaceSet {
                                    coord Coordinate {
                                        point [
                                            -2.0 0.0 0.5,
                                             2.0 0.0 0.5,
                                             2.0 0.0 -0.5,
                                            -2.0 0.0 -0.5,
                                        ]
                                    }
                                    coordIndex [ 0, 1, 2, 3, -1, ]
                                }
                            }
                        ]
                    }
                    Transform {
                        rotation 0.0 1.0 0.0 -0.26
                        children [ USE OffToLeftShadow ]
                    }
                    Transform {
                        rotation 0.0 1.0 0.0 -0.52
                        children [ USE OffToLeftShadow ]
                    }
                ]
            }
        ]
    }
    Transform {
        translation 2.5 -8.5 -5.0
        rotation 0.0 1.0 0.0 1.57
        children [
            Transform {
                rotation 0.0 1.0 0.0 -0.785
                center 5.0 0.0 0.0
                children [ USE ThreeUpRightCurvingShadows ]
            }
        ]
    }
    Transform {
        translation 2.5 -8.5 -5.0
        rotation 0.0 1.0 0.0 1.57
        children [
            Transform {
                rotation 0.0 1.0 0.0 -1.57
                center 5.0 0.0 0.0
                children [ USE ThreeUpRightCurvingShadows ]
            }
        ]
```

```
        }
        Transform {
            translation 7.5 -8.5 -5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                DEF ThreeDownLeftCurvingShadows Transform {
                    translation 5.0 0.0 0.0
                    children [
                        USE OffToLeftShadow,
                        Transform {
                            translation 0.0 0.0 0.0
                            rotation 0.0 1.0 0.0 0.26
                            children [ USE OffToLeftShadow ]
                        }
                        Transform {
                            translation 0.0 0.0 0.0
                            rotation 0.0 1.0 0.0 0.52
                            children [ USE OffToLeftShadow ]
                        }
                    ]
                }
            ]
        }
        Transform {
            translation 7.5 -8.5 -5.0
            rotation 0.0 1.0 0.0 1.57
            children [
                Transform {
                    rotation 0.0 1.0 0.0 0.785
                    center 5.0 0.0 0.0
                    children [ USE ThreeDownLeftCurvingShadows ]
                }
            ]
        }
        Transform {
            translation -2.0 -8.5 2.5
            children [
                DEF ThreeUpLeftCurvingShadows Transform {
                    translation -5.0 0.0 0.0
                    children [
                        DEF OffToRightShadow Transform {
                            translation 5.0 0.0 0.0
                            children [ USE Shadow ]
                        }
                        Transform {
                            translation 0.0 0.0 0.0
                            rotation 0.0 1.0 0.0 0.26
                            children [ USE OffToRightShadow ]
                        }
                        Transform {
                            translation 0.0 0.0 0.0
                            rotation 0.0 1.0 0.0 0.52
                            children [ USE OffToRightShadow ]
                        }
                    ]
                }
            ]
        }
        Transform {
            translation 5.0 -8.5 7.5
```

```
            rotation 0.0 1.0 0.0 3.14
            children [
                DEF ThreeDownRightCurvingShadows Transform {
                    translation 5.0 0.0 0.0
                    children [
                        USE OffToLeftShadow,
                        Transform {
                            translation 0.0 0.0 0.0
                            rotation 0.0 1.0 0.0 -0.26
                            children [ USE OffToLeftShadow ]
                        }
                        Transform {
                            translation 0.0 0.0 0.0
                            rotation 0.0 1.0 0.0 -0.52
                            children [ USE OffToLeftShadow ]
                        }
                    ]
                }
            ]
        }
        Transform {
            translation 5.0 -8.5 7.5
            rotation 0.0 1.0 0.0 3.14
            children [
                Transform {
                    rotation 0.0 1.0 0.0 -0.785
                    center 5.0 0.0 0.0
                    children [ USE ThreeDownRightCurvingShadows ]
                }
            ]
        }
    ]
}
```

```
#VRML V2.0 utf8
#
#  A Dinette
#  dinette.wrl
#        by David R. Nadeau
#
#  This example illustrates the use of the Inline node.
#
#  Two separate world components are inlined:  a table, and a chair.  The
#  chair is then instanced three times more to put one chair on each side
#  of the table.  The completed shape is a dinette, suitable for inlining
#  in to further worlds.
#
#  Things to experiment with:
#        Add things to the table, such as plates, glasses, candles, etc.
#
#        Add a point light source at the tip of a candle on the table
#
#        Change the table to use a formica table top texture.  Change the
#        table and chair legs to chrome by using a specular color and
#        shininess factor in their Material nodes.  Add a cloth texture to
#        the chair seats.
#
#        Place the dinette in a room by inlining it.
#

#
#  Table
#
Inline { url "table.wrl" }

#
#  Chairs
#
Transform {
    translation 0.95 0.0 0.0
    children DEF Chair Inline { url "chair.wrl" }
}
Transform {
    translation -0.95 0.0 0.0
    rotation 0.0 1.0 0.0 3.14
    children USE Chair
}
Transform {
    translation 0.0 0.0 0.95
    rotation 0.0 1.0 0.0 -1.57
    children USE Chair
}
Transform {
    translation 0.0 0.0 -0.95
    rotation 0.0 1.0 0.0 1.57
    children USE Chair
}
```

```
#VRML V2.0 utf8
#
#  table.wrl
#        by David R. Nadeau
#
#  This table is inlined in to the dinette.wrl file to illustrate inline
#  groups.
#

#
#  Table top
#
Transform {
    translation 0.0 0.615 0.0
    children [
        Shape {
            appearance DEF Brown Appearance {
                material Material {
                    diffuseColor 0.6 0.35 0.0
                }
            }
            geometry Cylinder {
                radius 0.7
                height 0.03
            }
        }
    ]
}

#
#  Table leg
#
Transform {
    translation 0.0 0.3075 0.0
    children [
        Shape {
            appearance USE Brown
            geometry Box {
                size 0.09 0.57 0.09
            }
        }
    ]
}

#
#  Table leg base
#
Transform {
    translation 0.0 0.015 0.0
    children [
        Shape {
            appearance USE Brown
            geometry Box {
                size 0.5 0.03 0.5
            }
        }
    ]
}
Transform {
    translation 0.0 0.045 0.0
    children [
```

```
        Shape {
            appearance USE Brown
            geometry Box {
                size 0.35 0.03 0.35
            }
        }
    ]
}
```

```
#VRML V2.0 utf8
#
#  Dinette Chair
#  chair.wrl
#        by David R. Nadeau
#
#  This chair is inlined in to the dinette.wrl file to illustrate inline
#  groups.
#

#
#  Chair seat
#
Transform {
    translation 0.0 0.5 0.0
    children [
        Shape {
            appearance DEF Brown Appearance {
                material Material {
                    diffuseColor 0.6 0.35 0.0
                }
            }
            geometry Box {
                size 0.39 0.03 0.41
            }
        }
    ]
}

#
#  Chair legs
#
Transform {
    translation 0.1575 0.2485 0.1575
    children [
        DEF Leg Shape {
            appearance USE Brown
            geometry Box {
                size 0.03 0.497 0.03
            }
        }
    ]
}
Transform {
    translation -0.1575 0.2485 0.1575
    children [ USE Leg ]
}
Transform {
    translation -0.1575 0.2485 -0.1575
    children [ USE Leg ]
}
Transform {
    translation 0.1575 0.2485 -0.1575
    children [ USE Leg ]
}

#
#  Chair back
#
Transform {
    translation 0.1875 0.5 0.0
```

```
    rotation 0.0 0.0 1.0 -0.17
    children [
        Transform {
            translation 0.0 0.54 0.0
            children [
                Shape {
                    appearance USE Brown
                    geometry Box {
                        size 0.06 0.17 0.43
                    }
                }
            ]
        }

        Transform {
            translation 0.0 0.2275 0.0
            children [
                DEF BackPole Shape {
                    appearance USE Brown
                    geometry Box {
                        size 0.02 0.455 0.02
                    }
                }
            ]
        }
        Transform {
            translation 0.0 0.2275 -0.083
            children [ USE BackPole ]
        }
        Transform {
            translation 0.0 0.2275 0.083
            children [ USE BackPole ]
        }
        Transform {
            translation 0.0 0.2275 -0.166
            children [ USE BackPole ]
        }
        Transform {
            translation 0.0 0.2275 0.166
            children [ USE BackPole ]
        }
    ]
}
```

Nomeando nós

Motivação

Sintaxe: DEF

Sintaxe: USE

Usando nós nomeados

Um exemplo do uso de nós nomeados

Resumo

Nomeando nós

Motivação

- **Se várias formas possuem a mesma geometria ou aparência, deve-se usar múltiplos nós duplicados, um para cada uso**
- **Define um nome para a primeira ocorrência de um nó**
- **Usa-se o nome para compartilhar o mesmo nó dentro de um novo contexto**

Nomeando nós

Sintaxe: DEF

- **A sintaxe `DEF` fornece um nome para um nó**

```
DEF RedColor Material {
    diffuseColor 1.0 0.0 0.0
}
```

- **Você pode nomear qualquer nó**
- **Nomes podem ser qualquer seqüência de letras e números**

. Nomes devem ser únicos no arquivo

Nomeando nós

Sintaxe: USE

- **A sintaxe `USE` utiliza um nó nomeado anteriormente**

```
Appearance {
    material USE RedColor
}
```

- **O re-uso de um nó nomeado é chamado de *instância***
- **Um nó nomeado pode ter qualquer número de instâncias**
- **Cada instância compartilha a mesma descrição do nó**

Nomeando nós

Usando nós nomeados

- **Nomeando e usando nós:**

. Diminui impressão

. Reduz o tamanho do arquivo

. Permite rápidas mudanças em formas com os mesmos atributos

. Acelera o processamento no browser

- **Nomes são necessários para animação...**

Nomeando nós

Um exemplo do uso de nós nomeados

[dinette.wrl]

Nomeando nós

Resumo

. **DEF** nomeia um nó

. **USE** utiliza um nó nomeado

Sumário dos exemplos

A fairy-tale castle

A bar plot

A simple spaceship

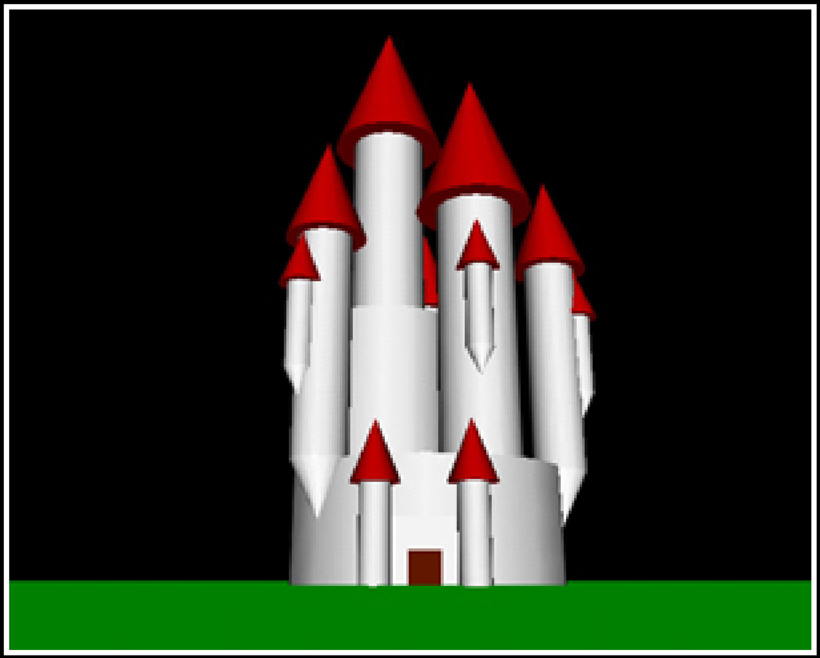
A juggling hand

Sumário dos exemplos

A fairy-tale castle

. `Cylinder` constrói as torres

. `Cone` constrói os telhados e algumas bases das torres

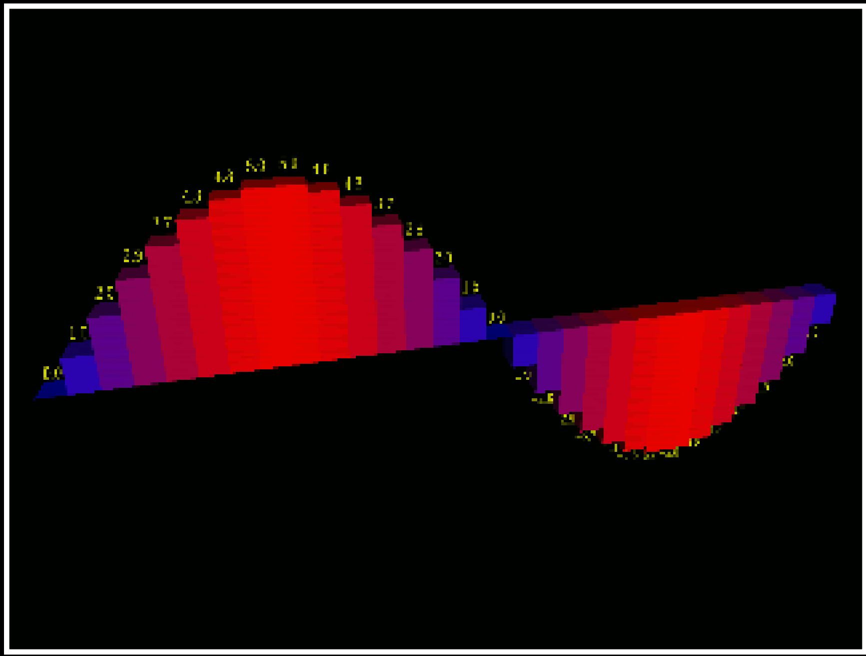


[castle.wrl]

Sumário dos exemplos

A bar plot

- **Box** cria as barras
- **Text** fornece os nomes das barras
- **Billboard** mantém os nomes das barras voltado para o observador

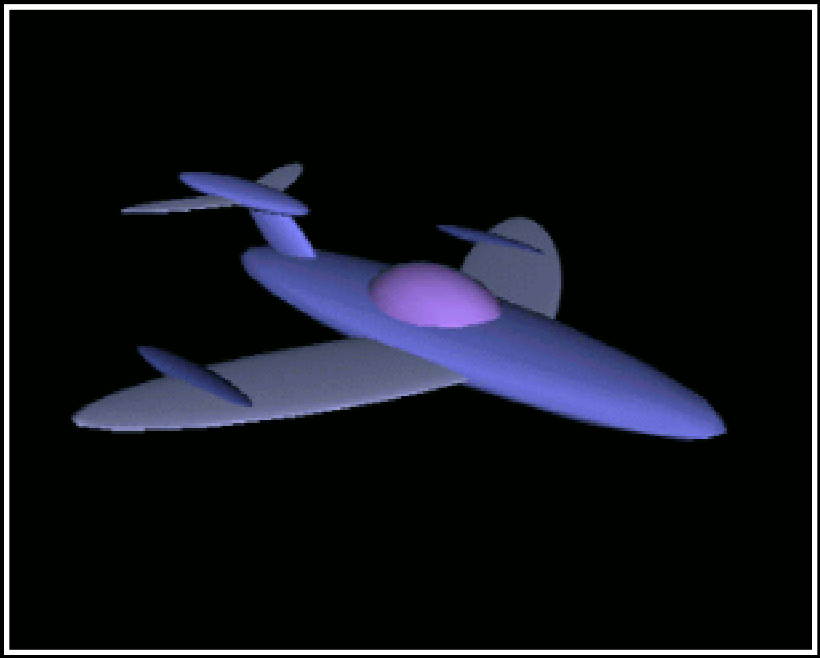


[**barplot.wrl**]

Sumário de exemplos

A simple spaceship

- `Sphere` **constrói todas as partes da nave**
- `Transform` **escalona as esferas da nave**

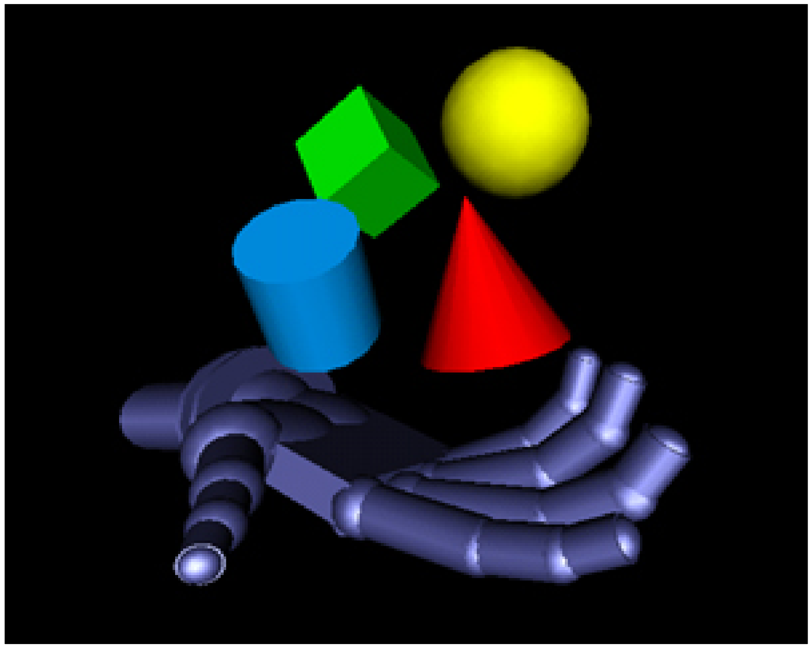


[space2.wrl]

Sumário de exemplos

A juggling hand

- **`Cylinder` e `Sphere` constróem os dedos e juntas**
- **`Transform` articulam a mão**



[hand.wrl]

```
#VRML V2.0 utf8
#
#  castle.wrl
#  Fairy-tale castle
#       by David R. Nadeau
#
#  This world builds a fairy-tale castle atop a green hill with
#  a blue sky behind.  There are no animations or fun things to
#  click on...  that's up to you.
#
#  This world illustrates how far you can go using the primitive
#  geometry nodes:  Box, Cone, and Cylinder.  An entire fairy-tale
#  castle is built by translating and rotating these shapes into
#  position.
#
#  Things to experiment with...
#       Change the color scheme to make the castle an evil lair.
#
#       Add texture maps to make the castle walls look like they are built
#       from stone.
#
#       Make the castle door an anchor.  Clicking on the door leads to
#       another world describing the castle interior.
#
#       Add panorama cloud and mountain textures to the Background node.
#
#       In Nintendo 64 style, add platforms that rise and fall and cannons
#       that rise up when you approach.  You can build these using
#       PositionInterpolators and TimeSensors to run the animations,
#       and ProximitySensors to trigger the actions.
#

Transform {
    translation 0.0 -1.6 0.0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 0.0 0.0
                    emissiveColor 0.0 0.5 0.0
                }
            }
            geometry Box { size 2000.0 0.01 2000.0 }
        }
    ]
}

#
#  Castle
#
DEF Castle Transform {
    translation 0.0 -1.6 -200.0
    children [
    #  Tower base
        Transform {
            translation 0.0 20.0 0.0
            children Shape {
                appearance DEF CastleColor Appearance {
                    material Material {
                        diffuseColor 1.0 1.0 1.0
                        ambientIntensity 0.5
```

```
                    }
                }
                geometry Cylinder {
                    radius 50.0
                    height 40.0
                    bottom FALSE
                }
            }
        }
    #  Entryway
        Transform {
            translation 0.0 0.0 52.0
            children [
            # Entry block
                Transform {
                    translation 0.0 10.0 0.0
                    children Shape {
                        appearance USE CastleColor
                        geometry Box { size 30.0 20.0 6.0 }
                    }
                }
            # Door
                Transform {
                    translation 0.0 5.0 3.0
                    children Shape {
                        appearance Appearance {
                            material Material {
                                diffuseColor 0.4 0.1 0.0
                            }
                        }
                        geometry Box { size 10.0 10.0 0.5 }
                    }
                }
            # Left entry tower
                Transform {
                    translation -15.0 0.0 3.0
                    children [
                    # Tower section
                        DEF EntryTower Transform {
                            translation 0.0 15.0 0.0
                            children Shape {
                                appearance USE CastleColor
                                geometry Cylinder {
                                    radius 5.0
                                    height 30.0
                                    bottom FALSE
                                    top FALSE
                                }
                            }
                        }
                    # Roof
                        DEF EntryRoof Transform {
                            translation 0.0 40.0 0.0
                            children Shape {
                                appearance DEF RoofColor Appearance {
                                    material Material {
                                        diffuseColor 1.0 0.0 0.0
                                        ambientIntensity 0.5
                                    }
                                }
                                geometry Cone {
```

```
                                    bottomRadius 8.0
                                    height 20.0
                                }
                            }
                        }
                    ]
                }
            # Right entry tower
                Transform {
                    translation 15.0 0.0 3.0
                    children [
                        USE EntryTower
                        USE EntryRoof
                    ]
                }
            ]
        }
    #  Left tower atop base
        Transform {
            translation -15.0 40.0 -5.0
            children [
            # Lower tower section
                Transform {
                    translation 0.0 30.0 0.0
                    children Shape {
                        appearance USE CastleColor
                        geometry Cylinder {
                            radius 25.0
                            height 60.0
                            bottom FALSE
                        }
                    }
                }
            # Upper tower section
                Transform {
                    translation 0.0 100.0 0.0
                    children Shape {
                        appearance USE CastleColor
                        geometry Cylinder {
                            radius 15.0
                            height 80.0
                            bottom FALSE
                            top FALSE
                        }
                    }
                }
            # Roof
                Transform {
                    translation 0.0 170.0 0.0
                    children DEF MainRoofCone Shape {
                        appearance USE RoofColor
                        geometry Cone {
                            bottomRadius 23.0
                            height 60.0
                        }
                    }
                }
            ]
        }
    #  Right tower atop base
```

```
        Transform {
            translation 20.0 40.0 15.0
            children [
            # Tower section
                Transform {
                    translation 0.0 50.0 0.0
                    children Shape {
                        appearance USE CastleColor
                        geometry Cylinder {
                            radius 15.0
                            height 100.0
                            bottom FALSE
                            top FALSE
                        }
                    }
                }
            # Roof
                Transform {
                    translation 0.0 130.0 0.0
                    children USE MainRoofCone
                }
            # Sub-tower
                Transform {
                    translation 0.0 30.0 15.0
                    children [
                    # Tower section
                        DEF SmallSubTower Transform {
                            translation 0.0 25.0 0.0
                            children Shape {
                                appearance USE CastleColor
                                geometry Cylinder {
                                    radius 5.0
                                    height 30.0
                                    bottom FALSE
                                    top FALSE
                                }
                            }
                        }
                    # Base cone
                        DEF SmallSubBase Transform {
                            translation 0.0 5.0 0.0
                            rotation 1.0 0.0 0.0 3.14159
                            children Shape {
                                appearance USE CastleColor
                                geometry Cone {
                                    bottomRadius 5.0
                                    height 10.0
                                    bottom FALSE
                                }
                            }
                        }
                    # Roof
                        DEF SmallSubRoof Transform {
                            translation 0.0 50.0 0.0
                            children Shape {
                                appearance USE RoofColor
                                geometry Cone {
                                    bottomRadius 8.0
                                    height 20.0
                                }
                            }
```

```
                    }
                ]
            }
        ]
    }
#  Right large sub tower
    Transform {
        translation 50.0 20.0 0.0
        children [
        # Tower section
            DEF LargeSubTower Transform {
                translation 0.0 60.0 0.0
                children Shape {
                    appearance USE CastleColor
                    geometry Cylinder {
                        radius 10.0
                        height 80.0
                        bottom FALSE
                        top FALSE
                    }
                }
            }
        # Base cone
            DEF LargeSubBase Transform {
                translation 0.0 10.0 0.0
                rotation 1.0 0.0 0.0 3.14159
                children Shape {
                    appearance USE CastleColor
                    geometry Cone {
                        bottomRadius 10.0
                        height 20.0
                        bottom FALSE
                    }
                }
            }
        # Roof
            DEF LargeSubRoof Transform {
                translation 0.0 120.0 0.0
                children Shape {
                    appearance USE RoofColor
                    geometry Cone {
                        bottomRadius 15.0
                        height 40.0
                    }
                }
            }
            Transform {
                translation 10.0 40.0 0.0
                children [
                    USE SmallSubTower
                    USE SmallSubBase
                    USE SmallSubRoof
                ]
            }
        ]
    }
#  Left large sub tower
    Transform {
        translation -35.36 20.0 35.36
        children [
```

```
                USE LargeSubTower
                USE LargeSubBase
                USE LargeSubRoof
                Transform {
                    translation -7.07 40.0 7.07
                    children [
                        USE SmallSubTower
                        USE SmallSubBase
                        USE SmallSubRoof
                    ]
                }
            ]
        }
    #  Back large sub tower
        Transform {
            translation 0.0 20.0 -50.0
            children [
                USE LargeSubTower
                USE LargeSubBase
                USE LargeSubRoof
                Transform {
                    translation 0.0 40.0 -10.0
                    children [
                        USE SmallSubTower
                        USE SmallSubBase
                        USE SmallSubRoof
                    ]
                }
            ]
        }
    ]
}
```

```
#VRML V2.0 utf8

# WHAT:   Bar Plot
# WHO:    John L. Moreland
# WHY:    Demonstrates Transform, Shape, Box, Text, Billboard

# Sample 0 of 29
Transform {
    translation 0.000000 0.000000 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.000000 0.000000 1.000000
                }
            }
            geometry Box {
                size 1.000000 0.000000 1.000000
            }
        }
    ]
}
Transform {
    translation 0.000000 0.476190 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "0.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 1 of 29
Transform {
    translation 1.000000 0.519779 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.207912 0.000000 0.792089
                }
            }
            geometry Box {
                size 1.000000 1.039558 1.000000
            }
        }
```

```
    ]
}
Transform {
    translation 1.000000 1.515748 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "1.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 2 of 29
Transform {
    translation 2.000000 1.016841 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.406736 0.000000 0.593264
                }
            }
            geometry Box {
                size 1.000000 2.033682 1.000000
            }
        }
    ]
}
Transform {
    translation 2.000000 2.509872 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "2.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
```

```
                    }
                }
            ]
        }
    ]
}

# Sample 3 of 29
Transform {
    translation 3.000000 1.469462 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.587785 0.000000 0.412215
                }
            }
            geometry Box {
                size 1.000000 2.938924 1.000000
            }
        }
    ]
}
Transform {
    translation 3.000000 3.415115 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "2.9"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 4 of 29
Transform {
    translation 4.000000 1.857861 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.743144 0.000000 0.256856
                }
            }
            geometry Box {
                size 1.000000 3.715722 1.000000
            }
```

```
        }
    ]
}
Transform {
    translation 4.000000 4.191912 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "3.7"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 5 of 29
Transform {
    translation 5.000000 2.165062 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.866025 0.000000 0.133975
                }
            }
            geometry Box {
                size 1.000000 4.330125 1.000000
            }
        }
    ]
}
Transform {
    translation 5.000000 4.806315 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "4.3"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
```

```
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 6 of 29
Transform {
    translation 6.000000 2.377640 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.951056 0.000000 0.048944
                }
            }
            geometry Box {
                size 1.000000 4.755281 1.000000
            }
        }
    ]
}
Transform {
    translation 6.000000 5.231471 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "4.8"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 7 of 29
Transform {
    translation 7.000000 2.486305 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.994522 0.000000 0.005478
                }
            }
            geometry Box {
                size 1.000000 4.972609 1.000000
```

```
            }
        }
    ]
}
Transform {
    translation 7.000000 5.448800 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "5.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 8 of 29
Transform {
    translation 8.000000 2.486305 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.994522 0.000000 0.005478
                }
            }
            geometry Box {
                size 1.000000 4.972610 1.000000
            }
        }
    ]
}
Transform {
    translation 8.000000 5.448800 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "5.0"
                        fontStyle FontStyle {
                            size 0.454545
```

```
                        justify [ "MIDDLE", "MIDDLE" ]
                    }
                }
            }
        ]
    }
  ]
}

# Sample 9 of 29
Transform {
    translation 9.000000 2.377643 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.951057 0.000000 0.048943
                }
            }
            geometry Box {
                size 1.000000 4.755285 1.000000
            }
        }
    ]
}
Transform {
    translation 9.000000 5.231476 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "4.8"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 10 of 29
Transform {
    translation 10.000000 2.165066 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.866026 0.000000 0.133974
                }
            }
            geometry Box {
```

```
                size 1.000000 4.330132 1.000000
            }
        }
    ]
}
Transform {
    translation 10.000000 4.806322 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "4.3"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 11 of 29
Transform {
    translation 11.000000 1.857865 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.743146 0.000000 0.256854
                }
            }
            geometry Box {
                size 1.000000 3.715730 1.000000
            }
        }
    ]
}
Transform {
    translation 11.000000 4.191921 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "3.7"
                        fontStyle FontStyle {
```

```
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 12 of 29
Transform {
    translation 12.000000 1.469467 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.587787 0.000000 0.412213
                }
            }
            geometry Box {
                size 1.000000 2.938935 1.000000
            }
        }
    ]
}
Transform {
    translation 12.000000 3.415125 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "2.9"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 13 of 29
Transform {
    translation 13.000000 1.016847 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.406739 0.000000 0.593261
                }
            }
```

```
            geometry Box {
                size 1.000000 2.033694 1.000000
            }
        }
    ]
}
Transform {
    translation 13.000000 2.509884 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "2.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 14 of 29
Transform {
    translation 14.000000 0.519785 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.207914 0.000000 0.792086
                }
            }
            geometry Box {
                size 1.000000 1.039570 1.000000
            }
        }
    ]
}
Transform {
    translation 14.000000 1.515760 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "1.0"
```

```
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 15 of 29
Transform {
    translation 15.000000 0.000006 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.000003 0.000000 0.999997
                }
            }
            geometry Box {
                size 1.000000 0.000013 1.000000
            }
        }
    ]
}
Transform {
    translation 15.000000 0.476203 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "0.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 16 of 29
Transform {
    translation 16.000000 -0.519772 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.207909 0.000000 0.792091
                }
```

```
            }
            geometry Box {
                size 1.000000 1.039545 1.000000
            }
        }
    ]
}
Transform {
    translation 16.000000 -1.515735 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-1.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 17 of 29
Transform {
    translation 17.000000 -1.016835 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.406734 0.000000 0.593266
                }
            }
            geometry Box {
                size 1.000000 2.033669 1.000000
            }
        }
    ]
}
Transform {
    translation 17.000000 -2.509860 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
```

```
                        string "-2.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 18 of 29
Transform {
    translation 18.000000 -1.469457 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.587783 0.000000 0.412217
                }
            }
            geometry Box {
                size 1.000000 2.938913 1.000000
            }
        }
    ]
}
Transform {
    translation 18.000000 -3.415104 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-2.9"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 19 of 29
Transform {
    translation 19.000000 -1.857857 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.743143 0.000000 0.256857
```

```
                }
            }
            geometry Box {
                size 1.000000 3.715713 1.000000
            }
        }
    ]
}
Transform {
    translation 19.000000 -4.191904 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-3.7"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 20 of 29
Transform {
    translation 20.000000 -2.165059 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.866024 0.000000 0.133976
                }
            }
            geometry Box {
                size 1.000000 4.330118 1.000000
            }
        }
    ]
}
Transform {
    translation 20.000000 -4.806308 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
```

```
                    geometry Text {
                        string "-4.3"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 21 of 29
Transform {
    translation 21.000000 -2.377639 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.951055 0.000000 0.048945
                }
            }
            geometry Box {
                size 1.000000 4.755277 1.000000
            }
        }
    ]
}
Transform {
    translation 21.000000 -5.231468 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-4.8"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 22 of 29
Transform {
    translation 22.000000 -2.486304 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
```

```
                    diffuseColor 0.994521 0.000000 0.005479
                }
            }
            geometry Box {
                size 1.000000 4.972608 1.000000
            }
        }
    ]
}
Transform {
    translation 22.000000 -5.448798 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-5.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 23 of 29
Transform {
    translation 23.000000 -2.486306 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.994522 0.000000 0.005478
                }
            }
            geometry Box {
                size 1.000000 4.972611 1.000000
            }
        }
    ]
}
Transform {
    translation 23.000000 -5.448802 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
```

```
                    }
                    geometry Text {
                        string "-5.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 24 of 29
Transform {
    translation 24.000000 -2.377645 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.951058 0.000000 0.048942
                }
            }
            geometry Box {
                size 1.000000 4.755289 1.000000
            }
        }
    ]
}
Transform {
    translation 24.000000 -5.231480 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-4.8"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 25 of 29
Transform {
    translation 25.000000 -2.165069 0.000000
    children [
        Shape {
            appearance Appearance {
```

```
                material Material {
                    diffuseColor 0.866028 0.000000 0.133972
                }
            }
            geometry Box {
                size 1.000000 4.330138 1.000000
            }
        }
    ]
}
Transform {
    translation 25.000000 -4.806329 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-4.3"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 26 of 29
Transform {
    translation 26.000000 -1.857870 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.743148 0.000000 0.256852
                }
            }
            geometry Box {
                size 1.000000 3.715739 1.000000
            }
        }
    ]
}
Transform {
    translation 26.000000 -4.191930 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
```

```
                        }
                    }
                    geometry Text {
                        string "-3.7"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 27 of 29
Transform {
    translation 27.000000 -1.469473 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.587789 0.000000 0.412211
                }
            }
            geometry Box {
                size 1.000000 2.938946 1.000000
            }
        }
    ]
}
Transform {
    translation 27.000000 -3.415136 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-2.9"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 28 of 29
Transform {
    translation 28.000000 -1.016852 0.000000
    children [
        Shape {
```

```
            appearance Appearance {
                material Material {
                    diffuseColor 0.406741 0.000000 0.593259
                }
            }
            geometry Box {
                size 1.000000 2.033704 1.000000
            }
        }
    ]
}
Transform {
    translation 28.000000 -2.509895 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.8 0.8 0
                        }
                    }
                    geometry Text {
                        string "-2.0"
                        fontStyle FontStyle {
                            size 0.454545
                            justify [ "MIDDLE", "MIDDLE" ]
                        }
                    }
                }
            ]
        }
    ]
}

# Sample 29 of 29
Transform {
    translation 29.000000 -0.519791 0.000000
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.207916 0.000000 0.792084
                }
            }
            geometry Box {
                size 1.000000 1.039582 1.000000
            }
        }
    ]
}
Transform {
    translation 29.000000 -1.515773 0.000000
    children [
        Billboard {
            axisOfRotation 0.0 0.0 0.0
            children [
                Shape {
                    appearance Appearance {
                        material Material {
```

```
                        diffuseColor 0.8 0.8 0
                    }
                }
                geometry Text {
                    string "-1.0"
                    fontStyle FontStyle {
                        size 0.454545
                        justify [ "MIDDLE", "MIDDLE" ]
                    }
                }
            }
        ]
    }
  ]
}
```

```
#VRML V2.0 utf8
#
#  space2.wrl
#  Simple spaceship
#       by David R. Nadeau
#
#  This file illustrates the use of VRML primitive shapes to build a
#  more complex shape.  Transform nodes are used to position and orient
#  the shapes.  Scaling in the Transform node squishes and stretches shapes,
#  enabling you to create flattened ellipsoids out of spheres, and more.
#


# Wing
DEF LeftWing Transform {
    translation 0.0 0.0 -0.9
    rotation 0.0 1.0 0.0 0.52
    scale 0.4 0.035 1.5
    children [
        DEF WingSphere Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.7 0.7 1.0
                    specularColor 1.0 1.0 1.0
                    shininess 0.1
                }
            }
            geometry Sphere { }
        }
    ]
}
DEF RightWing Transform {
    translation 0.0 0.0 0.9
    rotation 0.0 1.0 0.0 -0.52
    scale 0.4 0.035 1.5
    children [ USE WingSphere ]
}

# Fuselage
DEF Fuselage Transform {
    scale 2.0 0.2 0.5
    children [
        DEF FuselageSphere Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.5 0.5 1.0
                    specularColor 0.4 0.7 1.0
                    shininess 0.1
                }
            }
            geometry Sphere { }
        }
    ]
}

# Dome
Transform {
    scale 0.6 0.4 0.375
    children [
        Shape {
            appearance Appearance {
```

```
                material Material {
                    ambientIntensity 0.5
                    diffuseColor 0.7 0.5 1.0
                    specularColor 1.0 0.0 0.3
                    shininess 0.5
                }
            }
            geometry Sphere { }
        }
    ]
}

# Engines
Transform {
    translation -0.6 0.0 -1.5
    scale 0.6 0.06 0.1
    children [
        DEF EngineSphere Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.3 0.3 0.7
                }
            }
            geometry Sphere { }
        }
    ]
}
Transform {
    translation -0.6 0.0 1.5
    scale 0.6 0.06 0.1
    children [ USE EngineSphere ]
}

# Tail
Transform {
    translation -2.0 0.5 0.0
    scale 0.4 0.4 0.4
    children [
        USE LeftWing
        USE RightWing
        USE Fuselage
    ]
}
Transform {
    translation -1.5 0.25 0.0
    rotation 0.0 0.0 1.0 -0.6
    scale 0.5 0.2 0.075
    children [ USE FuselageSphere ]
}
```

Introdução a animação

Motivação

Construindo circuitos de animação

Exemplos

Roteandos eventos

Usando entrada e saídas de nós

Exemplos de entradas

Exemplos de saídas

Sintaxe: ROUTE

Tipos de dados de eventos

Seguindo a convenção dos nomes

Um exemplo de animação

Um exemplo de animação

Usando múltiplas rotas

Resumo

```
#VRML V2.0 utf8

#  hand.wrl
#  Robotic hand juggling primitives
#       by David R. Nadeau
#
#  This file illustrates the use of the primitive shapes and a lot of
#  nested Transform nodes to created articulated structures... in
#  particular, a robotic hand.  Each of the finger joints are controlled
#  by a separate rotation.  Changing the rotation angles (by editing
#  this file) will open and close the hand.
#
#  For convenience, each of the hand's parts are spearately authored
#  within a Switch node whose 'whichChoice' field is set to -1...
#  forcing the parts to be skipped during drawing.  The parts in the
#  Switch, however, are assembled later in a 'hand' shape at the end
#  of the Switch, and that shape is then instanced outside of the
#  Switch.  The effect is to create a file structure that enables you
#  to build things piece by piece, then use Transform's later on to
#  assemble those pieces into a drawn shape.
#

#
#  Hand components
#
Switch {
        whichChoice -1
        choice [

        #
        #  Generic color
        #
        Shape {
                appearance DEF HandColor Appearance {
                        material Material {
                                ambientIntensity 0.3
                                diffuseColor 0.3 0.3 0.5
                                specularColor 0.7 0.7 0.8
                                shininess 0.1
                        }
                }
        }


        #
        # Parts of a finger
        #

        # Base of a finger
        DEF finger_base Group {
                children [
                        Shape {
                                appearance USE HandColor
                                geometry Sphere { radius 0.41 }
                        }
                        Transform {
                                translation 0.0 0.875 0.0
                                children [
                                        Shape {
                                                appearance USE HandColor
                                                geometry Cylinder {
                                                        height 1.75
```

```
                                                radius 0.35
                                        }
                                }
                                Transform {
                                        translation 0.0 0.835 0.0
                                        children Shape {
                                                appearance USE HandColor
                                                geometry Sphere { radius 0.35 }
                                        }
                                }
                        ]
                }
        ]
}

# Middle part of a finger
DEF finger_middle Transform {
        translation 0.0 0.5 0.0
        children [
                Shape {
                        appearance USE HandColor
                        geometry Cylinder {
                                height 1.0
                                radius 0.315
                        }
                }
                Transform {
                        translation 0.0 0.5 0.0
                        children Shape {
                                appearance USE HandColor
                                geometry Sphere { radius 0.32 }
                        }
                }
        ]
}

# Tip of a finger
DEF finger_tip Transform {
        rotation 1.0 0.0 0.0  -0.1
        children Transform {
                translation 0.0 0.4 0.0
                children [
                        Shape {
                                appearance USE HandColor
                                geometry Cylinder {
                                        height 0.75
                                        radius 0.27
                                }
                        }
                        Transform {
                                translation 0.0 0.35 0.0
                                scale 1.0 0.7 1.0
                                children Shape {
                                        appearance USE HandColor
                                        geometry Sphere { radius 0.235 }
                                }
                        }
                ]
        }
}
```

```
        #
        # Fingers built from the finger parts
        #

        # Index finger
        DEF finger1 Transform {
                # Finger spread
                #                Z    +2 degrees
                rotation 0.0 0.0 1.0  0.035
                children Transform {
                        # Finger twist in
                        rotation 0.0 1.0 0.0  -0.03
                        children Transform {
                                # base rotate
#                               rotation 1.0 0.0 0.0  -1.1
                                rotation 1.0 0.0 0.0  -0.2
                                children [
                                        USE finger_base
                                        Transform {
                                                translation 0.0 1.71 0.0
                                                # middle rotate
#                                               rotation 1.0 0.0 0.0  -1.7
                                                rotation 1.0 0.0 0.0  -0.2
                                                children [
                                                        USE finger_middle
                                                        Transform {
                                                                translation 0.0 1.0 0.0
                                                                # tip rotate
#                                                               rotation 1.0 0.0 0.0  -1.0
                                                                rotation 1.0 0.0 0.0  -0.2
                                                                children USE finger_tip
                                                        }
                                                ]
                                        }
                                ]
                        }
                }
        }

        # Middle finger
        DEF finger2 Transform {
                # Finger spread
                rotation 0.0 0.0 1.0  0.0
                scale 1.01 1.15 1.01
                children Transform {
                        # Finger twist in
                        rotation 0.0 1.0 0.0  0.0
                        children Transform {
                                translation 0.0 0.2 0.0
                                # base rotate
#                               rotation 1.0 0.0 0.0  -1.05
                                rotation 1.0 0.0 0.0  -0.4
                                children [
                                        USE finger_base
                                        Transform {
                                                translation 0.0 1.71 0.0
                                                # middle rotate
#                                               rotation 1.0 0.0 0.0  -1.7
                                                rotation 1.0 0.0 0.0  -0.4
```

```
                                                children [
                                                        USE finger_middle
                                                        Transform {
                                                                translation 0.0 1.0 0.0
                                                                # tip rotate
#                                                               rotation 1.0 0.0 0.0  -1.0
                                                                rotation 1.0 0.0 0.0  -0.4
                                                                children USE finger_tip
                                                        }
                                                ]
                                        }
                                ]
                        }
                }
        }

        # Ring finger
        DEF finger3 Transform {
                # Finger spread
                rotation 0.0 0.0 1.0  -0.0175
                scale 1.0 1.05 1.0
                children Transform {
                        # Finger twist in
                        rotation 0.0 1.0 0.0  0.065
                        children Transform {
                                translation 0.0 0.1 0.0
                                # base rotate
#                               rotation 1.0 0.0 0.0  -1.1
                                rotation 1.0 0.0 0.0  -0.6
                                children [
                                        USE finger_base
                                        Transform {
                                                translation 0.0 1.71 0.0
                                                # middle rotate
#                                               rotation 1.0 0.0 0.0  -1.7
                                                rotation 1.0 0.0 0.0  -0.4
                                                children [
                                                        USE finger_middle
                                                        Transform {
                                                                translation 0.0 1.0 0.0
                                                                # tip rotate
#                                                               rotation 1.0 0.0 0.0  -1.0
                                                                rotation 1.0 0.0 0.0  -0.4
                                                                children USE finger_tip
                                                        }
                                                ]
                                        }
                                ]
                        }
                }
        }

        # Little finger
        DEF finger4 Transform {
                # Finger spread
                rotation 0.0 0.0 1.0  -0.0698
                scale 0.9 0.9 0.9
                children Transform {
                        # Finger twist in
                        rotation 0.0 1.0 0.0  0.1
```

```
                        children Transform {
                                translation 0.0 -0.2 0.0
                                # base rotate
#                               rotation 1.0 0.0 0.0  -1.2
                                rotation 1.0 0.0 0.0  -0.7
                                children [
                                        USE finger_base
                                        Transform {
                                                translation 0.0 1.71 0.0
                                                # middle rotate
#                                               rotation 1.0 0.0 0.0  -1.7
                                                rotation 1.0 0.0 0.0  -0.4
                                                children [
                                                        USE finger_middle
                                                        Transform {
                                                                translation 0.0 1.0 0.0
                                                                # tip rotate
#                                                               rotation 1.0 0.0 0.0  -1.0
                                                                rotation 1.0 0.0 0.0  -0.4
                                                                children USE finger_tip
                                                        }
                                                ]
                                        }
                                ]
                        }
                }
        }

        # Thumb
        DEF thumb Transform {
                # Thumb spread
                rotation 0.0 0.0 1.0  0.7
                children Transform {
                        # Thumb twist
                        rotation 0.0 0.0 1.0  0.1
                        children Transform {
                                # Thumb twist
                                rotation 0.0 1.0 0.0 -0.9
                                scale 0.7 0.7 0.8
                                children [
                                        # Thumb base
                                        Shape {
                                                appearance USE HandColor
                                                geometry Sphere { radius 0.9 }
                                        }
                                        Transform {
                                                translation 0.0 1.1 0.0
                                                children [
                                                        Shape {
                                                                appearance USE HandColor
                                                                geometry Cylinder {
                                                                        height 2.0
                                                                        radius 0.55
                                                                }
                                                        }
                                                        Transform {
                                                                translation 0.0 0.85 0.0
                                                                children [
Shape {
        appearance USE HandColor
        geometry Sphere { radius 0.58 }
```

```
}
Transform {
        rotation 1.0 0.0 0.0  0.08
        children [
                # Thumb middle
                Transform {
                        translation 0.0 0.6 0.0
                        children [
                                Shape {
                                        appearance USE HandColor
                                        geometry Cylinder {
                                                height 1.2
                                                radius 0.41
                                        }
                                }
                                Transform {
                                        translation 0.0 0.6 0.0
                                        children [
                                                Shape {
                                                        appearance USE HandColor
                                                        geometry Sphere { radius 0.45 }
                                                }
                                                Transform {
                                                        rotation 1.0 0.0 0.0  0.13
                                                        children [
                                                                # Thumb tip
Transform {
        rotation 1.0 0.0 0.0 -0.1
        children [
                Transform {
                        translation 0.0 0.5 0.05
                        children [
                                Shape {
                                        appearance USE HandColor
                                        geometry Cylinder {
                                                height 1.0
                                                radius 0.325
                                        }
                                }
                                Transform {
                                        translation 0.0 0.5 0.0
                                        scale 1.0 0.7 1.0
                                        children Shape {
                                                appearance USE HandColor
                                                geometry Sphere { radius 0.29 }
                                        }
                                }
                        ]
                }
        ]
}
                                                        ]
                                                }
                                        ]
                                }
                        ]
                }
        ]
}
                                                                ]
                                                        }
```

```
                                        ]
                                }
                        ]
                }
        }
}

# Palm
DEF palm Group {
        children [
                # Main body of hand
                Transform {
                        translation -0.05 0.875 -0.03
                        children Shape {
                                appearance USE HandColor
                                geometry Box { size 2.85 2.45 0.6 }
                        }
                }



                # Thumb bulge
                Transform {
                        translation -0.9 0.2 -0.2
                        scale 1.1 0.7 0.3
                        children Shape {
                                appearance USE HandColor
                                geometry Sphere { radius 1.1 }
                        }
                }

                # Side of hand bulge
                Transform {
                        translation 0.5 0.0 -0.2
                        scale 0.9 0.8 0.3
                        children Shape {
                                appearance USE HandColor
                                geometry Sphere { radius 1.0 }
                        }
                }

                # Rings, wrist, and arm
                Transform {
                        translation 0.0 -0.6 0.0
                        scale 1.5 1.0 1.0
                        children Transform {
                                rotation 1.0 0.0 0.0 -0.4
                                children [
                                        Shape {
                                                appearance USE HandColor
                                                geometry Cylinder {
                                                        height 0.1
                                                        radius 1.1
                                                }
                                        }
                                        Transform {
                                                translation 0.0 -0.3 0.0
                                                children [
                                                        Shape {
                                                                appearance USE HandColor
                                                                geometry Sphere { radius
```

```
0.8 }
                                                                }
                                                                Shape {
                                                                        appearance USE HandColor
                                                                        geometry Cylinder {
                                                                                height 0.1
                                                                                radius 1.1
                                                                        }
                                                                }
                                                                Transform {
                                                                        translation 0.0 -1.3 0.0
                                                                        children Shape {
                                                                                appearance USE
HandColor

                                                                                geometry Cylinder

{
                                                                                        height 2.0
                                                                                        radius 0.6
                                                                                }
                                                                        }
                                                                }
                                                        ]
                                                }
                                        ]
                                }
                        }


                # Veins on the back of the hand
                        Transform {
                                translation 0.0 -0.475 0.345
                                children [
                                        Transform {
                                                rotation 0.0 0.0 1.0 0.15
                                                children DEF vein Transform {
                                                        translation 0.0 1.375 0.0
                                                        children Shape {
                                                                appearance USE HandColor
                                                                geometry Cylinder {
                                                                        height 2.85
                                                                        radius 0.15
                                                                }
                                                        }
                                                }
                                        }
                                        Transform {
                                                rotation 0.0 0.0 1.0 0.45
                                                children USE vein
                                        }
                                        Transform {
                                                rotation 0.0 0.0 1.0 -0.15
                                                children USE vein
                                        }
                                        Transform {
                                                rotation 0.0 0.0 1.0 -0.45
                                                children USE vein
                                        }
                                        Transform {
                                                rotation 0.0 0.0 1.0 1.15
                                                children Transform {
                                                        translation 0.0 0.875 0.0
```

```
                                                        children Shape {
                                                                appearance USE HandColor
                                                                geometry Cylinder {
                                                                        height 1.75
                                                                        radius 0.15
                                                                }
                                                        }
                                                }
                                        }
                                ]
                        }
                ]
        }


        #
        #  Hand
        #
        DEF hand Group {
                children [
                        USE finger1
                        Transform {
                                translation 0.8 0.0 0.0
                                children [
                                        USE finger2
                                        Transform {
                                                translation 0.8 0.0 0.0
                                                children [
                                                        USE finger3
                                                        Transform {
                                                                translation 0.8 0.0 0.0
                                                                children USE finger4
                                                        }
                                                ]
                                        }
                                ]
                        }
                        Transform {
                                translation -0.4 -2.0 0.0
                                children USE thumb
                        }
                        Transform {
                                translation 1.2 -2.2 0.0
                                children USE palm
                        }
                ]
        }
        ]
}



#
#  The hand!
#
Transform {
        translation 0.0 -2.0 0.0
        rotation 0.0 1.0 0.0  1.0
        children Transform {
                rotation 1.0 0.0 0.0 1.9
                children USE hand
```

```
        }
}


#
#  Primitives in the hand
#
Transform {
        translation 1.5 1.0 0.0
        rotation 1.0 0.0 1.0 0.3
        children Shape {
                appearance Appearance {
                        material Material {
                                diffuseColor 1.0 0.0 0.0
                        }
                }
                geometry Cone { }
        }
}

Transform {
        translation 2.0 3.0 -0.5
        children Shape {
                appearance Appearance {
                        material Material {
                                diffuseColor 1.0 1.0 0.0
                        }
                }
                geometry Sphere { }
        }
}

Transform {
        translation -0.6 0.8 0.5
        rotation 1.0 0.0 0.2 0.82
        children Shape {
                appearance Appearance {
                        material Material {
                                diffuseColor 0.0 0.7 1.0
                        }
                }
                geometry Cylinder { radius 0.75 height 1.5 }
        }
}

Transform {
        translation 0.0 2.6 -1.1
        rotation 1.0 0.0 0.7 1.3
        children Shape {
                appearance Appearance {
                        material Material {
                                diffuseColor 0.0 1.0 0.0
                        }
                }
                geometry Box { size 1.25 1.25 1.25 }
        }
}
```

Introdução à animação

Motivação

- **Nós do tipo `Billboard` e `Anchor` tem comportamento embutido**
- **Você pode criar seus próprios comportamentos para fazer formas se moverem, rotacionarem, escalonarem, piscar e mais**
- **Nós precisamos de um meio para ligar, determinar o tempo e responder para uma seqüência de eventos em ordem, para fornecer melhor interação entre o usuário e o mundo**

Introdução à animação

Construindo circuitos de animação

- **Quase todos os nós podem ser um componente em um *circuito de animação***
  - **Nós agem como partes eletrônicas virtuais**
  - **Nós podem enviar e receber *eventos***
  - **Rotas ligadas conectam nós juntos**
- **Um evento é uma mensagem enviada entre nós**
  - **Um valor de dado (tal como uma translação)**
  - **A time stamp (Quando o evento for disparado)**

Introdução à animação

Exemplos

- **Para girar uma forma:**
  - **Conectar um nó que envia eventos de rotação para o campo `rotation` do nó `Transform`**
- **Para piscar uma forma:**
  - **Conectar um nó que envia um evento de cor para o campo `diffuseColor` do nó `Material`**

Introdução à animação

Roteando eventos

- **Para ativar um circuito de animação, você precisa:**
  - **Um nó que envia eventos**
    - **O nó deve ser nomeado com DEF**
  - **Um nó que recebe eventos**
    - **O nó deve ser nomeado com DEF**
  - **Uma rota conectando eles**

Introdução à animação

Usando entrada e saída de nós

- **Todo nó possui campos, entradas e saídas:**
  - ***field*: um valor armazenado**
  - ***eventIn*: uma entrada**
  - ***eventOut*: uma saída**
- **Um *exposedField* é um short-hand para um *field*, *eventIn*, and *eventOut***

Introdução à animação

Exemplo de entradas

- **Algumas entradas do nó `Transform`:**
  - `set_translation`
  - `set_rotation`
  - `set_scale`
- **Algumas entradas do nó `Material`:**
  - `set_diffuseColor`
  - `set_emissiveColor`
  - `set_transparency`

Introdução à animação

Exemplo de saídas

- **Algumas saídas do nó `TouchSensor`:**
  - `isOver`
  - `isActive`
  - `touchTime`
- **Uma saída do nó `OrientationInterpolator`:**
  - `value_changed`
- **Uma saída do nó `PositionInterpolator`:**
  - `value_changed`

Introdução à animação

Sintaxe: ROUTE

- **Uma expressão `ROUTE` conecta dois nós juntos usando**
. O nome do nó que envia e o nome *eventOut*
. O nome do nó que recebe e o nome *eventIn*

```
ROUTE MySender.rotation_changed
   TO MyReceiver.set_rotation
```

- **Tipos de dados do eventos devem combinar!**

Introdução à animação

Tipos de dados de eventos

SFBool	SFRotation / MFRotation
SFColor / MFColor	SFString / MFString
SFFloat / MFFloat	SFTime
SFImage	SFVec2f / MFVec2f
SFInt32 / MFInt32	SFVec3f / MFVec3f
SFNode / MFNode	

Introdução à animação

Seguindo a convenção dos nomes

- **Muitos nós tem *exposedFields***
- **Se o nome do campo exposto é `xxx`, então:**
  - **`set_xxx` é um *eventIn* para ajustar o campo**
  - **`xxx_changed` é um *eventOut* que envia uma mensagem quando o campo é alterado**
  - **Os sufixos `set_` e `_changed` são opcionais, mas recomendados para melhor entendimento**
- **O nó `Transform` possui:**
  - **Campo `rotation`**
  - **eventIn `set_rotation`**
  - **eventOut `rotation_changed`**

Introdução à animação

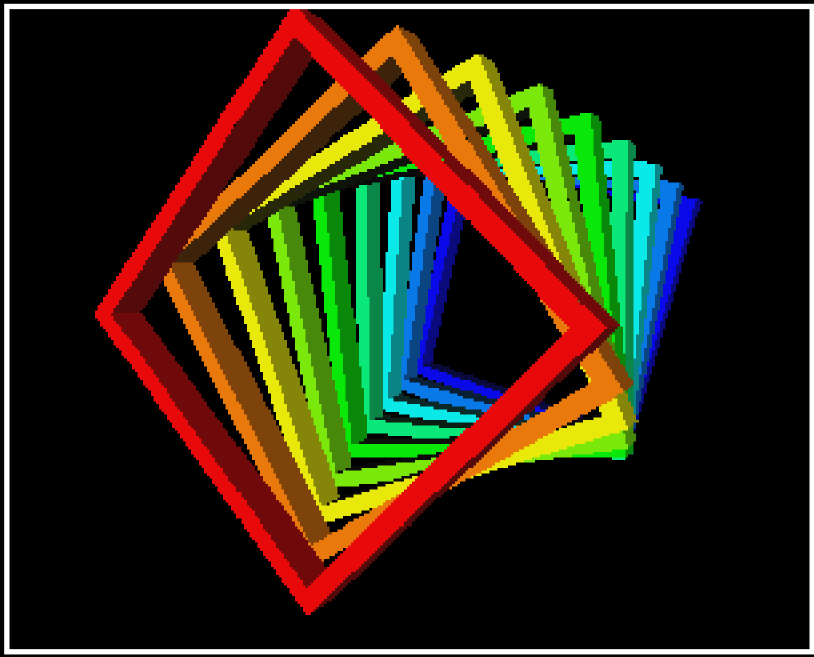
Um exemplo de animação

```
DEF RotateMe Transform {
    rotation 0.0 1.0 0.0 0.0
    children [ . . . ]
}
DEF Rotator OrientationInterpolator { .
. .  }

ROUTE Rotator.value_changed
   TO RotateMe.set_rotation
```

Introdução à animação

Um exemplo de animação



[colors.wrl]

Introdução à animação

Usando múltiplas rotas

- **Você deve ter *fan-out***
  - **Múltiplas rotas saem do mesmo remetente**
- **Você deve ter *fan-in***
  - **Múltipla rotas entram no mesmo destinatário**

Sumário Índice do Capítulo Slide Anterior Slide Seguinte 120

Introdução à animação

Resumo

- **Conecta remetentes e destinatários usando rotas**
- ***eventIns* são entradas, e *eventOuts* são saídas**
- **Uma rota nomeia o *sender.eventOut*, e o *receiver.eventIn***
  - **Tipos de dados devem ser compatíveis**
- **Você pode ter múltiplas rotas entrando ou saindo de um nó**

Animando transformações

Motivação

Exemplo

Controlando o tempo

Usando o tempo absoluto

Usando tempo fracionário

Sintaxe: TimeSensor

Usando temporizadores (timers)

Usando temporizadores (timers)

Usando temporizadores cíclicos

Usando saídas de temporizadores

Um exemplo de sensor de tempo

Um exemplo de sensor de tempo

Convertendo tempo para posição

Interpolando posições

Sintaxe: PositionInterpolator

Usando a entrada e saída do interpolador de pontos

Um exemplo usando interpoladores de posição

Um exemplo usando interpoladores de posição

Usando outros tipos de interpoladores

Sintaxe: OrientationInterpolator

Sintaxe: ColorInterpolator

Sintaxe: ScalarInterpolator

Sintaxe: PositionInterpolator

Um exemplo usando outro interpoladores

Resumo

Resumo

Resumo

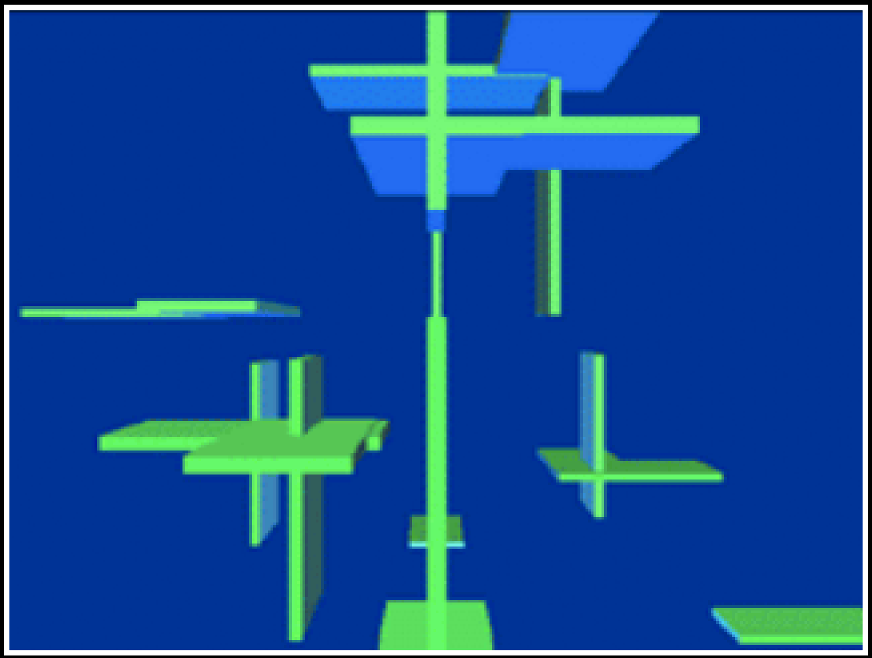
Animando transformações

Motivação

- **Uma *animação* muda algo em função do tempo:**
  - ***position* - uma direção de carro**
  - ***orientation* - um avião voando**
  - ***color* - mudando estações**
- **Animação requer o controle sobre o tempo:**
  - **Quando iniciar e quando parar**
  - **Qual velocidade**

Animando transformações

Exemplo



[floater.wrl]

Animando transformações

Controlando o tempo

- Um nó `TimeSensor` é similar a um relógio parado
- Você controla o tempo inicial e final
- O sensor gera eventos de tempo enquanto está rodando
- Para animar, rotas de eventos de tempo devem entrar em outros nós

Animando transformações

Usando tempo absoluto

- **Um nó `TimeSensor` gera eventos de tempo *absoluto* e *fracionado***
- **Eventos de tempo absoluto fornecem o tempo visto em relógios**

. O tempo absoluto é medido em segundos desde as 12:00am de 1 de Janeiro de 1970!

. Usados para ativar eventos em datas e épocas especificas

Animando transformações

Usando tempo fracionado

- **Eventos de tempo fracionado fornecem um número de 0.0 a 1.0**
  - **Valores variam de 0.0 a 1.0, depois repetem-se**
  - **O número de segundos entre 0.0 e 1.0 é controlado pelo *intervalo do ciclo* (*cycle interval)***
- **O sensor pode entrar em loop infinito, ou ser executado somente uma vez e parar**

Animando transformações

Sintaxe: TimeSensor

. **Um nó `TimeSensor` gera eventos baseados no tempo**

- ***start* and *stop* time - para executar**
- ***cycle interval* time - tamanho do ciclo**
- ***looping* - para repetir ou não**

```
TimeSensor {
    cycleInterval 1.0
    loop FALSE
    startTime 0.0
    stopTime 0.0
}
```

Animando transformações

Usando temporizadores

. **Criando temporizadores contínuos:**

```
loop TRUE
stopTime <= startTime
```

. **Executando um ciclo e parando:**

```
loop FALSE
stopTime <= startTime
```

. **Executando até parar, ou após o ciclo terminar:**

```
loop TRUE or FALSE
stopTime > startTime
```

Animando transformações

Usando temporizadores

- **O evento de entrada `set_startTime`:**
  - **Ativado quando o temporizador é iniciado**
- **The `set_stopTime` input event:**
  - **Ativado quando o temporizador é parado**

Animando transformações

Usando temporizadores cíclicos

- **O primeiro ciclo começa no *start time***
- **O intervalo do ciclo é o comprimento (em segundos) do ciclo**
- **Cada ciclo varia uma fração de 0.0 a 1.0**
- **Se `loop` é FALSE, ocorre somente um ciclo, de outra forma o temporizador entre em loop contínuo**

Animando transformações

Usando saídas de temporizadores

- **O evento de saída `isActive`:**
  - **Emite `TRUE` no tempo inicial**
  - **Emite `FALSE` no tempo final**
- **Um evento de saída `time`:**
  - **Emite o tempo absoluto**
- **O evento de saída `fraction_changed`:**
  - **Emite valores entre 0.0 e 1.0 durante um ciclo**
  - **Retorna para 0.0 no início de cada ciclo**

Animando transformações

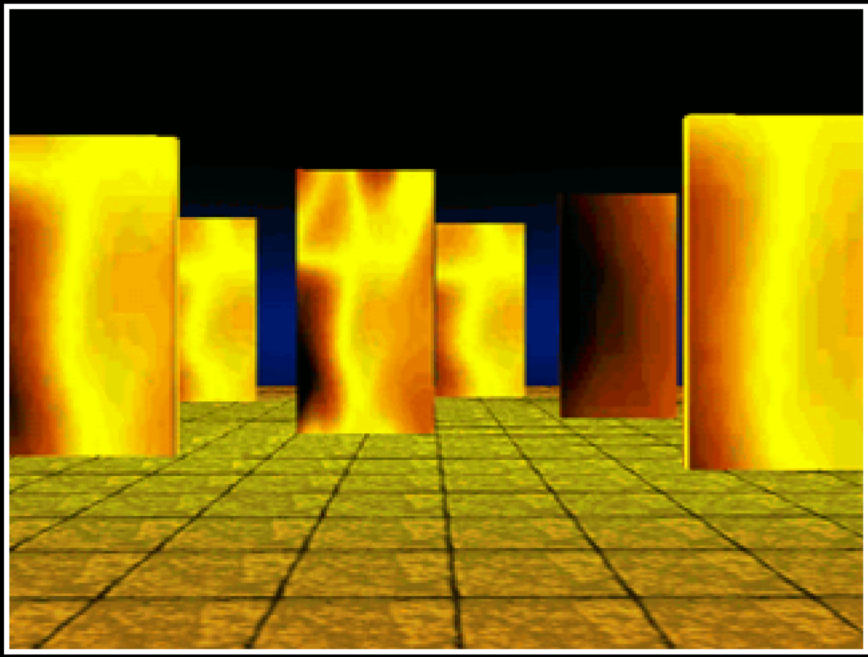
Um exemplo de sensor de tempo

```
DEF Monolith1Timer TimeSensor {
    cycleInterval 4.0
    loop FALSE
    startTime 0.0
    stopTime  1.0
}
ROUTE Monolith1Touch.touchTime
   TO Monolith1Timer.set_startTime

ROUTE Monolith1Timer.fraction_changed
   TO Monolith1Light.set_intensity
```

Animando transformações

Um exemplo de sensor de tempo



[monolith.wrl]

Animando transformações

Convertendo tempo para posição

- **Para animar a posição de uma forma, você deve fornecer:**
  - **Uma lista de posições chaves para um caminho**
  - **Um tempo que deverá estar associado a cada posição**
- **Um nó interpolador que converte uma entrada de tempo para uma saída de posição**
  - **Quando o tempo está entre duas posições chaves, o interpolador computa uma posição intermediária**

Animando transformações

Interpolando posições

- **Cada posição chave ao longo de um caminho possui:**

. Um *valor chave* (tal como uma posição)

. Um tempo fracionário chave

- **Interpolação preenche valores entre seus valores chaves:**

Tempo		Posição
0.0		0.0 0.0 0.0
	0.1	*0.4 0.1 0.0*
	0.2	*0.8 0.2 0.0*
	. . .	. . .
0.5		4.0 1.0 0.0
	. . .	. . .

Animando transformações

Sintaxe: PositionInterpolator

- **Um nó `PositionInterpolator` descreve um caminho de posição**
  - ***keys* - tempo fracionário chave**
  - ***key values* - posições chaves**

```
PositionInterpolator {
    key [ 0.0, . . . ]
    keyValue [ 0.0 0.0 0.0, . . . ]
}
```

- **Route into a `Transform` node's `set_translation` input**

Animando transformações

Usando entradas e saídas de interpoladores de posição

- **A entrada `set_fraction`:**
  - **Ativa o tempo fracionário corrente ao longo do caminho chave**
- **A saída `value_changed`:**
  - **Emite a posição ao longo do caminho cada vez que a fração é ativada**

Animando transformações

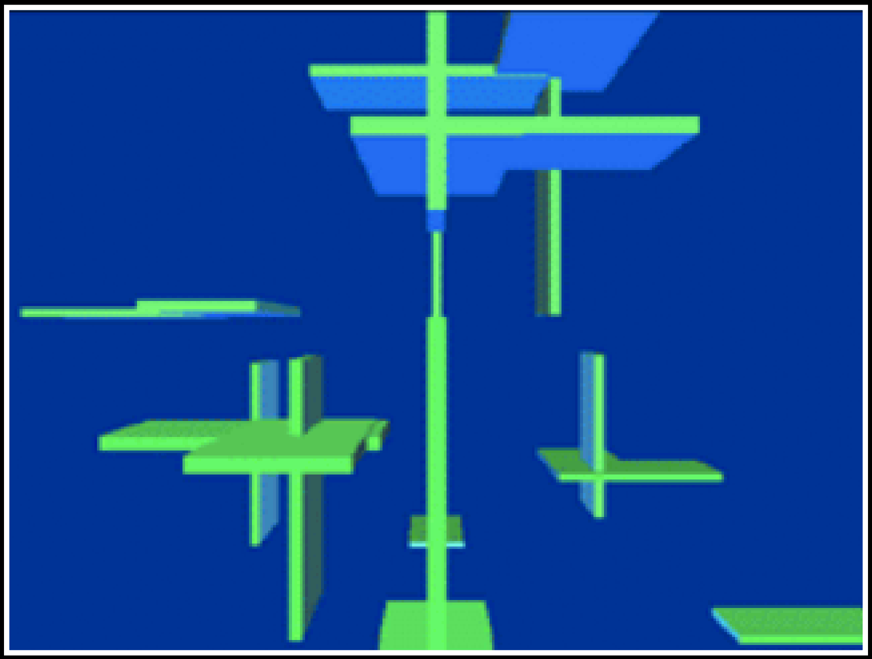
Um exemplo usando interpoladores de posição

```
DEF Mover PositionInterpolator {
    key  [ 0.0, . . . ]
    keyValue [ 0.0 0.0 0.0, . . .]
}
ROUTE Clock.fraction_changed
   TO Mover.set_fraction

ROUTE Mover.value_changed
   TO Movee.set_translation
```

Animando transformações

Um exemplo usando interpoladores de posição



[floater.wrl]

Animando transformações

Usando outros tipos de interpoladores

- **Para animar orientação de formas, usar `OrientationInterpolator`**
- **Para animar cores em formas, usar `ColorInterpolator`**
- **Para animar transparência em formas, usar `ScalarInterpolator`**
- **Para animar escala em formas, usar um truque e `PositionInterpolator`**

Animando transformações

Sintaxe: OrientationInterpolator

- **Um nó `OrientationInterpolator` descreve um caminho orientado**
  - ***keys* - frações de chave**
  - ***key values* - rotações chaves(eixos e ângulo)**

```
OrientationInterpolator {
    key [ 0.0, . . . ]
    keyValue [ 0.0 1.0 0.0 0.0, . . . ]
}
```

- **Route into a `Transform` node's `set_rotation` input**

Animando transformações

Sintaxe: ColorInterpolator

- **`ColorInterpolator` descreve um caminho de cor**
  - ***keys* - frações chave**
  - ***values* - cores chaves (red, green, blue)**

```
ColorInterpolator {
    key [ 0.0, . . . ]
    keyValue [ 1.0 1.0 0.0, . . . ]
}
```

- **Route into a `Material` node's `set_diffuseColor` or `set_emissiveColor` inputs**

Animando transformações

Sintaxe: ScalarInterpolator

- **`ScalarInterpolator` descreve um caminho escalar**
  - ***keys* - frações chaves**
  - ***values* - escalas chaves (usado de forma geral)**

```
ScalarInterpolator {
    key [ 0.0, . . . ]
    keyValue [ 4.5, . . . ]
}
```

- **Route into a `Material` node's `set_transparency` input**

Animando transformações

Sintaxe: PositionInterpolator

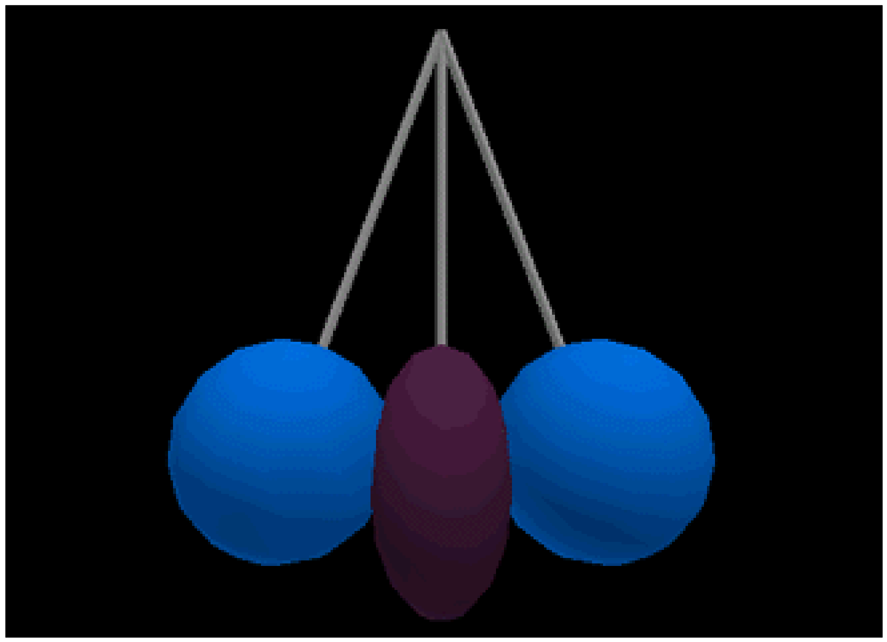
- **Um nó `PositionInterpolator` descreve um caminho de posição ou escala**
  - ***keys* - tempo fracionário chave**
  - ***key values* - posições chaves (ou escalas)**

```
PositionInterpolator {
    key [ 0.0, . . . ]
    keyValue [ 0.0 0.0 0.0, . . . ]
}
```

- **Route into a `Transform` node's `set_scale` input**

Animando transformações

Um exemplo usando outros interpoladores



[squisher.wrl]

Animando transformações

Resumo

- Os campos do nó `TimeSensor` controlam
  - Tempo inicial e final
  - Intervalo do ciclo
  - Quando é um loop ou não
- As saídas do sensor
  - true/false em `isActive` no início e no final
  - tempo absoluto em `time` enquanto é executado
  - tempo fracionário em `fraction_changed` enquanto é executado

Animando transformações

Resumo

. Interpoladores usam tempos chaves e valores, computando um valor intermediário

. Todos os interpoladores possuem:

◦ Uma entrada `set_fraction` para ativar o tempo fracionário

◦ Uma saída `value_changed` para enviar novos valores

Animando transformações

Resumo

. O nó `PositionInterpolator` converte tempo para posição (ou escala)

. O nó `OrientationInterpolator` tempo para rotações

. O nó `ColorInterpolator` converte tempo para cores

. O nó `ScalarInterpolator` converte tempo para escalares (tal como transparências)

```
#VRML V2.0 utf8
#
#  floater.wrl
#  Floating Pads
#       by David R. Nadeau
#
#  A collection of white pads are created.  Each pad
#  slides back and forth horizontally or vertically at its
#  own speed.  A green light illuminates the tops of the
#  pads while a blue light illuminates the bottoms.
#
#  Complexity is created from simplicity via two tricks:
#
#           1.  Create a set of simple shapes, then repeat the
#               same shapes several times.  In this case, there
#               are only 7 original pads moving left and right.
#               Those 7 are repeated again, but turned around the
#               Y axis so that they slide front to back.  The
#               same 7 are repeated again, but rotated around the
#               Z axis so that they slide up and down.  The
#               effect is one of a complex bunch of moving shapes
#               when in reality there are only 7.
#
#           2.  Give each moving shape a slightly different cycle
#               interval for its animation.  This makes all the
#               movement out of sync, which again creates the
#               appearance of complexity.
#
#  A PROTO is used to create a new Pad node that incorporates a
#  box shape, a time sensor, a position interpolator, and routes to
#  connect them together.  Fields to the Pad node control the initial
#  placement of the pad, and it's animation cycle interval.
#
#  Things to experiment with...
#    Extend the PROTO node to include a color field for selecting
#    the color of a pad.
#
#    Add textures.  Particularly try adding a texture with a
#    transparency (alpha) channel.  Using transparency textures
#    you can add stripes, checks, or whatever to the pads.  One
#    approach that looks interesting is to add a cloud texture,
#    but make the sky behind the clouds transparent in the
#    texture.  When mapped, this creates a blobby indistinct
#    shape on each pad.
#
#    Add a Transform node around the entire group of pads, then
#    use an OrientationInterpolator and TimeSensor to slowly
#    spin the world.  Remembering the complexity tricks above,
#    you can create a complex tumble by using three nested
#    Transform nodes (X, Y, and Z), each with its own
#    OrientationInterpolator and TimeSensor.  Give the three
#    TimeSensors slightly different cycleInterval values (such
#    as 5.0, 7.0, and 11.0).  Prime numbers are good choices.
#    Since the cycle times are different for the three spins,
#    they don't sync up, and it looks like a chaotic tumble.
#

WorldInfo {
    title "Floating Pads"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}
```

```
DEF Entry Viewpoint {
    position 0.0 1.6 10.0
    description "Entry View"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight FALSE
}

Background {
    skyColor [ 0.0 0.2 0.6 ]
}

PointLight {
    location 0.0 8.0 10.0
    color 0.3 1.0 0.3
}

PointLight {
    location 0.0 -8.0 -10.0
    color 0.0 0.3 1.0
}


#
#  Define the Pad node type
#
PROTO Pad [
    exposedField SFVec3f    translation    0.0 0.0 0.0
    exposedField SFRotation rotation       0.0 0.0 1.0 0.0
    exposedField SFTime     cycleInterval 10.0
] {
    Transform {
        translation IS translation
        rotation    IS rotation
        children DEF Movee Transform {
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 1.0 1.0 1.0
                        }
                    }
                    geometry Box { size 4.0 0.2 2.0 }
                }
                DEF Clock TimeSensor {
                    cycleInterval IS cycleInterval
                    loop TRUE
                    startTime 1.0
                    stopTime  0.0
                }
                DEF Mover PositionInterpolator {
                    key [ 0.0, 0.5, 1.0 ]
                    keyValue [
                        -10.0 0.0 0.0,
                         10.0 0.0 0.0,
                        -10.0 0.0 0.0,
                    ]
                }
            ]
```

```
        }
    }
    ROUTE Clock.fraction_changed TO Mover.set_fraction
    ROUTE Mover.value_changed    TO Movee.set_translation
}


#
#  Make a bunch of moving pads
#
Pad { translation  0.0  4.0   0.0   cycleInterval 10.0 }
Pad { translation -2.0  0.0  -2.0   cycleInterval 12.0 }
Pad { translation  2.0  6.0  -4.0   cycleInterval 14.0 }
Pad { translation -4.0  2.0  -8.0   cycleInterval 16.0 }
Pad { translation  0.0 -6.0 -10.0   cycleInterval 18.0 }
Pad { translation  4.0 -2.0 -10.0   cycleInterval 10.0 }
Pad { translation  0.0 -4.0  -0.0   cycleInterval  8.0 }

Pad { translation  0.0  4.0   0.0   cycleInterval 10.0   rotation 0.0 1.0 0.0 1.571 }
Pad { translation -2.0  0.0  -2.0   cycleInterval 12.0   rotation 0.0 1.0 0.0 1.571 }
Pad { translation  2.0  6.0  -4.0   cycleInterval 14.0   rotation 0.0 1.0 0.0 1.571 }
Pad { translation -4.0  2.0  -8.0   cycleInterval 16.0   rotation 0.0 1.0 0.0 1.571 }
Pad { translation  0.0 -6.0 -10.0   cycleInterval 18.0   rotation 0.0 1.0 0.0 1.571 }
Pad { translation  4.0 -2.0 -10.0   cycleInterval 10.0   rotation 0.0 1.0 0.0 1.571 }
Pad { translation  0.0 -4.0  -0.0   cycleInterval  8.0   rotation 0.0 1.0 0.0 1.571 }

Pad { translation  0.0  4.0   0.0   cycleInterval 10.0   rotation 0.0 0.0 1.0 1.571 }
Pad { translation -2.0  0.0  -2.0   cycleInterval 12.0   rotation 0.0 0.0 1.0 1.571 }
Pad { translation  2.0  6.0  -4.0   cycleInterval 14.0   rotation 0.0 0.0 1.0 1.571 }
Pad { translation -4.0  2.0  -8.0   cycleInterval 16.0   rotation 0.0 0.0 1.0 1.571 }
Pad { translation  0.0 -6.0 -10.0   cycleInterval 18.0   rotation 0.0 0.0 1.0 1.571 }
Pad { translation  4.0 -2.0 -10.0   cycleInterval 10.0   rotation 0.0 0.0 1.0 1.571 }
Pad { translation  0.0 -4.0  -0.0   cycleInterval  8.0   rotation 0.0 0.0 1.0 1.571 }
```

```
#VRML V2.0 utf8
#
#  Monolith World
#  monolith.wrl
#        by David R. Nadeau
#
#  Six vertical monoliths stand in a world.  Touching a monolith starts
#  it glowing in a pulsing fashion.
#
#  TouchSensors on the monoliths route to TimeSensors, one per monolith.
#  Each TimeSensor routes to the intensity field of a PointLight placed
#  just in front of the monolith.  Simultaneously, a monolith's TimeSensor
#  routes to the transparency field on a Material for a Box surrounding
#  the monolith.
#
#  As a monolith's TimeSensor fraction ramps up from 0.0 to 1.0, the
#  PointLight intensity ramps up from 0.0 (off) to 1.0 (full on).  At
#  the same time, the monolith's outer box transparency ramps from
#  0.0 (opaque) to 1.0 (transparent).  As the transparency increases,
#  an emissive textured inner box for the monolith becomes visible.
#  The inner box's emissive shading makes it appear to be a light source,
#  and the plausible source of the PointLight source's illumination.
#  The overall effect is one of pulsing glowing monoliths.
#
#  A grid mesh floor is used so that localized lighting effects will be
#  obvious.
#
#  Things to experiment with:
#        Try adding a ScalarInterpolator to each monolith and route it's
#        output in to the rotation field of the monolith's TextureTransform.
#        This will rotate the texture on the monolith.
#
#        Instead of the transparent box trick, try using a ColorInterpolator
#        to gradually increase the emissiveColor of a single monolith box
#        while decreasing the box's diffuseColor.
#
#        Try routing the TimeSensor's outputs in to a ScalarInterpolator
#        that varies up and down smoothly, then route this output in to the
#        light source's intensity, etc.  Instead of the 0.1-1.0 ramp, then
#        instant fall to 0.0 that the TimeSensor's fraction output gives, you
#        can shape an arbitrary smooth rise and fall curve in the Scalar
#        Interpolator's key values.
#

WorldInfo {
    title "Monolith World"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 1.6 9.0
    description "Entry View"
}

Background {
    skyColor [
        0.0 0.0 0.0,
        0.0 0.0 0.0,
        0.0 0.1 0.5,
        0.0 0.0 0.0,
    ]
```

```
    skyAngle [
        1.37,
        1.57,
        1.77,
    ]
}


#
#  Generic light source used to give the world ambience before any
#  monoliths start glowing
#
PointLight {
    ambientIntensity 0.1
    color 0.2 0.2 0.2
    location 0.0 8.0 10.0
}


#
#  Glowing monoliths and their light sources
#        The monoliths are all structurally the same, but they have
#        different positions, and slightly different texture transforms
#        This latter difference insures that the texture on each
#        monolith is different.
#
#        A better way to do this would be to create a PROTO for the
#        monolith, then provide an argument to the PROTO node that
#        alters the texture transform.
#
DEF Monolith1Light PointLight {
    intensity 0.0
    ambientIntensity 0.2
    color 0.8 0.4 0.0
    location 0.0 2.0 0.5
}
Transform {
    translation 0.0 2.0 0.0
    children [
        # Outer
        Shape {
            appearance Appearance {
                material DEF Monolith1Facade Material {
                    diffuseColor 0.2 0.2 0.2
                    transparency 0.0
                }
            }
            geometry DEF MonolithOuterBox Box {
                size 2.0 4.0 0.3
            }
        }
        # Inner
        Shape {
            appearance Appearance {
                material NULL # emissive texturing
                texture DEF MonolithInnerFire ImageTexture {
                    url "fire.jpg"
                }
                textureTransform TextureTransform {
                    scale 0.5 0.5
                }
```

```
            }
            geometry DEF MonolithInnerBox Box {
                size 1.9 3.9 0.2
            }
        }
        DEF Monolith1Touch TouchSensor {
        }
    ]
}

DEF Monolith2Light PointLight {
    intensity 0.0
    ambientIntensity 0.2
    color 0.8 0.4 0.0
    location -3.0 2.0 2.5
}
Transform {
    translation -3.0 2.0 2.0
    children [
        # Outer
        Shape {
            appearance Appearance {
                material DEF Monolith2Facade Material {
                    diffuseColor 0.2 0.2 0.2
                    transparency 0.0
                }
            }
            geometry USE MonolithOuterBox
        }
        # Inner
        Shape {
            appearance Appearance {
                material NULL # emissive texturing
                texture USE MonolithInnerFire
                textureTransform TextureTransform {
                    scale 0.3 0.3
                    translation 0.2 0.2
                }
            }
            geometry USE MonolithInnerBox
        }
        DEF Monolith2Touch TouchSensor {
        }
    ]
}

DEF Monolith3Light PointLight {
    intensity 0.0
    ambientIntensity 0.2
    color 0.8 0.4 0.0
    location 4.0 2.0 -1.5
}
Transform {
    translation 4.0 2.0 -2.0
    children [
        # Outer
        Shape {
            appearance Appearance {
                material DEF Monolith3Facade Material {
                    diffuseColor 0.2 0.2 0.2
                    transparency 0.0
```

```
                }
            }
            geometry USE MonolithOuterBox
        }
        # Inner
        Shape {
            appearance Appearance {
                material NULL # emissive texturing
                texture USE MonolithInnerFire
                textureTransform TextureTransform {
                    scale 0.1 0.1
                    translation 0.3 0.5
                }
            }
            geometry USE MonolithInnerBox
        }
        DEF Monolith3Touch TouchSensor {
        }
    ]
}

DEF Monolith4Light PointLight {
    intensity 0.0
    ambientIntensity 0.2
    color 0.8 0.4 0.0
    location 4.5 2.0 3.5
}
Transform {
    translation 4.5 2.0 3.0
    children [
        # Outer
        Shape {
            appearance Appearance {
                material DEF Monolith4Facade Material {
                    diffuseColor 0.2 0.2 0.2
                    transparency 0.0
                }
            }
            geometry USE MonolithOuterBox
        }
        # Inner
        Shape {
            appearance Appearance {
                material NULL # emissive texturing
                texture USE MonolithInnerFire
                textureTransform TextureTransform {
                    scale 0.2 0.2
                    translation 0.4 0.6
                }
            }
            geometry USE MonolithInnerBox
        }
        DEF Monolith4Touch TouchSensor {
        }
    ]
}

DEF Monolith5Light PointLight {
    intensity 0.0
    ambientIntensity 0.2
    color 0.8 0.4 0.0
```

```
    location -3.5 2.0 -4.5
}
Transform {
    translation -3.5 2.0 -5.0
    children [
        # Outer
        Shape {
            appearance Appearance {
                material DEF Monolith5Facade Material {
                    diffuseColor 0.2 0.2 0.2
                    transparency 0.0
                }
            }
            geometry USE MonolithOuterBox
        }
        # Inner
        Shape {
            appearance Appearance {
                material NULL # emissive texturing
                texture USE MonolithInnerFire
                textureTransform TextureTransform {
                    scale 0.2 0.4
                    translation 0.5 0.1
                }
            }
            geometry USE MonolithInnerBox
        }
        DEF Monolith5Touch TouchSensor {
        }
    ]
}

DEF Monolith6Light PointLight {
    intensity 0.0
    ambientIntensity 0.2
    color 0.8 0.4 0.0
    location 2.0 2.0 -5.5
}
Transform {
    translation 2.0 2.0 -6.0
    children [
        # Outer
        Shape {
            appearance Appearance {
                material DEF Monolith6Facade Material {
                    diffuseColor 0.2 0.2 0.2
                    transparency 0.0
                }
            }
            geometry USE MonolithOuterBox
        }
        # Inner
        Shape {
            appearance Appearance {
                material NULL # emissive texturing
                texture USE MonolithInnerFire
                textureTransform TextureTransform {
                    scale 0.2 0.4
                    translation 0.5 0.1
                }
            }
```

```
            geometry USE MonolithInnerBox
        }
        DEF Monolith6Touch TouchSensor {
        }
    ]
}


#
#  Floor mesh
#
Inline {
    url "floor.wrl"
}


#
#  Animation control
#
#  For each monolith, route its touch sensor in to the timer to start it.
#  Route the timer's fraction output in to the facade box's transparency
#  level and the light's intensity level.
#
#  Use different cycle intervals for each time sensor so that they
#  monoliths pulse in an apparently random, out of sync manner.
#
DEF Monolith1Timer TimeSensor {
    cycleInterval 4.0
    loop FALSE
    startTime 0.0
    stopTime 1.0
}
ROUTE Monolith1Touch.touchTime        TO Monolith1Timer.set_startTime
ROUTE Monolith1Timer.fraction_changed TO Monolith1Facade.set_transparency
ROUTE Monolith1Timer.fraction_changed TO Monolith1Light.set_intensity


DEF Monolith2Timer TimeSensor {
    cycleInterval 4.5
    loop FALSE
    startTime 0.0
    stopTime 1.0
}
ROUTE Monolith2Touch.touchTime        TO Monolith2Timer.set_startTime
ROUTE Monolith2Timer.fraction_changed TO Monolith2Facade.set_transparency
ROUTE Monolith2Timer.fraction_changed TO Monolith2Light.set_intensity


DEF Monolith3Timer TimeSensor {
    cycleInterval 5.0
    loop FALSE
    startTime 0.0
    stopTime 1.0
}
ROUTE Monolith3Touch.touchTime        TO Monolith3Timer.set_startTime
ROUTE Monolith3Timer.fraction_changed TO Monolith3Facade.set_transparency
ROUTE Monolith3Timer.fraction_changed TO Monolith3Light.set_intensity


DEF Monolith4Timer TimeSensor {
    cycleInterval 5.5
    loop FALSE
```

```
    startTime 0.0
    stopTime 1.0
}
ROUTE Monolith4Touch.touchTime        TO Monolith4Timer.set_startTime
ROUTE Monolith4Timer.fraction_changed TO Monolith4Facade.set_transparency
ROUTE Monolith4Timer.fraction_changed TO Monolith4Light.set_intensity


DEF Monolith5Timer TimeSensor {
    cycleInterval 6.0
    loop FALSE
    startTime 0.0
    stopTime 1.0
}
ROUTE Monolith5Touch.touchTime        TO Monolith5Timer.set_startTime
ROUTE Monolith5Timer.fraction_changed TO Monolith5Facade.set_transparency
ROUTE Monolith5Timer.fraction_changed TO Monolith5Light.set_intensity


DEF Monolith6Timer TimeSensor {
    cycleInterval 6.5
    loop FALSE
    startTime 0.0
    stopTime 1.0
}
ROUTE Monolith6Touch.touchTime        TO Monolith6Timer.set_startTime
ROUTE Monolith6Timer.fraction_changed TO Monolith6Facade.set_transparency
ROUTE Monolith6Timer.fraction_changed TO Monolith6Light.set_intensity
```

```
#VRML V2.0 utf8
#
#  squisher.wrl
#       by David R. Nadeau
#
#  This world illustrates the use of orientation, color, and position
#  interpolators.  The world contains left, center, and right balls.
#  The left and right balls swing back and forth, pivoting about a
#  common point above the center ball.  Each time the left and right
#  balls swing inward, striking the center ball, the center ball squishes
#  and turns red.
#
#  To make this effect, two orientation interpolators swing the left
#  and right balls back and forth.  A position interpolator's output
#  is used as a scale value to squish the center ball.  Finally, a
#  color interpolator repeatedly turns the center ball red, then blue.

WorldInfo {
    title "Squishy ball"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 1.6 10.0
    description "Entry View"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight TRUE
}

Background {
    skyColor [ 0.0 0.0 0.0 ]
}


#
# Center ball
#
Transform {
        center 0.0 4.0 0.0
        children [
                DEF Squishee Transform {
                        children Shape {
                                appearance Appearance {
                                        material DEF Coloree Material { }
                                }
                                geometry Sphere { }
                        }
                }
                DEF Stick Transform {
                        translation 0.0 2.0 0.0
                        children Shape {
                                appearance Appearance {
                                        material Material { }
                                }
                                geometry Cylinder {
                                        height 4.0
                                        radius 0.1
                                }
                        }
```

```
                }
        ]
}

#
#  Left and right balls
#
DEF Swingee1 Transform {
        center 0.0 4.0 0.0
        children [
                DEF Ball Shape {
                        appearance Appearance {
                                material Material {
                                        diffuseColor 0.0 0.5 1.0
                                }
                        }
                        geometry Sphere { }
                }
                USE Stick
        ]
}
DEF Swingee2 Transform {
        center 0.0 4.0 0.0
        children [
                USE Ball
                USE Stick
        ]
}


#
#  Animation
#
DEF Clock TimeSensor {
        cycleInterval 2.0
        loop TRUE
        startTime 1.0
        stopTime 0.0
}

DEF Swinger1 OrientationInterpolator {
        key [ 0.0, 0.5, 1.0 ]
        keyValue [
                0.0 0.0 1.0 -0.5,
                0.0 0.0 1.0 -0.2,
                0.0 0.0 1.0 -0.5,
        ]
}

ROUTE Clock.fraction_changed TO Swinger1.set_fraction
ROUTE Swinger1.value_changed TO Swingee1.set_rotation

DEF Swinger2 OrientationInterpolator {
        key [ 0.0, 0.5, 1.0 ]
        keyValue [
                0.0 0.0 1.0  0.5,
                0.0 0.0 1.0  0.2,
                0.0 0.0 1.0  0.5,
        ]
}

ROUTE Clock.fraction_changed TO Swinger2.set_fraction
```

```
ROUTE Swinger2.value_changed TO Swingee2.set_rotation

DEF Squisher PositionInterpolator {
        key [ 0.0, 0.5, 1.0 ]
        keyValue [
                1.0 1.0 1.0,
                0.5 1.4 1.4,
                1.0 1.0 1.0,
        ]
}

ROUTE Clock.fraction_changed TO Squisher.set_fraction
ROUTE Squisher.value_changed TO Squishee.set_scale

DEF Colorer ColorInterpolator {
        key [ 0.0, 0.5, 1.0 ]
        keyValue [
                0.0 0.5 1.0,
                1.0 0.0 0.0,
                0.0 0.5 1.0,
        ]
}

ROUTE Clock.fraction_changed TO Colorer.set_fraction
ROUTE Colorer.value_changed  TO Coloree.set_diffuseColor
```

Sentido ações do observador

Motivação

Usando sensores de ação

Sentindo formas

Sintaxe: TouchSensor

Um exemplo de uso do nó TouchSensor

Sintaxe: SphereSensor

Sintaxe: CylinderSensor

Sintaxe: PlaneSensor

Usando múltiplos sensores

Um exemplo do uso de múltiplos sensores

Sumário

Sentindo ações do observador

Motivação

- **Você pode sentir quando o cursor do observador:**
  - **Está *sobre* uma forma**
  - ***Tocou* uma forma**
  - **Está *arrastando* uma forma**
- **Você pode ativar animações no toque do observador**
- **Você pode abilitar o observador a mover e rotacionar formas**

Sentindo ações do observador

Usando sensores de ação

. Existem quatro tipos principais de sensores de ação:

- `TouchSensor`: **sensor de toque**
- `SphereSensor`: **sensor de arrasto**
- `CylinderSensor`: **sensor de arrasto**
- `PlaneSensor`: **sensor de arrasto**

. O nó `Anchor` é um sensor de ação de propósito especial com uma resposta embutida

Sentindo ações do observador

Sentindo formas

. Todos os sensores de ação sentem todas as formas no mesmo grupo

. Sensores são ativados quando o cursor do observador toca uma forma sensível

Sentindo ações do observador

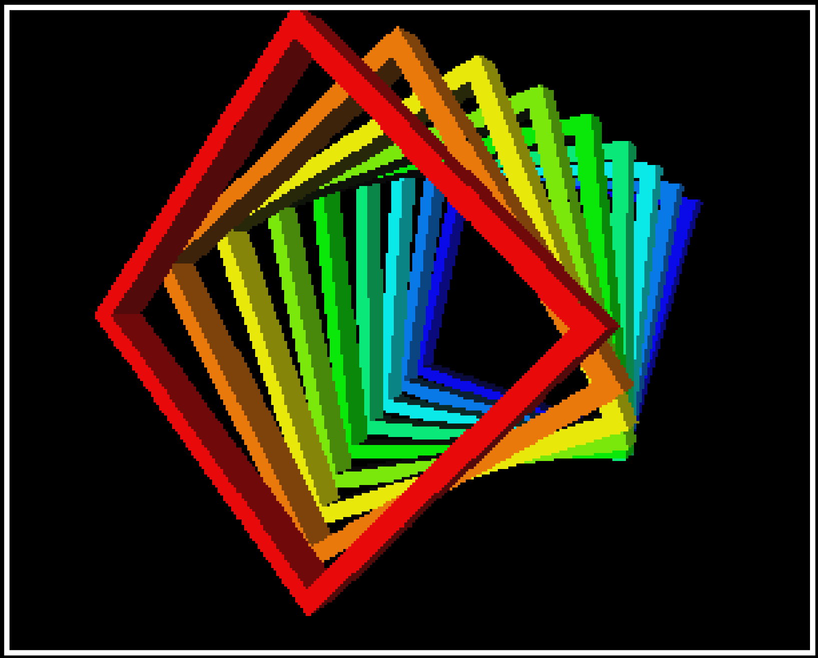
Sintaxe: TouchSensor

- **O nó TouchSensor sente o *toque* do cursor**
  - ***isOver* - envia true/false quando o cursor está/não está sobre a forma**
  - ***isActive* - envia true/false quando o butão do mouse está pressionado/liberado**
  - ***touchTime* - envia o tempo quando o botão do mouse está liberado**

```
Transform {
    children [
        . . .
        DEF Touched TouchSensor { }
    ]
}
```

Sentindo ações do observador

Um exemplo de uso do nó TouchSensor



[colors.wrl]

Sentindo ações do observador

Sintaxe: SphereSensor

. Um nó SphereSensor sente um arrasto do cursor e gera rotações como a rotação de uma bola

- ***isActive*** **- envia true/false quando o botão do mouse está pressionado/liberado**
- ***rotation_changed*** **- envia rotação durante durante o arrasto**

```
Transform {
   children [
      DEF RotateMe Transform { . . . }
      DEF Rotator  SphereSensor { }
   ]
}
ROUTE Rotator.rotation_changed
  TO RotateMe.set_rotation
```

Sentindo ações do observador

Sintaxe: CylinderSensor

. Um nó CylinderSensor sente um cursor arrastar e gera rotações como a rotação de um cilindro

- ***isActive*** **- envia true/false quando o botão do mouse está pressionado/liberado**
- ***rotation_changed*** **- envia a rotação durante o arrasto**

```
Transform {
    children [
        DEF RotateMe Transform { . . . }
        DEF Rotator  CylinderSensor { }
    ]
}
ROUTE Rotator.rotation_changed
   TO RotateMe.set_rotation
```

Sentindo ações do observador

Sintaxe: PlaneSensor

. Um nó PlaneSensor sente o cursor arrastar e gera translações como deslizamentos em um plano

- ***isActive*** **- envia true/false quando o botão do mouse está pressionado/liberado**
- ***translation_changed*** **- envia as translações durante o arrasto**

```
Transform {
    children [
        DEF MoveMe Transform { . . . }
        DEF Mover  PlaneSensor { }
    ]
}
ROUTE Mover.translation_changed
   TO MoveMe.set_translation
```

Sentindo ações do observador

Usando múltiplos sensores

. Múltplos sensores podem sentir a mesma forma mas...

○ Se os sensores estão no mesmo grupo:

. Todos respondem

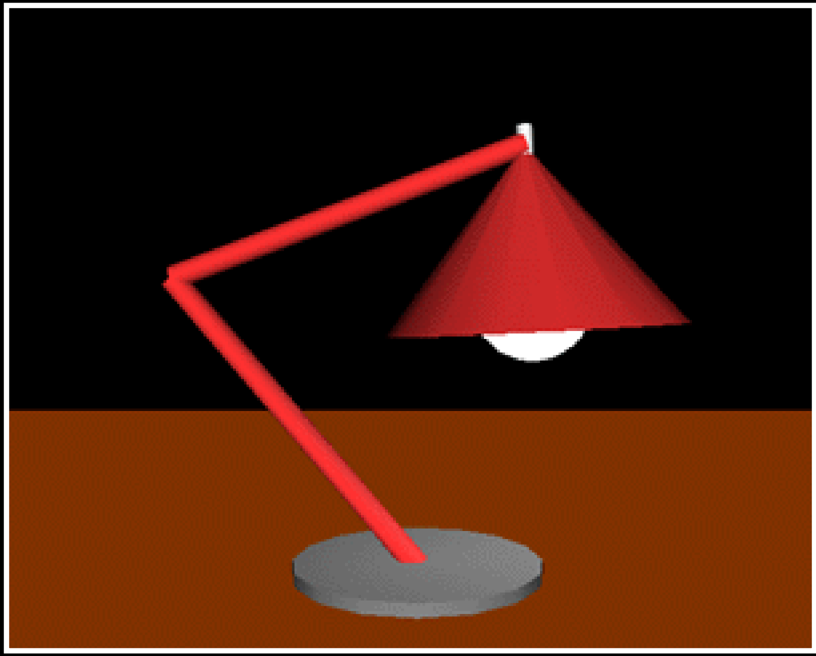
○ Se os sensores estão em diferentes pronfundidades na hierarquia:

. O sensor mais profundo responde

. Os outros sensores não respondem

Sentindo ações do observador

Um exemplo do uso de múltiplos sensores



[lamp.wrl]

Sentindo ações do observador

Resumo

- **Sensores de ação sentem quando o cursor do observador:**
  - **Está sobre uma forma**
  - **Toca uma forma**
  - **Arrasta uma forma**
- **Sensores convertem ações do observador em eventos para:**
  - **Começar e parar animações**
  - **Orientar formas**
  - **Posicionar formas**

```
#VRML V2.0 utf8
#
#  lamp.wrl
#  Desk lamp
#       by David R. Nadeau
#
#  This file builds a desk lamp that can be moved about.  Dragging
#  on the lamp base slides the lamp back and forth.  Dragging on the
#  lamp arms turns the lamp arm.  Dragging on the lamp shade turns
#  the lamp shade.
#

Transform {
  translation 0.0 -1.6 5.0
  scale 8.0 8.0 8.0
  children [

#
#  Table
#
Shape {
  appearance Appearance {
    material Material {
      emissiveColor 0.5 0.2 0.0
      diffuseColor 0.0 0.0 0.0
    }
  }
  geometry Box { size 2.5 0.001 2.5 }
}


Transform {
  translation 0.0 0.01 0.0
  rotation 1.0 0.0 0.0 1.571
  children [
  # Lamp
    DEF MoveLamp PlaneSensor { }
    DEF Lamp Transform {
      rotation 1.0 0.0 0.0 -1.571
      children [
      # Lamp base
        Shape {
          appearance Appearance {
            material Material { diffuseColor 0.5 0.5 0.5 }
          }
          geometry Cylinder {
            radius 0.1
            height 0.01
          }
        }
      # Base - First arm joint
        Group {
          children [
            DEF MoveFirstArm CylinderSensor { }
            DEF FirstArm Transform {
              children [
Transform {
translation 0.0 0.15 0.0
rotation    1.0 0.0 0.0  -0.7
center      0.0 -0.15 0.0
children [
```

```
              # Lower arm
                DEF LampArm Shape {
                  appearance DEF Red Appearance {
                    material Material { diffuseColor 1.0 0.2 0.2 }
                  }
                  geometry Cylinder {
                    radius 0.01
                    height 0.3
                  }
                }
              # First arm - second arm joint
                Group {
                  children [
                    DEF MoveSecondArm CylinderSensor { }
                    DEF SecondArm Transform {
                      children [
Transform {
translation 0.0 0.3 0.0
rotation  1.0 0.0 0.0  1.9
center    0.0 -0.15 0.0
children [
                      # Second arm
                        USE LampArm,
                      # Second arm - shade joint
                        Group {
                          children [
                            DEF MoveLampShade SphereSensor {
                              offset 1.0 0.0 0.0 -1.25
                            }
                            DEF LampShade Transform {
                              translation 0.0 0.075 0.0
                              rotation  1.0 0.0 0.0  -1.25
                              center    0.0 0.075 0.0
                              children [
                              # Shade
                                Shape {
                                  appearance USE Red
                                  geometry Cone {
                                    height 0.15
                                    bottomRadius 0.12
                                    bottom FALSE
                                  }
                                }
                              # Switch
                                Transform {
                                  translation 0.0 0.075 0.0
                                  children [
                                    Shape {
                                      appearance Appearance {
                                        material Material {
                                          diffuseColor 1.0 1.0 1.0
                                        }
                                      }
                                      geometry Cylinder {
                                        radius 0.007
                                        height 0.03
                                      }
                                    }
                                  ]
                                }
                              # Light bulb
```

```
                                Transform {
                                  translation 0.0 -0.05 0.0
                                  children [
                                    DEF LampLight SpotLight {
                                      ambientIntensity 0.2
                                      intensity 1.0
                                      direction 0.0 -1.0 0.0
                                      location 0.0 0.0 0.0
                                      beamWidth 1.571
                                      cutOffAngle 1.571
                                    }
                                    Shape {
                                      appearance Appearance {
                                        material DEF BulbMaterial Material {
                                           diffuseColor 0.0 0.0 0.0
                                           emissiveColor 1.0 1.0 1.0
                                        }
                                      }
                                      geometry Sphere {
                                        radius 0.05
                                      }
                                    }
                                  ]
                                }
                              ]
                            }
                          ]
                        }
] }
                      ]
                    }
                  ]
                }
] }
              ]
            }
          ]
        }
      ]
    }
  ]
}

  ]
}
ROUTE MoveLamp.translation_changed   TO Lamp.set_translation
ROUTE MoveFirstArm.rotation_changed  TO FirstArm.set_rotation
ROUTE MoveSecondArm.rotation_changed TO SecondArm.set_rotation
ROUTE MoveLampShade.rotation_changed TO LampShade.set_rotation
```

Construindo formas fora de pontos, linhas e superfície

Motivação

Exemplo

Construindo formas usando coordenadas

Sintaxe: Coordinate

Usando geometria de coordenadas

Sintaxe: PointSet

Um exemplo de forma com PointSet

Sintaxe: IndexedLineSet

Usando índices de coordenadas de conjunto de linhas

Usando listas de índices de coordenadas de conjunto de linhas

Um exemplo de forma com IndexedLineSet

Sintaxe: IndexedFaceSet

Usando listas de índices de coordenadas de conjunto de superfícies

Um exemplo de forma com IndexedFaceSet

Sintaxe: CoordinateInterpolator

Resumo

Resumo

Resumo

Construindo formas com pontos, linhas e superfícies

Motivação

- **Formas complexas são difíceis de construir com formas primitivas**
  - **Terreno**
  - **Animais**
  - **Plantas**
  - **Maquinários**
- **Em lugar, construir formas com componentes atômicos:**
  - **Pontos, linhas, e superfícies**

Construindo formas com pontos, linhas e superfícies

Exemplo

Construindo formas com pontos, linhas e superfícies

Construindo formas usando coordenadas

- **Construir formas é como ligar pontos em 3D:**
  - **Colocar os pontos nas posições 3D**
  - **Conectar os pontos para construir a forma**
- **Uma coordenadas determina a localização de um ponto em 3D**
  - **Medida em relação a origem do sistema de coordenadas**
- **A geometria do nó indica como conectar os pontos**

Construindo formas com pontos, linhas e superfícies

Sintaxe: Coordinate

. Um nó `Coordinate` contém uma lista de coordenadas para ser usada na cosntrução de uma forma

```
Coordinate {
    point [
#       X   Y   Z
        2.0 1.0 3.0,
        4.0 2.5 5.3,
        . . .
    ]
}
```

Construindo formas com pontos, linhas e superfícies

Usando geometria de coordenadas

- **Construir formas usando nós geométricos:**
  - **`PointSet`**
  - **`IndexedLineSet`**
  - **`IndexedFaceSet`**
- **Para todos os três nós, usar um nó `Coordinate` com o valor do campo `coord`**

Construindo formas com pontos, linhas e superfícies

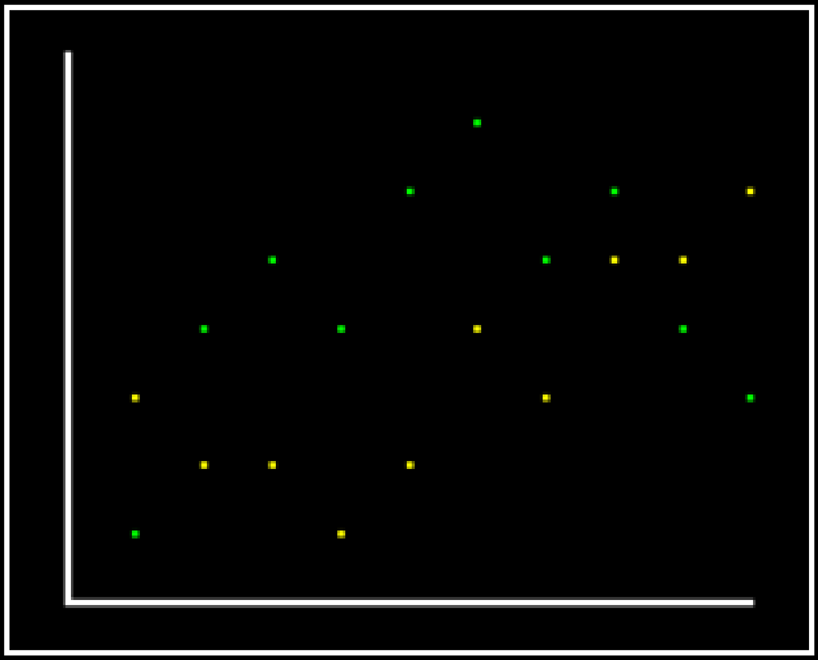
Sintaxe: PointSet

- **Um nó geométrico `PointSet` cria geometria dentre pontos**
  - **Um ponto é posicionado em cada coordenada**

```
PointSet {
    coord Coordinate {
        point [ . . . ]
    }
}
```

Construindo formas com pontos, linhas e superfícies

Um exemplo de forma com PointSet



[ptplot.wrl]

Construindo formas com pontos, linhas e superfícies

Sintaxe: IndexedLineSet

- Um nó geométrico `IndexedLineSet` cria geometria dentre linhas
  - Uma linha reta é desenhada entre os pares de coordenadas selecionadas

```
IndexedLineSet {
    coord Coordinate {
        point [ . . . ]
    }
    coordIndex [ . . . ]
}
```

Construindo formas com pontos, linhas e superfícies

Usando índices de coordenadas de conjunto de linhas

- **Cada coordenada em um nó `Coordinate` é implicitamente numerada**
  - **Índice *0* é a primeira coordenada**
  - **Índice *1* é a segunda coordenada, etc.**
- **Para construir formas com linhas**
  - **Fazer uma lista de coordenada, usando seus índices**
  - **Usar o nó `IndexedLineSet` para desenhar uma linha de uma coordenada para outra em uma lista**

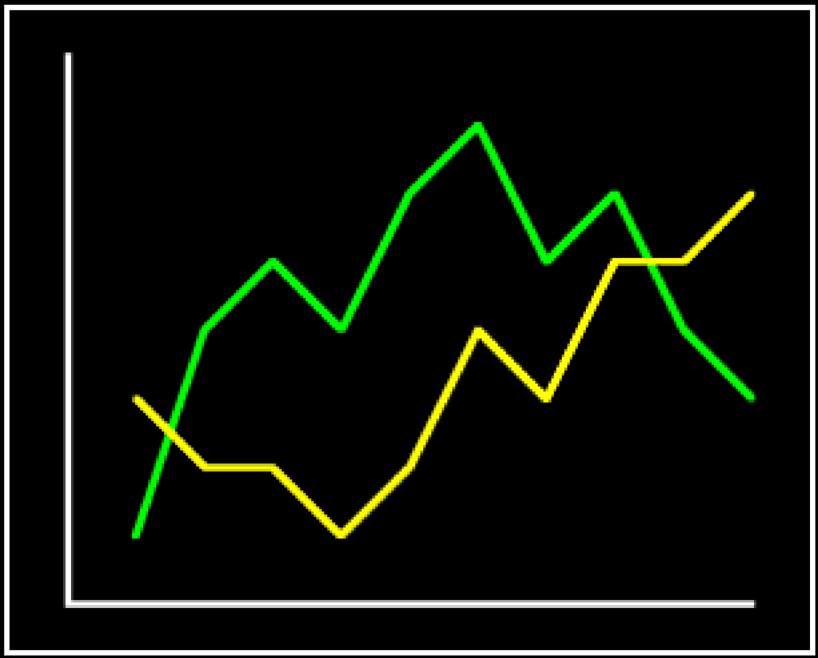
Construindo formas com pontos, linhas e superfícies

Usando lista de índices de coordenadas de conunto de linhas

- `1, 0, 3, -1, ...`
  - `1, 0,` **Desenha de 1 para 0**
  - `0, 3,` **Desenha de 0 para 3**
  - `-1,` **Finaliza a senqüência**
- **Lista de índices de coordenadas no campo `coordIndex` do nó `IndexedLineSet`**

Construindo formas dentre pontos, linhas e superfícies

Um exemplo de forma com IndexedLineSet



[lnplot.wrl]

Construindo formas dentre pontos, linhas e superfícies

Sintaxe: IndexedFaceSet

- **Um nó geométrico IndexedFaceSet cria geometria dentre *superfícies***
  - **Uma superfície lisa (polígono) é desenhada usando um esboço especificado pelas coordenada**

```
IndexedFaceSet {
    coord Coordinate {
        point [  . . .  ]
    }
    coordIndex [ . . . ]
}
```

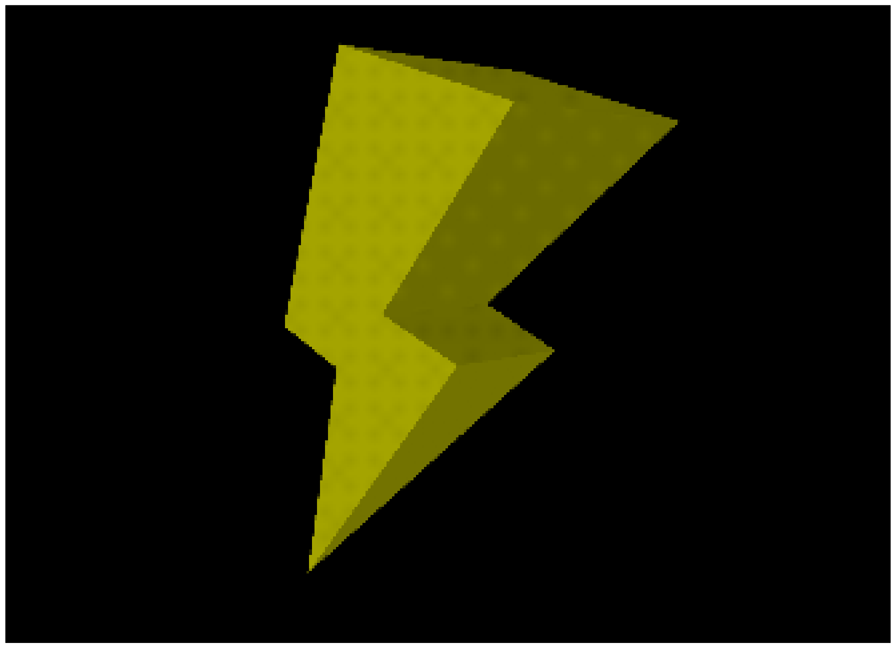
Construindo formas dentre pontos, linhas e superfícies

Usando lista de índices de coordenadas de conjunto de superfície

- **Para construir uma forma com superfícies**
  - **Fazer uma lista de coordenadas, usando seus índices**
  - **Usando um nó `IndexedFaceSet` para desenhar uma superfície esquematizada pelas coordenadas na lista**
- **Lista de índices de coordenadas no campo `coordIndex` do nó `IndexedFaceSet`**

Construindo formas dentre pontos, linhas e superfícies

Um exemplo de forma com IndexedFaceSet



[lightng.wrl]

Construindo formas dentre pontos, linhas e superfícies

Sintaxe: CoordinateInterpolator

- **Um nó `CoordinateInterpolator` descreve um caminho de coordenadas**
  - ***keys* - frações chave**
  - ***values* - lista de coordenadas chave(listas de X,Y,Z)**

```
CoordinateInterpolator {
    key [ 0.0, . . . ]
    keyValue [ 0.0 1.0 0.0, . . . ]
}
```

Construindo formas dentre pontos, linhas e superfícies

Resumo

. Formas são construídas através da conecção de coordenadas

. Coordenadas são listadas em um nó `Coordinate`

. Coordenadas são implicitamente numeradas, começanco em 0

. Lista de índices de coordenadas fornecem a ordem do uso das coordenadas

Construindo formas dentre pontos, linhas e superfícies

Resumo

- **O nó `PointSet` desenha um ponto em todas as coordenadas**
  - **O valor do campo `coord` é um nó `Coordinate`**
- **O nó `IndexedLineSet` desenha linhas entre coordenadas**
  - **O valor do campo `coord` é um nó `Coordinate`**
  - **O valor do campo `coordIndex` é uma lista de índices de coordenadas**

Construindo formas dentre pontos, linhas e superfícies

Resumo

- **O nó `IndexedFaceSet` desenha superfícies esquematizada por coordenadas**
  - **O valor do campo `coord` é um nó `Coordinate`**
  - **O valor do campo `coordIndex` é uma lista de índices de coordenadas**
- **O nó `CoordinateInterpolator` converte tempo para coordenadas**

```
#VRML V2.0 utf8
Group {
    children [
        # Axes
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 1.0 1.0 1.0
                    emissiveColor 1.0 1.0 1.0
                }
            }
            geometry IndexedLineSet {
                coord Coordinate {
                    point [
                        0.0 0.0 0.0, 10.0 0.0 0.0, 0.0 8.0 0.0
                    ]
                }
                coordIndex [
                    0, 1, -1,  0, 2, -1
                ]
            }
        }

        # Scatter plot with different color points
        Shape {
            appearance Appearance {
            }
            geometry PointSet {
                coord Coordinate {
                    point [
                        # Green points
                        1.0 1.0 0.0,   2.0 4.0 0.0,
                        3.0 5.0 0.0,   4.0 4.0 0.0,
                        5.0 6.0 0.0,   6.0 7.0 0.0,
                        7.0 5.0 0.0,   8.0 6.0 0.0,
                        9.0 4.0 0.0,  10.0 3.0 0.0,
                        # Yellow points
                        1.0 3.0 0.0,   2.0 2.0 0.0,
                        3.0 2.0 0.0,   4.0 1.0 0.0,
                        5.0 2.0 0.0,   6.0 4.0 0.0,
                        7.0 3.0 0.0,   8.0 5.0 0.0,
                        9.0 5.0 0.0,  10.0 6.0 0.0,
                    ]
                }
                color Color {
                    color [
                        # Green points
                        0.0 1.0 0.0,   0.0 1.0 0.0,
                        0.0 1.0 0.0,   0.0 1.0 0.0,
                        0.0 1.0 0.0,   0.0 1.0 0.0,
                        0.0 1.0 0.0,   0.0 1.0 0.0,
                        0.0 1.0 0.0,   0.0 1.0 0.0,
                        # Yellow points
                        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,
                    ]
                }
            }
```

```
            }
        ]
}
```

```
#VRML V2.0 utf8
Group {
    children [
        # Axes
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 1.0 1.0 1.0
                    emissiveColor 1.0 1.0 1.0
                }
            }
            geometry IndexedLineSet {
                coord Coordinate {
                    point [
                        0.0 0.0 0.0, 10.0 0.0 0.0, 0.0 8.0 0.0
                    ]
                }
                coordIndex [
                    0, 1, -1,  0, 2, -1
                ]
            }
        }

        # Line plot with different color lines
        Shape {
            appearance Appearance {
            }
            geometry IndexedLineSet {
                coord Coordinate {
                    point [
                        # Green line
                        1.0 1.0 0.0,   2.0 4.0 0.0,
                        3.0 5.0 0.0,   4.0 4.0 0.0,
                        5.0 6.0 0.0,   6.0 7.0 0.0,
                        7.0 5.0 0.0,   8.0 6.0 0.0,
                        9.0 4.0 0.0,  10.0 3.0 0.0,
                        # Yellow line
                        1.0 3.0 0.0,   2.0 2.0 0.0,
                        3.0 2.0 0.0,   4.0 1.0 0.0,
                        5.0 2.0 0.0,   6.0 4.0 0.0,
                        7.0 3.0 0.0,   8.0 5.0 0.0,
                        9.0 5.0 0.0,  10.0 6.0 0.0,
                    ]
                }
                color Color {
                    color [
                        0.0 1.0 0.0,   1.0 1.0 0.0,
                    ]
                }
                coordIndex [
                    # Green line
                    0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -1,
                    # Yellow line
                    10, 11, 12, 13, 14, 15, 16, 17, 18, 19, -1
                ]
                colorIndex [
                    0, 1
                ]
                colorPerVertex FALSE
            }
        }
```

```
    ]
}
```

```
#VRML V2.0 utf8
Shape {
	appearance Appearance {
		material Material {
			diffuseColor 1.0 1.0 0.0
		}
	}
	geometry IndexedFaceSet {
		coord Coordinate {
			point [
			# Lighting bolt tip
				0.0  0.0  0.0,
			# Front perimeter
				5.5  5.0  0.88,
				4.0  5.5  0.968,
				7.0  8.0  1.408,
				4.0  9.0  1.584,
				1.0  5.0  0.88,
				2.5  4.5  0.792,
			# Back perimeter
				5.5  5.0 -0.88,
				4.0  5.5 -0.968,
				7.0  8.0 -1.408,
				4.0  9.0 -1.584,
				1.0  5.0 -0.88,
				2.5  4.5 -0.792,
			]
		}
		coordIndex [
		# Front
			0,  1,  2,  3,  4,  5,  6, -1,
		# Back
			0, 12, 11, 10,  9,  8,  7, -1,
		# Sides
			0,  7,  1, -1,
			1,  7,  8,  2, -1,
			2,  8,  9,  3, -1,
			3,  9, 10,  4, -1,
			4, 10, 11,  5, -1,
			5, 11, 12,  6, -1,
			6, 12,  0, -1,
		]
		convex FALSE
	}
}
```

Construindo grids elevados

Motivação

Exemplo

Sintaxe: ElevationGrid

Sintaxe: ElevationGrid

Um exemplo de elevação com grid

Um exemplo de elevação com grid

Resumo

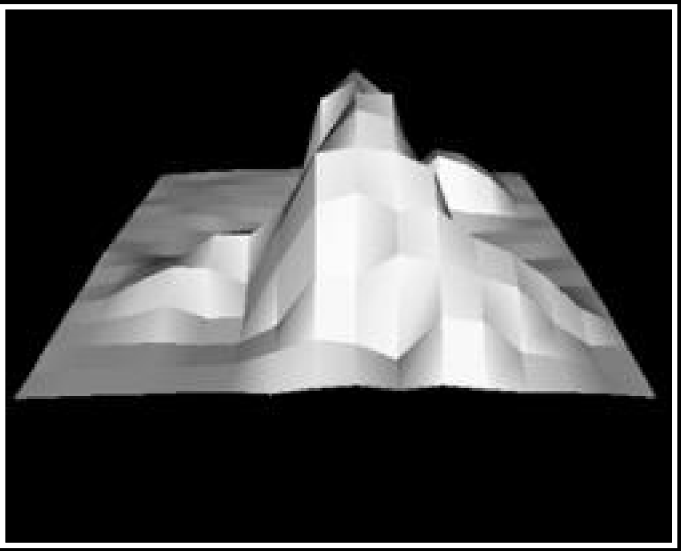
Construindo grids elevados

Motivação

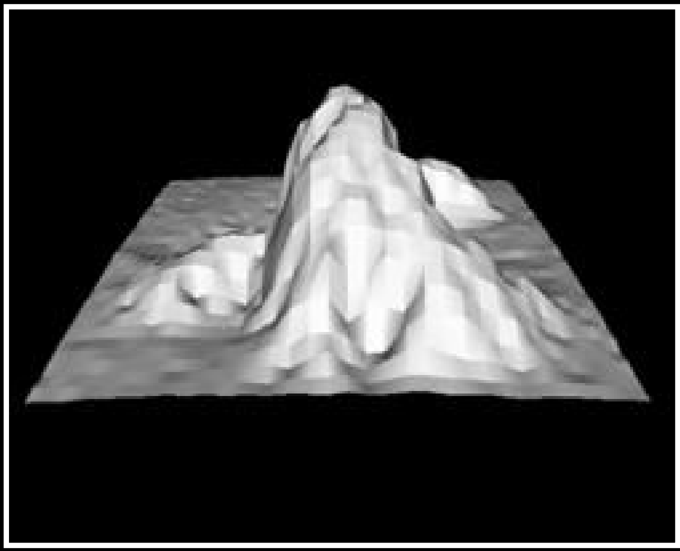
- **Construindo terrenos é muito comum**
  - **Colinas, vales, montanhas**
  - **Outros usos...**
- **Você pode construir um terreno usando um nó `IndexedFaceSet`**
- **Você pode construir terrenos mais eficientemente usando um nó `ElevationGrid`**

Construindo grids elevados

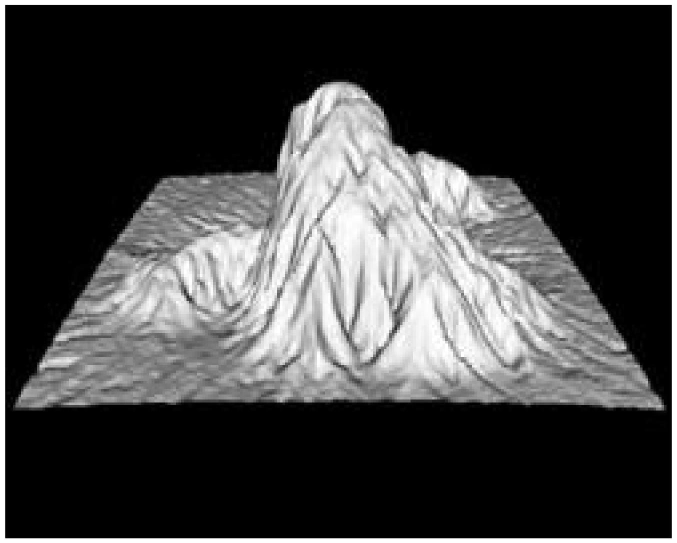
Exemplo



[mount16.wrl]



[mount32.wrl]



[mount128.wrl]

Construindo grids elevados

Sintaxe: ElevationGrid

- **Um nó geométrico `ElevationGrid` cria terrenos**
  - ***X & Z dimensions* - tamanho do grid**
  - ***X & Z spacings* - distâncias de linha e coluna**
  - ***mais* . . .**

```
ElevationGrid {
    xDimension 3
    zDimension 2
    xSpacing   1.0
    zSpacing   1.0
    . . .
}
```

Construindo grids elevados

Sintaxe: ElevationGrid

- **Um nó geométrico `ElevationGrid` cria terrenos**
  - ***height* - elevações nos pontos do grid**

```
ElevationGrid {
    . . .
    height [
        0.0, -0.5, 0.0,
        0.2,  4.0, 0.0
    ]
}
```

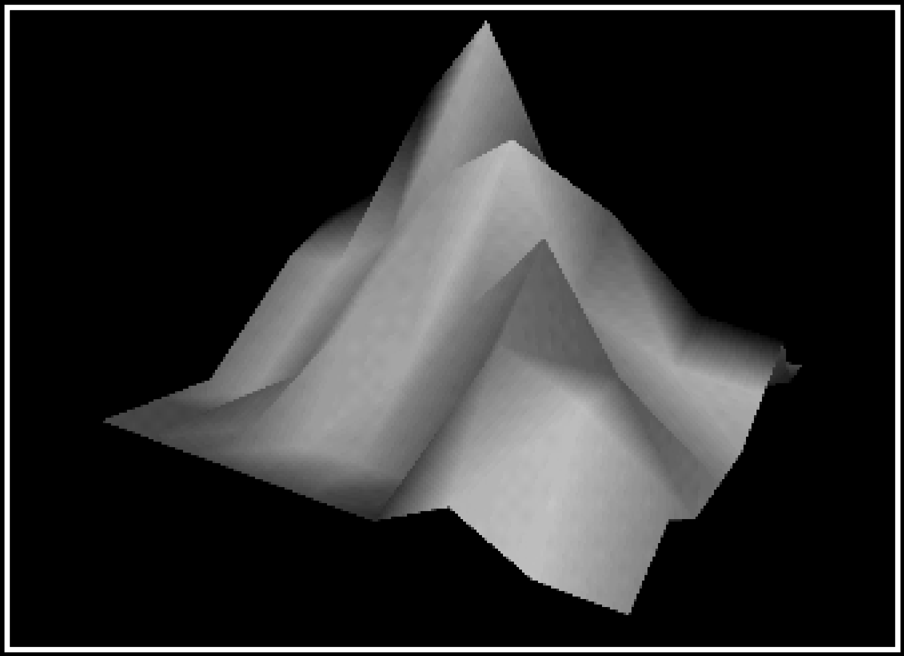
Construindo grids elevados

Um exemplo de grid elevado

```
Shape {
    . . .
    geometry ElevationGrid {
        xDimension 9
        zDimension 9
        xSpacing   1.0
        zSpacing   1.0
        height [ . . . ]
    }
}
```

Construindo grids elevados

Um exemplo de grid elevado



[**mount.wrl**]

Construindo grids elevados

Resumo

- **Um nó `ElevationGrid` cria, eficientemente, um terreno**
- **O tamanho do grid é especificado nos campos `xDimension` e `zDimension`**
- **Espaçamento do grid é especificado nos campos `Spacing` e `zSpacing`**
- **Elevações em cada ponto do grid são especificados no campo `height`**

```
#VRML V2.0 utf8

# WHO:    John L. Moreland
# WHAT:   mount16x16
# WHY:    Mountain ElevationGrid generated by im2wrl
# WHERE:  San Diego Supercomputer Center
# WHEN:   Tue Apr 29 09:10:45 PDT 1997
# HOW:    VRML 2.0

Shape {
        appearance Appearance {
                material Material {
                }
        }
        geometry ElevationGrid {
                xDimension 16
                zDimension 16
                xSpacing 0.16
                zSpacing 0.16
                solid TRUE
                height [
                        0.019608,
                        0.031373,
                        0.043137,
                        0.050980,
                        0.054902,
                        0.054902,
                        0.066667,
                        0.062745,
                        0.062745,
                        0.062745,
                        0.058824,
                        0.054902,
                        0.047059,
                        0.047059,
                        0.035294,
                        0.031373,
                        0.035294,
                        0.035294,
                        0.039216,
                        0.054902,
                        0.062745,
                        0.066667,
                        0.070588,
                        0.070588,
                        0.070588,
                        0.070588,
                        0.098039,
                        0.129412,
                        0.086275,
                        0.078431,
                        0.066667,
                        0.043137,
                        0.039216,
                        0.050980,
                        0.058824,
                        0.074510,
                        0.066667,
                        0.062745,
                        0.054902,
                        0.062745,
```

```
0.058824,
0.066667,
0.145098,
0.274510,
0.227451,
0.168627,
0.074510,
0.050980,
0.035294,
0.058824,
0.070588,
0.074510,
0.062745,
0.058824,
0.047059,
0.039216,
0.054902,
0.058824,
0.211765,
0.356863,
0.325490,
0.223529,
0.074510,
0.050980,
0.054902,
0.070588,
0.074510,
0.078431,
0.054902,
0.047059,
0.035294,
0.027451,
0.086275,
0.129412,
0.231373,
0.372549,
0.321569,
0.231373,
0.082353,
0.058824,
0.054902,
0.066667,
0.082353,
0.074510,
0.035294,
0.023529,
0.027451,
0.058824,
0.454902,
0.517647,
0.239216,
0.121569,
0.094118,
0.078431,
0.058824,
0.058824,
0.062745,
0.066667,
0.074510,
0.047059,
0.023529,
```

```
0.027451,
0.129412,
0.474510,
0.815686,
0.701961,
0.317647,
0.054902,
0.035294,
0.050980,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.050980,
0.031373,
0.082353,
0.274510,
0.780392,
0.956863,
0.827451,
0.250980,
0.054902,
0.035294,
0.047059,
0.062745,
0.058824,
0.078431,
0.078431,
0.058824,
0.054902,
0.113725,
0.074510,
0.282353,
0.858824,
0.945098,
0.745098,
0.203922,
0.031373,
0.027451,
0.031373,
0.043137,
0.062745,
0.070588,
0.094118,
0.109804,
0.117647,
0.192157,
0.282353,
0.231373,
0.517647,
0.925490,
0.929412,
0.423529,
0.027451,
0.031373,
0.050980,
0.058824,
0.066667,
0.058824,
```

```
0.082353,
0.160784,
0.223529,
0.329412,
0.329412,
0.458824,
0.733333,
0.901961,
0.858824,
0.607843,
0.164706,
0.054902,
0.047059,
0.062745,
0.062745,
0.058824,
0.090196,
0.180392,
0.254902,
0.243137,
0.164706,
0.443137,
0.768627,
0.796078,
0.772549,
0.701961,
0.298039,
0.290196,
0.152941,
0.098039,
0.066667,
0.050980,
0.066667,
0.101961,
0.152941,
0.086275,
0.094118,
0.341176,
0.611765,
0.662745,
0.584314,
0.596078,
0.470588,
0.396078,
0.282353,
0.137255,
0.058824,
0.047059,
0.054902,
0.062745,
0.070588,
0.058824,
0.066667,
0.152941,
0.388235,
0.368627,
0.474510,
0.439216,
0.380392,
0.325490,
0.164706,
```

```
                        0.058824,
                        0.050980,
                        0.035294,
                        0.050980,
                        0.058824,
                        0.062745,
                        0.066667,
                        0.074510,
                        0.133333,
                        0.192157,
                        0.172549,
                        0.098039,
                        0.290196,
                        0.274510,
                        0.133333,
                        0.105882,
                        0.054902,
                        0.047059,
                        0.035294,
                        0.043137,
                        0.050980,
                        0.054902,
                        0.062745,
                        0.066667,
                        0.054902,
                        0.082353,
                        0.098039,
                        0.074510,
                        0.098039,
                        0.078431,
                        0.062745,
                        0.062745,
                        0.054902,
                        0.039216,
                ]
        }
}
```

```
#VRML V2.0 utf8

# WHO:    John L. Moreland
# WHAT:   mount32x32
# WHY:    Mountain ElevationGrid generated by im2wrl
# WHERE:  San Diego Supercomputer Center
# WHEN:   Tue Apr 29 09:10:45 PDT 1997
# HOW:    VRML 2.0

Shape {
        appearance Appearance {
                material Material {
                }
        }
        geometry ElevationGrid {
                xDimension 32
                zDimension 32
                xSpacing 0.08
                zSpacing 0.08
                solid TRUE
                height [
                        0.019608,
                        0.027451,
                        0.031373,
                        0.035294,
                        0.043137,
                        0.043137,
                        0.050980,
                        0.047059,
                        0.054902,
                        0.054902,
                        0.054902,
                        0.054902,
                        0.066667,
                        0.062745,
                        0.062745,
                        0.062745,
                        0.062745,
                        0.062745,
                        0.062745,
                        0.058824,
                        0.058824,
                        0.062745,
                        0.054902,
                        0.054902,
                        0.047059,
                        0.050980,
                        0.047059,
                        0.039216,
                        0.035294,
                        0.031373,
                        0.031373,
                        0.019608,
                        0.019608,
                        0.027451,
                        0.039216,
                        0.039216,
                        0.047059,
                        0.047059,
                        0.050980,
                        0.047059,
```

```
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.070588,
0.066667,
0.070588,
0.070588,
0.066667,
0.062745,
0.066667,
0.074510,
0.070588,
0.078431,
0.074510,
0.062745,
0.062745,
0.054902,
0.047059,
0.043137,
0.039216,
0.031373,
0.027451,
0.035294,
0.039216,
0.035294,
0.050980,
0.039216,
0.058824,
0.054902,
0.050980,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.070588,
0.074510,
0.098039,
0.121569,
0.129412,
0.113725,
0.086275,
0.101961,
0.078431,
0.066667,
0.066667,
0.047059,
0.043137,
0.035294,
0.031373,
0.043137,
0.047059,
0.050980,
0.058824,
```

```
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.058824,
0.070588,
0.058824,
0.070588,
0.070588,
0.062745,
0.062745,
0.070588,
0.070588,
0.082353,
0.113725,
0.160784,
0.227451,
0.196078,
0.196078,
0.145098,
0.125490,
0.105882,
0.062745,
0.050980,
0.047059,
0.031373,
0.039216,
0.047059,
0.050980,
0.047059,
0.058824,
0.070588,
0.074510,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.062745,
0.062745,
0.054902,
0.058824,
0.062745,
0.066667,
0.082353,
0.145098,
0.247059,
0.274510,
0.270588,
0.227451,
0.219608,
0.168627,
0.121569,
0.074510,
0.058824,
0.050980,
0.043137,
0.047059,
```

```
0.050980,
0.054902,
0.062745,
0.062745,
0.074510,
0.070588,
0.070588,
0.058824,
0.058824,
0.062745,
0.050980,
0.047059,
0.054902,
0.039216,
0.058824,
0.054902,
0.050980,
0.058824,
0.086275,
0.129412,
0.254902,
0.333333,
0.305882,
0.294118,
0.250980,
0.211765,
0.129412,
0.070588,
0.058824,
0.050980,
0.047059,
0.035294,
0.050980,
0.058824,
0.050980,
0.070588,
0.078431,
0.074510,
0.066667,
0.062745,
0.054902,
0.058824,
0.047059,
0.047059,
0.039216,
0.039216,
0.035294,
0.054902,
0.066667,
0.058824,
0.090196,
0.211765,
0.301961,
0.356863,
0.364706,
0.325490,
0.298039,
0.223529,
0.121569,
0.074510,
0.058824,
```

```
0.050980,
0.047059,
0.047059,
0.058824,
0.058824,
0.066667,
0.070588,
0.078431,
0.078431,
0.062745,
0.058824,
0.050980,
0.039216,
0.047059,
0.039216,
0.039216,
0.043137,
0.050980,
0.082353,
0.117647,
0.074510,
0.152941,
0.219608,
0.325490,
0.384314,
0.372549,
0.356863,
0.317647,
0.247059,
0.105882,
0.066667,
0.054902,
0.054902,
0.043137,
0.054902,
0.050980,
0.070588,
0.066667,
0.074510,
0.086275,
0.078431,
0.058824,
0.054902,
0.043137,
0.047059,
0.035294,
0.035294,
0.031373,
0.027451,
0.050980,
0.086275,
0.235294,
0.129412,
0.160784,
0.231373,
0.286275,
0.372549,
0.329412,
0.321569,
0.254902,
```

```
0.231373,
0.172549,
0.082353,
0.050980,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.078431,
0.082353,
0.086275,
0.078431,
0.062745,
0.039216,
0.039216,
0.035294,
0.035294,
0.027451,
0.027451,
0.035294,
0.098039,
0.203922,
0.243137,
0.305882,
0.270588,
0.258824,
0.254902,
0.239216,
0.160784,
0.152941,
0.141176,
0.129412,
0.117647,
0.090196,
0.062745,
0.058824,
0.050980,
0.054902,
0.062745,
0.066667,
0.070588,
0.082353,
0.078431,
0.074510,
0.047059,
0.035294,
0.031373,
0.023529,
0.027451,
0.027451,
0.035294,
0.058824,
0.239216,
0.454902,
0.541176,
0.517647,
0.400000,
0.239216,
0.129412,
0.121569,
```

```
0.086275,
0.094118,
0.086275,
0.078431,
0.070588,
0.058824,
0.070588,
0.058824,
0.062745,
0.050980,
0.062745,
0.062745,
0.070588,
0.082353,
0.078431,
0.066667,
0.054902,
0.043137,
0.027451,
0.023529,
0.031373,
0.043137,
0.098039,
0.152941,
0.419608,
0.580392,
0.796078,
0.709804,
0.427451,
0.227451,
0.137255,
0.070588,
0.050980,
0.047059,
0.047059,
0.054902,
0.058824,
0.066667,
0.074510,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.074510,
0.078431,
0.047059,
0.031373,
0.023529,
0.023529,
0.027451,
0.039216,
0.129412,
0.227451,
0.474510,
0.733333,
0.815686,
0.854902,
0.701961,
```

```
0.552941,
0.317647,
0.145098,
0.054902,
0.027451,
0.035294,
0.043137,
0.050980,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.074510,
0.062745,
0.058824,
0.043137,
0.031373,
0.031373,
0.047059,
0.058824,
0.137255,
0.494118,
0.803922,
0.913725,
0.917647,
0.890196,
0.807843,
0.486275,
0.184314,
0.039216,
0.031373,
0.027451,
0.031373,
0.035294,
0.047059,
0.050980,
0.062745,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.066667,
0.054902,
0.050980,
0.050980,
0.031373,
0.027451,
0.082353,
0.211765,
0.274510,
0.537255,
0.780392,
0.929412,
```

```
0.956863,
0.909804,
0.827451,
0.541176,
0.250980,
0.117647,
0.054902,
0.031373,
0.035294,
0.039216,
0.047059,
0.050980,
0.062745,
0.062745,
0.058824,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.027451,
0.035294,
0.203922,
0.576471,
0.780392,
0.831373,
0.956863,
0.972549,
0.933333,
0.878431,
0.678431,
0.505882,
0.149020,
0.058824,
0.031373,
0.035294,
0.039216,
0.039216,
0.050980,
0.062745,
0.062745,
0.062745,
0.070588,
0.078431,
0.086275,
0.078431,
0.070588,
0.058824,
0.058824,
0.054902,
0.066667,
0.113725,
0.066667,
0.074510,
0.086275,
```

```
0.282353,
0.666667,
0.858824,
0.925490,
0.945098,
0.937255,
0.745098,
0.545098,
0.203922,
0.113725,
0.031373,
0.031373,
0.027451,
0.027451,
0.031373,
0.039216,
0.043137,
0.054902,
0.062745,
0.058824,
0.070588,
0.086275,
0.094118,
0.086275,
0.082353,
0.058824,
0.062745,
0.094118,
0.121569,
0.172549,
0.156863,
0.117647,
0.176471,
0.403922,
0.611765,
0.780392,
0.909804,
0.862745,
0.850980,
0.650980,
0.286275,
0.101961,
0.043137,
0.031373,
0.023529,
0.035294,
0.043137,
0.050980,
0.050980,
0.054902,
0.066667,
0.066667,
0.070588,
0.074510,
0.094118,
0.121569,
0.109804,
0.090196,
0.117647,
0.133333,
0.192157,
```

```
0.258824,
0.282353,
0.258824,
0.231373,
0.286275,
0.517647,
0.803922,
0.925490,
0.945098,
0.929412,
0.803922,
0.423529,
0.145098,
0.027451,
0.023529,
0.031373,
0.043137,
0.050980,
0.054902,
0.058824,
0.066667,
0.066667,
0.062745,
0.062745,
0.070588,
0.082353,
0.121569,
0.176471,
0.133333,
0.192157,
0.219608,
0.270588,
0.325490,
0.352941,
0.325490,
0.356863,
0.435294,
0.682353,
0.847059,
0.941176,
0.925490,
0.901961,
0.647059,
0.149020,
0.066667,
0.027451,
0.031373,
0.039216,
0.031373,
0.043137,
0.047059,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.070588,
0.082353,
0.098039,
0.160784,
```

```
0.235294,
0.223529,
0.290196,
0.329412,
0.345098,
0.329412,
0.352941,
0.458824,
0.568627,
0.733333,
0.874510,
0.901961,
0.882353,
0.858824,
0.741176,
0.607843,
0.384314,
0.164706,
0.066667,
0.054902,
0.043137,
0.047059,
0.070588,
0.062745,
0.070588,
0.062745,
0.062745,
0.054902,
0.074510,
0.094118,
0.149020,
0.207843,
0.231373,
0.266667,
0.301961,
0.286275,
0.294118,
0.270588,
0.262745,
0.447059,
0.709804,
0.792157,
0.839216,
0.850980,
0.819608,
0.819608,
0.776471,
0.741176,
0.572549,
0.258824,
0.121569,
0.101961,
0.086275,
0.105882,
0.109804,
0.078431,
0.074510,
0.066667,
0.058824,
0.058824,
0.062745,
```

```
0.090196,
0.129412,
0.180392,
0.231373,
0.254902,
0.262745,
0.243137,
0.184314,
0.164706,
0.266667,
0.443137,
0.654902,
0.768627,
0.788235,
0.796078,
0.741176,
0.772549,
0.709804,
0.701961,
0.545098,
0.298039,
0.298039,
0.290196,
0.215686,
0.152941,
0.101961,
0.098039,
0.086275,
0.066667,
0.066667,
0.054902,
0.062745,
0.078431,
0.078431,
0.145098,
0.203922,
0.207843,
0.121569,
0.149020,
0.121569,
0.082353,
0.188235,
0.411765,
0.580392,
0.698039,
0.729412,
0.701961,
0.690196,
0.690196,
0.674510,
0.631373,
0.576471,
0.486275,
0.462745,
0.396078,
0.235294,
0.152941,
0.152941,
0.113725,
0.070588,
```

```
0.070588,
0.062745,
0.050980,
0.058824,
0.066667,
0.094118,
0.101961,
0.137255,
0.152941,
0.137255,
0.086275,
0.078431,
0.094118,
0.141176,
0.341176,
0.529412,
0.611765,
0.619608,
0.662745,
0.654902,
0.584314,
0.600000,
0.596078,
0.529412,
0.470588,
0.462745,
0.396078,
0.270588,
0.282353,
0.211765,
0.137255,
0.078431,
0.058824,
0.062745,
0.050980,
0.050980,
0.066667,
0.070588,
0.078431,
0.074510,
0.101961,
0.082353,
0.066667,
0.062745,
0.070588,
0.156863,
0.278431,
0.329412,
0.505882,
0.568627,
0.576471,
0.576471,
0.545098,
0.545098,
0.513726,
0.501961,
0.466667,
0.407843,
0.360784,
0.282353,
0.227451,
```

```
0.129412,
0.098039,
0.066667,
0.066667,
0.050980,
0.047059,
0.043137,
0.054902,
0.054902,
0.062745,
0.066667,
0.070588,
0.062745,
0.058824,
0.058824,
0.066667,
0.101961,
0.152941,
0.301961,
0.388235,
0.501961,
0.368627,
0.376471,
0.474510,
0.439216,
0.439216,
0.443137,
0.380392,
0.368627,
0.325490,
0.239216,
0.164706,
0.098039,
0.058824,
0.058824,
0.050980,
0.054902,
0.043137,
0.050980,
0.058824,
0.058824,
0.058824,
0.066667,
0.070588,
0.066667,
0.062745,
0.062745,
0.078431,
0.101961,
0.156863,
0.262745,
0.305882,
0.321569,
0.192157,
0.258824,
0.270588,
0.243137,
0.376471,
0.380392,
0.341176,
```

```
0.266667,
0.227451,
0.203922,
0.121569,
0.094118,
0.062745,
0.050980,
0.050980,
0.035294,
0.035294,
0.047059,
0.050980,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.074510,
0.098039,
0.133333,
0.129412,
0.192157,
0.192157,
0.172549,
0.090196,
0.098039,
0.223529,
0.290196,
0.266667,
0.274510,
0.200000,
0.133333,
0.117647,
0.105882,
0.066667,
0.054902,
0.047059,
0.047059,
0.047059,
0.023529,
0.043137,
0.043137,
0.047059,
0.054902,
0.058824,
0.058824,
0.054902,
0.062745,
0.066667,
0.070588,
0.082353,
0.086275,
0.070588,
0.121569,
0.160784,
0.098039,
0.066667,
0.113725,
0.164706,
```

```
0.184314,
0.149020,
0.117647,
0.113725,
0.090196,
0.054902,
0.082353,
0.062745,
0.043137,
0.047059,
0.043137,
0.035294,
0.035294,
0.035294,
0.043137,
0.043137,
0.050980,
0.050980,
0.054902,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.054902,
0.078431,
0.082353,
0.090196,
0.098039,
0.054902,
0.074510,
0.098039,
0.098039,
0.086275,
0.078431,
0.066667,
0.062745,
0.047059,
0.062745,
0.062745,
0.054902,
0.047059,
0.039216,
0.031373,
0.023529,
0.031373,
0.031373,
0.043137,
0.047059,
0.050980,
0.054902,
0.062745,
0.058824,
0.062745,
0.058824,
0.062745,
0.058824,
0.066667,
0.066667,
0.062745,
```

```
                                0.078431,
                                0.062745,
                                0.066667,
                                0.078431,
                                0.078431,
                                0.062745,
                                0.062745,
                                0.062745,
                                0.050980,
                                0.043137,
                                0.054902,
                                0.050980,
                                0.047059,
                                0.039216,
                                0.039216,
                                0.031373,
                      ]
          }
}
```

```
#VRML V2.0 utf8

# WHO:    John L. Moreland
# WHAT:   mount128x128
# WHY:    Mountain ElevationGrid generated by im2wrl
# WHERE:  San Diego Supercomputer Center
# WHEN:   Tue Apr 29 09:10:45 PDT 1997
# HOW:    VRML 2.0

Viewpoint { position 1.253 1.7 4.5  orientation 1.0 0.0 0.0 -0.5 }
Shape {
        appearance Appearance {
                material Material {
                }
        }
        geometry ElevationGrid {
                xDimension 128
                zDimension 128
                xSpacing 0.02
                zSpacing 0.02
                solid TRUE
                height [
                        0.019608,
                        0.015686,
                        0.019608,
                        0.023529,
                        0.027451,
                        0.031373,
                        0.031373,
                        0.031373,
                        0.031373,
                        0.027451,
                        0.035294,
                        0.035294,
                        0.035294,
                        0.031373,
                        0.031373,
                        0.039216,
                        0.043137,
                        0.043137,
                        0.039216,
                        0.043137,
                        0.043137,
                        0.043137,
                        0.047059,
                        0.050980,
                        0.050980,
                        0.050980,
                        0.047059,
                        0.047059,
                        0.047059,
                        0.050980,
                        0.050980,
                        0.054902,
                        0.054902,
                        0.058824,
                        0.066667,
                        0.062745,
                        0.054902,
                        0.050980,
                        0.054902,
```

```
0.054902,
0.054902,
0.058824,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.058824,
0.054902,
0.050980,
0.054902,
0.054902,
0.054902,
0.050980,
0.054902,
0.058824,
0.058824,
0.054902,
0.047059,
0.047059,
0.047059,
0.050980,
```

```
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.027451,
0.031373,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.023529,
0.019608,
0.019608,
0.019608,
0.015686,
0.015686,
0.019608,
0.019608,
0.015686,
0.019608,
0.027451,
0.031373,
0.031373,
0.035294,
0.035294,
0.031373,
0.035294,
0.039216,
0.039216,
0.035294,
0.035294,
0.039216,
0.043137,
0.043137,
0.039216,
0.047059,
0.047059,
0.047059,
0.047059,
0.050980,
0.054902,
0.054902,
0.050980,
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
```

```
0.058824,
0.062745,
0.066667,
0.062745,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
```

```
0.058824,
0.058824,
0.058824,
0.054902,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.039216,
0.039216,
0.035294,
0.027451,
0.027451,
0.035294,
0.035294,
0.031373,
0.031373,
0.027451,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.019608,
0.019608,
0.019608,
0.023529,
0.023529,
0.019608,
0.023529,
0.031373,
0.031373,
0.035294,
0.039216,
0.035294,
0.035294,
0.039216,
0.039216,
0.035294,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.050980,
0.054902,
```

```
0.054902,
0.054902,
0.050980,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.066667,
0.066667,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.062745,
0.066667,
```

```
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.047059,
0.039216,
0.039216,
0.043137,
0.039216,
0.043137,
0.047059,
0.043137,
0.039216,
0.035294,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.027451,
0.023529,
0.019608,
0.023529,
0.027451,
0.027451,
0.023529,
0.019608,
0.023529,
0.019608,
0.027451,
0.031373,
0.027451,
0.027451,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
0.035294,
0.039216,
0.039216,
0.035294,
0.043137,
0.047059,
0.047059,
0.043137,
```

```
0.043137,
0.043137,
0.047059,
0.050980,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.054902,
0.047059,
0.050980,
0.054902,
0.054902,
0.062745,
0.066667,
0.062745,
0.054902,
0.054902,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
```

```
0.066667,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.047059,
0.043137,
0.047059,
0.047059,
0.043137,
0.047059,
0.047059,
0.039216,
0.035294,
0.039216,
0.039216,
0.035294,
0.035294,
0.035294,
0.031373,
0.027451,
0.023529,
0.019608,
0.023529,
0.027451,
0.027451,
0.023529,
0.019608,
0.023529,
0.019608,
0.027451,
0.031373,
0.031373,
0.027451,
0.027451,
0.031373,
0.035294,
0.039216,
0.039216,
0.035294,
```

```
0.039216,
0.039216,
0.035294,
0.039216,
0.047059,
0.047059,
0.047059,
0.047059,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.054902,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
```

```
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.066667,
0.070588,
0.074510,
0.074510,
0.074510,
0.074510,
0.070588,
0.074510,
0.074510,
0.074510,
0.078431,
0.074510,
0.062745,
0.062745,
0.074510,
0.074510,
0.070588,
0.070588,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.054902,
0.047059,
0.050980,
0.050980,
0.047059,
0.050980,
0.043137,
0.039216,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.023529,
0.023529,
0.027451,
0.023529,
0.027451,
0.031373,
0.031373,
```

```
0.027451,
0.027451,
0.031373,
0.035294,
0.039216,
0.039216,
0.039216,
0.043137,
0.039216,
0.031373,
0.039216,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.054902,
0.054902,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
```

```
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
0.078431,
0.078431,
0.078431,
0.074510,
0.082353,
0.086275,
0.086275,
0.086275,
0.078431,
0.066667,
0.066667,
0.082353,
0.086275,
0.078431,
0.070588,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.070588,
0.066667,
0.066667,
0.058824,
0.050980,
0.054902,
0.054902,
0.054902,
0.047059,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.047059,
0.039216,
0.031373,
0.035294,
0.039216,
0.035294,
0.035294,
0.031373,
0.027451,
0.031373,
```

```
0.027451,
0.023529,
0.027451,
0.027451,
0.023529,
0.027451,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.043137,
0.043137,
0.047059,
0.043137,
0.035294,
0.039216,
0.047059,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.054902,
0.058824,
0.054902,
0.050980,
0.054902,
0.058824,
0.054902,
0.047059,
0.047059,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
```

```
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
0.082353,
0.086275,
0.090196,
0.090196,
0.086275,
0.082353,
0.090196,
0.101961,
0.101961,
0.098039,
0.082353,
0.074510,
0.078431,
0.086275,
0.090196,
0.082353,
0.070588,
0.062745,
0.074510,
0.074510,
0.070588,
0.074510,
0.078431,
0.074510,
0.074510,
0.062745,
0.054902,
0.058824,
0.058824,
0.054902,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.050980,
0.043137,
0.035294,
0.035294,
```

```
0.039216,
0.043137,
0.039216,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.027451,
0.027451,
0.031373,
0.031373,
0.027451,
0.031373,
0.035294,
0.035294,
0.039216,
0.035294,
0.035294,
0.039216,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
0.043137,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.054902,
0.062745,
0.058824,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
```

```
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.078431,
0.086275,
0.090196,
0.094118,
0.101961,
0.109804,
0.101961,
0.094118,
0.101961,
0.117647,
0.125490,
0.109804,
0.094118,
0.098039,
0.101961,
0.094118,
0.086275,
0.086275,
0.078431,
0.070588,
0.082353,
0.090196,
0.082353,
0.086275,
0.090196,
0.082353,
0.078431,
0.070588,
0.062745,
0.062745,
0.058824,
0.054902,
0.062745,
0.066667,
```

```
0.066667,
0.062745,
0.058824,
0.050980,
0.043137,
0.039216,
0.039216,
0.043137,
0.043137,
0.039216,
0.039216,
0.035294,
0.031373,
0.035294,
0.031373,
0.027451,
0.027451,
0.035294,
0.035294,
0.031373,
0.035294,
0.039216,
0.039216,
0.043137,
0.039216,
0.035294,
0.035294,
0.043137,
0.047059,
0.050980,
0.047059,
0.039216,
0.039216,
0.039216,
0.039216,
0.047059,
0.054902,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.058824,
0.050980,
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
```

```
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.078431,
0.086275,
0.090196,
0.098039,
0.113725,
0.125490,
0.125490,
0.121569,
0.121569,
0.133333,
0.145098,
0.129412,
0.105882,
0.113725,
0.125490,
0.113725,
0.101961,
0.101961,
0.094118,
0.086275,
0.098039,
0.101961,
0.098039,
0.101961,
0.101961,
0.094118,
0.086275,
```

```
0.078431,
0.074510,
0.062745,
0.054902,
0.066667,
0.078431,
0.078431,
0.074510,
0.066667,
0.058824,
0.050980,
0.047059,
0.047059,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.043137,
0.050980,
0.050980,
0.047059,
0.043137,
0.043137,
0.039216,
0.039216,
0.050980,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.066667,
0.062745,
0.062745,
0.066667,
```

```
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.070588,
0.062745,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
0.074510,
0.078431,
0.082353,
0.086275,
0.094118,
0.109804,
0.133333,
0.149020,
0.152941,
0.149020,
0.149020,
0.160784,
0.152941,
0.121569,
0.125490,
0.149020,
0.141176,
0.125490,
0.121569,
0.121569,
0.121569,
```

```
0.125490,
0.121569,
0.121569,
0.121569,
0.113725,
0.094118,
0.090196,
0.090196,
0.078431,
0.062745,
0.062745,
0.078431,
0.090196,
0.082353,
0.078431,
0.066667,
0.058824,
0.054902,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
0.039216,
0.039216,
0.039216,
0.035294,
0.035294,
0.035294,
0.031373,
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.043137,
0.050980,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
```

```
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
0.082353,
0.086275,
0.090196,
0.098039,
0.105882,
0.117647,
0.145098,
0.168627,
0.176471,
0.176471,
0.176471,
0.180392,
0.176471,
0.152941,
```

```
0.152941,
0.180392,
0.176471,
0.149020,
0.149020,
0.152941,
0.152941,
0.149020,
0.145098,
0.149020,
0.137255,
0.113725,
0.098039,
0.094118,
0.090196,
0.082353,
0.078431,
0.082353,
0.090196,
0.094118,
0.082353,
0.074510,
0.066667,
0.058824,
0.054902,
0.050980,
0.050980,
0.047059,
0.047059,
0.043137,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.031373,
0.035294,
0.035294,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.050980,
0.050980,
0.054902,
0.062745,
0.062745,
0.058824,
```

```
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.062745,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.070588,
0.066667,
0.058824,
0.062745,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.070588,
0.074510,
0.074510,
0.078431,
0.086275,
0.090196,
0.098039,
0.105882,
0.113725,
0.133333,
0.156863,
```

```
0.172549,
0.180392,
0.188235,
0.203922,
0.207843,
0.203922,
0.192157,
0.184314,
0.200000,
0.196078,
0.168627,
0.172549,
0.180392,
0.176471,
0.172549,
0.168627,
0.164706,
0.141176,
0.121569,
0.117647,
0.109804,
0.098039,
0.098039,
0.101961,
0.105882,
0.101961,
0.090196,
0.078431,
0.070588,
0.062745,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
```

```
0.058824,
0.058824,
0.054902,
0.058824,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.070588,
0.066667,
0.062745,
0.062745,
0.058824,
0.062745,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.074510,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
0.078431,
```

```
0.082353,
0.086275,
0.090196,
0.098039,
0.113725,
0.129412,
0.149020,
0.160784,
0.160784,
0.176471,
0.207843,
0.231373,
0.227451,
0.219608,
0.211765,
0.211765,
0.196078,
0.180392,
0.192157,
0.200000,
0.196078,
0.196078,
0.184314,
0.164706,
0.145098,
0.141176,
0.145098,
0.133333,
0.125490,
0.125490,
0.129412,
0.121569,
0.105882,
0.094118,
0.082353,
0.070588,
0.062745,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
0.039216,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
```

```
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.070588,
0.070588,
0.066667,
0.062745,
0.066667,
0.070588,
0.066667,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
```

```
0.070588,
0.066667,
0.062745,
0.066667,
0.070588,
0.078431,
0.082353,
0.086275,
0.090196,
0.094118,
0.109804,
0.125490,
0.129412,
0.137255,
0.145098,
0.149020,
0.164706,
0.203922,
0.235294,
0.247059,
0.250980,
0.243137,
0.227451,
0.203922,
0.200000,
0.211765,
0.211765,
0.211765,
0.207843,
0.184314,
0.164706,
0.164706,
0.168627,
0.160784,
0.152941,
0.152941,
0.152941,
0.149020,
0.129412,
0.113725,
0.098039,
0.090196,
0.074510,
0.062745,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.043137,
0.039216,
0.039216,
0.039216,
0.039216,
0.035294,
0.035294,
0.039216,
```

```
0.043137,
0.039216,
0.039216,
0.043137,
0.047059,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.050980,
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.070588,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.054902,
0.054902,
0.058824,
```

```
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.078431,
0.082353,
0.086275,
0.094118,
0.101961,
0.117647,
0.137255,
0.149020,
0.152941,
0.160784,
0.176471,
0.196078,
0.215686,
0.239216,
0.258824,
0.270588,
0.266667,
0.247059,
0.231373,
0.235294,
0.231373,
0.223529,
0.219608,
0.207843,
0.188235,
0.188235,
0.192157,
0.184314,
0.176471,
0.172549,
0.172549,
0.172549,
0.164706,
0.145098,
0.121569,
0.101961,
0.090196,
0.074510,
0.066667,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
```

```
0.047059,
0.043137,
0.039216,
0.035294,
0.031373,
0.035294,
0.035294,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.050980,
0.054902,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.062745,
0.058824,
0.058824,
```

```
0.062745,
0.062745,
0.062745,
0.058824,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.074510,
0.078431,
0.086275,
0.094118,
0.109804,
0.133333,
0.164706,
0.184314,
0.196078,
0.219608,
0.243137,
0.243137,
0.243137,
0.262745,
0.278431,
0.274510,
0.258824,
0.258824,
0.258824,
0.247059,
0.231373,
0.219608,
0.207843,
0.207843,
0.211765,
0.207843,
0.200000,
0.192157,
0.188235,
0.180392,
0.172549,
0.164706,
0.149020,
0.129412,
0.109804,
0.090196,
0.074510,
0.066667,
0.058824,
0.058824,
0.058824,
```

```
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.035294,
0.027451,
0.019608,
0.039216,
0.039216,
0.043137,
0.047059,
0.047059,
0.043137,
0.043137,
0.047059,
0.050980,
0.054902,
0.050980,
0.047059,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.070588,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.050980,
0.047059,
0.047059,
```

```
0.054902,
0.058824,
0.062745,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.054902,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.082353,
0.090196,
0.101961,
0.117647,
0.145098,
0.184314,
0.211765,
0.227451,
0.247059,
0.266667,
0.270588,
0.262745,
0.274510,
0.286275,
0.282353,
0.270588,
0.270588,
0.266667,
0.250980,
0.235294,
0.227451,
0.227451,
0.231373,
0.227451,
0.219608,
0.215686,
0.207843,
0.196078,
0.168627,
0.145098,
0.141176,
0.133333,
0.121569,
```

```
0.121569,
0.105882,
0.086275,
0.074510,
0.066667,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.043137,
0.039216,
0.031373,
0.023529,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.047059,
0.050980,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.062745,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
0.070588,
0.070588,
0.074510,
0.074510,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
```

```
0.062745,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.047059,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.058824,
0.054902,
0.047059,
0.047059,
0.054902,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.082353,
0.094118,
0.109804,
0.129412,
0.160784,
0.192157,
0.215686,
0.239216,
0.254902,
0.270588,
0.274510,
0.282353,
0.290196,
0.294118,
0.290196,
0.286275,
0.282353,
0.274510,
0.258824,
0.247059,
0.250980,
0.250980,
0.243137,
0.235294,
0.231373,
0.219608,
```

```
0.207843,
0.188235,
0.149020,
0.105882,
0.094118,
0.098039,
0.098039,
0.105882,
0.105882,
0.098039,
0.086275,
0.074510,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.047059,
0.047059,
0.043137,
0.043137,
0.043137,
0.035294,
0.027451,
0.047059,
0.047059,
0.047059,
0.047059,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.070588,
0.074510,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.070588,
0.070588,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
```

```
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.050980,
0.047059,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.070588,
0.074510,
0.082353,
0.094118,
0.109804,
0.137255,
0.160784,
0.188235,
0.219608,
0.239216,
0.262745,
0.282353,
0.298039,
0.305882,
0.301961,
0.301961,
0.298039,
0.294118,
0.286275,
0.274510,
```

```
0.270588,
0.270588,
0.262745,
0.254902,
0.247059,
0.239216,
0.223529,
0.203922,
0.176471,
0.141176,
0.105882,
0.090196,
0.086275,
0.082353,
0.082353,
0.086275,
0.090196,
0.086275,
0.074510,
0.066667,
0.062745,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.043137,
0.047059,
0.047059,
0.043137,
0.031373,
0.031373,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.070588,
0.074510,
0.074510,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
```

```
0.070588,
0.070588,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.058824,
0.058824,
0.050980,
0.047059,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.050980,
0.047059,
0.043137,
0.047059,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.054902,
0.054902,
0.054902,
0.062745,
0.070588,
0.074510,
0.074510,
0.078431,
0.090196,
0.105882,
0.117647,
0.133333,
0.176471,
0.215686,
0.235294,
0.266667,
0.294118,
0.313726,
```

```
0.321569,
0.317647,
0.313726,
0.309804,
0.301961,
0.294118,
0.294118,
0.290196,
0.282353,
0.274510,
0.266667,
0.258824,
0.247059,
0.227451,
0.207843,
0.188235,
0.172549,
0.156863,
0.137255,
0.121569,
0.105882,
0.090196,
0.074510,
0.074510,
0.078431,
0.074510,
0.062745,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.035294,
0.035294,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
```

```
0.074510,
0.074510,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.054902,
0.058824,
0.054902,
0.047059,
0.043137,
0.043137,
0.050980,
0.054902,
0.054902,
0.054902,
0.050980,
0.039216,
0.043137,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.070588,
0.082353,
0.086275,
0.090196,
0.105882,
0.121569,
0.129412,
```

```
0.145098,
0.184314,
0.231373,
0.254902,
0.278431,
0.305882,
0.325490,
0.333333,
0.329412,
0.321569,
0.317647,
0.305882,
0.305882,
0.305882,
0.298039,
0.294118,
0.290196,
0.282353,
0.270588,
0.250980,
0.231373,
0.219608,
0.219608,
0.211765,
0.196078,
0.176471,
0.152941,
0.129412,
0.109804,
0.086275,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.043137,
0.043137,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
```

```
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
0.074510,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.050980,
0.047059,
0.054902,
0.058824,
0.050980,
0.050980,
0.047059,
0.043137,
0.047059,
0.050980,
0.054902,
0.054902,
0.047059,
0.039216,
0.043137,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
```

```
0.066667,
0.082353,
0.101961,
0.113725,
0.121569,
0.141176,
0.164706,
0.180392,
0.200000,
0.239216,
0.278431,
0.298039,
0.313726,
0.329412,
0.337255,
0.333333,
0.325490,
0.317647,
0.313726,
0.321569,
0.313726,
0.305882,
0.305882,
0.301961,
0.290196,
0.270588,
0.250980,
0.235294,
0.223529,
0.215686,
0.203922,
0.184314,
0.164706,
0.141176,
0.117647,
0.101961,
0.082353,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.039216,
0.039216,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
```

```
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.066667,
0.070588,
0.074510,
0.078431,
0.078431,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.047059,
0.043137,
0.050980,
0.054902,
0.050980,
0.050980,
0.054902,
0.047059,
0.043137,
0.047059,
0.050980,
0.050980,
0.047059,
0.039216,
0.043137,
0.054902,
0.054902,
0.047059,
0.043137,
0.043137,
0.043137,
0.047059,
0.054902,
0.058824,
```

```
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.082353,
0.105882,
0.129412,
0.145098,
0.160784,
0.188235,
0.219608,
0.235294,
0.258824,
0.290196,
0.317647,
0.329412,
0.333333,
0.341176,
0.345098,
0.337255,
0.329412,
0.329412,
0.333333,
0.325490,
0.317647,
0.313726,
0.309804,
0.290196,
0.274510,
0.262745,
0.239216,
0.211765,
0.192157,
0.176471,
0.160784,
0.141176,
0.113725,
0.090196,
0.082353,
0.074510,
0.062745,
0.058824,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.043137,
```

```
0.039216,
0.035294,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.078431,
0.078431,
0.078431,
0.074510,
0.074510,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.050980,
0.043137,
0.047059,
0.050980,
0.047059,
0.050980,
0.050980,
0.050980,
0.043137,
0.039216,
0.043137,
0.047059,
0.043137,
0.039216,
0.043137,
0.050980,
0.047059,
```

```
0.039216,
0.039216,
0.039216,
0.039216,
0.047059,
0.054902,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.070588,
0.082353,
0.101961,
0.133333,
0.164706,
0.184314,
0.203922,
0.235294,
0.258824,
0.274510,
0.294118,
0.325490,
0.337255,
0.341176,
0.349020,
0.352941,
0.352941,
0.349020,
0.349020,
0.349020,
0.337255,
0.329412,
0.321569,
0.309804,
0.294118,
0.290196,
0.286275,
0.262745,
0.235294,
0.215686,
0.196078,
0.176471,
0.152941,
0.117647,
0.098039,
0.094118,
0.090196,
0.078431,
0.066667,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
```

```
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.047059,
0.039216,
0.035294,
0.035294,
0.039216,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.078431,
0.078431,
0.082353,
0.078431,
0.078431,
0.074510,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.054902,
0.058824,
0.054902,
0.047059,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
```

```
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.047059,
0.035294,
0.035294,
0.043137,
0.047059,
0.050980,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.070588,
0.078431,
0.090196,
0.113725,
0.149020,
0.188235,
0.211765,
0.227451,
0.254902,
0.282353,
0.301961,
0.325490,
0.337255,
0.349020,
0.356863,
0.360784,
0.360784,
0.364706,
0.364706,
0.360784,
0.349020,
0.337255,
0.325490,
0.313726,
0.309804,
0.309804,
0.298039,
0.282353,
0.262745,
0.247059,
0.223529,
0.196078,
0.168627,
0.145098,
0.121569,
0.109804,
0.098039,
0.086275,
0.074510,
```

```
0.066667,
0.062745,
0.058824,
0.058824,
0.062745,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.039216,
0.039216,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.070588,
0.074510,
0.078431,
0.078431,
0.078431,
0.082353,
0.078431,
0.078431,
0.074510,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.054902,
0.050980,
0.054902,
0.054902,
0.050980,
0.043137,
```

```
0.039216,
0.043137,
0.043137,
0.047059,
0.047059,
0.043137,
0.043137,
0.039216,
0.035294,
0.039216,
0.043137,
0.047059,
0.043137,
0.035294,
0.039216,
0.050980,
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.070588,
0.070588,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.082353,
0.101961,
0.125490,
0.160784,
0.184314,
0.203922,
0.231373,
0.266667,
0.298039,
0.325490,
0.345098,
0.356863,
0.368627,
0.368627,
0.376471,
0.380392,
0.376471,
0.368627,
0.352941,
0.341176,
0.329412,
0.321569,
0.321569,
0.321569,
0.305882,
0.290196,
0.270588,
0.239216,
0.215686,
0.196078,
```

```
0.180392,
0.156863,
0.129412,
0.105882,
0.090196,
0.082353,
0.074510,
0.070588,
0.062745,
0.062745,
0.062745,
0.062745,
0.054902,
0.047059,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.043137,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.054902,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
0.078431,
0.078431,
0.082353,
0.078431,
0.078431,
0.078431,
0.074510,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.047059,
```

```
0.050980,
0.050980,
0.047059,
0.050980,
0.050980,
0.050980,
0.047059,
0.039216,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.035294,
0.043137,
0.047059,
0.047059,
0.039216,
0.035294,
0.043137,
0.054902,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
0.078431,
0.078431,
0.070588,
0.070588,
0.074510,
0.078431,
0.082353,
0.086275,
0.098039,
0.109804,
0.125490,
0.141176,
0.152941,
0.156863,
0.168627,
0.207843,
0.247059,
0.282353,
0.313726,
0.337255,
0.356863,
0.372549,
0.384314,
0.396078,
0.392157,
0.380392,
0.372549,
0.356863,
0.345098,
0.333333,
0.333333,
0.333333,
```

```
0.329412,
0.313726,
0.301961,
0.274510,
0.239216,
0.207843,
0.188235,
0.164706,
0.141176,
0.113725,
0.090196,
0.074510,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.047059,
0.043137,
0.047059,
0.050980,
0.047059,
0.043137,
0.039216,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
0.078431,
0.078431,
0.082353,
0.082353,
0.082353,
0.078431,
0.074510,
0.070588,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
```

```
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.050980,
0.047059,
0.043137,
0.047059,
0.047059,
0.047059,
0.050980,
0.050980,
0.047059,
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.043137,
0.039216,
0.035294,
0.035294,
0.043137,
0.047059,
0.043137,
0.035294,
0.035294,
0.047059,
0.054902,
0.058824,
0.066667,
0.074510,
0.074510,
0.074510,
0.086275,
0.098039,
0.090196,
0.082353,
0.078431,
0.078431,
0.094118,
0.113725,
0.133333,
0.149020,
0.160784,
0.180392,
0.192157,
0.180392,
0.184314,
0.227451,
0.270588,
0.290196,
0.309804,
0.333333,
0.349020,
0.368627,
0.396078,
0.403922,
0.392157,
```

```
0.380392,
0.372549,
0.364706,
0.352941,
0.349020,
0.345098,
0.341176,
0.329412,
0.321569,
0.309804,
0.286275,
0.258824,
0.219608,
0.176471,
0.137255,
0.113725,
0.094118,
0.074510,
0.070588,
0.070588,
0.062745,
0.058824,
0.054902,
0.050980,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.054902,
0.050980,
0.047059,
0.039216,
0.043137,
0.050980,
0.050980,
0.047059,
0.047059,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
0.074510,
0.078431,
0.082353,
0.082353,
0.082353,
0.078431,
```

```
0.078431,
0.070588,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.043137,
0.039216,
0.039216,
0.039216,
0.043137,
0.047059,
0.047059,
0.043137,
0.039216,
0.035294,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.035294,
0.035294,
0.039216,
0.043137,
0.039216,
0.039216,
0.047059,
0.050980,
0.050980,
0.058824,
0.070588,
0.082353,
0.086275,
0.082353,
0.094118,
0.117647,
0.113725,
0.094118,
0.078431,
0.074510,
0.086275,
0.109804,
0.137255,
0.152941,
0.160784,
0.184314,
0.211765,
0.219608,
0.231373,
0.270588,
0.309804,
0.325490,
```

```
0.341176,
0.356863,
0.368627,
0.384314,
0.403922,
0.396078,
0.380392,
0.372549,
0.368627,
0.368627,
0.360784,
0.356863,
0.356863,
0.345098,
0.329412,
0.317647,
0.309804,
0.301961,
0.282353,
0.247059,
0.192157,
0.152941,
0.129412,
0.105882,
0.086275,
0.082353,
0.078431,
0.066667,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.043137,
0.043137,
0.043137,
0.050980,
0.050980,
0.050980,
0.043137,
0.043137,
0.043137,
0.047059,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
```

```
0.074510,
0.074510,
0.078431,
0.082353,
0.082353,
0.078431,
0.078431,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.047059,
0.047059,
0.050980,
0.047059,
0.043137,
0.035294,
0.035294,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.035294,
0.043137,
0.050980,
0.050980,
0.054902,
0.058824,
0.070588,
0.082353,
0.098039,
0.101961,
0.113725,
0.137255,
0.129412,
0.101961,
0.082353,
0.078431,
0.082353,
0.090196,
0.109804,
0.121569,
0.129412,
```

```
0.149020,
0.180392,
0.211765,
0.239216,
0.274510,
0.309804,
0.329412,
0.341176,
0.352941,
0.364706,
0.388235,
0.407843,
0.388235,
0.364706,
0.364706,
0.364706,
0.360784,
0.356863,
0.360784,
0.356863,
0.341176,
0.325490,
0.313726,
0.301961,
0.305882,
0.298039,
0.266667,
0.227451,
0.192157,
0.156863,
0.121569,
0.109804,
0.101961,
0.090196,
0.078431,
0.070588,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.047059,
0.047059,
0.050980,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.047059,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
```

```
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.074510,
0.078431,
0.082353,
0.082353,
0.082353,
0.082353,
0.078431,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.047059,
0.047059,
0.050980,
0.050980,
0.047059,
0.039216,
0.031373,
0.039216,
0.043137,
0.043137,
0.039216,
0.039216,
0.039216,
0.031373,
0.031373,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.039216,
0.043137,
0.043137,
0.047059,
0.058824,
0.074510,
0.086275,
0.105882,
0.121569,
0.145098,
0.164706,
0.145098,
0.109804,
```

```
0.090196,
0.086275,
0.094118,
0.098039,
0.098039,
0.101961,
0.113725,
0.137255,
0.164706,
0.192157,
0.219608,
0.258824,
0.294118,
0.305882,
0.305882,
0.317647,
0.345098,
0.380392,
0.392157,
0.368627,
0.352941,
0.352941,
0.349020,
0.345098,
0.352941,
0.364706,
0.352941,
0.333333,
0.317647,
0.301961,
0.286275,
0.290196,
0.294118,
0.286275,
0.254902,
0.215686,
0.172549,
0.137255,
0.125490,
0.113725,
0.098039,
0.086275,
0.074510,
0.062745,
0.054902,
0.050980,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.058824,
0.050980,
0.043137,
```

```
0.047059,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.078431,
0.078431,
0.082353,
0.082353,
0.082353,
0.082353,
0.082353,
0.078431,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.047059,
0.043137,
0.047059,
0.047059,
0.050980,
0.047059,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.039216,
0.031373,
0.027451,
0.031373,
0.035294,
0.031373,
0.027451,
0.031373,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.043137,
0.050980,
0.058824,
0.070588,
```

```
0.090196,
0.105882,
0.133333,
0.180392,
0.200000,
0.172549,
0.129412,
0.105882,
0.109804,
0.117647,
0.121569,
0.117647,
0.121569,
0.137255,
0.164706,
0.192157,
0.215686,
0.231373,
0.258824,
0.282353,
0.290196,
0.290196,
0.313726,
0.352941,
0.384314,
0.372549,
0.345098,
0.341176,
0.345098,
0.333333,
0.325490,
0.341176,
0.349020,
0.341176,
0.325490,
0.301961,
0.282353,
0.266667,
0.266667,
0.274510,
0.274510,
0.258824,
0.227451,
0.188235,
0.156863,
0.129412,
0.109804,
0.094118,
0.082353,
0.074510,
0.062745,
0.047059,
0.043137,
0.047059,
0.047059,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
```

```
0.054902,
0.047059,
0.043137,
0.054902,
0.054902,
0.054902,
0.047059,
0.050980,
0.058824,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
0.074510,
0.078431,
0.082353,
0.086275,
0.082353,
0.078431,
0.078431,
0.078431,
0.070588,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.047059,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.043137,
0.035294,
0.031373,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
```

```
0.031373,
0.031373,
0.035294,
0.050980,
0.066667,
0.070588,
0.070588,
0.086275,
0.117647,
0.149020,
0.200000,
0.235294,
0.215686,
0.168627,
0.137255,
0.129412,
0.137255,
0.141176,
0.145098,
0.160784,
0.168627,
0.180392,
0.203922,
0.231373,
0.247059,
0.258824,
0.274510,
0.286275,
0.298039,
0.325490,
0.356863,
0.372549,
0.345098,
0.317647,
0.321569,
0.329412,
0.325490,
0.313726,
0.309804,
0.321569,
0.317647,
0.294118,
0.266667,
0.254902,
0.243137,
0.243137,
0.239216,
0.231373,
0.235294,
0.231373,
0.207843,
0.172549,
0.137255,
0.109804,
0.094118,
0.082353,
0.074510,
0.062745,
0.050980,
0.050980,
0.054902,
```

```
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.047059,
0.043137,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.082353,
0.086275,
0.086275,
0.082353,
0.082353,
0.078431,
0.074510,
0.070588,
0.070588,
0.066667,
0.066667,
0.062745,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.043137,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.035294,
0.031373,
0.027451,
0.031373,
0.031373,
0.031373,
0.031373,
0.027451,
```

```
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.043137,
0.058824,
0.078431,
0.090196,
0.086275,
0.086275,
0.121569,
0.168627,
0.215686,
0.258824,
0.262745,
0.223529,
0.184314,
0.152941,
0.152941,
0.160784,
0.164706,
0.192157,
0.207843,
0.207843,
0.211765,
0.223529,
0.243137,
0.262745,
0.274510,
0.286275,
0.301961,
0.321569,
0.337255,
0.341176,
0.321569,
0.298039,
0.298039,
0.305882,
0.305882,
0.286275,
0.270588,
0.278431,
0.278431,
0.243137,
0.211765,
0.203922,
0.203922,
0.207843,
0.192157,
0.188235,
0.215686,
0.223529,
0.203922,
0.176471,
0.145098,
0.117647,
```

```
0.098039,
0.086275,
0.078431,
0.066667,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.050980,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.074510,
0.078431,
0.078431,
0.086275,
0.086275,
0.086275,
0.086275,
0.082353,
0.078431,
0.074510,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.054902,
0.047059,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
0.039216,
0.039216,
0.043137,
0.039216,
0.035294,
```

```
0.031373,
0.027451,
0.027451,
0.031373,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.039216,
0.043137,
0.050980,
0.066667,
0.086275,
0.109804,
0.117647,
0.101961,
0.113725,
0.156863,
0.203922,
0.250980,
0.294118,
0.298039,
0.254902,
0.200000,
0.188235,
0.188235,
0.188235,
0.215686,
0.239216,
0.239216,
0.235294,
0.231373,
0.247059,
0.266667,
0.274510,
0.282353,
0.290196,
0.290196,
0.290196,
0.298039,
0.294118,
0.274510,
0.270588,
0.270588,
0.258824,
0.239216,
0.227451,
0.231373,
0.243137,
0.223529,
0.176471,
0.149020,
0.168627,
0.172549,
0.145098,
```

```
0.164706,
0.207843,
0.203922,
0.184314,
0.172549,
0.149020,
0.121569,
0.101961,
0.086275,
0.074510,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.047059,
0.047059,
0.054902,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.074510,
0.074510,
0.074510,
0.074510,
0.078431,
0.078431,
0.078431,
0.082353,
0.082353,
0.086275,
0.086275,
0.086275,
0.086275,
0.086275,
0.082353,
0.074510,
0.070588,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.047059,
0.039216,
0.035294,
0.043137,
0.043137,
```

```
0.039216,
0.039216,
0.039216,
0.035294,
0.039216,
0.035294,
0.035294,
0.031373,
0.031373,
0.027451,
0.027451,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.023529,
0.027451,
0.027451,
0.027451,
0.035294,
0.043137,
0.054902,
0.066667,
0.078431,
0.098039,
0.137255,
0.156863,
0.152941,
0.133333,
0.149020,
0.180392,
0.219608,
0.298039,
0.360784,
0.317647,
0.250980,
0.235294,
0.223529,
0.207843,
0.239216,
0.258824,
0.254902,
0.250980,
0.247059,
0.250980,
0.262745,
0.266667,
0.270588,
0.270588,
0.247059,
0.235294,
0.258824,
0.266667,
0.250980,
0.231373,
0.211765,
0.188235,
0.184314,
0.192157,
0.188235,
```

```
0.207843,
0.219608,
0.176471,
0.133333,
0.145098,
0.145098,
0.117647,
0.145098,
0.192157,
0.180392,
0.156863,
0.149020,
0.145098,
0.125490,
0.101961,
0.086275,
0.074510,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.050980,
0.047059,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
0.078431,
0.078431,
0.082353,
0.082353,
0.086275,
0.082353,
0.082353,
0.086275,
0.086275,
0.082353,
0.082353,
0.078431,
0.074510,
0.070588,
0.066667,
0.062745,
```

```
0.058824,
0.054902,
0.047059,
0.039216,
0.031373,
0.035294,
0.039216,
0.039216,
0.035294,
0.039216,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.035294,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.047059,
0.062745,
0.090196,
0.098039,
0.109804,
0.156863,
0.192157,
0.203922,
0.207843,
0.207843,
0.219608,
0.243137,
0.309804,
0.388235,
0.360784,
0.305882,
0.294118,
0.262745,
0.235294,
0.270588,
0.290196,
0.282353,
0.270588,
0.258824,
0.247059,
0.250980,
0.254902,
0.254902,
0.243137,
0.207843,
0.207843,
0.239216,
0.239216,
```

```
0.215686,
0.192157,
0.160784,
0.133333,
0.137255,
0.160784,
0.152941,
0.168627,
0.196078,
0.176471,
0.141176,
0.121569,
0.113725,
0.105882,
0.129412,
0.160784,
0.152941,
0.129412,
0.117647,
0.125490,
0.117647,
0.105882,
0.090196,
0.078431,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.058824,
0.054902,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.066667,
0.074510,
0.078431,
0.078431,
0.078431,
0.078431,
0.082353,
0.086275,
0.082353,
0.082353,
0.082353,
0.086275,
```

```
0.082353,
0.078431,
0.078431,
0.078431,
0.070588,
0.066667,
0.062745,
0.058824,
0.054902,
0.050980,
0.047059,
0.039216,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.027451,
0.027451,
0.031373,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.047059,
0.074510,
0.105882,
0.125490,
0.129412,
0.164706,
0.227451,
0.274510,
0.301961,
0.305882,
0.301961,
0.317647,
0.364706,
0.423529,
0.411765,
0.372549,
0.345098,
0.294118,
0.262745,
0.301961,
0.329412,
0.309804,
0.286275,
0.262745,
0.239216,
0.227451,
```

```
0.231373,
0.227451,
0.207843,
0.180392,
0.196078,
0.227451,
0.215686,
0.188235,
0.164706,
0.145098,
0.113725,
0.113725,
0.133333,
0.125490,
0.129412,
0.149020,
0.152941,
0.137255,
0.101961,
0.086275,
0.105882,
0.121569,
0.133333,
0.125490,
0.105882,
0.094118,
0.094118,
0.098039,
0.094118,
0.090196,
0.086275,
0.078431,
0.074510,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.047059,
0.047059,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.074510,
0.078431,
0.082353,
0.078431,
```

```
0.078431,
0.082353,
0.082353,
0.082353,
0.078431,
0.082353,
0.082353,
0.082353,
0.078431,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.043137,
0.035294,
0.027451,
0.027451,
0.031373,
0.035294,
0.031373,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.035294,
0.050980,
0.090196,
0.137255,
0.168627,
0.184314,
0.203922,
0.278431,
0.368627,
0.392157,
0.392157,
0.388235,
0.388235,
0.443137,
0.501961,
0.486275,
0.439216,
0.384314,
0.317647,
0.294118,
```

```
0.329412,
0.337255,
0.309804,
0.286275,
0.262745,
0.227451,
0.200000,
0.188235,
0.196078,
0.188235,
0.160784,
0.176471,
0.203922,
0.196078,
0.168627,
0.137255,
0.117647,
0.101961,
0.101961,
0.121569,
0.105882,
0.098039,
0.113725,
0.121569,
0.121569,
0.094118,
0.074510,
0.098039,
0.109804,
0.105882,
0.101961,
0.086275,
0.078431,
0.078431,
0.078431,
0.082353,
0.082353,
0.082353,
0.082353,
0.078431,
0.074510,
0.070588,
0.066667,
0.062745,
0.058824,
0.050980,
0.047059,
0.054902,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.066667,
```

```
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.082353,
0.082353,
0.078431,
0.078431,
0.086275,
0.086275,
0.082353,
0.078431,
0.078431,
0.082353,
0.078431,
0.074510,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.050980,
0.043137,
0.039216,
0.035294,
0.027451,
0.027451,
0.027451,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.023529,
0.023529,
0.027451,
0.027451,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.039216,
0.047059,
0.062745,
0.094118,
0.160784,
0.223529,
0.258824,
0.282353,
0.337255,
0.419608,
0.462745,
0.466667,
0.462745,
0.462745,
```

```
0.517647,
0.576471,
0.549020,
0.486275,
0.415686,
0.364706,
0.364706,
0.364706,
0.333333,
0.294118,
0.270588,
0.250980,
0.215686,
0.172549,
0.149020,
0.164706,
0.168627,
0.145098,
0.145098,
0.164706,
0.168627,
0.152941,
0.117647,
0.090196,
0.078431,
0.094118,
0.113725,
0.101961,
0.082353,
0.090196,
0.098039,
0.101961,
0.086275,
0.062745,
0.078431,
0.094118,
0.086275,
0.078431,
0.074510,
0.074510,
0.074510,
0.070588,
0.066667,
0.070588,
0.074510,
0.074510,
0.074510,
0.074510,
0.066667,
0.062745,
0.058824,
0.054902,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
```

```
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.078431,
0.082353,
0.078431,
0.082353,
0.086275,
0.086275,
0.082353,
0.078431,
0.070588,
0.074510,
0.074510,
0.074510,
0.066667,
0.054902,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.035294,
0.031373,
0.031373,
0.027451,
0.031373,
0.031373,
0.027451,
0.027451,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.023529,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.035294,
0.039216,
0.047059,
0.050980,
0.058824,
0.086275,
0.105882,
0.152941,
0.239216,
0.321569,
```

```
0.372549,
0.411765,
0.454902,
0.505882,
0.541176,
0.537255,
0.541176,
0.592157,
0.631373,
0.596078,
0.517647,
0.458824,
0.450980,
0.443137,
0.400000,
0.345098,
0.298039,
0.262745,
0.239216,
0.207843,
0.156863,
0.121569,
0.129412,
0.141176,
0.117647,
0.105882,
0.121569,
0.133333,
0.133333,
0.113725,
0.086275,
0.070588,
0.086275,
0.101961,
0.094118,
0.074510,
0.074510,
0.082353,
0.086275,
0.074510,
0.054902,
0.062745,
0.078431,
0.070588,
0.062745,
0.066667,
0.070588,
0.074510,
0.062745,
0.054902,
0.058824,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.058824,
0.058824,
0.058824,
0.058824,
```

```
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.078431,
0.082353,
0.082353,
0.082353,
0.086275,
0.086275,
0.086275,
0.078431,
0.070588,
0.066667,
0.070588,
0.070588,
0.062745,
0.054902,
0.047059,
0.043137,
0.039216,
0.039216,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.035294,
0.031373,
0.027451,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.031373,
0.031373,
0.035294,
0.039216,
0.047059,
0.062745,
```

```
0.062745,
0.066667,
0.101961,
0.145098,
0.176471,
0.235294,
0.337255,
0.443137,
0.513726,
0.537255,
0.560784,
0.603922,
0.623529,
0.627451,
0.654902,
0.670588,
0.623529,
0.560784,
0.541176,
0.533333,
0.490196,
0.423529,
0.364706,
0.309804,
0.254902,
0.211765,
0.180392,
0.145098,
0.113725,
0.105882,
0.105882,
0.090196,
0.082353,
0.094118,
0.105882,
0.113725,
0.101961,
0.082353,
0.066667,
0.074510,
0.086275,
0.082353,
0.066667,
0.070588,
0.074510,
0.074510,
0.062745,
0.050980,
0.054902,
0.066667,
0.058824,
0.050980,
0.058824,
0.062745,
0.058824,
0.047059,
0.039216,
0.050980,
0.062745,
0.066667,
```

```
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.054902,
0.050980,
0.047059,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.078431,
0.082353,
0.082353,
0.082353,
0.082353,
0.086275,
0.086275,
0.082353,
0.078431,
0.070588,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.047059,
0.047059,
0.043137,
0.035294,
0.031373,
0.031373,
0.031373,
0.035294,
0.035294,
0.031373,
0.027451,
0.023529,
0.023529,
0.023529,
0.027451,
0.031373,
0.027451,
0.027451,
0.027451,
0.031373,
```

```
0.035294,
0.039216,
0.035294,
0.039216,
0.047059,
0.066667,
0.090196,
0.086275,
0.074510,
0.109804,
0.176471,
0.247059,
0.294118,
0.341176,
0.447059,
0.568627,
0.627451,
0.631373,
0.658824,
0.698039,
0.705882,
0.709804,
0.701961,
0.658824,
0.631373,
0.619608,
0.580392,
0.513726,
0.443137,
0.376471,
0.301961,
0.239216,
0.192157,
0.156863,
0.145098,
0.137255,
0.121569,
0.098039,
0.078431,
0.070588,
0.074510,
0.086275,
0.094118,
0.082353,
0.066667,
0.062745,
0.066667,
0.074510,
0.070588,
0.062745,
0.062745,
0.066667,
0.066667,
0.054902,
0.047059,
0.054902,
0.066667,
0.058824,
0.050980,
0.050980,
```

```
0.047059,
0.043137,
0.043137,
0.043137,
0.050980,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.047059,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.082353,
0.086275,
0.082353,
0.078431,
0.078431,
0.086275,
0.086275,
0.082353,
0.078431,
0.070588,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.050980,
0.047059,
0.043137,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.027451,
0.023529,
0.023529,
0.023529,
```

```
0.023529,
0.027451,
0.031373,
0.031373,
0.027451,
0.031373,
0.031373,
0.043137,
0.054902,
0.054902,
0.054902,
0.066667,
0.090196,
0.121569,
0.121569,
0.101961,
0.133333,
0.203922,
0.286275,
0.372549,
0.415686,
0.439216,
0.525490,
0.627451,
0.682353,
0.713726,
0.741176,
0.760784,
0.760784,
0.741176,
0.709804,
0.682353,
0.643137,
0.584314,
0.509804,
0.439216,
0.364706,
0.290196,
0.239216,
0.203922,
0.176471,
0.168627,
0.160784,
0.141176,
0.113725,
0.094118,
0.074510,
0.070588,
0.078431,
0.078431,
0.066667,
0.058824,
0.058824,
0.054902,
0.058824,
0.058824,
0.054902,
0.058824,
0.062745,
0.054902,
```

```
0.047059,
0.047059,
0.054902,
0.062745,
0.058824,
0.047059,
0.043137,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.074510,
0.070588,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.050980,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.078431,
0.082353,
0.082353,
0.078431,
0.074510,
0.074510,
0.078431,
0.078431,
0.078431,
0.074510,
0.066667,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.047059,
0.043137,
0.039216,
```

```
0.035294,
0.031373,
0.027451,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.027451,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.043137,
0.062745,
0.078431,
0.086275,
0.098039,
0.121569,
0.160784,
0.160784,
0.152941,
0.211765,
0.290196,
0.349020,
0.419608,
0.494118,
0.517647,
0.509804,
0.580392,
0.701961,
0.772549,
0.784314,
0.796078,
0.796078,
0.780392,
0.756863,
0.709804,
0.647059,
0.576471,
0.501961,
0.427451,
0.356863,
0.301961,
0.262745,
0.227451,
0.207843,
0.192157,
0.164706,
0.137255,
0.117647,
0.101961,
0.082353,
0.070588,
0.070588,
0.066667,
0.058824,
0.050980,
0.050980,
```

```
0.043137,
0.043137,
0.047059,
0.054902,
0.054902,
0.050980,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.050980,
0.043137,
0.047059,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.074510,
0.070588,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.050980,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.078431,
0.082353,
0.082353,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.070588,
0.062745,
0.062745,
```

```
0.062745,
0.058824,
0.050980,
0.043137,
0.039216,
0.039216,
0.039216,
0.035294,
0.031373,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.027451,
0.031373,
0.035294,
0.039216,
0.043137,
0.043137,
0.047059,
0.058824,
0.086275,
0.125490,
0.145098,
0.172549,
0.215686,
0.219608,
0.215686,
0.305882,
0.419608,
0.490196,
0.505882,
0.545098,
0.619608,
0.603922,
0.592157,
0.713726,
0.807843,
0.823529,
0.823529,
0.815686,
0.803922,
0.780392,
0.733333,
0.662745,
0.596078,
0.533333,
0.458824,
0.388235,
0.341176,
0.290196,
0.247059,
0.231373,
0.207843,
0.172549,
0.137255,
0.109804,
0.090196,
```

```
0.078431,
0.062745,
0.054902,
0.058824,
0.054902,
0.047059,
0.039216,
0.035294,
0.035294,
0.043137,
0.050980,
0.047059,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.047059,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.054902,
0.050980,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.082353,
0.086275,
0.082353,
0.074510,
```

```
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.058824,
0.054902,
0.050980,
0.047059,
0.039216,
0.035294,
0.031373,
0.035294,
0.035294,
0.031373,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.031373,
0.035294,
0.043137,
0.050980,
0.058824,
0.066667,
0.074510,
0.094118,
0.145098,
0.203922,
0.247059,
0.298039,
0.305882,
0.298039,
0.380392,
0.505882,
0.600000,
0.619608,
0.627451,
0.682353,
0.674510,
0.658824,
0.752941,
0.831373,
0.847059,
0.843137,
0.827451,
0.796078,
0.764706,
0.737255,
0.682353,
0.611765,
0.560784,
0.505882,
0.450980,
0.388235,
0.321569,
```

```
0.278431,
0.247059,
0.211765,
0.176471,
0.145098,
0.109804,
0.082353,
0.066667,
0.054902,
0.043137,
0.047059,
0.047059,
0.043137,
0.031373,
0.031373,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
```

```
0.070588,
0.074510,
0.074510,
0.078431,
0.086275,
0.086275,
0.078431,
0.070588,
0.070588,
0.062745,
0.054902,
0.054902,
0.050980,
0.047059,
0.039216,
0.039216,
0.039216,
0.035294,
0.031373,
0.027451,
0.031373,
0.031373,
0.027451,
0.027451,
0.023529,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.047059,
0.054902,
0.074510,
0.098039,
0.117647,
0.133333,
0.160784,
0.219608,
0.274510,
0.345098,
0.388235,
0.400000,
0.470588,
0.576471,
0.650980,
0.690196,
0.713726,
0.725490,
0.713726,
0.737255,
0.807843,
0.854902,
0.862745,
0.854902,
0.827451,
0.784314,
0.741176,
0.721569,
```

```
0.694118,
0.635294,
0.580392,
0.537255,
0.494118,
0.431373,
0.360784,
0.298039,
0.250980,
0.211765,
0.180392,
0.145098,
0.113725,
0.090196,
0.066667,
0.047059,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.039216,
0.047059,
0.043137,
0.039216,
0.039216,
0.039216,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
```

```
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.078431,
0.082353,
0.082353,
0.078431,
0.074510,
0.066667,
0.054902,
0.047059,
0.043137,
0.035294,
0.031373,
0.031373,
0.035294,
0.031373,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.027451,
0.031373,
0.031373,
0.031373,
0.039216,
0.050980,
0.070588,
0.098039,
0.129412,
0.164706,
0.192157,
0.207843,
0.227451,
0.274510,
0.352941,
0.427451,
0.474510,
0.560784,
0.654902,
0.709804,
0.733333,
0.768627,
0.792157,
0.796078,
0.815686,
0.831373,
```

```
0.858824,
0.874510,
0.854902,
0.823529,
0.788235,
0.749020,
0.701961,
0.682353,
0.670588,
0.627451,
0.552941,
0.486275,
0.439216,
0.384314,
0.317647,
0.258824,
0.223529,
0.188235,
0.145098,
0.113725,
0.098039,
0.074510,
0.054902,
0.043137,
0.039216,
0.031373,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.039216,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.043137,
0.050980,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
```

```
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.066667,
0.066667,
0.066667,
0.074510,
0.074510,
0.070588,
0.070588,
0.078431,
0.078431,
0.078431,
0.070588,
0.066667,
0.062745,
0.058824,
0.054902,
0.047059,
0.039216,
0.039216,
0.035294,
0.035294,
0.031373,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.027451,
0.031373,
0.035294,
0.039216,
0.039216,
0.054902,
0.082353,
0.121569,
0.156863,
0.200000,
0.250980,
0.278431,
0.301961,
0.364706,
0.431373,
0.466667,
0.521569,
0.635294,
0.741176,
```

```
0.780392,
0.780392,
0.800000,
0.839216,
0.862745,
0.862745,
0.854902,
0.866667,
0.878431,
0.854902,
0.827451,
0.803922,
0.764706,
0.709804,
0.658824,
0.627451,
0.588235,
0.513726,
0.454902,
0.423529,
0.384314,
0.309804,
0.235294,
0.203922,
0.176471,
0.129412,
0.098039,
0.086275,
0.070588,
0.058824,
0.050980,
0.043137,
0.035294,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.039216,
0.043137,
0.043137,
0.039216,
0.039216,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
```

```
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.074510,
0.074510,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.054902,
0.050980,
0.047059,
0.043137,
0.039216,
0.035294,
0.031373,
0.027451,
0.027451,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.039216,
0.047059,
0.050980,
0.058824,
0.082353,
0.129412,
0.180392,
0.227451,
0.294118,
0.349020,
```

```
0.384314,
0.466667,
0.552941,
0.588235,
0.623529,
0.705882,
0.792157,
0.835294,
0.835294,
0.827451,
0.847059,
0.878431,
0.890196,
0.890196,
0.886275,
0.882353,
0.862745,
0.843137,
0.815686,
0.780392,
0.733333,
0.662745,
0.580392,
0.505882,
0.419608,
0.368627,
0.368627,
0.337255,
0.266667,
0.203922,
0.160784,
0.133333,
0.094118,
0.066667,
0.062745,
0.058824,
0.050980,
0.043137,
0.039216,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
0.035294,
0.035294,
0.039216,
0.047059,
0.047059,
0.047059,
0.050980,
0.054902,
0.054902,
0.050980,
0.050980,
0.054902,
0.062745,
0.062745,
```

```
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.070588,
0.074510,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.066667,
0.058824,
0.054902,
0.058824,
0.058824,
0.050980,
0.047059,
0.043137,
0.039216,
0.035294,
0.031373,
0.031373,
0.027451,
0.027451,
0.027451,
0.031373,
0.031373,
0.031373,
0.035294,
0.039216,
0.047059,
0.054902,
0.062745,
```

```
0.062745,
0.074510,
0.109804,
0.168627,
0.219608,
0.294118,
0.376471,
0.435294,
0.517647,
0.627451,
0.709804,
0.745098,
0.764706,
0.815686,
0.870588,
0.882353,
0.874510,
0.870588,
0.890196,
0.909804,
0.909804,
0.901961,
0.886275,
0.874510,
0.854902,
0.831373,
0.803922,
0.764706,
0.709804,
0.635294,
0.529412,
0.411765,
0.333333,
0.313726,
0.290196,
0.243137,
0.180392,
0.121569,
0.082353,
0.058824,
0.047059,
0.047059,
0.047059,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.031373,
0.035294,
0.043137,
0.047059,
0.047059,
0.047059,
0.050980,
```

```
0.050980,
0.050980,
0.047059,
0.047059,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.062745,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.047059,
0.043137,
0.039216,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
```

```
0.035294,
0.043137,
0.047059,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.070588,
0.090196,
0.137255,
0.207843,
0.294118,
0.403922,
0.494118,
0.572549,
0.654902,
0.745098,
0.803922,
0.819608,
0.854902,
0.894118,
0.913725,
0.913725,
0.905882,
0.905882,
0.917647,
0.921569,
0.913725,
0.901961,
0.890196,
0.874510,
0.858824,
0.839216,
0.807843,
0.760784,
0.690196,
0.592157,
0.486275,
0.400000,
0.333333,
0.262745,
0.184314,
0.117647,
0.082353,
0.054902,
0.039216,
0.031373,
0.035294,
0.035294,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.031373,
0.027451,
```

```
0.031373,
0.035294,
0.039216,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
0.047059,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.062745,
0.062745,
0.062745,
0.054902,
0.047059,
0.043137,
0.043137,
0.039216,
0.039216,
```

```
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.047059,
0.058824,
0.070588,
0.074510,
0.070588,
0.070588,
0.070588,
0.086275,
0.105882,
0.149020,
0.231373,
0.325490,
0.423529,
0.505882,
0.584314,
0.674510,
0.756863,
0.803922,
0.843137,
0.890196,
0.913725,
0.925490,
0.929412,
0.929412,
0.925490,
0.929412,
0.929412,
0.925490,
0.917647,
0.905882,
0.898039,
0.890196,
0.874510,
0.835294,
0.764706,
0.694118,
0.639216,
0.572549,
0.494118,
0.376471,
0.231373,
0.141176,
0.109804,
0.094118,
0.062745,
0.039216,
0.031373,
0.031373,
0.031373,
0.027451,
0.027451,
0.023529,
0.023529,
```

```
0.023529,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.035294,
0.039216,
0.047059,
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.066667,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.058824,
0.058824,
```

```
0.058824,
0.058824,
0.050980,
0.050980,
0.047059,
0.047059,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.043137,
0.054902,
0.066667,
0.086275,
0.098039,
0.105882,
0.105882,
0.098039,
0.105882,
0.137255,
0.172549,
0.243137,
0.325490,
0.384314,
0.443137,
0.521569,
0.623529,
0.705882,
0.768627,
0.839216,
0.890196,
0.917647,
0.929412,
0.937255,
0.941176,
0.941176,
0.941176,
0.941176,
0.937255,
0.925490,
0.917647,
0.905882,
0.890196,
0.862745,
0.815686,
0.721569,
0.643137,
0.615686,
0.572549,
0.501961,
0.388235,
0.254902,
0.200000,
0.176471,
0.121569,
0.074510,
0.058824,
```

```
0.047059,
0.035294,
0.035294,
0.035294,
0.031373,
0.027451,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.039216,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
```

```
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.054902,
0.050980,
0.047059,
0.043137,
0.039216,
0.031373,
0.031373,
0.031373,
0.027451,
0.027451,
0.031373,
0.047059,
0.062745,
0.074510,
0.101961,
0.125490,
0.152941,
0.168627,
0.152941,
0.145098,
0.176471,
0.215686,
0.266667,
0.333333,
0.388235,
0.450980,
0.517647,
0.572549,
0.647059,
0.749020,
0.839216,
0.882353,
0.909804,
0.929412,
0.941176,
0.949020,
0.949020,
0.949020,
0.949020,
0.941176,
0.929412,
0.913725,
0.890196,
0.866667,
0.835294,
0.788235,
0.709804,
0.635294,
0.572549,
0.494118,
0.407843,
```

```
0.345098,
0.305882,
0.266667,
0.200000,
0.129412,
0.098039,
0.090196,
0.070588,
0.047059,
0.047059,
0.047059,
0.039216,
0.031373,
0.027451,
0.027451,
0.027451,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.050980,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
```

```
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.043137,
0.039216,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.047059,
0.062745,
0.082353,
0.113725,
0.145098,
0.176471,
0.211765,
0.223529,
0.211765,
0.231373,
0.274510,
0.309804,
0.360784,
0.443137,
0.537255,
0.619608,
0.654902,
0.694118,
0.780392,
0.858824,
0.894118,
0.917647,
0.929412,
0.945098,
0.952941,
0.956863,
0.956863,
0.949020,
0.941176,
0.929412,
0.909804,
0.894118,
0.882353,
```

```
0.862745,
0.827451,
0.784314,
0.725490,
0.647059,
0.541176,
0.415686,
0.317647,
0.286275,
0.250980,
0.200000,
0.172549,
0.149020,
0.117647,
0.082353,
0.058824,
0.054902,
0.054902,
0.043137,
0.039216,
0.035294,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.043137,
0.043137,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.054902,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
```

```
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.031373,
0.027451,
0.023529,
0.027451,
0.023529,
0.027451,
0.039216,
0.054902,
0.074510,
0.098039,
0.117647,
0.145098,
0.203922,
0.274510,
0.290196,
0.298039,
0.333333,
0.380392,
0.431373,
0.498039,
0.576471,
0.666667,
0.741176,
0.780392,
0.835294,
0.886275,
0.913725,
0.925490,
0.937255,
0.949020,
0.960784,
0.960784,
```

```
0.960784,
0.956863,
0.952941,
0.945098,
0.933333,
0.921569,
0.909804,
0.894118,
0.870588,
0.835294,
0.780392,
0.709804,
0.639216,
0.525490,
0.388235,
0.305882,
0.274510,
0.266667,
0.247059,
0.192157,
0.145098,
0.113725,
0.086275,
0.070588,
0.058824,
0.050980,
0.043137,
0.039216,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
```

```
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.070588,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.050980,
0.043137,
0.035294,
0.031373,
0.031373,
0.027451,
0.023529,
0.023529,
0.027451,
0.043137,
0.058824,
0.070588,
0.086275,
0.121569,
0.200000,
0.301961,
0.368627,
0.376471,
0.400000,
0.450980,
0.509804,
0.560784,
0.607843,
0.662745,
0.745098,
0.819608,
0.866667,
```

```
0.882353,
0.901961,
0.921569,
0.941176,
0.956863,
0.964706,
0.964706,
0.964706,
0.960784,
0.960784,
0.956863,
0.949020,
0.937255,
0.921569,
0.905882,
0.878431,
0.847059,
0.792157,
0.725490,
0.666667,
0.611765,
0.541176,
0.470588,
0.415686,
0.376471,
0.309804,
0.227451,
0.180392,
0.152941,
0.113725,
0.086275,
0.066667,
0.050980,
0.043137,
0.039216,
0.039216,
0.035294,
0.035294,
0.031373,
0.035294,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
```

```
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.074510,
0.074510,
0.070588,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.054902,
0.047059,
0.043137,
0.043137,
0.039216,
0.031373,
0.023529,
0.023529,
0.027451,
0.031373,
0.043137,
0.062745,
0.090196,
0.129412,
0.207843,
0.313726,
0.403922,
0.458824,
0.486275,
0.509804,
```

```
0.564706,
0.631373,
0.686275,
0.721569,
0.756863,
0.811765,
0.843137,
0.858824,
0.894118,
0.925490,
0.945098,
0.960784,
0.968627,
0.968627,
0.968627,
0.968627,
0.964706,
0.956863,
0.941176,
0.913725,
0.901961,
0.901961,
0.886275,
0.843137,
0.784314,
0.733333,
0.698039,
0.674510,
0.647059,
0.592157,
0.513726,
0.427451,
0.321569,
0.227451,
0.180392,
0.149020,
0.121569,
0.090196,
0.066667,
0.050980,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
```

```
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.058824,
0.058824,
0.047059,
0.035294,
0.027451,
0.023529,
0.023529,
0.023529,
0.035294,
0.054902,
0.090196,
```

```
0.133333,
0.203922,
0.294118,
0.388235,
0.498039,
0.576471,
0.603922,
0.643137,
0.717647,
0.780392,
0.811765,
0.807843,
0.807843,
0.831373,
0.866667,
0.909804,
0.941176,
0.956863,
0.964706,
0.968627,
0.972549,
0.972549,
0.972549,
0.968627,
0.960784,
0.933333,
0.898039,
0.862745,
0.862745,
0.878431,
0.854902,
0.784314,
0.713726,
0.678431,
0.654902,
0.611765,
0.568627,
0.505882,
0.403922,
0.282353,
0.196078,
0.149020,
0.117647,
0.098039,
0.074510,
0.058824,
0.047059,
0.031373,
0.027451,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.047059,
```

```
0.043137,
0.039216,
0.039216,
0.043137,
0.047059,
0.050980,
0.054902,
0.050980,
0.054902,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.062745,
0.070588,
0.074510,
0.074510,
0.074510,
0.074510,
0.074510,
0.074510,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.066667,
0.062745,
0.050980,
0.035294,
```

```
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.043137,
0.070588,
0.113725,
0.176471,
0.235294,
0.313726,
0.450980,
0.584314,
0.647059,
0.686275,
0.760784,
0.831373,
0.862745,
0.862745,
0.850980,
0.862745,
0.901961,
0.933333,
0.952941,
0.960784,
0.964706,
0.968627,
0.972549,
0.972549,
0.968627,
0.968627,
0.960784,
0.945098,
0.921569,
0.866667,
0.815686,
0.815686,
0.831373,
0.807843,
0.709804,
0.600000,
0.556863,
0.521569,
0.482353,
0.450980,
0.380392,
0.278431,
0.211765,
0.168627,
0.121569,
0.078431,
0.054902,
0.047059,
0.039216,
0.027451,
0.023529,
0.027451,
0.027451,
0.027451,
0.031373,
```

```
0.035294,
0.035294,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.047059,
0.047059,
0.047059,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.074510,
0.074510,
0.078431,
0.078431,
0.074510,
0.078431,
0.078431,
0.074510,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
```

```
0.066667,
0.070588,
0.074510,
0.082353,
0.074510,
0.058824,
0.043137,
0.035294,
0.031373,
0.031373,
0.039216,
0.035294,
0.031373,
0.047059,
0.074510,
0.109804,
0.137255,
0.207843,
0.352941,
0.486275,
0.545098,
0.580392,
0.674510,
0.776471,
0.839216,
0.870588,
0.878431,
0.890196,
0.913725,
0.937255,
0.945098,
0.949020,
0.960784,
0.964706,
0.964706,
0.964706,
0.964706,
0.964706,
0.960784,
0.952941,
0.941176,
0.898039,
0.815686,
0.745098,
0.737255,
0.756863,
0.701961,
0.552941,
0.462745,
0.450980,
0.403922,
0.368627,
0.356863,
0.309804,
0.243137,
0.188235,
0.133333,
0.082353,
0.050980,
0.035294,
```

```
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.027451,
0.031373,
0.031373,
0.031373,
0.027451,
0.031373,
0.035294,
0.035294,
0.035294,
0.031373,
0.035294,
0.035294,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.047059,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.074510,
0.078431,
0.078431,
0.078431,
0.082353,
0.082353,
0.078431,
0.078431,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
```

```
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.066667,
0.074510,
0.082353,
0.090196,
0.098039,
0.094118,
0.070588,
0.054902,
0.047059,
0.039216,
0.043137,
0.058824,
0.047059,
0.035294,
0.031373,
0.047059,
0.070588,
0.082353,
0.137255,
0.274510,
0.364706,
0.388235,
0.454902,
0.572549,
0.705882,
0.788235,
0.827451,
0.862745,
0.894118,
0.901961,
0.901961,
0.917647,
0.937255,
0.949020,
0.952941,
0.956863,
0.956863,
0.960784,
0.956863,
0.952941,
0.949020,
0.937255,
0.913725,
0.854902,
0.741176,
0.639216,
0.647059,
0.678431,
0.564706,
0.411765,
0.372549,
0.325490,
0.266667,
0.274510,
```

```
0.274510,
0.231373,
0.164706,
0.109804,
0.074510,
0.047059,
0.031373,
0.023529,
0.023529,
0.027451,
0.027451,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.031373,
0.031373,
0.027451,
0.027451,
0.031373,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.043137,
0.047059,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.054902,
0.054902,
0.054902,
0.078431,
0.082353,
0.082353,
0.082353,
0.086275,
0.086275,
0.086275,
0.082353,
0.078431,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
```

```
0.066667,
0.058824,
0.058824,
0.058824,
0.062745,
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.066667,
0.078431,
0.090196,
0.101961,
0.113725,
0.109804,
0.090196,
0.074510,
0.066667,
0.054902,
0.062745,
0.082353,
0.074510,
0.050980,
0.039216,
0.050980,
0.086275,
0.113725,
0.145098,
0.219608,
0.282353,
0.329412,
0.419608,
0.545098,
0.666667,
0.729412,
0.764706,
0.827451,
0.858824,
0.850980,
0.854902,
0.894118,
0.925490,
0.929412,
0.929412,
0.937255,
0.945098,
0.952941,
0.945098,
0.937255,
0.937255,
0.933333,
0.917647,
0.874510,
0.745098,
0.580392,
0.541176,
```

```
0.592157,
0.545098,
0.411765,
0.317647,
0.262745,
0.203922,
0.180392,
0.184314,
0.168627,
0.113725,
0.066667,
0.050980,
0.039216,
0.031373,
0.027451,
0.027451,
0.031373,
0.031373,
0.027451,
0.023529,
0.023529,
0.027451,
0.027451,
0.031373,
0.031373,
0.027451,
0.027451,
0.027451,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.050980,
0.050980,
0.047059,
0.078431,
0.082353,
0.086275,
0.086275,
0.086275,
0.090196,
0.086275,
0.082353,
```

```
0.074510,
0.070588,
0.074510,
0.078431,
0.074510,
0.074510,
0.070588,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.054902,
0.054902,
0.050980,
0.054902,
0.058824,
0.062745,
0.066667,
0.070588,
0.078431,
0.090196,
0.105882,
0.121569,
0.125490,
0.113725,
0.101961,
0.086275,
0.074510,
0.090196,
0.105882,
0.101961,
0.078431,
0.058824,
0.054902,
0.086275,
0.137255,
0.180392,
0.231373,
0.278431,
0.313726,
0.380392,
0.501961,
0.600000,
0.654902,
0.721569,
0.776471,
0.776471,
0.780392,
0.827451,
0.874510,
0.901961,
0.905882,
0.901961,
0.917647,
0.937255,
0.945098,
0.937255,
0.925490,
```

```
0.925490,
0.921569,
0.909804,
0.866667,
0.737255,
0.588235,
0.529412,
0.525490,
0.509804,
0.466667,
0.400000,
0.341176,
0.270588,
0.203922,
0.164706,
0.129412,
0.086275,
0.062745,
0.054902,
0.047059,
0.035294,
0.027451,
0.031373,
0.031373,
0.031373,
0.027451,
0.023529,
0.023529,
0.027451,
0.031373,
0.035294,
0.031373,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.035294,
0.035294,
0.039216,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.074510,
```

```
0.078431,
0.082353,
0.086275,
0.090196,
0.094118,
0.090196,
0.086275,
0.078431,
0.074510,
0.078431,
0.082353,
0.082353,
0.078431,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.066667,
0.074510,
0.074510,
0.074510,
0.082353,
0.101961,
0.121569,
0.133333,
0.137255,
0.125490,
0.105882,
0.098039,
0.121569,
0.129412,
0.117647,
0.105882,
0.098039,
0.078431,
0.070588,
0.105882,
0.172549,
0.227451,
0.258824,
0.270588,
0.325490,
0.431373,
0.521569,
0.603922,
0.666667,
0.674510,
0.686275,
0.745098,
0.811765,
0.850980,
0.874510,
```

```
0.878431,
0.878431,
0.901961,
0.929412,
0.933333,
0.925490,
0.901961,
0.878431,
0.874510,
0.862745,
0.819608,
0.737255,
0.670588,
0.635294,
0.584314,
0.541176,
0.525490,
0.486275,
0.447059,
0.380392,
0.294118,
0.231373,
0.168627,
0.117647,
0.090196,
0.070588,
0.054902,
0.039216,
0.031373,
0.027451,
0.031373,
0.031373,
0.027451,
0.023529,
0.023529,
0.027451,
0.027451,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.031373,
0.039216,
0.043137,
0.043137,
0.039216,
0.039216,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
```

```
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.074510,
0.074510,
0.078431,
0.082353,
0.090196,
0.094118,
0.094118,
0.094118,
0.086275,
0.074510,
0.082353,
0.086275,
0.086275,
0.078431,
0.070588,
0.074510,
0.078431,
0.074510,
0.066667,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.058824,
0.066667,
0.078431,
0.082353,
0.082353,
0.090196,
0.105882,
0.117647,
0.137255,
0.152941,
0.145098,
0.133333,
0.141176,
0.156863,
0.149020,
0.133333,
0.133333,
0.145098,
0.125490,
0.078431,
0.078431,
0.121569,
0.172549,
0.215686,
0.262745,
0.313726,
0.384314,
0.466667,
0.533333,
```

```
0.556863,
0.572549,
0.643137,
0.717647,
0.760784,
0.792157,
0.823529,
0.839216,
0.854902,
0.894118,
0.917647,
0.925490,
0.909804,
0.870588,
0.839216,
0.831373,
0.823529,
0.807843,
0.792157,
0.760784,
0.709804,
0.647059,
0.596078,
0.533333,
0.447059,
0.403922,
0.368627,
0.313726,
0.243137,
0.180392,
0.141176,
0.109804,
0.074510,
0.054902,
0.039216,
0.031373,
0.031373,
0.031373,
0.031373,
0.027451,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.031373,
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.050980,
0.050980,
0.047059,
0.043137,
0.047059,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
```

```
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.070588,
0.070588,
0.070588,
0.078431,
0.086275,
0.090196,
0.098039,
0.098039,
0.094118,
0.082353,
0.086275,
0.090196,
0.086275,
0.074510,
0.078431,
0.082353,
0.082353,
0.078431,
0.066667,
0.058824,
0.058824,
0.058824,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.078431,
0.094118,
0.101961,
0.105882,
0.113725,
0.121569,
0.137255,
0.160784,
0.168627,
0.172549,
0.184314,
0.188235,
0.176471,
0.156863,
0.152941,
0.180392,
0.168627,
0.117647,
0.101961,
0.101961,
```

```
0.117647,
0.176471,
0.239216,
0.286275,
0.349020,
0.403922,
0.407843,
0.427451,
0.509804,
0.611765,
0.666667,
0.698039,
0.737255,
0.780392,
0.811765,
0.850980,
0.890196,
0.909804,
0.921569,
0.917647,
0.886275,
0.862745,
0.870588,
0.874510,
0.862745,
0.850980,
0.807843,
0.752941,
0.709804,
0.650980,
0.556863,
0.447059,
0.352941,
0.286275,
0.235294,
0.168627,
0.117647,
0.101961,
0.082353,
0.058824,
0.050980,
0.043137,
0.039216,
0.035294,
0.031373,
0.031373,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.027451,
0.035294,
0.039216,
0.039216,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
```

```
0.050980,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.062745,
0.066667,
0.070588,
0.074510,
0.078431,
0.090196,
0.094118,
0.101961,
0.098039,
0.094118,
0.098039,
0.098039,
0.086275,
0.086275,
0.090196,
0.086275,
0.082353,
0.078431,
0.066667,
0.062745,
0.066667,
0.066667,
0.074510,
0.082353,
0.078431,
0.074510,
0.070588,
0.082353,
0.101961,
0.121569,
0.125490,
0.133333,
0.141176,
0.145098,
0.160784,
0.184314,
0.203922,
0.219608,
0.223529,
0.211765,
```

```
0.180392,
0.168627,
0.203922,
0.203922,
0.160784,
0.160784,
0.137255,
0.098039,
0.125490,
0.192157,
0.258824,
0.325490,
0.341176,
0.317647,
0.364706,
0.466667,
0.549020,
0.615686,
0.662745,
0.701961,
0.749020,
0.796078,
0.839216,
0.878431,
0.901961,
0.917647,
0.925490,
0.921569,
0.909804,
0.917647,
0.925490,
0.905882,
0.870588,
0.839216,
0.815686,
0.780392,
0.694118,
0.580392,
0.474510,
0.352941,
0.243137,
0.172549,
0.113725,
0.078431,
0.070588,
0.054902,
0.039216,
0.039216,
0.039216,
0.039216,
0.035294,
0.031373,
0.031373,
0.027451,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
```

```
0.035294,
0.039216,
0.043137,
0.039216,
0.039216,
0.043137,
0.047059,
0.050980,
0.047059,
0.043137,
0.043137,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.082353,
0.086275,
0.094118,
0.098039,
0.101961,
0.101961,
0.105882,
0.101961,
0.098039,
0.098039,
0.094118,
0.090196,
0.086275,
0.078431,
0.066667,
0.066667,
0.070588,
0.070588,
0.078431,
0.094118,
0.098039,
0.098039,
0.086275,
0.086275,
0.105882,
0.133333,
0.145098,
0.156863,
0.168627,
```

```
0.164706,
0.164706,
0.192157,
0.227451,
0.250980,
0.254902,
0.239216,
0.207843,
0.192157,
0.223529,
0.227451,
0.200000,
0.200000,
0.168627,
0.109804,
0.125490,
0.192157,
0.243137,
0.282353,
0.278431,
0.270588,
0.337255,
0.411765,
0.466667,
0.541176,
0.607843,
0.666667,
0.729412,
0.780392,
0.823529,
0.862745,
0.898039,
0.917647,
0.929412,
0.933333,
0.937255,
0.937255,
0.929412,
0.909804,
0.886275,
0.874510,
0.862745,
0.807843,
0.717647,
0.600000,
0.478431,
0.380392,
0.309804,
0.243137,
0.180392,
0.137255,
0.101961,
0.066667,
0.047059,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.027451,
0.023529,
```

```
0.023529,
0.027451,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.035294,
0.039216,
0.043137,
0.047059,
0.047059,
0.050980,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.054902,
0.050980,
0.050980,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.070588,
0.074510,
0.070588,
0.074510,
0.078431,
0.082353,
0.086275,
0.094118,
0.098039,
0.105882,
0.113725,
0.113725,
0.109804,
0.105882,
0.098039,
0.101961,
0.098039,
0.086275,
0.078431,
0.070588,
0.078431,
0.078431,
0.082353,
0.098039,
0.113725,
0.121569,
```

```
0.113725,
0.101961,
0.109804,
0.129412,
0.152941,
0.172549,
0.188235,
0.192157,
0.192157,
0.211765,
0.247059,
0.274510,
0.282353,
0.266667,
0.247059,
0.231373,
0.231373,
0.243137,
0.235294,
0.219608,
0.180392,
0.152941,
0.180392,
0.200000,
0.211765,
0.239216,
0.254902,
0.278431,
0.337255,
0.400000,
0.458824,
0.513726,
0.580392,
0.686275,
0.772549,
0.815686,
0.850980,
0.882353,
0.909804,
0.929412,
0.937255,
0.937255,
0.941176,
0.945098,
0.937255,
0.917647,
0.913725,
0.905882,
0.882353,
0.831373,
0.772549,
0.670588,
0.549020,
0.474510,
0.419608,
0.349020,
0.266667,
0.196078,
0.145098,
0.098039,
0.066667,
```

```
0.043137,
0.027451,
0.023529,
0.027451,
0.027451,
0.023529,
0.023529,
0.023529,
0.027451,
0.027451,
0.031373,
0.035294,
0.039216,
0.043137,
0.043137,
0.047059,
0.043137,
0.039216,
0.039216,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.070588,
0.070588,
0.070588,
0.070588,
0.074510,
0.078431,
0.082353,
0.086275,
0.094118,
0.105882,
0.113725,
0.121569,
0.121569,
0.113725,
0.105882,
0.113725,
0.109804,
0.098039,
0.082353,
```

```
0.078431,
0.090196,
0.098039,
0.098039,
0.101961,
0.117647,
0.141176,
0.145098,
0.133333,
0.133333,
0.133333,
0.145098,
0.168627,
0.192157,
0.207843,
0.219608,
0.235294,
0.258824,
0.286275,
0.301961,
0.294118,
0.282353,
0.270588,
0.250980,
0.254902,
0.258824,
0.231373,
0.203922,
0.219608,
0.231373,
0.207843,
0.203922,
0.239216,
0.286275,
0.337255,
0.396078,
0.454902,
0.517647,
0.568627,
0.627451,
0.725490,
0.803922,
0.847059,
0.882353,
0.909804,
0.925490,
0.937255,
0.945098,
0.945098,
0.945098,
0.945098,
0.941176,
0.933333,
0.929412,
0.917647,
0.894118,
0.854902,
0.803922,
0.717647,
0.600000,
0.494118,
```

```
0.423529,
0.349020,
0.262745,
0.184314,
0.145098,
0.109804,
0.070588,
0.043137,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.023529,
0.027451,
0.027451,
0.031373,
0.031373,
0.035294,
0.039216,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.058824,
0.070588,
0.066667,
0.070588,
0.074510,
0.074510,
0.078431,
0.082353,
0.082353,
0.090196,
0.098039,
0.105882,
0.117647,
```

```
0.125490,
0.121569,
0.117647,
0.125490,
0.121569,
0.105882,
0.086275,
0.082353,
0.101961,
0.117647,
0.121569,
0.121569,
0.125490,
0.149020,
0.168627,
0.172549,
0.164706,
0.164706,
0.156863,
0.164706,
0.184314,
0.211765,
0.243137,
0.258824,
0.270588,
0.294118,
0.313726,
0.317647,
0.309804,
0.298039,
0.282353,
0.278431,
0.270588,
0.250980,
0.258824,
0.282353,
0.270588,
0.235294,
0.227451,
0.254902,
0.294118,
0.349020,
0.419608,
0.470588,
0.517647,
0.572549,
0.639216,
0.705882,
0.768627,
0.827451,
0.878431,
0.921569,
0.937255,
0.945098,
0.949020,
0.949020,
0.949020,
0.945098,
0.941176,
0.937255,
0.929412,
```

```
0.917647,
0.894118,
0.850980,
0.784314,
0.698039,
0.584314,
0.454902,
0.352941,
0.282353,
0.207843,
0.137255,
0.105882,
0.082353,
0.054902,
0.031373,
0.023529,
0.019608,
0.019608,
0.023529,
0.023529,
0.027451,
0.027451,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.058824,
0.058824,
0.062745,
0.066667,
0.062745,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
```

```
0.074510,
0.078431,
0.078431,
0.082353,
0.090196,
0.098039,
0.113725,
0.125490,
0.133333,
0.137255,
0.137255,
0.133333,
0.129412,
0.109804,
0.090196,
0.109804,
0.133333,
0.145098,
0.149020,
0.145098,
0.141176,
0.160784,
0.184314,
0.196078,
0.207843,
0.203922,
0.196078,
0.203922,
0.227451,
0.266667,
0.286275,
0.282353,
0.298039,
0.325490,
0.333333,
0.329412,
0.321569,
0.309804,
0.290196,
0.278431,
0.290196,
0.305882,
0.309804,
0.286275,
0.262745,
0.278431,
0.301961,
0.313726,
0.352941,
0.439216,
0.509804,
0.541176,
0.584314,
0.643137,
0.705882,
0.776471,
0.835294,
0.878431,
0.921569,
0.949020,
0.949020,
```

```
0.945098,
0.945098,
0.945098,
0.941176,
0.937255,
0.929412,
0.917647,
0.898039,
0.866667,
0.796078,
0.705882,
0.635294,
0.576471,
0.470588,
0.309804,
0.180392,
0.117647,
0.086275,
0.062745,
0.047059,
0.035294,
0.027451,
0.023529,
0.023529,
0.023529,
0.023529,
0.027451,
0.031373,
0.031373,
0.035294,
0.031373,
0.027451,
0.027451,
0.031373,
0.035294,
0.039216,
0.039216,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
```

```
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.074510,
0.078431,
0.082353,
0.082353,
0.090196,
0.101961,
0.117647,
0.129412,
0.141176,
0.156863,
0.160784,
0.156863,
0.152941,
0.133333,
0.101961,
0.109804,
0.141176,
0.160784,
0.172549,
0.168627,
0.152941,
0.152941,
0.172549,
0.215686,
0.247059,
0.254902,
0.247059,
0.235294,
0.250980,
0.282353,
0.309804,
0.305882,
0.309804,
0.333333,
0.345098,
0.345098,
0.337255,
0.321569,
0.298039,
0.301961,
0.325490,
0.333333,
0.325490,
0.305882,
0.298039,
0.329412,
0.364706,
0.380392,
0.392157,
0.454902,
0.564706,
0.639216,
0.670588,
0.694118,
```

```
0.749020,
0.815686,
0.858824,
0.894118,
0.929412,
0.949020,
0.945098,
0.941176,
0.937255,
0.937255,
0.937255,
0.933333,
0.921569,
0.905882,
0.882353,
0.835294,
0.749020,
0.623529,
0.509804,
0.458824,
0.372549,
0.215686,
0.098039,
0.074510,
0.070588,
0.047059,
0.031373,
0.027451,
0.027451,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.027451,
0.031373,
0.035294,
0.043137,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.066667,
0.066667,
0.058824,
0.062745,
0.062745,
0.062745,
```

```
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.058824,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.078431,
0.078431,
0.082353,
0.090196,
0.101961,
0.113725,
0.121569,
0.133333,
0.156863,
0.172549,
0.176471,
0.172549,
0.156863,
0.129412,
0.133333,
0.152941,
0.168627,
0.184314,
0.192157,
0.192157,
0.176471,
0.176471,
0.219608,
0.258824,
0.278431,
0.282353,
0.270588,
0.266667,
0.294118,
0.325490,
0.325490,
0.325490,
0.341176,
0.352941,
0.352941,
0.352941,
0.333333,
0.317647,
0.325490,
0.341176,
0.345098,
0.352941,
0.356863,
0.356863,
0.360784,
0.392157,
```

```
0.435294,
0.450980,
0.466667,
0.560784,
0.682353,
0.741176,
0.756863,
0.796078,
0.847059,
0.886275,
0.917647,
0.937255,
0.941176,
0.941176,
0.937255,
0.933333,
0.925490,
0.925490,
0.925490,
0.913725,
0.901961,
0.886275,
0.847059,
0.784314,
0.647059,
0.466667,
0.364706,
0.266667,
0.149020,
0.121569,
0.141176,
0.113725,
0.066667,
0.047059,
0.039216,
0.027451,
0.027451,
0.031373,
0.039216,
0.039216,
0.031373,
0.035294,
0.039216,
0.039216,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.035294,
0.039216,
0.043137,
0.043137,
0.047059,
0.043137,
0.039216,
0.047059,
0.054902,
0.058824,
0.058824,
```

```
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
0.086275,
0.094118,
0.101961,
0.105882,
0.121569,
0.145098,
0.168627,
0.188235,
0.192157,
0.180392,
0.168627,
0.180392,
0.184314,
0.168627,
0.176471,
0.200000,
0.219608,
0.211765,
0.203922,
0.223529,
0.250980,
0.282353,
0.305882,
0.301961,
0.290196,
0.309804,
0.329412,
0.333333,
0.333333,
0.349020,
0.360784,
0.360784,
0.356863,
0.345098,
0.333333,
0.341176,
```

```
0.352941,
0.356863,
0.380392,
0.407843,
0.411765,
0.403922,
0.400000,
0.435294,
0.474510,
0.490196,
0.556863,
0.670588,
0.764706,
0.811765,
0.843137,
0.878431,
0.905882,
0.921569,
0.929412,
0.933333,
0.933333,
0.929412,
0.921569,
0.913725,
0.913725,
0.913725,
0.905882,
0.894118,
0.886275,
0.870588,
0.831373,
0.733333,
0.603922,
0.498039,
0.345098,
0.239216,
0.258824,
0.243137,
0.156863,
0.109804,
0.098039,
0.070588,
0.050980,
0.043137,
0.047059,
0.054902,
0.047059,
0.035294,
0.035294,
0.039216,
0.043137,
0.039216,
0.035294,
0.031373,
0.035294,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
```

```
0.039216,
0.035294,
0.035294,
0.043137,
0.058824,
0.066667,
0.066667,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.074510,
0.078431,
0.082353,
0.086275,
0.094118,
0.105882,
0.117647,
0.133333,
0.156863,
0.180392,
0.200000,
0.200000,
0.200000,
0.211765,
0.207843,
0.184314,
0.176471,
0.196078,
0.231373,
0.247059,
0.250980,
0.250980,
0.262745,
0.294118,
0.317647,
0.317647,
0.317647,
0.329412,
0.333333,
0.333333,
0.341176,
```

```
0.349020,
0.352941,
0.352941,
0.352941,
0.349020,
0.341176,
0.345098,
0.356863,
0.368627,
0.400000,
0.435294,
0.454902,
0.454902,
0.450980,
0.474510,
0.509804,
0.521569,
0.572549,
0.674510,
0.772549,
0.835294,
0.870588,
0.886275,
0.901961,
0.913725,
0.917647,
0.921569,
0.921569,
0.917647,
0.909804,
0.901961,
0.901961,
0.901961,
0.894118,
0.878431,
0.874510,
0.870588,
0.839216,
0.760784,
0.666667,
0.545098,
0.392157,
0.360784,
0.403922,
0.337255,
0.235294,
0.192157,
0.145098,
0.109804,
0.101961,
0.082353,
0.074510,
0.066667,
0.050980,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.035294,
```

```
0.035294,
0.039216,
0.039216,
0.035294,
0.043137,
0.047059,
0.047059,
0.047059,
0.043137,
0.043137,
0.047059,
0.062745,
0.074510,
0.070588,
0.062745,
0.058824,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.074510,
0.078431,
0.078431,
0.082353,
0.090196,
0.101961,
0.117647,
0.129412,
0.149020,
0.168627,
0.188235,
0.203922,
0.219608,
0.231373,
0.219608,
0.203922,
0.200000,
0.203922,
0.231373,
0.266667,
0.282353,
0.278431,
0.278431,
0.301961,
```

```
0.321569,
0.325490,
0.333333,
0.345098,
0.341176,
0.341176,
0.345098,
0.345098,
0.341176,
0.337255,
0.341176,
0.345098,
0.349020,
0.352941,
0.360784,
0.376471,
0.411765,
0.454902,
0.486275,
0.505882,
0.525490,
0.556863,
0.603922,
0.619608,
0.635294,
0.709804,
0.792157,
0.835294,
0.862745,
0.882353,
0.894118,
0.905882,
0.909804,
0.909804,
0.913725,
0.901961,
0.890196,
0.894118,
0.894118,
0.890196,
0.882353,
0.870588,
0.858824,
0.858824,
0.831373,
0.729412,
0.600000,
0.505882,
0.450980,
0.470588,
0.501961,
0.458824,
0.384314,
0.298039,
0.196078,
0.160784,
0.156863,
0.125490,
0.105882,
0.086275,
```

```
0.054902,
0.054902,
0.058824,
0.054902,
0.047059,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.043137,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.066667,
0.078431,
0.074510,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.074510,
0.078431,
0.082353,
0.086275,
0.090196,
0.090196,
0.098039,
0.113725,
0.133333,
0.149020,
0.160784,
0.176471,
0.196078,
0.219608,
0.235294,
0.231373,
0.219608,
0.227451,
```

```
0.223529,
0.239216,
0.270588,
0.290196,
0.290196,
0.286275,
0.305882,
0.325490,
0.329412,
0.333333,
0.341176,
0.345098,
0.345098,
0.341176,
0.337255,
0.329412,
0.329412,
0.333333,
0.337255,
0.345098,
0.352941,
0.364706,
0.380392,
0.411765,
0.458824,
0.509804,
0.533333,
0.549020,
0.568627,
0.623529,
0.694118,
0.717647,
0.733333,
0.784314,
0.827451,
0.850980,
0.874510,
0.890196,
0.894118,
0.898039,
0.901961,
0.901961,
0.890196,
0.874510,
0.882353,
0.890196,
0.882353,
0.870588,
0.858824,
0.850980,
0.839216,
0.819608,
0.741176,
0.647059,
0.627451,
0.635294,
0.607843,
0.580392,
0.564706,
0.505882,
```

```
0.384314,
0.270588,
0.235294,
0.207843,
0.164706,
0.129412,
0.086275,
0.058824,
0.066667,
0.074510,
0.062745,
0.054902,
0.054902,
0.058824,
0.054902,
0.047059,
0.043137,
0.039216,
0.035294,
0.043137,
0.047059,
0.058824,
0.062745,
0.066667,
0.070588,
0.078431,
0.082353,
0.074510,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.078431,
0.086275,
0.090196,
0.098039,
0.101961,
0.105882,
0.101961,
0.109804,
0.133333,
0.149020,
0.160784,
```

```
0.176471,
0.192157,
0.211765,
0.235294,
0.239216,
0.235294,
0.243137,
0.250980,
0.262745,
0.278431,
0.294118,
0.301961,
0.305882,
0.317647,
0.329412,
0.329412,
0.325490,
0.329412,
0.333333,
0.333333,
0.337255,
0.333333,
0.329412,
0.325490,
0.321569,
0.325490,
0.337255,
0.345098,
0.352941,
0.372549,
0.411765,
0.474510,
0.545098,
0.580392,
0.588235,
0.584314,
0.615686,
0.701961,
0.764706,
0.772549,
0.780392,
0.807843,
0.843137,
0.870588,
0.882353,
0.882353,
0.890196,
0.894118,
0.890196,
0.874510,
0.858824,
0.866667,
0.878431,
0.874510,
0.862745,
0.850980,
0.847059,
0.831373,
0.815686,
0.780392,
```

```
0.745098,
0.749020,
0.745098,
0.690196,
0.647059,
0.623529,
0.549020,
0.443137,
0.388235,
0.349020,
0.266667,
0.180392,
0.117647,
0.070588,
0.066667,
0.082353,
0.074510,
0.062745,
0.066667,
0.070588,
0.070588,
0.058824,
0.054902,
0.050980,
0.043137,
0.047059,
0.050980,
0.054902,
0.062745,
0.074510,
0.074510,
0.078431,
0.086275,
0.086275,
0.070588,
0.062745,
0.066667,
0.070588,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.082353,
0.086275,
0.094118,
0.101961,
```

```
0.109804,
0.117647,
0.121569,
0.117647,
0.125490,
0.149020,
0.172549,
0.192157,
0.203922,
0.211765,
0.235294,
0.250980,
0.250980,
0.254902,
0.262745,
0.274510,
0.290196,
0.301961,
0.309804,
0.317647,
0.325490,
0.325490,
0.317647,
0.313726,
0.321569,
0.321569,
0.317647,
0.321569,
0.329412,
0.325490,
0.313726,
0.309804,
0.317647,
0.321569,
0.317647,
0.325490,
0.360784,
0.411765,
0.482353,
0.564706,
0.631373,
0.662745,
0.654902,
0.674510,
0.721569,
0.776471,
0.807843,
0.807843,
0.811765,
0.839216,
0.862745,
0.870588,
0.870588,
0.874510,
0.878431,
0.878431,
0.862745,
0.843137,
0.854902,
0.866667,
```

```
0.862745,
0.854902,
0.847059,
0.839216,
0.827451,
0.807843,
0.788235,
0.784314,
0.780392,
0.756863,
0.713726,
0.686275,
0.643137,
0.564706,
0.509804,
0.478431,
0.407843,
0.282353,
0.176471,
0.113725,
0.094118,
0.101961,
0.101961,
0.078431,
0.070588,
0.082353,
0.082353,
0.078431,
0.066667,
0.066667,
0.062745,
0.050980,
0.062745,
0.074510,
0.070588,
0.074510,
0.082353,
0.086275,
0.090196,
0.094118,
0.086275,
0.066667,
0.066667,
0.074510,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
```

```
0.070588,
0.074510,
0.078431,
0.082353,
0.086275,
0.094118,
0.101961,
0.109804,
0.121569,
0.137255,
0.145098,
0.145098,
0.164706,
0.192157,
0.207843,
0.215686,
0.219608,
0.231373,
0.250980,
0.254902,
0.258824,
0.266667,
0.278431,
0.294118,
0.301961,
0.305882,
0.309804,
0.313726,
0.309804,
0.301961,
0.298039,
0.309804,
0.313726,
0.305882,
0.305882,
0.305882,
0.294118,
0.290196,
0.294118,
0.298039,
0.290196,
0.278431,
0.305882,
0.352941,
0.400000,
0.458824,
0.537255,
0.623529,
0.694118,
0.713726,
0.737255,
0.760784,
0.780392,
0.803922,
0.819608,
0.831373,
0.843137,
0.854902,
0.858824,
0.858824,
```

```
0.862745,
0.866667,
0.866667,
0.850980,
0.831373,
0.839216,
0.850980,
0.850980,
0.843137,
0.835294,
0.823529,
0.811765,
0.796078,
0.780392,
0.780392,
0.784314,
0.760784,
0.737255,
0.709804,
0.658824,
0.607843,
0.560784,
0.498039,
0.400000,
0.290196,
0.215686,
0.176471,
0.156863,
0.145098,
0.117647,
0.086275,
0.094118,
0.101961,
0.090196,
0.082353,
0.086275,
0.090196,
0.074510,
0.058824,
0.074510,
0.090196,
0.090196,
0.086275,
0.094118,
0.101961,
0.101961,
0.098039,
0.082353,
0.066667,
0.070588,
0.078431,
0.074510,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
```

```
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.058824,
0.062745,
0.070588,
0.074510,
0.078431,
0.086275,
0.090196,
0.094118,
0.101961,
0.113725,
0.129412,
0.149020,
0.164706,
0.172549,
0.188235,
0.207843,
0.219608,
0.227451,
0.227451,
0.231373,
0.243137,
0.250980,
0.258824,
0.266667,
0.278431,
0.290196,
0.298039,
0.301961,
0.301961,
0.294118,
0.294118,
0.286275,
0.274510,
0.282353,
0.290196,
0.294118,
0.282353,
0.270588,
0.266667,
0.270588,
0.274510,
0.278431,
0.266667,
0.262745,
0.313726,
0.368627,
0.400000,
0.447059,
0.513726,
0.588235,
0.658824,
0.709804,
0.752941,
0.776471,
0.780392,
```

```
0.792157,
0.811765,
0.831373,
0.839216,
0.839216,
0.839216,
0.843137,
0.850980,
0.850980,
0.850980,
0.839216,
0.823529,
0.819608,
0.831373,
0.831373,
0.827451,
0.819608,
0.807843,
0.792157,
0.780392,
0.776471,
0.768627,
0.768627,
0.760784,
0.741176,
0.717647,
0.690196,
0.639216,
0.572549,
0.494118,
0.407843,
0.325490,
0.258824,
0.219608,
0.184314,
0.152941,
0.121569,
0.109804,
0.121569,
0.117647,
0.101961,
0.094118,
0.105882,
0.109804,
0.086275,
0.066667,
0.086275,
0.105882,
0.105882,
0.101961,
0.109804,
0.117647,
0.109804,
0.090196,
0.078431,
0.074510,
0.078431,
0.078431,
0.078431,
0.074510,
0.074510,
```

```
0.070588,
0.066667,
0.066667,
0.066667,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.070588,
0.074510,
0.078431,
0.082353,
0.082353,
0.090196,
0.101961,
0.121569,
0.141176,
0.156863,
0.172549,
0.180392,
0.192157,
0.207843,
0.219608,
0.227451,
0.231373,
0.231373,
0.235294,
0.247059,
0.258824,
0.266667,
0.278431,
0.286275,
0.290196,
0.286275,
0.282353,
0.278431,
0.282353,
0.274510,
0.254902,
0.247059,
0.254902,
0.270588,
0.258824,
0.235294,
0.247059,
0.258824,
0.250980,
0.247059,
0.243137,
0.239216,
0.294118,
0.341176,
0.364706,
0.407843,
```

```
0.454902,
0.501961,
0.556863,
0.627451,
0.694118,
0.733333,
0.756863,
0.788235,
0.807843,
0.819608,
0.827451,
0.823529,
0.819608,
0.831373,
0.835294,
0.835294,
0.831373,
0.827451,
0.819608,
0.803922,
0.800000,
0.807843,
0.811765,
0.807843,
0.800000,
0.780392,
0.764706,
0.768627,
0.764706,
0.752941,
0.749020,
0.733333,
0.717647,
0.690196,
0.643137,
0.576471,
0.505882,
0.431373,
0.356863,
0.278431,
0.215686,
0.164706,
0.129412,
0.133333,
0.149020,
0.145098,
0.137255,
0.133333,
0.129412,
0.141176,
0.137255,
0.098039,
0.078431,
0.101961,
0.113725,
0.113725,
0.121569,
0.129412,
0.125490,
0.105882,
0.086275,
```

```
0.086275,
0.086275,
0.082353,
0.082353,
0.078431,
0.074510,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.066667,
0.070588,
0.074510,
0.074510,
0.078431,
0.086275,
0.101961,
0.117647,
0.137255,
0.152941,
0.164706,
0.176471,
0.188235,
0.200000,
0.215686,
0.223529,
0.223529,
0.223529,
0.235294,
0.243137,
0.254902,
0.266667,
0.274510,
0.278431,
0.282353,
0.278431,
0.262745,
0.258824,
0.274510,
0.274510,
0.250980,
0.235294,
0.227451,
0.239216,
0.231373,
0.207843,
0.227451,
0.243137,
0.223529,
```

```
0.207843,
0.215686,
0.235294,
0.262745,
0.298039,
0.329412,
0.368627,
0.388235,
0.415686,
0.474510,
0.560784,
0.639216,
0.694118,
0.741176,
0.784314,
0.800000,
0.800000,
0.811765,
0.811765,
0.807843,
0.815686,
0.823529,
0.819608,
0.811765,
0.807843,
0.807843,
0.796078,
0.772549,
0.784314,
0.796078,
0.792157,
0.788235,
0.768627,
0.745098,
0.756863,
0.764706,
0.749020,
0.737255,
0.725490,
0.713726,
0.686275,
0.647059,
0.600000,
0.537255,
0.470588,
0.396078,
0.321569,
0.239216,
0.172549,
0.160784,
0.192157,
0.203922,
0.180392,
0.172549,
0.176471,
0.172549,
0.176471,
0.156863,
0.105882,
0.094118,
0.129412,
```

```
0.133333,
0.121569,
0.137255,
0.141176,
0.125490,
0.101961,
0.086275,
0.094118,
0.094118,
0.086275,
0.082353,
0.078431,
0.074510,
0.074510,
0.074510,
0.074510,
0.074510,
0.070588,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.070588,
0.074510,
0.082353,
0.086275,
0.098039,
0.109804,
0.125490,
0.141176,
0.149020,
0.156863,
0.172549,
0.192157,
0.207843,
0.211765,
0.215686,
0.227451,
0.235294,
0.243137,
0.250980,
0.258824,
0.258824,
0.262745,
0.274510,
0.274510,
0.254902,
0.235294,
0.250980,
0.262745,
0.243137,
0.227451,
```

```
0.215686,
0.207843,
0.196078,
0.180392,
0.200000,
0.211765,
0.192157,
0.188235,
0.223529,
0.266667,
0.298039,
0.337255,
0.372549,
0.411765,
0.447059,
0.466667,
0.509804,
0.603922,
0.682353,
0.725490,
0.756863,
0.784314,
0.788235,
0.784314,
0.792157,
0.803922,
0.800000,
0.800000,
0.807843,
0.807843,
0.796078,
0.780392,
0.784314,
0.776471,
0.756863,
0.772549,
0.788235,
0.780392,
0.772549,
0.749020,
0.725490,
0.733333,
0.749020,
0.741176,
0.725490,
0.709804,
0.694118,
0.674510,
0.639216,
0.588235,
0.529412,
0.458824,
0.392157,
0.329412,
0.262745,
0.223529,
0.243137,
0.262745,
0.235294,
0.211765,
0.223529,
```

```
0.235294,
0.211765,
0.188235,
0.164706,
0.137255,
0.141176,
0.160784,
0.152941,
0.141176,
0.145098,
0.137255,
0.117647,
0.098039,
0.094118,
0.098039,
0.098039,
0.090196,
0.086275,
0.078431,
0.074510,
0.078431,
0.082353,
0.078431,
0.074510,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.078431,
0.082353,
0.090196,
0.098039,
0.109804,
0.121569,
0.129412,
0.133333,
0.137255,
0.156863,
0.180392,
0.188235,
0.196078,
0.211765,
0.231373,
0.235294,
0.239216,
0.247059,
0.254902,
0.247059,
0.243137,
0.258824,
```

```
0.262745,
0.239216,
0.207843,
0.223529,
0.243137,
0.235294,
0.211765,
0.192157,
0.184314,
0.168627,
0.152941,
0.168627,
0.164706,
0.156863,
0.184314,
0.227451,
0.266667,
0.309804,
0.341176,
0.376471,
0.443137,
0.521569,
0.560784,
0.584314,
0.654902,
0.725490,
0.756863,
0.760784,
0.768627,
0.768627,
0.768627,
0.776471,
0.788235,
0.792157,
0.788235,
0.792157,
0.796078,
0.788235,
0.764706,
0.749020,
0.741176,
0.737255,
0.760784,
0.776471,
0.772549,
0.760784,
0.741176,
0.717647,
0.709804,
0.717647,
0.725490,
0.717647,
0.701961,
0.678431,
0.647059,
0.603922,
0.545098,
0.482353,
0.419608,
0.356863,
0.298039,
```

```
0.278431,
0.301961,
0.321569,
0.298039,
0.258824,
0.254902,
0.286275,
0.290196,
0.254902,
0.215686,
0.207843,
0.215686,
0.203922,
0.176471,
0.160784,
0.152941,
0.145098,
0.125490,
0.109804,
0.101961,
0.101961,
0.109804,
0.105882,
0.098039,
0.090196,
0.082353,
0.086275,
0.086275,
0.086275,
0.082353,
0.074510,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.074510,
0.082353,
0.090196,
0.098039,
0.109804,
0.113725,
0.125490,
0.121569,
0.117647,
0.137255,
0.160784,
0.172549,
0.192157,
0.215686,
0.227451,
```

```
0.227451,
0.231373,
0.243137,
0.247059,
0.235294,
0.223529,
0.227451,
0.231373,
0.215686,
0.188235,
0.192157,
0.219608,
0.223529,
0.196078,
0.164706,
0.160784,
0.149020,
0.129412,
0.133333,
0.137255,
0.145098,
0.172549,
0.203922,
0.239216,
0.270588,
0.301961,
0.349020,
0.419608,
0.494118,
0.545098,
0.580392,
0.631373,
0.698039,
0.733333,
0.745098,
0.752941,
0.752941,
0.752941,
0.764706,
0.776471,
0.780392,
0.776471,
0.776471,
0.780392,
0.776471,
0.756863,
0.725490,
0.701961,
0.713726,
0.741176,
0.756863,
0.752941,
0.749020,
0.741176,
0.725490,
0.709804,
0.698039,
0.698039,
0.701961,
0.694118,
0.670588,
```

```
0.639216,
0.592157,
0.541176,
0.490196,
0.443137,
0.388235,
0.337255,
0.341176,
0.372549,
0.368627,
0.325490,
0.298039,
0.321569,
0.345098,
0.325490,
0.286275,
0.262745,
0.274510,
0.282353,
0.239216,
0.192157,
0.172549,
0.152941,
0.133333,
0.113725,
0.105882,
0.109804,
0.117647,
0.117647,
0.113725,
0.101961,
0.094118,
0.090196,
0.090196,
0.090196,
0.082353,
0.078431,
0.074510,
0.070588,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.078431,
0.082353,
0.090196,
0.094118,
0.098039,
0.105882,
0.117647,
0.109804,
```

```
0.094118,
0.105882,
0.137255,
0.168627,
0.192157,
0.207843,
0.215686,
0.215686,
0.219608,
0.235294,
0.247059,
0.231373,
0.207843,
0.196078,
0.192157,
0.192157,
0.180392,
0.168627,
0.180392,
0.200000,
0.180392,
0.145098,
0.137255,
0.141176,
0.117647,
0.105882,
0.121569,
0.137255,
0.149020,
0.180392,
0.211765,
0.254902,
0.313726,
0.372549,
0.415686,
0.462745,
0.517647,
0.568627,
0.619608,
0.670588,
0.705882,
0.729412,
0.737255,
0.737255,
0.745098,
0.752941,
0.760784,
0.764706,
0.760784,
0.756863,
0.756863,
0.764706,
0.752941,
0.721569,
0.694118,
0.709804,
0.733333,
0.741176,
0.733333,
0.725490,
0.725490,
```

```
0.721569,
0.709804,
0.694118,
0.678431,
0.678431,
0.670588,
0.650980,
0.627451,
0.592157,
0.545098,
0.513726,
0.482353,
0.443137,
0.411765,
0.407843,
0.411765,
0.396078,
0.368627,
0.356863,
0.372549,
0.372549,
0.349020,
0.329412,
0.321569,
0.317647,
0.290196,
0.247059,
0.215686,
0.188235,
0.149020,
0.117647,
0.109804,
0.117647,
0.125490,
0.125490,
0.121569,
0.113725,
0.105882,
0.098039,
0.090196,
0.082353,
0.082353,
0.078431,
0.074510,
0.074510,
0.070588,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.078431,
```

```
0.082353,
0.086275,
0.086275,
0.086275,
0.086275,
0.098039,
0.090196,
0.078431,
0.098039,
0.137255,
0.164706,
0.180392,
0.196078,
0.207843,
0.200000,
0.200000,
0.215686,
0.231373,
0.219608,
0.200000,
0.172549,
0.152941,
0.164706,
0.188235,
0.176471,
0.156863,
0.160784,
0.160784,
0.137255,
0.125490,
0.125490,
0.109804,
0.090196,
0.098039,
0.109804,
0.109804,
0.141176,
0.188235,
0.250980,
0.329412,
0.388235,
0.431373,
0.474510,
0.525490,
0.580392,
0.623529,
0.662745,
0.690196,
0.705882,
0.713726,
0.725490,
0.733333,
0.741176,
0.745098,
0.745098,
0.741176,
0.729412,
0.729412,
0.741176,
0.745098,
0.721569,
```

```
0.698039,
0.709804,
0.721569,
0.721569,
0.713726,
0.705882,
0.701961,
0.698039,
0.694118,
0.686275,
0.670588,
0.654902,
0.647059,
0.627451,
0.607843,
0.584314,
0.552941,
0.533333,
0.498039,
0.466667,
0.458824,
0.450980,
0.443137,
0.435294,
0.423529,
0.407843,
0.392157,
0.384314,
0.380392,
0.376471,
0.364706,
0.317647,
0.262745,
0.223529,
0.200000,
0.176471,
0.149020,
0.125490,
0.121569,
0.137255,
0.141176,
0.137255,
0.125490,
0.117647,
0.109804,
0.101961,
0.086275,
0.078431,
0.074510,
0.078431,
0.078431,
0.074510,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
```

```
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.070588,
0.074510,
0.078431,
0.078431,
0.078431,
0.070588,
0.070588,
0.078431,
0.078431,
0.090196,
0.121569,
0.145098,
0.152941,
0.164706,
0.184314,
0.203922,
0.196078,
0.180392,
0.196078,
0.207843,
0.196078,
0.192157,
0.164706,
0.121569,
0.125490,
0.168627,
0.176471,
0.149020,
0.129412,
0.137255,
0.141176,
0.121569,
0.105882,
0.090196,
0.078431,
0.082353,
0.094118,
0.105882,
0.137255,
0.188235,
0.247059,
0.305882,
0.364706,
0.411765,
0.443137,
0.490196,
0.541176,
0.580392,
0.615686,
0.647059,
0.674510,
0.698039,
0.713726,
0.717647,
0.725490,
0.729412,
```

```
0.721569,
0.709804,
0.694118,
0.701961,
0.725490,
0.725490,
0.709804,
0.690196,
0.698039,
0.709804,
0.705882,
0.690196,
0.678431,
0.682353,
0.682353,
0.674510,
0.666667,
0.658824,
0.647059,
0.631373,
0.623529,
0.611765,
0.596078,
0.576471,
0.552941,
0.521569,
0.494118,
0.486275,
0.482353,
0.478431,
0.474510,
0.462745,
0.439216,
0.411765,
0.396078,
0.396078,
0.388235,
0.345098,
0.286275,
0.235294,
0.192157,
0.168627,
0.160784,
0.152941,
0.149020,
0.149020,
0.156863,
0.152941,
0.141176,
0.125490,
0.121569,
0.113725,
0.101961,
0.086275,
0.074510,
0.070588,
0.074510,
0.074510,
0.074510,
0.070588,
```

```
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.066667,
0.070588,
0.074510,
0.070588,
0.070588,
0.074510,
0.078431,
0.090196,
0.101961,
0.121569,
0.133333,
0.137255,
0.141176,
0.152941,
0.172549,
0.196078,
0.184314,
0.168627,
0.184314,
0.200000,
0.184314,
0.180392,
0.168627,
0.125490,
0.105882,
0.141176,
0.156863,
0.133333,
0.109804,
0.113725,
0.133333,
0.113725,
0.086275,
0.074510,
0.074510,
0.090196,
0.105882,
0.133333,
0.164706,
0.188235,
0.219608,
0.254902,
0.301961,
0.337255,
0.368627,
0.431373,
0.490196,
0.521569,
0.564706,
```

```
0.615686,
0.658824,
0.686275,
0.694118,
0.698039,
0.709804,
0.713726,
0.705882,
0.682353,
0.666667,
0.690196,
0.709804,
0.705882,
0.694118,
0.686275,
0.690196,
0.698039,
0.690196,
0.670588,
0.650980,
0.662745,
0.670588,
0.658824,
0.643137,
0.635294,
0.639216,
0.627451,
0.611765,
0.603922,
0.600000,
0.584314,
0.552941,
0.529412,
0.513726,
0.501961,
0.498039,
0.501961,
0.498039,
0.474510,
0.447059,
0.427451,
0.407843,
0.396078,
0.364706,
0.290196,
0.235294,
0.203922,
0.176471,
0.180392,
0.188235,
0.180392,
0.176471,
0.180392,
0.176471,
0.164706,
0.145098,
0.133333,
0.125490,
0.113725,
0.101961,
```

```
0.086275,
0.078431,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.062745,
0.070588,
0.078431,
0.098039,
0.117647,
0.125490,
0.125490,
0.121569,
0.129412,
0.141176,
0.160784,
0.180392,
0.168627,
0.152941,
0.172549,
0.188235,
0.176471,
0.168627,
0.168627,
0.145098,
0.113725,
0.117647,
0.129412,
0.113725,
0.094118,
0.101961,
0.121569,
0.105882,
0.078431,
0.070588,
0.078431,
0.090196,
0.109804,
0.133333,
0.145098,
0.152941,
0.172549,
0.207843,
```

```
0.250980,
0.286275,
0.333333,
0.403922,
0.458824,
0.494118,
0.549020,
0.615686,
0.650980,
0.670588,
0.678431,
0.678431,
0.690196,
0.698039,
0.690196,
0.666667,
0.662745,
0.682353,
0.694118,
0.686275,
0.678431,
0.682353,
0.682353,
0.678431,
0.678431,
0.658824,
0.635294,
0.647059,
0.650980,
0.643137,
0.627451,
0.611765,
0.623529,
0.619608,
0.588235,
0.568627,
0.580392,
0.568627,
0.533333,
0.509804,
0.501961,
0.498039,
0.498039,
0.501961,
0.498039,
0.470588,
0.447059,
0.431373,
0.427451,
0.407843,
0.356863,
0.286275,
0.227451,
0.184314,
0.180392,
0.211765,
0.231373,
0.219608,
0.207843,
0.203922,
```

```
0.192157,
0.176471,
0.160784,
0.149020,
0.137255,
0.121569,
0.105882,
0.094118,
0.086275,
0.078431,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.050980,
0.050980,
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.074510,
0.094118,
0.109804,
0.113725,
0.109804,
0.109804,
0.117647,
0.129412,
0.149020,
0.164706,
0.149020,
0.137255,
0.152941,
0.168627,
0.160784,
0.156863,
0.156863,
0.149020,
0.125490,
0.109804,
0.105882,
0.094118,
0.082353,
0.090196,
0.101961,
0.094118,
0.082353,
0.074510,
0.082353,
```

```
0.094118,
0.101961,
0.101961,
0.105882,
0.129412,
0.160784,
0.203922,
0.254902,
0.305882,
0.356863,
0.400000,
0.447059,
0.501961,
0.556863,
0.600000,
0.623529,
0.647059,
0.654902,
0.658824,
0.670588,
0.666667,
0.654902,
0.647059,
0.658824,
0.674510,
0.674510,
0.670588,
0.666667,
0.670588,
0.666667,
0.650980,
0.643137,
0.627451,
0.607843,
0.627451,
0.631373,
0.615686,
0.603922,
0.592157,
0.603922,
0.607843,
0.572549,
0.549020,
0.552941,
0.549020,
0.525490,
0.501961,
0.482353,
0.478431,
0.490196,
0.494118,
0.490196,
0.466667,
0.447059,
0.443137,
0.435294,
0.411765,
0.364706,
0.317647,
0.266667,
```

```
0.215686,
0.219608,
0.254902,
0.262745,
0.250980,
0.235294,
0.227451,
0.215686,
0.200000,
0.184314,
0.168627,
0.149020,
0.133333,
0.113725,
0.098039,
0.090196,
0.082353,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.082353,
0.094118,
0.098039,
0.098039,
0.098039,
0.101961,
0.109804,
0.121569,
0.133333,
0.137255,
0.133333,
0.125490,
0.145098,
0.152941,
0.141176,
0.141176,
0.141176,
0.137255,
0.129412,
0.109804,
0.098039,
0.086275,
```

```
0.078431,
0.078431,
0.078431,
0.078431,
0.074510,
0.074510,
0.082353,
0.094118,
0.094118,
0.082353,
0.094118,
0.141176,
0.184314,
0.227451,
0.286275,
0.341176,
0.376471,
0.415686,
0.474510,
0.529412,
0.556863,
0.568627,
0.588235,
0.611765,
0.627451,
0.647059,
0.643137,
0.619608,
0.615686,
0.635294,
0.654902,
0.662745,
0.658824,
0.662745,
0.658824,
0.654902,
0.647059,
0.631373,
0.611765,
0.584314,
0.568627,
0.596078,
0.615686,
0.600000,
0.580392,
0.576471,
0.588235,
0.596078,
0.576471,
0.549020,
0.533333,
0.529412,
0.529412,
0.517647,
0.490196,
0.470588,
0.474510,
0.482353,
0.482353,
0.462745,
```

```
0.447059,
0.439216,
0.427451,
0.396078,
0.360784,
0.329412,
0.298039,
0.270588,
0.282353,
0.298039,
0.294118,
0.282353,
0.262745,
0.247059,
0.231373,
0.211765,
0.192157,
0.172549,
0.152941,
0.137255,
0.117647,
0.098039,
0.086275,
0.078431,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.070588,
0.074510,
0.078431,
0.082353,
0.090196,
0.086275,
0.082353,
0.090196,
0.098039,
0.105882,
0.109804,
0.113725,
0.117647,
0.113725,
0.121569,
0.137255,
0.141176,
0.133333,
```

```
0.121569,
0.121569,
0.121569,
0.117647,
0.105882,
0.098039,
0.090196,
0.082353,
0.074510,
0.070588,
0.070588,
0.070588,
0.074510,
0.082353,
0.086275,
0.082353,
0.082353,
0.117647,
0.176471,
0.219608,
0.266667,
0.313726,
0.341176,
0.372549,
0.439216,
0.501961,
0.525490,
0.517647,
0.525490,
0.549020,
0.580392,
0.607843,
0.619608,
0.603922,
0.592157,
0.611765,
0.635294,
0.639216,
0.643137,
0.643137,
0.643137,
0.647059,
0.639216,
0.631373,
0.619608,
0.607843,
0.576471,
0.549020,
0.576471,
0.603922,
0.596078,
0.576471,
0.568627,
0.576471,
0.580392,
0.568627,
0.552941,
0.529412,
0.513726,
0.509804,
```

```
0.513726,
0.501961,
0.478431,
0.458824,
0.462745,
0.470588,
0.462745,
0.443137,
0.427451,
0.411765,
0.388235,
0.368627,
0.352941,
0.341176,
0.329412,
0.321569,
0.317647,
0.305882,
0.290196,
0.270588,
0.239216,
0.211765,
0.192157,
0.176471,
0.152941,
0.137255,
0.125490,
0.105882,
0.090196,
0.082353,
0.074510,
0.074510,
0.070588,
0.066667,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.066667,
0.070588,
0.074510,
0.078431,
0.082353,
0.078431,
0.074510,
0.082353,
0.094118,
0.094118,
0.094118,
```

```
0.098039,
0.098039,
0.101961,
0.117647,
0.125490,
0.129412,
0.121569,
0.113725,
0.105882,
0.105882,
0.105882,
0.098039,
0.086275,
0.078431,
0.074510,
0.070588,
0.066667,
0.066667,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.090196,
0.129412,
0.168627,
0.203922,
0.239216,
0.262745,
0.282353,
0.360784,
0.447059,
0.482353,
0.482353,
0.470588,
0.478431,
0.517647,
0.568627,
0.584314,
0.564706,
0.564706,
0.596078,
0.619608,
0.619608,
0.619608,
0.623529,
0.623529,
0.627451,
0.631373,
0.627451,
0.619608,
0.603922,
0.603922,
0.576471,
0.541176,
0.568627,
0.588235,
0.576471,
0.564706,
0.560784,
```

```
0.560784,
0.556863,
0.549020,
0.541176,
0.533333,
0.509804,
0.486275,
0.486275,
0.494118,
0.486275,
0.462745,
0.447059,
0.447059,
0.450980,
0.435294,
0.419608,
0.407843,
0.388235,
0.372549,
0.364706,
0.352941,
0.345098,
0.333333,
0.313726,
0.290196,
0.274510,
0.250980,
0.219608,
0.184314,
0.164706,
0.152941,
0.137255,
0.121569,
0.105882,
0.094118,
0.082353,
0.078431,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.054902,
0.058824,
0.058824,
0.062745,
0.066667,
0.070588,
0.074510,
```

```
0.078431,
0.074510,
0.070588,
0.078431,
0.082353,
0.086275,
0.086275,
0.086275,
0.086275,
0.098039,
0.113725,
0.117647,
0.113725,
0.113725,
0.101961,
0.094118,
0.094118,
0.094118,
0.090196,
0.078431,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.070588,
0.082353,
0.105882,
0.121569,
0.137255,
0.156863,
0.184314,
0.200000,
0.250980,
0.349020,
0.415686,
0.419608,
0.415686,
0.423529,
0.447059,
0.501961,
0.549020,
0.545098,
0.537255,
0.564706,
0.596078,
0.596078,
0.592157,
0.596078,
0.603922,
0.607843,
0.607843,
0.607843,
0.607843,
0.600000,
0.588235,
0.580392,
```

```
0.552941,
0.525490,
0.556863,
0.576471,
0.556863,
0.545098,
0.549020,
0.545098,
0.533333,
0.525490,
0.521569,
0.521569,
0.509804,
0.478431,
0.462745,
0.478431,
0.482353,
0.466667,
0.443137,
0.423529,
0.431373,
0.427451,
0.407843,
0.400000,
0.388235,
0.364706,
0.341176,
0.321569,
0.305882,
0.298039,
0.282353,
0.262745,
0.247059,
0.227451,
0.200000,
0.168627,
0.149020,
0.137255,
0.129412,
0.117647,
0.101961,
0.090196,
0.082353,
0.074510,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
```

```
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.070588,
0.078431,
0.078431,
0.078431,
0.074510,
0.074510,
0.090196,
0.101961,
0.105882,
0.101961,
0.098039,
0.094118,
0.086275,
0.082353,
0.086275,
0.082353,
0.074510,
0.066667,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.070588,
0.090196,
0.109804,
0.129412,
0.156863,
0.180392,
0.203922,
0.231373,
0.278431,
0.337255,
0.356863,
0.333333,
0.329412,
0.372549,
0.435294,
0.490196,
0.505882,
0.513726,
0.545098,
0.576471,
0.568627,
0.549020,
0.560784,
0.576471,
0.576471,
```

```
0.576471,
0.580392,
0.576471,
0.576471,
0.576471,
0.576471,
0.572549,
0.545098,
0.521569,
0.537255,
0.556863,
0.545098,
0.529412,
0.529412,
0.529412,
0.513726,
0.501961,
0.505882,
0.505882,
0.501961,
0.478431,
0.454902,
0.458824,
0.466667,
0.458824,
0.435294,
0.407843,
0.407843,
0.407843,
0.388235,
0.372549,
0.360784,
0.341176,
0.321569,
0.305882,
0.282353,
0.266667,
0.254902,
0.239216,
0.227451,
0.207843,
0.184314,
0.156863,
0.129412,
0.113725,
0.113725,
0.109804,
0.098039,
0.086275,
0.082353,
0.074510,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.050980,
0.054902,
```

```
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.074510,
0.074510,
0.070588,
0.070588,
0.074510,
0.082353,
0.094118,
0.098039,
0.094118,
0.090196,
0.086275,
0.078431,
0.074510,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.070588,
0.082353,
0.098039,
0.129412,
0.160784,
0.192157,
0.223529,
0.239216,
0.254902,
0.274510,
0.274510,
0.262745,
0.290196,
0.356863,
0.427451,
0.458824,
0.474510,
0.513726,
```

```
0.556863,
0.564706,
0.541176,
0.509804,
0.521569,
0.552941,
0.545098,
0.533333,
0.541176,
0.541176,
0.549020,
0.556863,
0.556863,
0.560784,
0.552941,
0.529412,
0.513726,
0.517647,
0.529412,
0.521569,
0.509804,
0.509804,
0.498039,
0.482353,
0.490196,
0.490196,
0.482353,
0.474510,
0.454902,
0.439216,
0.439216,
0.443137,
0.427451,
0.400000,
0.388235,
0.384314,
0.368627,
0.352941,
0.337255,
0.313726,
0.294118,
0.290196,
0.290196,
0.274510,
0.254902,
0.231373,
0.203922,
0.188235,
0.172549,
0.149020,
0.117647,
0.098039,
0.094118,
0.094118,
0.090196,
0.082353,
0.078431,
0.074510,
0.070588,
0.062745,
0.062745,
```

```
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.066667,
0.066667,
0.070588,
0.078431,
0.082353,
0.086275,
0.090196,
0.090196,
0.086275,
0.082353,
0.074510,
0.070588,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.066667,
0.070588,
0.078431,
0.101961,
0.125490,
0.152941,
0.184314,
0.184314,
0.172549,
0.184314,
0.223529,
```

```
0.278431,
0.329412,
0.376471,
0.407843,
0.427451,
0.466667,
0.513726,
0.533333,
0.537255,
0.537255,
0.505882,
0.478431,
0.494118,
0.501961,
0.498039,
0.501961,
0.498039,
0.513726,
0.533333,
0.533333,
0.541176,
0.541176,
0.525490,
0.498039,
0.478431,
0.498039,
0.509804,
0.494118,
0.486275,
0.474510,
0.458824,
0.470588,
0.482353,
0.470588,
0.454902,
0.443137,
0.431373,
0.419608,
0.419608,
0.419608,
0.407843,
0.388235,
0.372549,
0.364706,
0.352941,
0.341176,
0.317647,
0.286275,
0.274510,
0.274510,
0.262745,
0.243137,
0.211765,
0.180392,
0.156863,
0.141176,
0.117647,
0.094118,
0.082353,
0.078431,
0.074510,
```

```
0.082353,
0.078431,
0.074510,
0.070588,
0.070588,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.070588,
0.074510,
0.078431,
0.082353,
0.082353,
0.078431,
0.078431,
0.074510,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.078431,
```

```
0.094118,
0.117647,
0.137255,
0.133333,
0.125490,
0.149020,
0.207843,
0.282353,
0.329412,
0.356863,
0.384314,
0.419608,
0.443137,
0.458824,
0.478431,
0.498039,
0.525490,
0.521569,
0.474510,
0.435294,
0.431373,
0.450980,
0.454902,
0.431373,
0.454902,
0.490196,
0.501961,
0.513726,
0.513726,
0.498039,
0.462745,
0.435294,
0.466667,
0.486275,
0.478431,
0.470588,
0.450980,
0.435294,
0.450980,
0.466667,
0.458824,
0.439216,
0.419608,
0.415686,
0.403922,
0.392157,
0.392157,
0.396078,
0.384314,
0.364706,
0.360784,
0.352941,
0.337255,
0.313726,
0.282353,
0.258824,
0.247059,
0.235294,
0.211765,
0.184314,
0.156863,
```

```
0.133333,
0.113725,
0.094118,
0.078431,
0.070588,
0.066667,
0.062745,
0.070588,
0.074510,
0.070588,
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.043137,
0.039216,
0.043137,
0.043137,
0.043137,
0.047059,
0.050980,
0.054902,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
0.070588,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
```

```
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.078431,
0.090196,
0.101961,
0.117647,
0.117647,
0.129412,
0.152941,
0.176471,
0.211765,
0.258824,
0.301961,
0.349020,
0.392157,
0.407843,
0.388235,
0.392157,
0.423529,
0.462745,
0.501961,
0.505882,
0.474510,
0.419608,
0.368627,
0.376471,
0.380392,
0.345098,
0.376471,
0.435294,
0.458824,
0.474510,
0.474510,
0.435294,
0.384314,
0.384314,
0.439216,
0.462745,
0.450980,
0.454902,
0.439216,
0.423529,
0.435294,
0.447059,
0.443137,
0.431373,
0.411765,
0.392157,
0.380392,
0.372549,
0.364706,
0.368627,
0.368627,
0.356863,
0.341176,
0.337255,
0.325490,
0.294118,
```

```
0.258824,
0.235294,
0.239216,
0.231373,
0.203922,
0.184314,
0.164706,
0.141176,
0.121569,
0.109804,
0.098039,
0.086275,
0.074510,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.054902,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.050980,
0.047059,
0.047059,
0.043137,
0.031373,
0.031373,
0.039216,
0.047059,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
```

```
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.070588,
0.078431,
0.086275,
0.094118,
0.105882,
0.125490,
0.137255,
0.145098,
0.164706,
0.192157,
0.211765,
0.231373,
0.270588,
0.325490,
0.376471,
0.388235,
0.356863,
0.349020,
0.376471,
0.407843,
0.450980,
0.474510,
0.458824,
0.431373,
0.400000,
0.337255,
0.305882,
0.301961,
0.282353,
0.341176,
0.403922,
0.415686,
0.431373,
0.427451,
0.368627,
0.298039,
0.317647,
0.400000,
0.427451,
0.423529,
0.435294,
0.427451,
0.411765,
0.427451,
0.435294,
0.427451,
0.415686,
0.403922,
0.380392,
0.352941,
0.349020,
0.349020,
```

```
0.341176,
0.337255,
0.329412,
0.305882,
0.305882,
0.309804,
0.290196,
0.250980,
0.215686,
0.211765,
0.207843,
0.180392,
0.176471,
0.172549,
0.152941,
0.129412,
0.117647,
0.105882,
0.098039,
0.082353,
0.070588,
0.054902,
0.050980,
0.054902,
0.058824,
0.054902,
0.058824,
0.062745,
0.058824,
0.050980,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.035294,
0.035294,
0.039216,
0.047059,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.074510,
```

```
0.074510,
0.074510,
0.066667,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.070588,
0.078431,
0.086275,
0.094118,
0.101961,
0.105882,
0.125490,
0.145098,
0.160784,
0.172549,
0.188235,
0.211765,
0.247059,
0.286275,
0.337255,
0.368627,
0.341176,
0.309804,
0.333333,
0.368627,
0.400000,
0.435294,
0.443137,
0.415686,
0.376471,
0.341176,
0.286275,
0.235294,
0.219608,
0.231373,
0.301961,
0.364706,
0.384314,
0.392157,
0.380392,
0.345098,
0.270588,
0.266667,
0.345098,
0.380392,
0.388235,
0.407843,
0.411765,
0.400000,
0.411765,
0.423529,
```

```
0.415686,
0.400000,
0.384314,
0.372549,
0.349020,
0.325490,
0.317647,
0.317647,
0.305882,
0.282353,
0.262745,
0.274510,
0.294118,
0.282353,
0.250980,
0.207843,
0.184314,
0.172549,
0.156863,
0.149020,
0.145098,
0.141176,
0.125490,
0.109804,
0.098039,
0.090196,
0.082353,
0.074510,
0.066667,
0.058824,
0.054902,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.047059,
0.043137,
0.047059,
0.050980,
0.050980,
0.047059,
0.047059,
0.039216,
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
```

```
0.058824,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.074510,
0.078431,
0.086275,
0.090196,
0.094118,
0.098039,
0.109804,
0.129412,
0.141176,
0.152941,
0.168627,
0.203922,
0.250980,
0.286275,
0.313726,
0.313726,
0.286275,
0.290196,
0.321569,
0.352941,
0.392157,
0.407843,
0.392157,
0.384314,
0.349020,
0.282353,
0.219608,
0.164706,
0.149020,
0.196078,
0.270588,
0.325490,
0.352941,
0.345098,
0.333333,
0.333333,
0.294118,
0.247059,
0.278431,
```

```
0.325490,
0.352941,
0.388235,
0.396078,
0.388235,
0.400000,
0.411765,
0.403922,
0.388235,
0.372549,
0.360784,
0.349020,
0.321569,
0.294118,
0.290196,
0.290196,
0.250980,
0.227451,
0.250980,
0.262745,
0.250980,
0.239216,
0.219608,
0.196078,
0.180392,
0.160784,
0.141176,
0.125490,
0.117647,
0.113725,
0.105882,
0.094118,
0.086275,
0.082353,
0.074510,
0.066667,
0.062745,
0.050980,
0.047059,
0.054902,
0.058824,
0.058824,
0.054902,
0.054902,
0.050980,
0.043137,
0.039216,
0.043137,
0.043137,
0.043137,
0.039216,
0.043137,
0.047059,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
```

```
0.058824,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.070588,
0.070588,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.066667,
0.070588,
0.078431,
0.078431,
0.078431,
0.082353,
0.090196,
0.101961,
0.105882,
0.117647,
0.133333,
0.156863,
0.196078,
0.235294,
0.262745,
0.262745,
0.243137,
0.243137,
0.270588,
0.305882,
0.345098,
0.376471,
0.356863,
0.321569,
0.325490,
0.317647,
0.258824,
0.192157,
0.137255,
0.129412,
0.192157,
0.258824,
0.286275,
```

```
0.305882,
0.290196,
0.270588,
0.294118,
0.282353,
0.231373,
0.243137,
0.294118,
0.341176,
0.372549,
0.376471,
0.384314,
0.392157,
0.388235,
0.380392,
0.364706,
0.356863,
0.349020,
0.341176,
0.329412,
0.294118,
0.266667,
0.266667,
0.247059,
0.215686,
0.223529,
0.227451,
0.211765,
0.211765,
0.215686,
0.203922,
0.188235,
0.164706,
0.141176,
0.121569,
0.105882,
0.098039,
0.098039,
0.094118,
0.086275,
0.078431,
0.070588,
0.062745,
0.054902,
0.047059,
0.043137,
0.050980,
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.047059,
0.039216,
0.035294,
0.035294,
0.031373,
0.031373,
0.039216,
0.043137,
```

```
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.074510,
0.082353,
0.094118,
0.098039,
0.105882,
0.117647,
0.137255,
0.160784,
0.184314,
0.207843,
0.219608,
0.207843,
0.200000,
0.215686,
0.247059,
0.286275,
0.325490,
0.341176,
0.305882,
0.266667,
0.266667,
0.270588,
```

```
0.243137,
0.192157,
0.137255,
0.121569,
0.176471,
0.223529,
0.239216,
0.254902,
0.239216,
0.219608,
0.247059,
0.247059,
0.215686,
0.250980,
0.305882,
0.337255,
0.356863,
0.360784,
0.372549,
0.372549,
0.360784,
0.345098,
0.329412,
0.329412,
0.333333,
0.325490,
0.313726,
0.298039,
0.262745,
0.235294,
0.231373,
0.215686,
0.196078,
0.188235,
0.184314,
0.184314,
0.188235,
0.188235,
0.176471,
0.152941,
0.129412,
0.113725,
0.098039,
0.082353,
0.078431,
0.086275,
0.082353,
0.074510,
0.066667,
0.062745,
0.058824,
0.050980,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
```

```
0.039216,
0.031373,
0.031373,
0.035294,
0.035294,
0.039216,
0.043137,
0.043137,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.074510,
0.082353,
0.090196,
0.098039,
0.109804,
0.129412,
0.145098,
0.160784,
0.172549,
0.180392,
0.176471,
0.172549,
0.184314,
0.207843,
0.235294,
```

```
0.262745,
0.290196,
0.305882,
0.270588,
0.235294,
0.227451,
0.227451,
0.223529,
0.203922,
0.152941,
0.113725,
0.133333,
0.164706,
0.188235,
0.207843,
0.196078,
0.180392,
0.207843,
0.203922,
0.196078,
0.250980,
0.294118,
0.317647,
0.337255,
0.345098,
0.352941,
0.349020,
0.337255,
0.313726,
0.290196,
0.298039,
0.309804,
0.313726,
0.298039,
0.282353,
0.262745,
0.227451,
0.200000,
0.196078,
0.184314,
0.164706,
0.156863,
0.156863,
0.160784,
0.172549,
0.164706,
0.145098,
0.125490,
0.109804,
0.094118,
0.074510,
0.066667,
0.070588,
0.074510,
0.070588,
0.066667,
0.062745,
0.062745,
0.058824,
0.054902,
```

```
0.047059,
0.043137,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.078431,
0.082353,
0.086275,
0.098039,
0.109804,
0.125490,
0.137255,
0.152941,
```

```
0.152941,
0.141176,
0.137255,
0.149020,
0.164706,
0.188235,
0.211765,
0.227451,
0.250980,
0.270588,
0.243137,
0.207843,
0.203922,
0.200000,
0.196078,
0.192157,
0.160784,
0.117647,
0.098039,
0.113725,
0.141176,
0.168627,
0.156863,
0.137255,
0.152941,
0.156863,
0.176471,
0.231373,
0.266667,
0.294118,
0.313726,
0.321569,
0.321569,
0.325490,
0.317647,
0.282353,
0.250980,
0.266667,
0.286275,
0.298039,
0.290196,
0.258824,
0.243137,
0.223529,
0.192157,
0.164706,
0.160784,
0.152941,
0.133333,
0.125490,
0.141176,
0.149020,
0.149020,
0.141176,
0.125490,
0.105882,
0.090196,
0.074510,
0.062745,
0.062745,
```

```
0.066667,
0.066667,
0.062745,
0.058824,
0.054902,
0.058824,
0.054902,
0.047059,
0.039216,
0.043137,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
0.035294,
0.039216,
0.043137,
0.047059,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.074510,
0.078431,
```

```
0.078431,
0.086275,
0.098039,
0.109804,
0.121569,
0.129412,
0.133333,
0.125490,
0.113725,
0.117647,
0.129412,
0.137255,
0.156863,
0.176471,
0.192157,
0.223529,
0.247059,
0.223529,
0.192157,
0.180392,
0.180392,
0.176471,
0.172549,
0.156863,
0.129412,
0.101961,
0.090196,
0.113725,
0.141176,
0.129412,
0.098039,
0.101961,
0.133333,
0.184314,
0.223529,
0.254902,
0.278431,
0.286275,
0.290196,
0.294118,
0.294118,
0.294118,
0.266667,
0.231373,
0.223529,
0.247059,
0.274510,
0.270588,
0.250980,
0.223529,
0.200000,
0.184314,
0.156863,
0.133333,
0.133333,
0.121569,
0.109804,
0.113725,
0.117647,
0.125490,
```

```
0.133333,
0.125490,
0.105882,
0.090196,
0.078431,
0.074510,
0.066667,
0.058824,
0.058824,
0.062745,
0.054902,
0.050980,
0.054902,
0.054902,
0.047059,
0.043137,
0.039216,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
0.039216,
0.035294,
0.035294,
0.031373,
0.035294,
0.035294,
0.035294,
0.035294,
0.039216,
0.043137,
0.047059,
0.050980,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.062745,
0.058824,
0.058824,
0.066667,
0.066667,
0.066667,
0.066667,
```

```
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.074510,
0.074510,
0.078431,
0.086275,
0.098039,
0.109804,
0.113725,
0.117647,
0.109804,
0.105882,
0.105882,
0.105882,
0.105882,
0.125490,
0.137255,
0.145098,
0.168627,
0.203922,
0.223529,
0.211765,
0.180392,
0.160784,
0.160784,
0.160784,
0.156863,
0.145098,
0.133333,
0.105882,
0.086275,
0.098039,
0.113725,
0.105882,
0.094118,
0.109804,
0.149020,
0.196078,
0.223529,
0.239216,
0.250980,
0.258824,
0.266667,
0.266667,
0.258824,
0.262745,
0.254902,
0.223529,
0.188235,
0.200000,
0.239216,
0.243137,
0.235294,
0.215686,
0.180392,
0.164706,
0.149020,
```

```
0.129412,
0.109804,
0.105882,
0.101961,
0.090196,
0.086275,
0.101961,
0.117647,
0.113725,
0.098039,
0.086275,
0.078431,
0.074510,
0.070588,
0.058824,
0.054902,
0.054902,
0.054902,
0.047059,
0.047059,
0.050980,
0.050980,
0.047059,
0.043137,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.023529,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.035294,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
```

```
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.078431,
0.086275,
0.098039,
0.101961,
0.098039,
0.094118,
0.090196,
0.098039,
0.098039,
0.086275,
0.098039,
0.113725,
0.109804,
0.117647,
0.145098,
0.176471,
0.203922,
0.203922,
0.176471,
0.149020,
0.141176,
0.137255,
0.137255,
0.133333,
0.121569,
0.098039,
0.078431,
0.082353,
0.086275,
0.090196,
0.109804,
0.141176,
0.172549,
0.200000,
0.211765,
0.215686,
0.223529,
0.235294,
0.243137,
0.235294,
0.223529,
0.223529,
0.223529,
0.207843,
0.172549,
0.160784,
```

```
0.188235,
0.203922,
0.203922,
0.188235,
0.164706,
0.145098,
0.133333,
0.125490,
0.105882,
0.090196,
0.082353,
0.074510,
0.062745,
0.078431,
0.101961,
0.101961,
0.090196,
0.078431,
0.074510,
0.070588,
0.066667,
0.058824,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.043137,
0.043137,
0.015686,
0.019608,
0.031373,
0.035294,
0.035294,
0.039216,
0.035294,
0.039216,
0.043137,
0.047059,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.062745,
0.058824,
0.058824,
```

```
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.078431,
0.082353,
0.090196,
0.090196,
0.082353,
0.078431,
0.086275,
0.094118,
0.086275,
0.074510,
0.086275,
0.098039,
0.090196,
0.105882,
0.129412,
0.152941,
0.184314,
0.192157,
0.168627,
0.145098,
0.129412,
0.117647,
0.117647,
0.117647,
0.109804,
0.090196,
0.070588,
0.070588,
0.074510,
0.090196,
0.117647,
0.145098,
0.164706,
0.184314,
0.188235,
0.188235,
0.200000,
0.211765,
0.215686,
```

```
0.203922,
0.188235,
0.188235,
0.188235,
0.184314,
0.176471,
0.149020,
0.133333,
0.141176,
0.152941,
0.145098,
0.137255,
0.129412,
0.117647,
0.109804,
0.101961,
0.086275,
0.070588,
0.058824,
0.054902,
0.074510,
0.094118,
0.094118,
0.082353,
0.078431,
0.074510,
0.070588,
0.062745,
0.054902,
0.047059,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.050980,
0.047059,
0.047059,
0.043137,
0.039216,
0.039216,
0.039216,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.023529,
0.027451,
0.035294,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.043137,
0.047059,
0.047059,
0.043137,
0.047059,
0.050980,
```

```
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.074510,
0.078431,
0.082353,
0.082353,
0.070588,
0.078431,
0.086275,
0.090196,
0.082353,
0.066667,
0.070588,
0.082353,
0.094118,
0.113725,
0.121569,
0.133333,
0.160784,
0.180392,
0.160784,
0.137255,
0.121569,
0.105882,
0.098039,
0.098039,
0.094118,
0.078431,
0.066667,
0.070588,
0.082353,
0.094118,
0.113725,
0.129412,
```

```
0.145098,
0.156863,
0.164706,
0.164706,
0.176471,
0.188235,
0.184314,
0.172549,
0.160784,
0.156863,
0.149020,
0.149020,
0.152941,
0.145098,
0.117647,
0.101961,
0.109804,
0.117647,
0.113725,
0.109804,
0.101961,
0.094118,
0.090196,
0.082353,
0.070588,
0.058824,
0.054902,
0.074510,
0.090196,
0.090196,
0.082353,
0.078431,
0.074510,
0.070588,
0.062745,
0.054902,
0.050980,
0.043137,
0.043137,
0.043137,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.043137,
0.043137,
0.039216,
0.039216,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.031373,
0.031373,
0.035294,
0.039216,
0.043137,
0.039216,
0.039216,
```

```
0.039216,
0.047059,
0.047059,
0.043137,
0.039216,
0.043137,
0.047059,
0.047059,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.066667,
0.066667,
0.062745,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.078431,
0.078431,
0.074510,
0.070588,
0.078431,
0.086275,
0.082353,
0.074510,
0.066667,
0.074510,
0.090196,
0.105882,
0.117647,
0.109804,
0.101961,
0.129412,
0.160784,
0.152941,
0.129412,
0.113725,
0.098039,
0.086275,
0.082353,
0.082353,
```

```
0.074510,
0.066667,
0.070588,
0.082353,
0.098039,
0.101961,
0.105882,
0.121569,
0.133333,
0.141176,
0.145098,
0.152941,
0.164706,
0.160784,
0.149020,
0.133333,
0.129412,
0.125490,
0.113725,
0.113725,
0.121569,
0.125490,
0.109804,
0.090196,
0.094118,
0.098039,
0.090196,
0.086275,
0.082353,
0.078431,
0.074510,
0.074510,
0.066667,
0.054902,
0.070588,
0.082353,
0.078431,
0.078431,
0.078431,
0.074510,
0.070588,
0.062745,
0.054902,
0.050980,
0.047059,
0.047059,
0.050980,
0.047059,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.043137,
0.043137,
0.039216,
0.039216,
0.035294,
0.035294,
0.035294,
0.035294,
```

```
0.031373,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.035294,
0.039216,
0.043137,
0.047059,
0.043137,
0.035294,
0.039216,
0.047059,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.066667,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.070588,
0.066667,
0.070588,
0.074510,
0.074510,
0.070588,
0.066667,
0.078431,
0.098039,
0.109804,
0.109804,
0.098039,
0.086275,
0.105882,
0.133333,
```

```
0.129412,
0.121569,
0.113725,
0.105882,
0.094118,
0.082353,
0.074510,
0.070588,
0.066667,
0.066667,
0.078431,
0.090196,
0.090196,
0.086275,
0.094118,
0.109804,
0.121569,
0.129412,
0.137255,
0.141176,
0.133333,
0.129412,
0.117647,
0.109804,
0.109804,
0.098039,
0.082353,
0.094118,
0.113725,
0.113725,
0.094118,
0.074510,
0.078431,
0.082353,
0.074510,
0.070588,
0.070588,
0.066667,
0.066667,
0.058824,
0.050980,
0.058824,
0.070588,
0.070588,
0.074510,
0.074510,
0.070588,
0.066667,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
```

```
0.043137,
0.043137,
0.039216,
0.035294,
0.031373,
0.031373,
0.035294,
0.031373,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.039216,
0.043137,
0.047059,
0.039216,
0.035294,
0.043137,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.062745,
0.062745,
0.062745,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.070588,
0.082353,
```

```
0.098039,
0.101961,
0.101961,
0.090196,
0.078431,
0.086275,
0.105882,
0.105882,
0.105882,
0.105882,
0.109804,
0.101961,
0.090196,
0.070588,
0.058824,
0.058824,
0.062745,
0.070588,
0.082353,
0.082353,
0.074510,
0.082353,
0.094118,
0.105882,
0.117647,
0.125490,
0.117647,
0.113725,
0.109804,
0.105882,
0.098039,
0.094118,
0.082353,
0.066667,
0.070588,
0.090196,
0.098039,
0.094118,
0.078431,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.058824,
0.050980,
0.043137,
0.054902,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.050980,
```

```
0.050980,
0.047059,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.035294,
0.027451,
0.027451,
0.027451,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.043137,
0.039216,
0.031373,
0.031373,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.054902,
0.050980,
0.047059,
0.050980,
0.054902,
0.050980,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
```

0.062745,
0.054902,
0.054902,
0.058824,
0.058824,
0.070588,
0.078431,
0.090196,
0.094118,
0.090196,
0.082353,
0.074510,
0.078431,
0.086275,
0.090196,
0.090196,
0.094118,
0.101961,
0.098039,
0.086275,
0.070588,
0.058824,
0.054902,
0.058824,
0.066667,
0.074510,
0.074510,
0.070588,
0.074510,
0.086275,
0.098039,
0.105882,
0.109804,
0.105882,
0.098039,
0.094118,
0.094118,
0.090196,
0.086275,
0.070588,
0.058824,
0.062745,
0.078431,
0.082353,
0.082353,
0.078431,
0.066667,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.050980,
0.047059,
0.054902,
0.066667,
0.066667,
0.062745,
0.066667,
0.066667,

```
0.066667,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.047059,
0.043137,
0.043137,
0.039216,
0.039216,
0.035294,
0.039216,
0.039216,
0.031373,
0.027451,
0.023529,
0.019608,
0.035294,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.031373,
0.027451,
0.031373,
0.039216,
0.043137,
0.047059,
0.050980,
0.050980,
0.047059,
0.043137,
0.043137,
0.047059,
0.050980,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.062745,
```

```
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.058824,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.074510,
0.082353,
0.086275,
0.082353,
0.074510,
0.074510,
0.074510,
0.074510,
0.074510,
0.086275,
0.094118,
0.094118,
0.090196,
0.082353,
0.070588,
0.062745,
0.058824,
0.054902,
0.062745,
0.070588,
0.066667,
0.066667,
0.070588,
0.078431,
0.086275,
0.094118,
0.098039,
0.094118,
0.086275,
0.078431,
0.078431,
0.078431,
0.078431,
0.070588,
0.058824,
0.058824,
0.070588,
0.074510,
0.074510,
0.074510,
0.070588,
0.062745,
0.054902,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
```

```
0.047059,
0.054902,
0.062745,
0.062745,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.043137,
0.039216,
0.035294,
0.031373,
0.035294,
0.035294,
0.035294,
0.031373,
0.023529,
0.015686,
0.031373,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.039216,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.035294,
0.043137,
0.047059,
0.047059,
0.047059,
0.043137,
0.043137,
0.047059,
0.047059,
0.047059,
0.050980,
0.054902,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
```

```
0.058824,
0.058824,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.062745,
0.066667,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.074510,
0.078431,
0.078431,
0.074510,
0.070588,
0.070588,
0.074510,
0.070588,
0.070588,
0.078431,
0.090196,
0.086275,
0.082353,
0.078431,
0.074510,
0.070588,
0.062745,
0.054902,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.074510,
0.082353,
0.090196,
0.090196,
0.086275,
0.082353,
0.074510,
0.070588,
0.070588,
0.070588,
0.066667,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
0.070588,
0.066667,
```

```
0.066667,
0.058824,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.058824,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.047059,
0.043137,
0.043137,
0.043137,
0.039216,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.023529,
0.027451,
0.027451,
0.027451,
0.031373,
0.031373,
0.035294,
0.039216,
0.039216,
0.031373,
0.027451,
0.027451,
0.031373,
0.039216,
0.047059,
0.047059,
0.043137,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
```

```
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.050980,
0.054902,
0.062745,
0.066667,
0.066667,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.058824,
0.054902,
0.054902,
0.058824,
0.070588,
0.078431,
0.074510,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.066667,
0.074510,
0.082353,
0.082353,
0.078431,
0.074510,
0.070588,
0.070588,
0.062745,
0.054902,
0.058824,
0.066667,
0.066667,
0.062745,
0.070588,
0.074510,
0.082353,
0.086275,
0.086275,
0.082353,
0.078431,
0.074510,
0.066667,
0.062745,
0.066667,
0.062745,
```

```
0.058824,
0.058824,
0.062745,
0.062745,
0.066667,
0.066667,
0.062745,
0.066667,
0.062745,
0.054902,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.050980,
0.058824,
0.058824,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.031373,
0.023529,
0.023529,
0.023529,
0.027451,
0.031373,
0.035294,
0.035294,
0.035294,
0.031373,
0.027451,
0.027451,
0.031373,
0.043137,
0.047059,
0.043137,
0.043137,
0.047059,
0.050980,
```

```
0.050980,
0.050980,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.050980,
0.050980,
0.054902,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.058824,
0.054902,
0.058824,
0.058824,
0.066667,
0.070588,
0.070588,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.066667,
0.074510,
0.078431,
0.078431,
0.070588,
0.066667,
0.066667,
0.062745,
0.054902,
0.058824,
0.066667,
0.066667,
0.062745,
0.066667,
0.074510,
0.078431,
0.078431,
0.078431,
```

```
0.078431,
0.078431,
0.074510,
0.066667,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.047059,
0.054902,
0.054902,
0.054902,
0.054902,
0.058824,
0.054902,
0.050980,
0.047059,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.019608,
0.023529,
0.027451,
0.027451,
0.031373,
0.035294,
0.031373,
0.027451,
0.027451,
0.027451,
0.035294,
```

```
0.039216,
0.043137,
0.047059,
0.043137,
0.039216,
0.043137,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.062745,
0.066667,
0.070588,
0.074510,
0.074510,
0.070588,
0.066667,
0.062745,
0.058824,
0.054902,
0.058824,
0.066667,
```

```
0.066667,
0.062745,
0.062745,
0.070588,
0.070588,
0.070588,
0.074510,
0.074510,
0.074510,
0.070588,
0.070588,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.062745,
0.058824,
0.054902,
0.047059,
0.047059,
0.043137,
0.043137,
0.047059,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.039216,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.031373,
0.027451,
0.027451,
0.019608,
0.023529,
0.027451,
0.027451,
```

```
0.031373,
0.031373,
0.027451,
0.031373,
0.031373,
0.035294,
0.039216,
0.043137,
0.043137,
0.043137,
0.043137,
0.039216,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.062745,
0.062745,
0.062745,
0.062745,
0.058824,
0.058824,
0.062745,
0.062745,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.070588,
```

```
0.066667,
0.066667,
0.062745,
0.054902,
0.050980,
0.058824,
0.062745,
0.066667,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.070588,
0.070588,
0.070588,
0.070588,
0.070588,
0.066667,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.062745,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.054902,
0.050980,
0.050980,
0.047059,
0.043137,
0.039216,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.043137,
0.043137,
0.043137,
0.039216,
0.035294,
0.031373,
0.031373,
0.035294,
0.035294,
0.035294,
0.035294,
0.031373,
0.031373,
0.031373,
0.027451,
```

```
0.027451,
0.027451,
0.023529,
0.023529,
0.023529,
0.027451,
0.027451,
0.027451,
0.027451,
0.027451,
0.031373,
0.035294,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.039216,
0.043137,
0.047059,
0.047059,
0.047059,
0.047059,
0.050980,
0.050980,
0.050980,
0.047059,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.054902,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.058824,
0.062745,
0.058824,
0.054902,
0.054902,
0.062745,
0.062745,
0.062745,
0.062745,
0.054902,
0.054902,
0.062745,
0.062745,
0.062745,
0.054902,
```

```
0.054902,
0.058824,
0.058824,
0.058824,
0.062745,
0.066667,
0.066667,
0.066667,
0.062745,
0.062745,
0.058824,
0.054902,
0.058824,
0.062745,
0.066667,
0.066667,
0.062745,
0.062745,
0.062745,
0.062745,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.066667,
0.058824,
0.054902,
0.054902,
0.058824,
0.058824,
0.058824,
0.054902,
0.050980,
0.054902,
0.054902,
0.054902,
0.054902,
0.050980,
0.050980,
0.047059,
0.039216,
0.047059,
0.050980,
0.050980,
0.050980,
0.050980,
0.050980,
0.047059,
0.047059,
0.047059,
0.043137,
0.043137,
0.043137,
0.039216,
0.039216,
0.031373,
0.027451,
0.031373,
```

```
                                0.035294,
                                0.031373,
                                0.031373,
                                0.031373,
                                0.027451,
                                0.027451,
                                0.023529,
                                0.023529,
                                0.019608,
                                0.023529,
                      ]
          }
}
```

```
#VRML V2.0 utf8
Shape {
        appearance Appearance {
                material Material { }
        }
        geometry ElevationGrid {
                xDimension  9
                zDimension  9
                xSpacing    1.0
                zSpacing    1.0
                solid       FALSE
                creaseAngle 0.785
                height [
                        0.0, 0.0, 0.5, 1.0, 0.5, 0.0, 0.0, 0.0, 0.0,
                        0.0, 0.0, 0.0, 0.0, 2.5, 0.5, 0.0, 0.0, 0.0,
                        0.0, 0.0, 0.5, 0.5, 3.0, 1.0, 0.5, 0.0, 1.0,
                        0.0, 0.0, 0.5, 2.0, 4.5, 2.5, 1.0, 1.5, 0.5,
                        1.0, 2.5, 3.0, 4.5, 5.5, 3.5, 3.0, 1.0, 0.0,
                        0.5, 2.0, 2.0, 2.5, 3.5, 4.0, 2.0, 0.5, 0.0,
                        0.0, 0.0, 0.5, 1.5, 1.0, 2.0, 3.0, 1.5, 0.0,
                        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, 1.5, 0.5,
                        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0,
                ]
        }
}
```

Construindo formas extrudadas

Motivação

Exemplos

Criando formas extrudadas

Extrusões ao longo de uma linha reta

Extrusões ao redor de um círculo

Extrusões ao longo de uma hélice

Sintaxe: Extrusion

Torcendo formas extrudadas

Sintaxe: Extrusion

Exemplo de extrusões com escala e rotação

Resumo

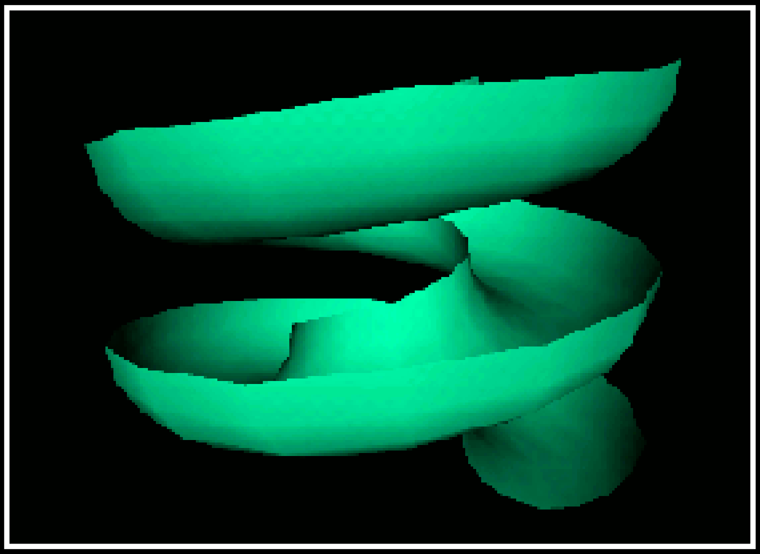
Construindo formas extrudadas

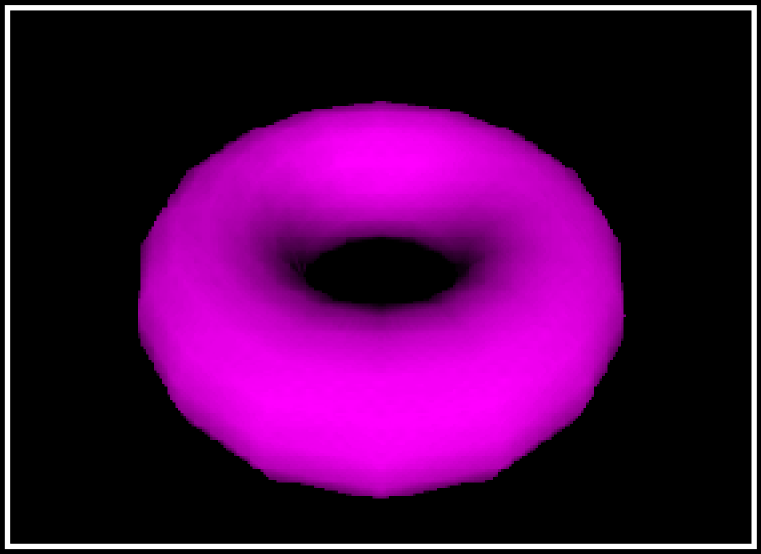
Motivação

- **Formas extrudadas são muito comuns**
  - **Tubos, canos, barreiras, vasos, donuts**
  - **Outros usos . . .**
- **Você pode construir formas extrudadas usando um nó `IndexedFaceSet`**
- **Você pode construir formas extrudadas mais facilmente e eficientemente usando o nó `Extrusion`**

Construindo formas extrudadas

Exemplos





[slide.wrl] [donut.wrl]

Construindo formas extrudadas

Criando formas extrudadas

. Formas extrudadas são descritas por:

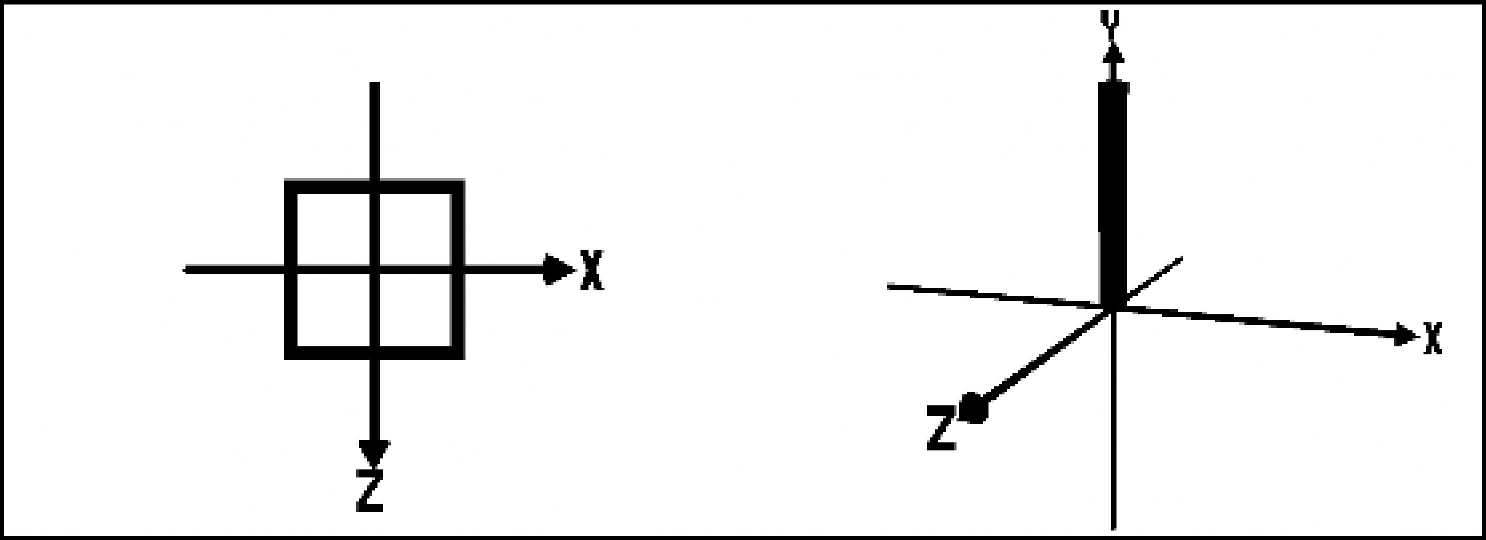
- **Um *corte trasnversal* 2D**
- **Uma *espinha* 3D ao longo da qual o corte transversal é varrido**

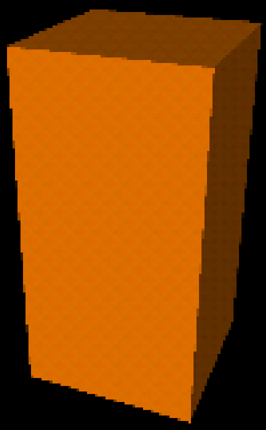
. Formas extrudadas são como bolhas longas criadas com um seção transversal comum

- **O caminho ao longo do qual você move o seção é a espinha**

Construindo formas extrudadas

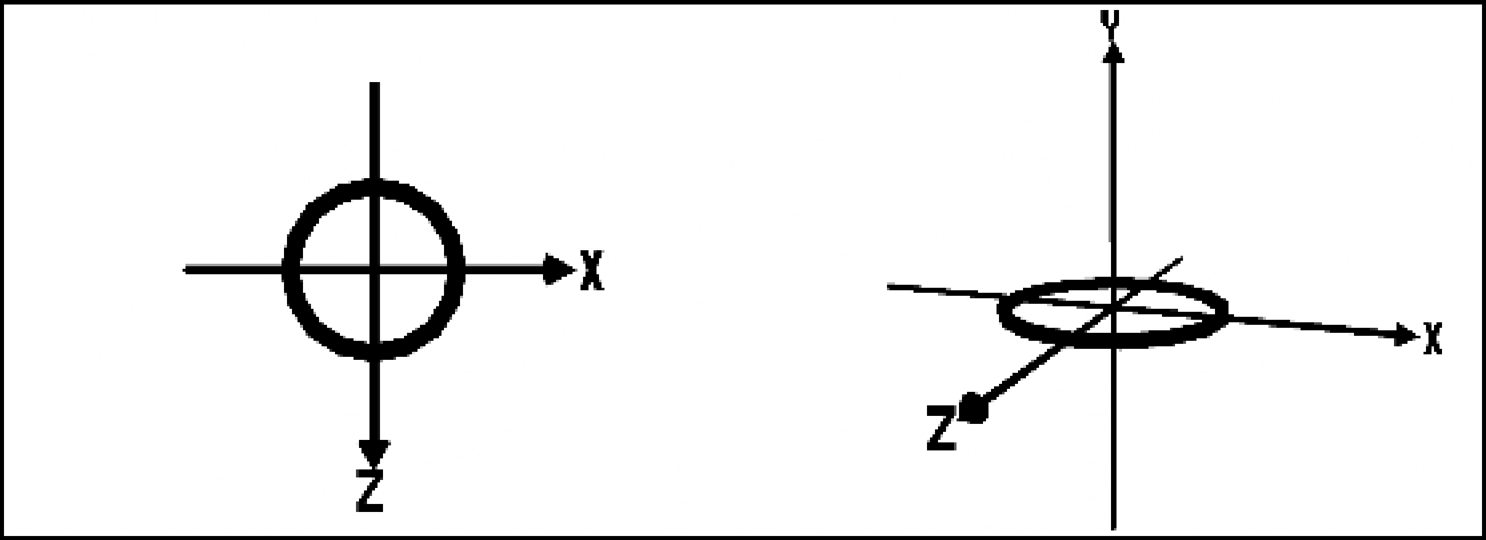
Extrusão ao longo de uma linha reta

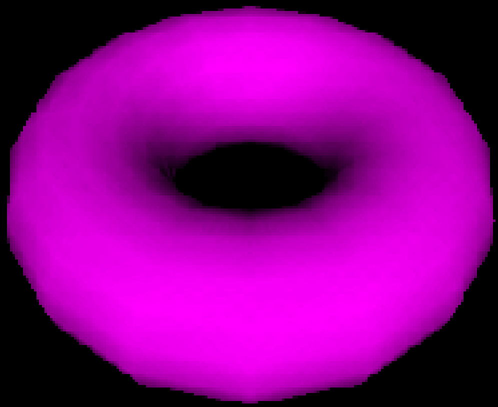




Construindo formas extrudadas

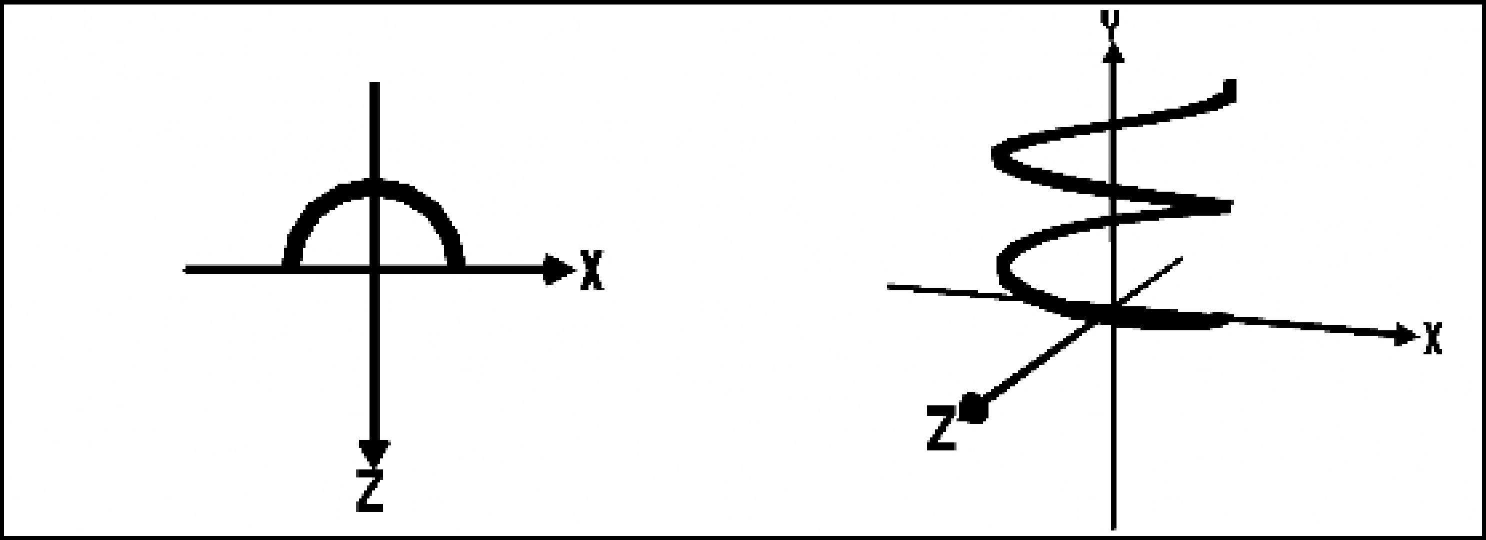
Extrusão ao redor de um círculo

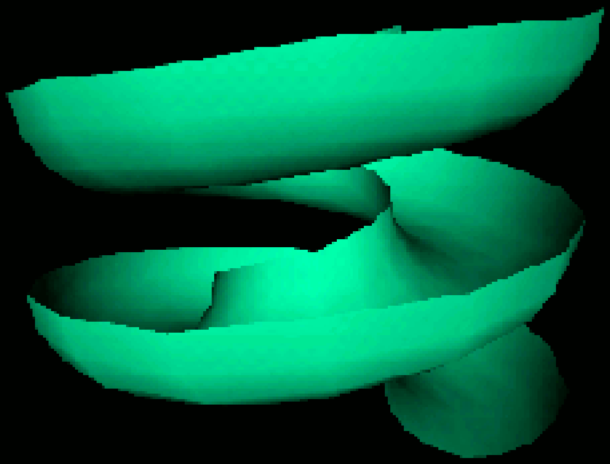




Construindo formas extrudadas

Extrusão ao longo de uma hélice





Construindo formas extrudadas

Sintaxe: Extrusion

- **Um nó geométrico `Extrusion` cria uma geometria extruded**
  - **2-D *cross-section* - corte transversal**
  - **3-D *spine* - caminho de varredura**
  - ***mais* . . .**

```
Extrusion {
    crossSection [ . . . ]
    spine [ . . .  ]
    . . .
}
```

Construindo formas extrudadas

Torcendo formas extrudadas

- **Você pode percorrer o corte transversal ao longo da espinha**
  - **Vasos, instrumentos musicais**
  - **Superfícies de revolução**
- **Você pode rotacionar o corte transversal ao longo da espinha**
  - **Fitas torcidas**

Construindo formas extrudadas

Sintaxe: Extrusion

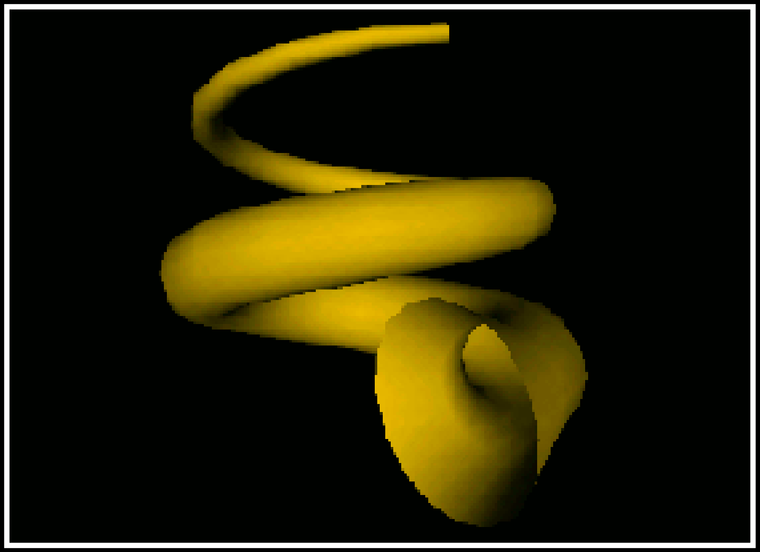
. Um nó geométrico `Extrusion` cria uma geometria usando

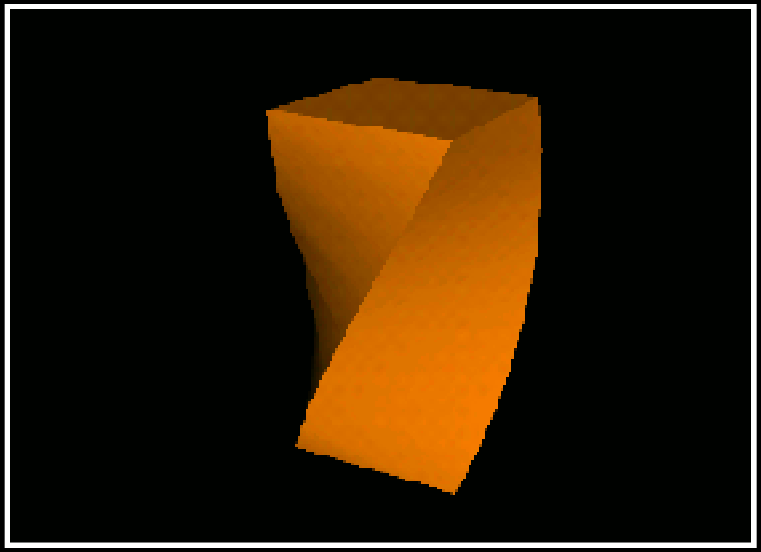
- ***scales* - subir o corte transversal pelos pontos da espinha**
- ***rotations* - rotacionar o corte transversal pelos pontos da espinha**

```
Extrusion {
    . . .
    scale [ . . . ]
    orientation [ . . . ]
}
```

Construindo formas extrudadas

Exemplos de extrusões com escala e rotação





[horn.wrl] [bartwist.wrl]

Construindo formas extrudadas

Resumo

. Um nó `Extrusion` cria formas extrudadas eficientemente

. O campo `crossSection` especifica o corte transversal

. O campo `spine` especifica o caminho a ser varrido

. Os campos `scale` e `orientation` especificam escala e rotação em cada ponto da espinha

```
#VRML V2.0 utf8
Shape {
        appearance Appearance {
                material Material {
                        diffuseColor 0.0 1.0 0.7
                }
        }
        geometry Extrusion {
                creaseAngle 1.57
                endCap   FALSE
                beginCap FALSE
                solid    FALSE
                crossSection [
                # Half-circle
                        -1.00  0.00,  -0.92 -0.38,
                        -0.71 -0.71,  -0.38 -0.92,
                         0.00 -1.00,   0.38 -0.92,
                         0.71 -0.71,   0.92 -0.38,
                         1.00  0.00,
                ]
                spine [
                # Helix
                         2.00 0.00 -0.00,   1.85 0.12 -0.77,
                         1.41 0.24 -1.41,   0.77 0.36 -1.85,
                         0.00 0.48 -2.00,  -0.77 0.61 -1.85,
                        -1.41 0.73 -1.41,  -1.85 0.85 -0.77,
                        -2.00 0.97  0.00,  -1.85 1.09  0.77,
                        -1.41 1.21  1.41,  -0.77 1.33  1.85,
                         0.00 1.45  2.00,   0.77 1.58  1.85,
                         1.41 1.70  1.41,   1.85 1.82  0.77,
                         2.00 1.94  0.00,   1.85 2.06 -0.77,
                         1.41 2.18 -1.41,   0.77 2.30 -1.85,
                         0.00 2.42 -2.00,  -0.77 2.55 -1.85,
                        -1.41 2.67 -1.41,  -1.85 2.79 -0.77,
                        -2.00 2.91  0.00,  -1.85 3.03  0.77,
                        -1.41 3.15  1.41,  -0.77 3.27  1.85,
                         0.00 3.39  2.00,   0.77 3.52  1.85,
                         1.41 3.64  1.41,   1.85 3.76  0.77,
                         2.00 3.88  0.00,
                ]
        }
}
```

```
#VRML V2.0 utf8
Shape {
        appearance Appearance {
                material Material {
                        diffuseColor 1.0 0.0 1.0
                }
        }
        geometry Extrusion {
                creaseAngle 1.57
                beginCap FALSE
                endCap   FALSE
                crossSection [
                # Circle
                         1.00  0.00,   0.92 -0.38,
                         0.71 -0.71,   0.38 -0.92,
                         0.00 -1.00,  -0.38 -0.92,
                        -0.71 -0.71,  -0.92 -0.38,
                        -1.00 -0.00,  -0.92  0.38,
                        -0.71  0.71,  -0.38  0.92,
                         0.00  1.00,   0.38  0.92,
                         0.71  0.71,   0.92  0.38,
                         1.00  0.00
                ]
                spine [
                # Circle
                         2.00 0.0  0.00,   1.85 0.0 0.77,
                         1.41 0.0  1.41,   0.77 0.0 1.85,
                         0.00 0.0  2.00,  -0.77 0.0 1.85,
                        -1.41 0.0  1.41,  -1.85 0.0 0.77,
                        -2.00 0.0  0.00,  -1.85 0.0 -0.77,
                        -1.41 0.0 -1.41,  -0.77 0.0 -1.85,
                         0.00 0.0 -2.00,   0.77 0.0 -1.85,
                         1.41 0.0 -1.41,   1.85 0.0 -0.77,
                         2.00 0.0  0.00,
                ]
        }
}
```

```
#VRML V2.0 utf8
Shape {
    appearance Appearance {
        material Material {
            diffuseColor 0.9 0.7 0.0
        }
    }
    geometry Extrusion {
        beginCap FALSE
        endCap FALSE
        creaseAngle 1.57
        solid FALSE
        crossSection [
        # Circle
             1.00  0.00,   0.92 -0.38,
             0.71 -0.71,   0.38 -0.92,
             0.00 -1.00,  -0.38 -0.92,
            -0.71 -0.71,  -0.92 -0.38,
            -1.00  0.00,  -0.92  0.38,
            -0.71  0.71,  -0.38  0.92,
             0.00  1.00,   0.38  0.92,
             0.71  0.71,   0.92  0.38,
             1.00  0.00
        ]
        spine [
        # Helix
             2.00 0.00 -0.00,   1.85 0.12 -0.77,
             1.41 0.24 -1.41,   0.77 0.36 -1.85,
             0.00 0.48 -2.00,  -0.77 0.61 -1.85,
            -1.41 0.73 -1.41,  -1.85 0.85 -0.77,
            -2.00 0.97  0.00,  -1.85 1.09  0.77,
            -1.41 1.21  1.41,  -0.77 1.33  1.85,
             0.00 1.45  2.00,   0.77 1.58  1.85,
             1.41 1.70  1.41,   1.85 1.82  0.77,
             2.00 1.94  0.00,   1.85 2.06 -0.77,
             1.41 2.18 -1.41,   0.77 2.30 -1.85,
             0.00 2.42 -2.00,  -0.77 2.55 -1.85,
            -1.41 2.67 -1.41,  -1.85 2.79 -0.77,
            -2.00 2.91  0.00,  -1.85 3.03  0.77,
            -1.41 3.15  1.41,  -0.77 3.27  1.85,
             0.00 3.39  2.00,   0.77 3.52  1.85,
             1.41 3.64  1.41,   1.85 3.76  0.77,
             2.00 3.88  0.00,
        ]
        scale [
            1.200 1.200,  0.778 0.778,  0.756 0.756,  0.734 0.734,
            0.712 0.712,  0.690 0.690,  0.668 0.668,  0.646 0.646,
            0.625 0.625,  0.603 0.603,  0.581 0.581,  0.559 0.559,
            0.537 0.537,  0.515 0.515,  0.493 0.493,  0.471 0.471,
            0.450 0.450,  0.428 0.428,  0.406 0.406,  0.384 0.384,
            0.362 0.362,  0.340 0.340,  0.318 0.318,  0.296 0.296,
            0.275 0.275,  0.253 0.253,  0.231 0.231,  0.209 0.209,
            0.187 0.187,  0.165 0.165,  0.143 0.143,  0.121 0.121,
            0.100 0.100,
        ]
    }
}
```

```
#VRML V2.0 utf8

Shape {
	appearance Appearance {
		material Material {
			diffuseColor 1.0 0.5 0.0
		}
	}
	geometry Extrusion {
		creaseAngle 0.785
		crossSection [
		# Square
			-1.0  1.0,   1.0  1.0,
			 1.0 -1.0,  -1.0 -1.0,
			-1.0  1.0
		]
		spine [
		# Straight-line
			0.0 0.0 0.0,
			0.0 0.5 0.0,
			0.0 1.0 0.0,
			0.0 1.5 0.0,
			0.0 2.0 0.0,
			0.0 2.5 0.0,
			0.0 3.0 0.0,
			0.0 3.5 0.0,
			0.0 4.0 0.0
		]
		orientation [
			0.0 1.0 0.0 0.0,
			0.0 1.0 0.0 0.175,
			0.0 1.0 0.0 0.349,
			0.0 1.0 0.0 0.524,
			0.0 1.0 0.0 0.698,
			0.0 1.0 0.0 0.873,
			0.0 1.0 0.0 1.047,
			0.0 1.0 0.0 1.222,
			0.0 1.0 0.0 1.396,
		]
	}
}
```

Controlando propriedades de geometria baseadas em coordenadas

---

Motivação

Exemplo

Sintaxe: Color

Cores forçadas

Sintaxe: PointSet

Um exemplo de forma com PointSet

Sintaxe: IndexedLineSet

Controlando faixa de cores para conjuntos de linhas

Um exemplo de forma com IndexedLineSet

Sintaxe: IndexedFaceSet

Controlando faixa de cores para conjuntos de faces

Um exemplo de forma com IndexedFaceSet

Sintaxe: ElevationGrid

Controlando faixa de cores para grids elevados

[Um exemplo de forma com ElevationGrid]

[Controlando nuanças usando o ângulo crease]

[Selecionando ângulos crease]

[Um exemplo usando ângulos crease]

[Sintaxe: Normal]

[Sintaxe: IndexedFaceSet]

[Controlando a faixa de normais para conjuntos de faces]

[Sintaxe: ElevationGrid]

[Controlando a faixa de normais para grids elevados]

[Sintaxe: NormalInterpolator]

[Resumo]

[Resumo]

Controlando propriedades de geometria baseadas em coordenadas

Motivação

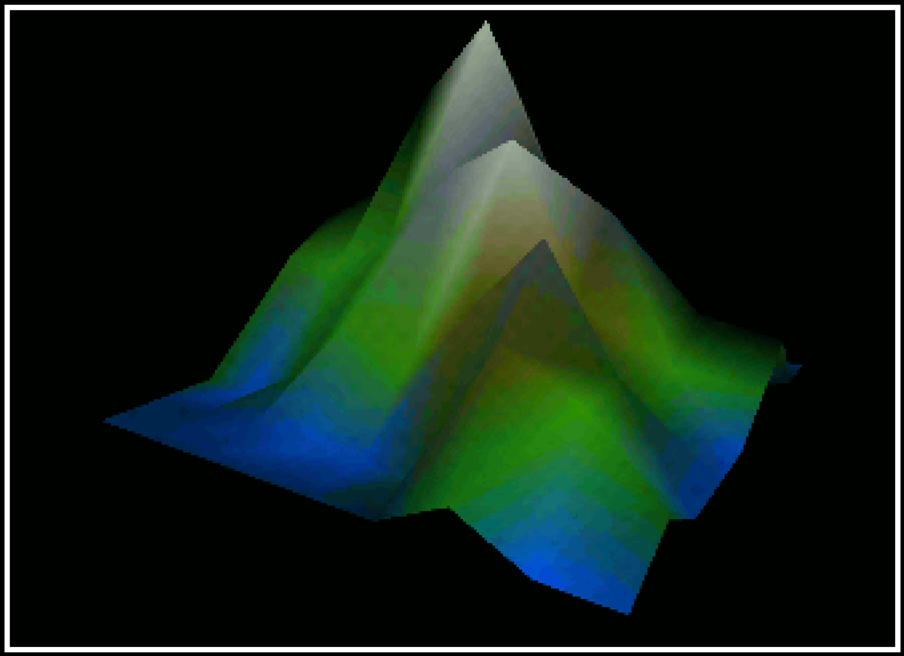
. O nó `Material` fornece uma forma inteira com a mesma cor

. Você pode fornecer cores para partes de uma forma usando o nó `Color`

. Você pode especificar sombras lisa ou facetada usando o valor do campo `creaseAngle`

Controlando propriedades de geometria baseadas em coordenadas

Exemplo



[cmount.wrl]

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: Color

. Um nó `Color` contém uma lista de valores RGB

```
Color {
    color [ 1.0 0.0 0.0, . . . ]
}
```

. Usado como o valor do campo `color` dos nós `IndexedFaceSet`, `IndexedLineSet`, `PointSet` ou `ElevationGrid`

Controlando propriedades de geometria baseadas em coordenadas

Cores forçadas

- **Cores no nó `Color` sobrescrevem aquelas no nó `Material`**
- **Você pode forçar cores**
  - **Para cada ponto, linha ou superfície**
  - **Para cada coordenada em uma linha ou superfície**

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: PointSet

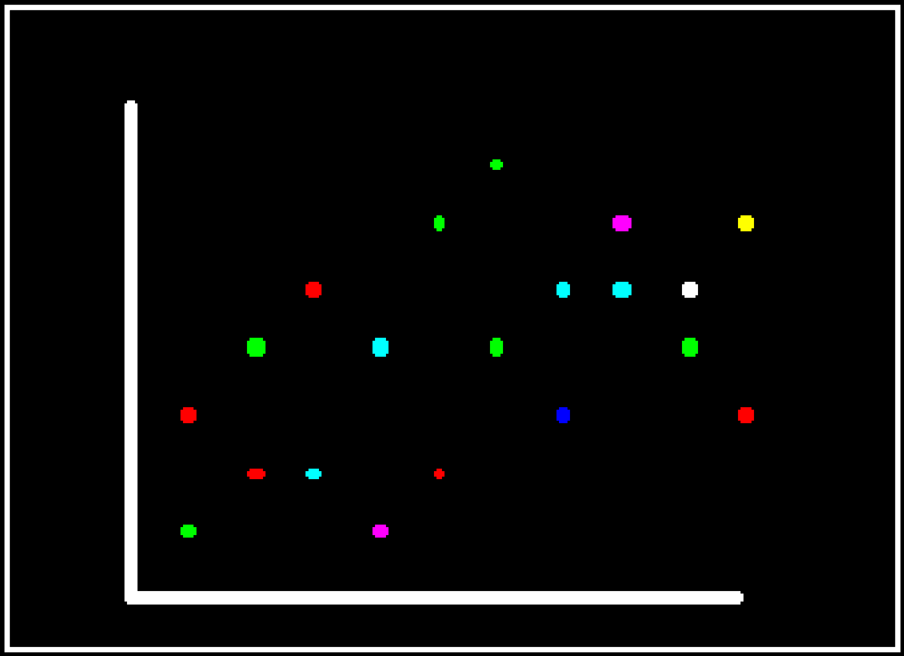
. Um nó geométrico `PointSet` cria geometria dentre pontos

- ***color*** **- fornece uma lista de cores**
- **Sempre força uma cor para cada ponto, em ordem**

```
PointSet {
    coord Coordinate { . . . }
    color Color { . . . }
}
```

Controlando propriedades de geometria baseadas em coordenadas

Um exemplo de forma com PointSet



[scatter.wrl]

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: IndexedLineSet

. Um nó geométrico `IndexedLineSet` cria linhas especificando

- *color* - uma lista de cores
- *color index* - seleciona cores de uma lista (tal qual a seleção de coordenadas)
- *color per vertex* - controla a obrigação de cores

```
IndexedLineSet {
    coord Coordinate { . . . }
    coordIndex [ . . . ]
    color Color { . . . }
    colorIndex [ . . . ]
    colorPerVertex TRUE
}
```

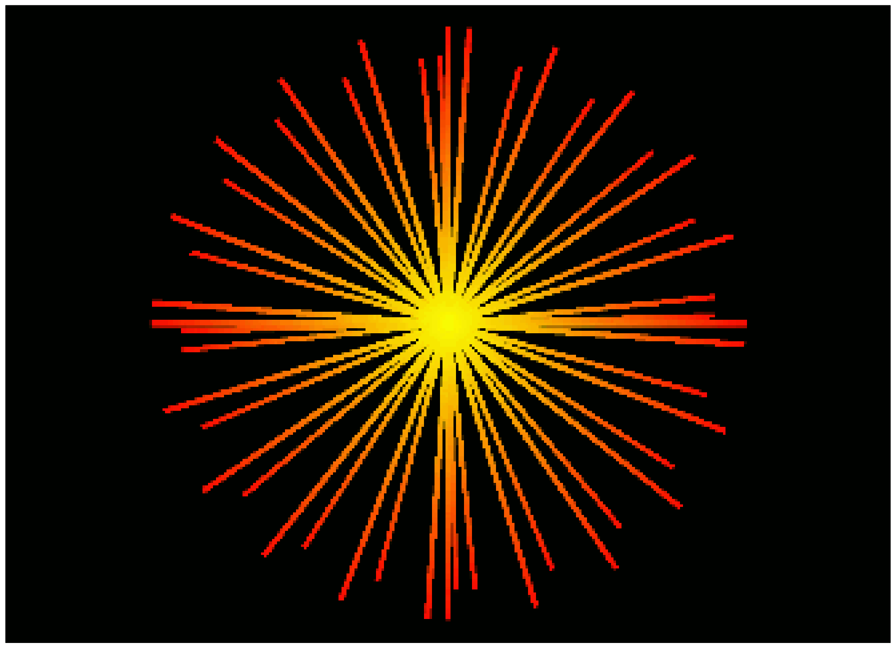
Controlando propriedades de geometria baseadas em coordenadas

Controlando faixa de cores para conjunto de linhas

- **O campo `colorPerVertex` controla como o índice de cores são usadas**
  - **`FALSE`: um índice de cor para cada linha (termina com -1 no índice de coordenadas)**
  - **`TRUE`: um índice de cor para cada índice de coordenada de cada linha (incluindo o índice de coordenadas -1)**

Controlando propriedades de geometria baseadas em coordenadas

Um exemplo de forma com IndexedLineSet



[**burst.wrl**]

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: IndexedFaceSet

. Um nó geométrico IndexedFaceSet cria geometria entre superfícies

- ***color*** **- uma lista de cores**
- ***color indexes*** **- seleciona cores de uma lista (como a seleção de coordenadas)**
- ***color per vertex*** **- controle de faixa de cores**

```
IndexedFaceSet {
    coord Coordinate { . . . }
    coordIndex [ . . . ]
    color Color { . . . }
    colorIndex [ . . . ]
    colorPerVertex TRUE
}
```

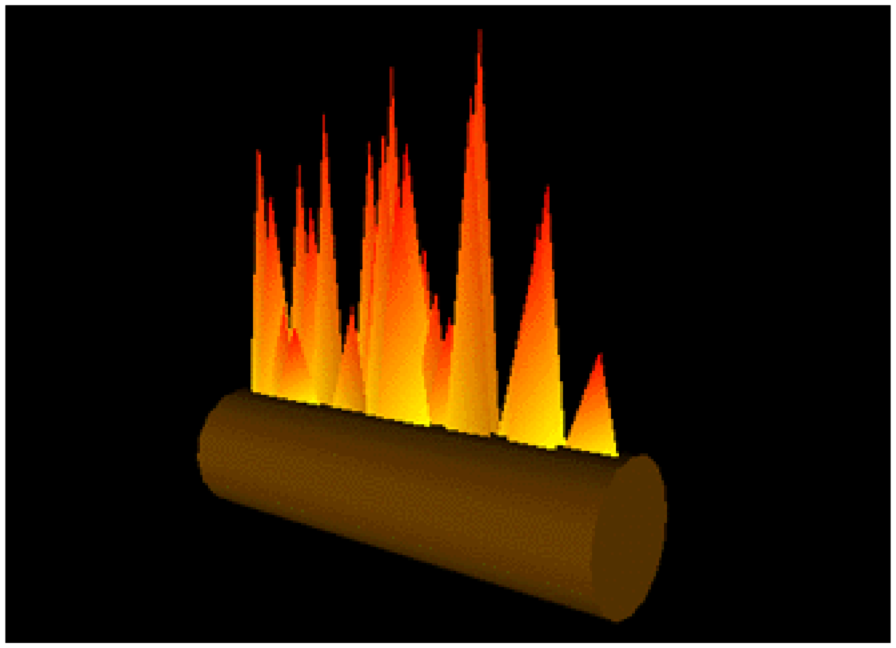
Controlando propriedades de geometria baseadas em coordenadas

Controlando a faixa de cores para conjunto de faces

- **O campo `colorPerVertex` controla quantos índices de cores são usados (similar ao conjunto de linhas)**
  - **`FALSE`: um índice de cor para cada face (finalizando com -1 o índice de coordenadas)**
  - **`TRUE`: um índice de cor para cada índice de coordenada de cada face (incluindo o índice de coordenada -1)**

Controlando propriedades de geometria baseadas em coordenadas

Um exemplo de forma com IndexedFaceSet



[log.wrl]

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: ElevationGrid

- **Um nó geométrico `ElevationGrid` cria terrenos**
  - ***color*** **- uma lista de cores**
  - ***color per vertex*** **- controla a faixa de cores**

```
ElevationGrid {
    height [ . . . ]
    color Color { . . . }
    colorPerVertex TRUE
}
```

- **O nó `ElevationGrid` não usa índice de cores**

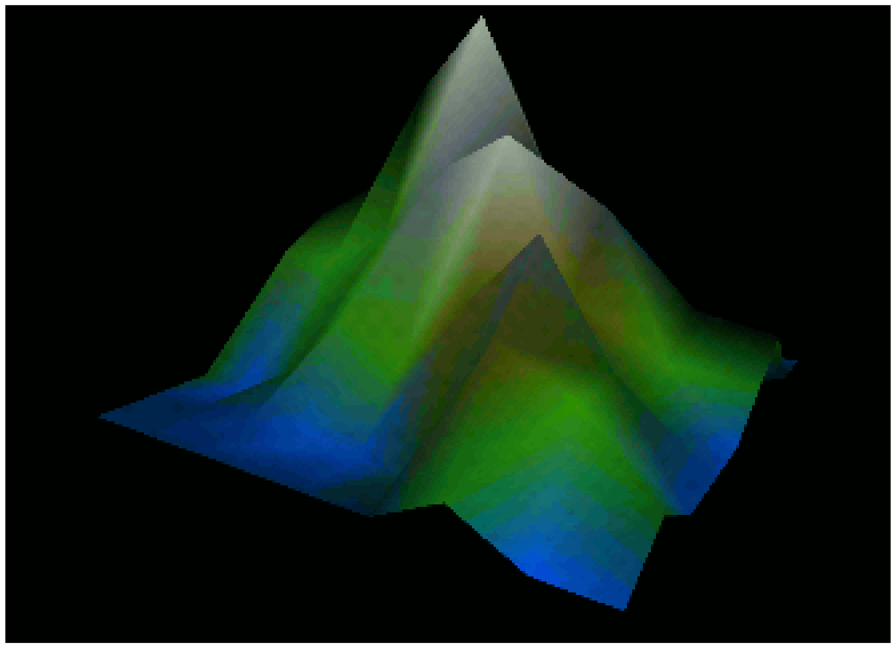
Controlando propriedades de geometria baseadas em coordenadas

Controlando faixa de cores de grids elevados

- **O campo `colorPerVertex` controla quanto índices de cores são usados (similar aos conjuntos de linhas e faces)**
  - **`FALSE`: uma cor para cada quadrado no grid**
  - **`TRUE`: uma cor para cada altura de cada quadrado no grid**

Controlando propriedades de geometria baseadas em coordenadas

Um exemplo de forma com ElevationGrid



[cmount.wrl]

Controlando propriedades de geometria baseadas em coordenadas

Controlando sombreados com o ângulo crease

. Inicialmente, faces são desenhadas com sombreados por face

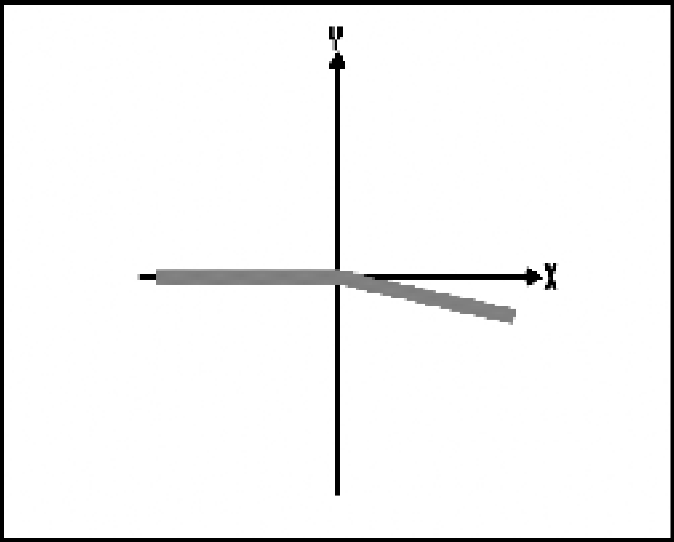
. Você pode fazer sombreados lisos usando o campo `creaseAngle` para

- `IndexedFaceSet`
- `ElevationGrid`
- `Extrusion`

Controlando propriedades de geometria baseadas em coordenadas

Selecionando ângulos crease

. Um ângulo *crease* é um ângulo limite entre duas superfícies



. Se o ângulo >= ângulo crease, usar sobreamento de face

. Se o ângulo < ângulo crease, usar sombreamento liso

Controlando propriedades de geometria baseadas em coordenadas

Um exemplo usando ângulos crease

ângulo crease = 0 **ângulo crease = 45 deg**

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: Normal

. Um nó `Normal` contém uma lista de vetores normais que sobreescrevem o uso de ângulos crease

```
Normal {
    vector [ 0.0 1.0 0.0, . . . ]
}
```

. Geralmente a geração automática de normal é bom o suficiente

. Normais podem ser fornecidas para os nós `IndexedFaceSet` e `ElevationGrid`

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: IndexedFaceSet

. Um nó geométrico `IndexedFaceSet` cria geometria dentre superfícies

- ***Normal vectors* - lista de normais**
- ***Normal indexes* - seleciona normais de uma lista (como a seleção de coordenadas)**
- ***Normal binding* - controla a faixa de normais**

```
IndexedFaceSet {
    coord Coordinate { . . . }
    coordIndex [ . . . ]
    normal Normal { . . . }
    normalIndex [ . . . ]
    normalPerVertex TRUE
}
```

Controlando propriedades de geometria baseadas em coordenadas

Controlando a faixa de normais para conjuntos de superfícies

- **O campo `normalPerVertex` controla como os índices de normais são usados**
  - **`FALSE`: um índice normal para cada superfície (finalizando com o índice de coordenada -1)**
  - **`TRUE`: um índice normal para cada índice de coordenada de cada face (incluindo o índice de coordenada -1)**

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: ElevationGrid

- **Um nó geométrico `ElevationGrid` cria terrenos**
  - ***Normal vectors* - lista de normais**
  - ***Normal indexes* - seleciona normais de uma lista (como a seleção de coordenadas)**
  - ***Normal binding* - controla a faixa de normais**

```
ElevationGrid {
    height [ . . . ]
    normal Normal { . . . }
    normalPerVertex TRUE
}
```

Controlando propriedades de geometria baseadas em coordenadas

Controlando a faixa de normais de grids elevados

- O campo `normalPerVertex` controla como os índices normais são usados (similar ao conjunto de superfícies)
  - `FALSE`: uma normal para cada quadrado no grid
  - `TRUE`: uma normal para cada altura de cada quadrado no grid

Controlando propriedades de geometria baseadas em coordenadas

Sintaxe: NormalInterpolator

- **Um nó `NormalInterpolator` descreve um caminho de normais**
  - ***keys* - fração chave**
  - ***values* - lista de normais chaves (listas X,Y,Z)**

```
NormalInterpolator {
    key [ 0.0, . . . ]
    keyValue [ 0.0 1.0 1.0, . . . ]
}
```

Controlando propriedades de geometria baseadas em coordenadas

Resumo

- **O nó `Color` lista cores para usar em partes de uma forma**
  - **Usado como o valor do campo `color`**
  - **Índice de cores selecionam cores para usar**
  - **Cores sobrescrevem o nó `Material`**
- **O campo `colorPerVertex` seleciona cores conforme linha/superfície/quadrado de grid ou cores conforme coordenadas**

Controlando propriedades de geometria baseadas em coordenadas

Resumo

- O campo `creaseAngle` controla sombreamento facetado ou liso
- O nó `Normal` lista vetores normais para uso como partes de uma forma
  - Usado como valor do campo `normal`
  - Índices normais selecionam normais para uso
  - Normais sobrescrevem o valor `creaseAngle`
- O campo `normalPerVertex` seleciona normais conforme superfície/quadrado do grid ou normais conforme coordenadas
- O nó `NormalInterpolator` descreve um caminho para normais

```
#VRML V2.0 utf8
Shape {
        appearance Appearance {
                material Material { }
        }
        geometry ElevationGrid {
                xDimension  9
                zDimension  9
                xSpacing    1.0
                zSpacing    1.0
                solid       FALSE
                creaseAngle 0.785
                height [
                        0.0, 0.0, 0.5, 1.0, 0.5, 0.0, 0.0, 0.0, 0.0,
                        0.0, 0.0, 0.0, 0.0, 2.5, 0.5, 0.0, 0.0, 0.0,
                        0.0, 0.0, 0.5, 0.5, 3.0, 1.0, 0.5, 0.0, 1.0,
                        0.0, 0.0, 0.5, 2.0, 4.5, 2.5, 1.0, 1.5, 0.5,
                        1.0, 2.5, 3.0, 4.5, 5.5, 3.5, 3.0, 1.0, 0.0,
                        0.5, 2.0, 2.0, 2.5, 3.5, 4.0, 2.0, 0.5, 0.0,
                        0.0, 0.0, 0.5, 1.5, 1.0, 2.0, 3.0, 1.5, 0.0,
                        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, 1.5, 0.5,
                        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0,
                ]
                colorPerVertex TRUE
                color Color {
                        color [
                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.5 0.1,
                                0.2 0.6 0.0,  0.0 0.5 0.1,  0.0 0.3 1.0,
                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.3 1.0,

                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.3 1.0,
                                0.0 0.3 1.0,  0.5 0.4 0.0,  0.0 0.5 0.1,
                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.3 1.0,

                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.5 0.1,
                                0.0 0.5 0.1,  0.5 0.4 0.0,  0.2 0.6 0.0,
                                0.0 0.5 0.1,  0.0 0.3 1.0,  0.2 0.6 0.0,

                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.5 0.1,
                                0.4 0.3 0.1,  0.7 0.7 0.7,  0.5 0.4 0.0,
                                0.2 0.6 0.1,  0.3 0.6 0.6,  0.0 0.5 0.1,

                                0.2 0.6 0.0,  0.5 0.4 0.0,  0.5 0.4 0.0,
                                0.7 0.7 0.7,  0.8 0.8 0.8,  0.5 0.5 0.7,
                                0.5 0.5 0.7,  0.2 0.6 0.0,  0.0 0.3 1.0,

                                0.0 0.5 0.1,  0.2 0.6 0.1,  0.2 0.6 0.1,
                                0.2 0.6 0.1,  0.5 0.5 0.7,  0.7 0.7 0.7,
                                0.5 0.4 0.0,  0.0 0.5 0.1,  0.0 0.3 1.0,

                                0.0 0.5 0.1,  0.0 0.3 1.0,  0.0 0.5 0.1,
                                0.2 0.6 0.1,  0.2 0.6 0.0,  0.5 0.4 0.0,
                                0.5 0.5 0.7,  0.2 0.6 0.0,  0.0 0.3 1.0,

                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.3 1.0,
                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.5 0.1,
                                0.5 0.4 0.0,  0.2 0.6 0.0,  0.0 0.5 0.1,

                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.3 1.0,
                                0.0 0.3 1.0,  0.0 0.3 1.0,  0.0 0.3 1.0,
                                0.0 0.5 0.1,  0.0 0.3 1.0,  0.0 0.3 1.0,
                        ]
```

```
                    }
          }
}
```

```
#VRML V2.0 utf8
Group {
    children [
    # Axes
        Shape {
            appearance Appearance {
                material Material {
                    emissiveColor 1.0 1.0 1.0
                    diffuseColor  1.0 1.0 1.0
                }
            }
            geometry IndexedLineSet {
                coord Coordinate {
                    point [
                        0.0 0.0 0.0, 10.0 0.0 0.0, 0.0 8.0 0.0
                    ]
                }
                coordIndex [ 0, 1, -1,  0, 2, -1 ]
            }
        }
    # Scatter plot with different color points
        Shape {
            # no appearance, use emissive coloring
            geometry PointSet {
                coord Coordinate {
                    point [
                        1.0 1.0 0.0,   2.0 4.0 0.0,
                        3.0 5.0 0.0,   4.0 4.0 0.0,
                        5.0 6.0 0.0,   6.0 7.0 0.0,
                        7.0 5.0 0.0,   8.0 6.0 0.0,
                        9.0 4.0 0.0,  10.0 3.0 0.0,
                        1.0 3.0 0.0,   2.0 2.0 0.0,
                        3.0 2.0 0.0,   4.0 1.0 0.0,
                        5.0 2.0 0.0,   6.0 4.0 0.0,
                        7.0 3.0 0.0,   8.0 5.0 0.0,
                        9.0 5.0 0.0,  10.0 6.0 0.0,
                    ]
                }
                color Color {
                    color [
                        0.0 1.0 0.0,   0.0 1.0 0.0,
                        1.0 0.0 0.0,   0.0 1.0 1.0,
                        0.0 1.0 0.0,   0.0 1.0 0.0,
                        0.0 1.0 1.0,   1.0 0.0 1.0,
                        0.0 1.0 0.0,   1.0 0.0 0.0,
                        1.0 0.0 0.0,   1.0 0.0 0.0,
                        0.0 1.0 1.0,   1.0 0.0 1.0,
                        1.0 0.0 0.0,   0.0 1.0 0.0,
                        0.0 0.0 1.0,   0.0 1.0 1.0,
                        1.0 1.0 1.0,   1.0 1.0 0.0,
                    ]
                }
            }
        }
    ]
}
```

```
#VRML V2.0 utf8
DEF Burst Shape {
    # no appearance, use emissive coloring
    geometry IndexedLineSet {
        coord Coordinate {
            point [
                 0.00  0.00  0.00,   1.00  0.00  0.00,
                 0.92  0.38  0.00,   0.71  0.71  0.00,
                 0.38  0.92  0.00,   0.00  1.00  0.00,
                -0.38  0.92  0.00,  -0.71  0.71  0.00,
                -0.92  0.38  0.00,  -1.00  0.00  0.00,
                -0.92 -0.38  0.00,  -0.71 -0.71  0.00,
                -0.38 -0.92  0.00,   0.00 -1.00  0.00,
                 0.38 -0.92  0.00,   0.71 -0.71  0.00,
                 0.92 -0.38  0.00,
            ]
        }
        coordIndex [
            0,  1, -1,   0,  2, -1,
            0,  3, -1,   0,  4, -1,
            0,  5, -1,   0,  6, -1,
            0,  7, -1,   0,  8, -1,
            0,  9, -1,   0, 10, -1,
            0, 11, -1,   0, 12, -1,
            0, 13, -1,   0, 14, -1,
            0, 15, -1,   0, 16, -1
        ]
        colorPerVertex TRUE
        color Color {
            color [
                1.0 1.0 0.0,  # burst center color
                1.0 0.3 0.3   # burst ends color
            ]
        }
        colorIndex [
            0,  1,  0,   0,  1,  0,
            0,  1,  0,   0,  1,  0,
            0,  1,  0,   0,  1,  0,
            0,  1,  0,   0,  1,  0,
            0,  1,  0,   0,  1,  0,
            0,  1,  0,   0,  1,  0,
            0,  1,  0,   0,  1,  0,
            0,  1,  0,   0,  1,  0,
        ]
    }
}
Transform { rotation 0.0 1.0 0.0  0.785 children USE Burst }
Transform { rotation 0.0 1.0 0.0  1.57  children USE Burst }
Transform { rotation 0.0 1.0 0.0  2.355 children USE Burst }
```

```
#VRML V2.0 utf8

# A log
Transform {
    translation 0.0 -0.4 0.0
    rotation    0.0 0.0 1.0 -1.57
    children Shape {
        appearance Appearance {
            material Material {
                diffuseColor 0.5 0.3 0.0
            }
        }
        geometry Cylinder {
            height 2.9
            radius 0.4
        }
    }
}
# A set of flames
DEF Flames Shape {
    # No appearance, use emissive coloring
    geometry IndexedFaceSet {
        coord Coordinate {
            point [
                -0.7 0.0 0.0,  -0.8 1.5 0.0,  -1.0 0.0 0.0,
                -0.5 0.0 0.01, -0.7 1.2 0.01, -0.9 0.0 0.01,
                -0.1 0.0 0.0,  -0.2 1.6 0.0,  -0.4 0.0 0.0,
                 0.3 0.0 0.01,  0.2 1.0 0.01,  0.0 0.0 0.0,
            ]
        }
        coordIndex [
            0, 1, 2, -1,  3,  4,  5, -1,
            6, 7, 8, -1,  9, 10, 11, -1
        ]
        solid FALSE
        colorPerVertex TRUE
        color Color {
            color [
                1.0 1.0 0.0,  1.0 0.0 0.0,  1.0 0.7 0.0,
                0.8 0.5 0.0,  1.0 0.1 0.0,  1.0 0.8 0.0,
            ]
        }
        colorIndex [
            3, 4, 5, 0,   0, 1, 2, 0,
            3, 4, 5, 0,   0, 1, 2, 0
        ]
    }
}
# Repeat the flames to make a roaring fire
Transform {
    translation 0.8 0.0 0.02
    scale 1.0 1.3 1.0
    children USE Flames
}
Transform {
    translation 1.1 0.0 0.04
    scale 1.0 0.5 1.0
    children USE Flames
}
Transform {
    translation -0.3 0.0 0.06
```

```
    scale 1.0 1.1 1.0
    children USE Flames
}
Transform {
    translation -0.1 0.0 0.08
    scale 1.0 0.4 1.0
    children USE Flames
}
Transform {
    translation 0.8 0.0 0.10
    scale 1.0 1.1 1.0
    children USE Flames
}
```

Exemplos

Terreno digital

Fita torcida

Relógio de tempo real

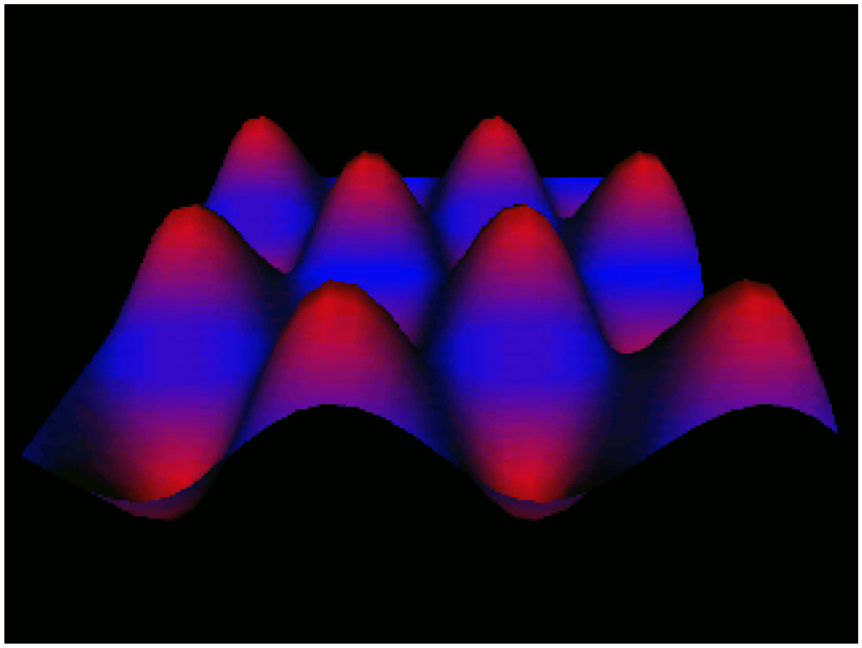
Animação

Forma de cobra

Exemplos

Terreno digital

. Um nó `ElevationGrid` cria uma terreno digital
. Um nó `Color` fornece as cores do terreno



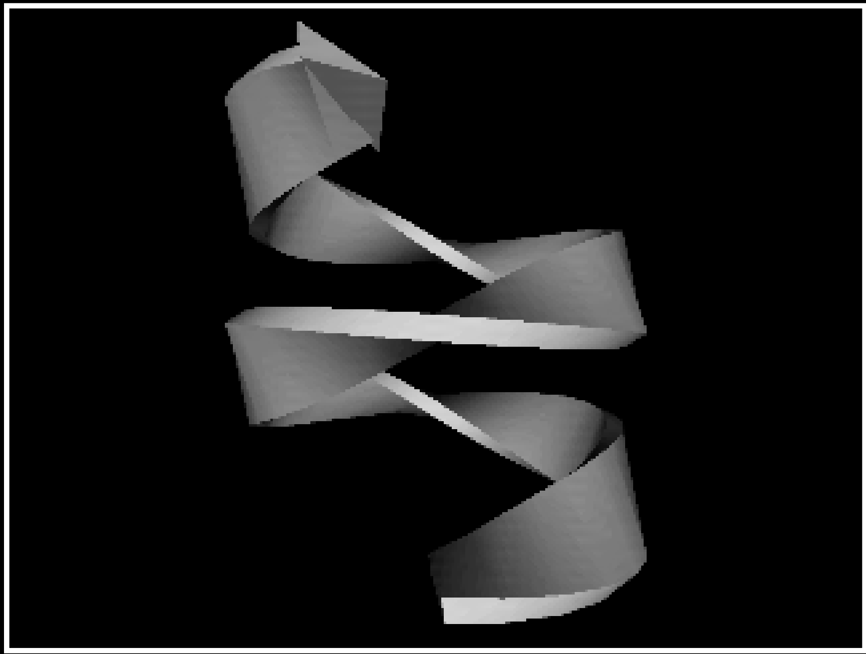
[terrain1.wrl]

Exemplos

Fita torcida

. Um nó `Extrusion` cria uma fita

. Os campos `orientation` e `scale` fazem a fita girar e mudar de tamanho



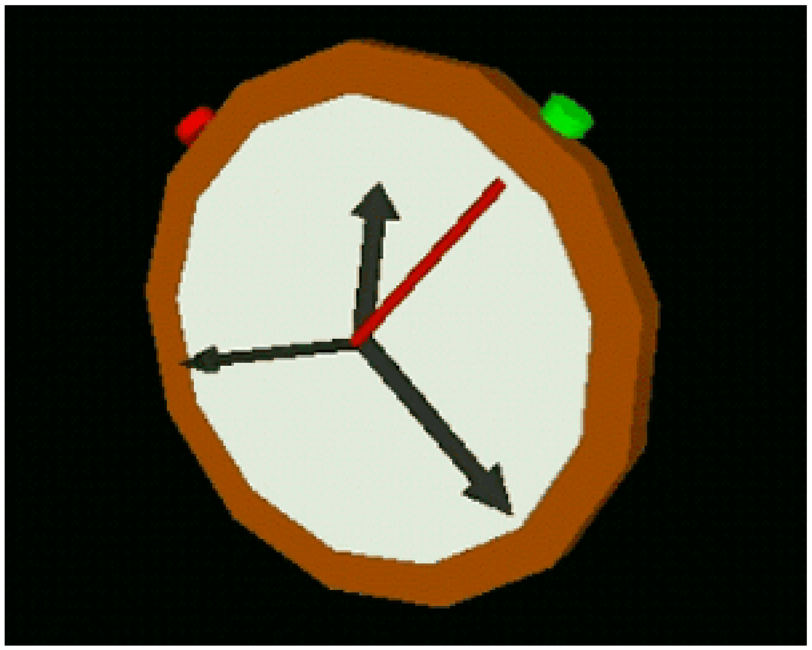
[ribbon2.wrl]

Exemplos

Relógio de tempo real

. Um conjunto de nós `TimeSensor` mostram o tempo

. Um conjunto de nós `OrientationInterpolator` giram os ponteiros do relógio

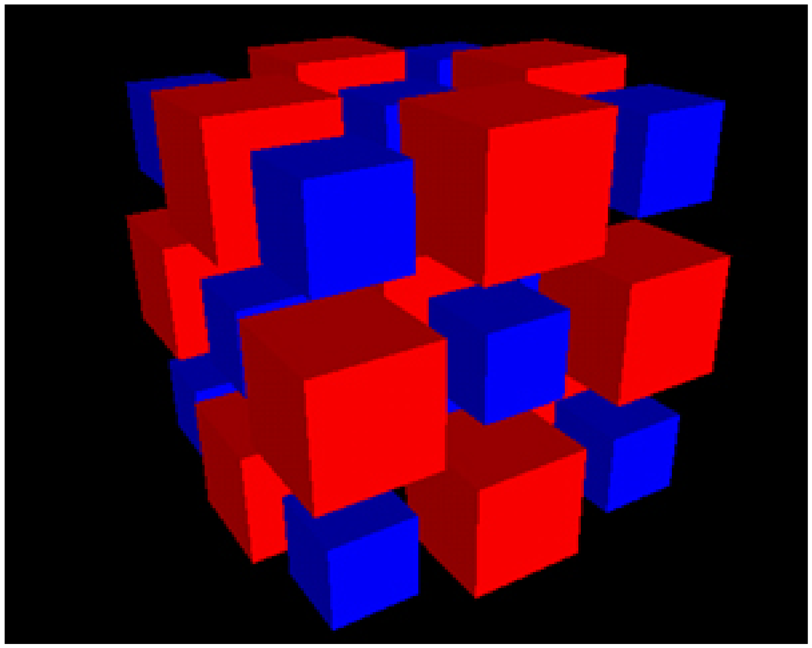


[stopwtch.wrl]

Exemplos

Animação

. O primeiro nó `TimeSensor` mede o tempo de um segundo nó `TimeSensor` para criar uma animação periódica

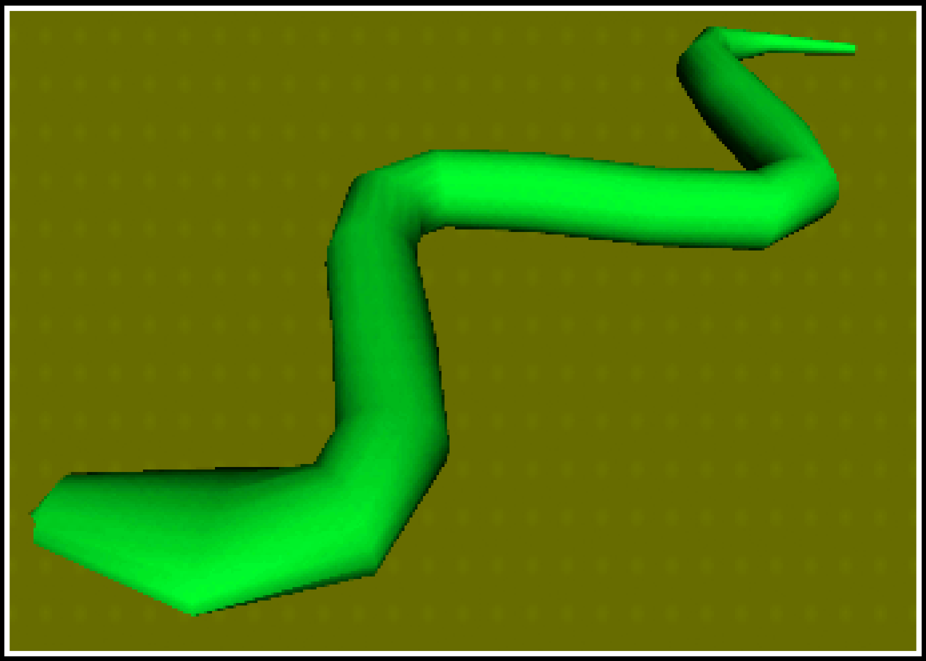


[timetime.wrl]

Exemplos

Forma de cobra

. Um nó `CoordinateInterpolator` anima a espinha de um nó `Extrusion`



[snake.wrl]

```
#VRML V2.0 utf8

# WHO:    John L. Moreland
# WHAT:   Terrain
# WHY:    Demonstrates ElevationGrid, Color
# WHERE:  San Diego Supercomputer Center
# WHEN:   Tue Apr 29 09:10:45 PDT 1997
# HOW:    VRML 2.0

Shape {
        appearance Appearance {
                material Material {}
        }
        geometry ElevationGrid {
                xDimension 30
                zDimension 30
                xSpacing 1.0
                zSpacing 1.0
                solid FALSE
                creaseAngle 0.785
                height [
                        0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
0.00, 0.00,
                        0.00, 0.83, 1.51, 1.93, 2.02, 1.76, 1.20, 0.42, -0.42, -1.20, -1.76, -
2.02, -1.93, -1.51, -0.83, 0.00, 0.83, 1.51, 1.93, 2.02, 1.76, 1.20, 0.42, -0.42, -1.20, -1.76, -
2.02, -1.93, -1.51, -0.83,
                        0.00, 1.51, 2.76, 3.53, 3.70, 3.22, 2.18, 0.77, -0.77, -2.18, -3.22, -
3.70, -3.53, -2.76, -1.51, 0.00, 1.51, 2.76, 3.53, 3.70, 3.22, 2.18, 0.77, -0.77, -2.18, -3.22, -
3.70, -3.53, -2.76, -1.51,
                        0.00, 1.93, 3.53, 4.52, 4.73, 4.12, 2.80, 0.99, -0.99, -2.80, -4.12, -
4.73, -4.52, -3.53, -1.93, 0.00, 1.93, 3.53, 4.52, 4.73, 4.12, 2.80, 0.99, -0.99, -2.80, -4.12, -
4.73, -4.52, -3.53, -1.93,
                        0.00, 2.02, 3.70, 4.73, 4.95, 4.31, 2.92, 1.03, -1.03, -2.92, -4.31, -
4.95, -4.73, -3.70, -2.02, 0.00, 2.02, 3.70, 4.73, 4.95, 4.31, 2.92, 1.03, -1.03, -2.92, -4.31, -
4.95, -4.73, -3.70, -2.02,
                        0.00, 1.76, 3.22, 4.12, 4.31, 3.75, 2.55, 0.90, -0.90, -2.55, -3.75, -
4.31, -4.12, -3.22, -1.76, 0.00, 1.76, 3.22, 4.12, 4.31, 3.75, 2.55, 0.90, -0.90, -2.55, -3.75, -
4.31, -4.12, -3.22, -1.76,
                        0.00, 1.20, 2.18, 2.80, 2.92, 2.55, 1.73, 0.61, -0.61, -1.73, -2.55, -
2.92, -2.80, -2.18, -1.20, 0.00, 1.20, 2.18, 2.80, 2.92, 2.55, 1.73, 0.61, -0.61, -1.73, -2.55, -
2.92, -2.80, -2.18, -1.20,
                        0.00, 0.42, 0.77, 0.99, 1.03, 0.90, 0.61, 0.22, -0.22, -0.61, -0.90, -
1.03, -0.99, -0.77, -0.42, 0.00, 0.42, 0.77, 0.99, 1.03, 0.90, 0.61, 0.22, -0.22, -0.61, -0.90, -
1.03, -0.99, -0.77, -0.42,
                        0.00, -0.42, -0.77, -0.99, -1.03, -0.90, -0.61, -0.22, 0.22, 0.61, 0.90,
1.03, 0.99, 0.77, 0.42, 0.00, -0.42, -0.77, -0.99, -1.03, -0.90, -0.61, -0.22, 0.22, 0.61, 0.90,
1.03, 0.99, 0.77, 0.42,
                        0.00, -1.20, -2.18, -2.80, -2.92, -2.55, -1.73, -0.61, 0.61, 1.73, 2.55,
2.92, 2.80, 2.18, 1.20, 0.00, -1.20, -2.18, -2.80, -2.92, -2.55, -1.73, -0.61, 0.61, 1.73, 2.55,
2.92, 2.80, 2.18, 1.20,
                        0.00, -1.76, -3.22, -4.12, -4.31, -3.75, -2.55, -0.90, 0.90, 2.55, 3.75,
4.31, 4.12, 3.22, 1.76, 0.00, -1.76, -3.22, -4.12, -4.31, -3.75, -2.55, -0.90, 0.90, 2.55, 3.75,
4.31, 4.12, 3.22, 1.76,
                        0.00, -2.02, -3.70, -4.73, -4.95, -4.31, -2.92, -1.03, 1.03, 2.92, 4.31,
4.95, 4.73, 3.70, 2.02, 0.00, -2.02, -3.70, -4.73, -4.95, -4.31, -2.92, -1.03, 1.03, 2.92, 4.31,
4.95, 4.73, 3.70, 2.02,
                        0.00, -1.93, -3.53, -4.52, -4.73, -4.12, -2.80, -0.99, 0.99, 2.80, 4.12,
4.73, 4.52, 3.53, 1.93, 0.00, -1.93, -3.53, -4.52, -4.73, -4.12, -2.80, -0.99, 0.99, 2.80, 4.12,
4.73, 4.52, 3.53, 1.93,
                        0.00, -1.51, -2.76, -3.53, -3.70, -3.22, -2.18, -0.77, 0.77, 2.18, 3.22,
```

```
3.70, 3.53, 2.76, 1.51, 0.00, -1.51, -2.76, -3.53, -3.70, -3.22, -2.18, -0.77, 0.77, 2.18, 3.22, 
3.70, 3.53, 2.76, 1.51,
                        0.00, -0.83, -1.51, -1.93, -2.02, -1.76, -1.20, -0.42, 0.42, 1.20, 1.76, 
2.02, 1.93, 1.51, 0.83, 0.00, -0.83, -1.51, -1.93, -2.02, -1.76, -1.20, -0.42, 0.42, 1.20, 1.76, 
2.02, 1.93, 1.51, 0.83,
                        0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 
0.00, 0.00,
                        0.00, 0.83, 1.51, 1.93, 2.02, 1.76, 1.20, 0.42, -0.42, -1.20, -1.76, -
2.02, -1.93, -1.51, -0.83, 0.00, 0.83, 1.51, 1.93, 2.02, 1.76, 1.20, 0.42, -0.42, -1.20, -1.76, -
2.02, -1.93, -1.51, -0.83,
                        0.00, 1.51, 2.76, 3.53, 3.70, 3.22, 2.18, 0.77, -0.77, -2.18, -3.22, -
3.70, -3.53, -2.76, -1.51, 0.00, 1.51, 2.76, 3.53, 3.70, 3.22, 2.18, 0.77, -0.77, -2.18, -3.22, -
3.70, -3.53, -2.76, -1.51,
                        0.00, 1.93, 3.53, 4.52, 4.73, 4.12, 2.80, 0.99, -0.99, -2.80, -4.12, -
4.73, -4.52, -3.53, -1.93, 0.00, 1.93, 3.53, 4.52, 4.73, 4.12, 2.80, 0.99, -0.99, -2.80, -4.12, -
4.73, -4.52, -3.53, -1.93,
                        0.00, 2.02, 3.70, 4.73, 4.95, 4.31, 2.92, 1.03, -1.03, -2.92, -4.31, -
4.95, -4.73, -3.70, -2.02, 0.00, 2.02, 3.70, 4.73, 4.95, 4.31, 2.92, 1.03, -1.03, -2.92, -4.31, -
4.95, -4.73, -3.70, -2.02,
                        0.00, 1.76, 3.22, 4.12, 4.31, 3.75, 2.55, 0.90, -0.90, -2.55, -3.75, -
4.31, -4.12, -3.22, -1.76, 0.00, 1.76, 3.22, 4.12, 4.31, 3.75, 2.55, 0.90, -0.90, -2.55, -3.75, -
4.31, -4.12, -3.22, -1.76,
                        0.00, 1.20, 2.18, 2.80, 2.92, 2.55, 1.73, 0.61, -0.61, -1.73, -2.55, -
2.92, -2.80, -2.18, -1.20, 0.00, 1.20, 2.18, 2.80, 2.92, 2.55, 1.73, 0.61, -0.61, -1.73, -2.55, -
2.92, -2.80, -2.18, -1.20,
                        0.00, 0.42, 0.77, 0.99, 1.03, 0.90, 0.61, 0.22, -0.22, -0.61, -0.90, -
1.03, -0.99, -0.77, -0.42, 0.00, 0.42, 0.77, 0.99, 1.03, 0.90, 0.61, 0.22, -0.22, -0.61, -0.90, -
1.03, -0.99, -0.77, -0.42,
                        0.00, -0.42, -0.77, -0.99, -1.03, -0.90, -0.61, -0.22, 0.22, 0.61, 0.90, 
1.03, 0.99, 0.77, 0.42, 0.00, -0.42, -0.77, -0.99, -1.03, -0.90, -0.61, -0.22, 0.22, 0.61, 0.90, 
1.03, 0.99, 0.77, 0.42,
                        0.00, -1.20, -2.18, -2.80, -2.92, -2.55, -1.73, -0.61, 0.61, 1.73, 2.55, 
2.92, 2.80, 2.18, 1.20, 0.00, -1.20, -2.18, -2.80, -2.92, -2.55, -1.73, -0.61, 0.61, 1.73, 2.55, 
2.92, 2.80, 2.18, 1.20,
                        0.00, -1.76, -3.22, -4.12, -4.31, -3.75, -2.55, -0.90, 0.90, 2.55, 3.75, 
4.31, 4.12, 3.22, 1.76, 0.00, -1.76, -3.22, -4.12, -4.31, -3.75, -2.55, -0.90, 0.90, 2.55, 3.75, 
4.31, 4.12, 3.22, 1.76,
                        0.00, -2.02, -3.70, -4.73, -4.95, -4.31, -2.92, -1.03, 1.03, 2.92, 4.31, 
4.95, 4.73, 3.70, 2.02, 0.00, -2.02, -3.70, -4.73, -4.95, -4.31, -2.92, -1.03, 1.03, 2.92, 4.31, 
4.95, 4.73, 3.70, 2.02,
                        0.00, -1.93, -3.53, -4.52, -4.73, -4.12, -2.80, -0.99, 0.99, 2.80, 4.12, 
4.73, 4.52, 3.53, 1.93, 0.00, -1.93, -3.53, -4.52, -4.73, -4.12, -2.80, -0.99, 0.99, 2.80, 4.12, 
4.73, 4.52, 3.53, 1.93,
                        0.00, -1.51, -2.76, -3.53, -3.70, -3.22, -2.18, -0.77, 0.77, 2.18, 3.22, 
3.70, 3.53, 2.76, 1.51, 0.00, -1.51, -2.76, -3.53, -3.70, -3.22, -2.18, -0.77, 0.77, 2.18, 3.22, 
3.70, 3.53, 2.76, 1.51,
                        0.00, -0.83, -1.51, -1.93, -2.02, -1.76, -1.20, -0.42, 0.42, 1.20, 1.76, 
2.02, 1.93, 1.51, 0.83, 0.00, -0.83, -1.51, -1.93, -2.02, -1.76, -1.20, -0.42, 0.42, 1.20, 1.76, 
2.02, 1.93, 1.51, 0.83,
                ]
                colorPerVertex TRUE
                color Color {
                        color [
                                0.00 0.00 1.00,
                                0.00 0.00 1.00,
                                0.00 0.00 1.00,
                                0.00 0.00 1.00,
                                0.00 0.00 1.00,
                                0.00 0.00 1.00,
                                0.00 0.00 1.00,
```

```
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
# sample_x 0
0.00 0.00 1.00,
0.17 0.00 0.83,
0.30 0.00 0.70,
0.39 0.00 0.61,
0.40 0.00 0.60,
0.35 0.00 0.65,
0.24 0.00 0.76,
0.08 0.00 0.92,
0.08 0.00 0.92,
0.24 0.00 0.76,
0.35 0.00 0.65,
0.40 0.00 0.60,
0.39 0.00 0.61,
0.30 0.00 0.70,
0.17 0.00 0.83,
0.00 0.00 1.00,
0.17 0.00 0.83,
0.30 0.00 0.70,
0.39 0.00 0.61,
0.40 0.00 0.60,
0.35 0.00 0.65,
0.24 0.00 0.76,
0.08 0.00 0.92,
0.08 0.00 0.92,
0.24 0.00 0.76,
0.35 0.00 0.65,
0.40 0.00 0.60,
0.39 0.00 0.61,
0.30 0.00 0.70,
0.17 0.00 0.83,
# sample_x 1
0.00 0.00 1.00,
0.30 0.00 0.70,
0.55 0.00 0.45,
0.71 0.00 0.29,
0.74 0.00 0.26,
```

```
0.64 0.00 0.36,
0.44 0.00 0.56,
0.15 0.00 0.85,
0.15 0.00 0.85,
0.44 0.00 0.56,
0.64 0.00 0.36,
0.74 0.00 0.26,
0.71 0.00 0.29,
0.55 0.00 0.45,
0.30 0.00 0.70,
0.00 0.00 1.00,
0.30 0.00 0.70,
0.55 0.00 0.45,
0.71 0.00 0.29,
0.74 0.00 0.26,
0.64 0.00 0.36,
0.44 0.00 0.56,
0.15 0.00 0.85,
0.15 0.00 0.85,
0.44 0.00 0.56,
0.64 0.00 0.36,
0.74 0.00 0.26,
0.71 0.00 0.29,
0.55 0.00 0.45,
0.30 0.00 0.70,
# sample_x 2
0.00 0.00 1.00,
0.39 0.00 0.61,
0.71 0.00 0.29,
0.90 0.00 0.10,
0.95 0.00 0.05,
0.82 0.00 0.18,
0.56 0.00 0.44,
0.20 0.00 0.80,
0.20 0.00 0.80,
0.56 0.00 0.44,
0.82 0.00 0.18,
0.95 0.00 0.05,
0.90 0.00 0.10,
0.71 0.00 0.29,
0.39 0.00 0.61,
0.00 0.00 1.00,
0.39 0.00 0.61,
0.71 0.00 0.29,
0.90 0.00 0.10,
0.95 0.00 0.05,
0.82 0.00 0.18,
0.56 0.00 0.44,
0.20 0.00 0.80,
0.20 0.00 0.80,
0.56 0.00 0.44,
0.82 0.00 0.18,
0.95 0.00 0.05,
0.90 0.00 0.10,
0.71 0.00 0.29,
0.39 0.00 0.61,
# sample_x 3
0.00 0.00 1.00,
0.40 0.00 0.60,
0.74 0.00 0.26,
0.95 0.00 0.05,
```

```
0.99 0.00 0.01,
0.86 0.00 0.14,
0.58 0.00 0.42,
0.21 0.00 0.79,
0.21 0.00 0.79,
0.58 0.00 0.42,
0.86 0.00 0.14,
0.99 0.00 0.01,
0.95 0.00 0.05,
0.74 0.00 0.26,
0.40 0.00 0.60,
0.00 0.00 1.00,
0.40 0.00 0.60,
0.74 0.00 0.26,
0.95 0.00 0.05,
0.99 0.00 0.01,
0.86 0.00 0.14,
0.58 0.00 0.42,
0.21 0.00 0.79,
0.21 0.00 0.79,
0.58 0.00 0.42,
0.86 0.00 0.14,
0.99 0.00 0.01,
0.95 0.00 0.05,
0.74 0.00 0.26,
0.40 0.00 0.60,
# sample_x 4
0.00 0.00 1.00,
0.35 0.00 0.65,
0.64 0.00 0.36,
0.82 0.00 0.18,
0.86 0.00 0.14,
0.75 0.00 0.25,
0.51 0.00 0.49,
0.18 0.00 0.82,
0.18 0.00 0.82,
0.51 0.00 0.49,
0.75 0.00 0.25,
0.86 0.00 0.14,
0.82 0.00 0.18,
0.64 0.00 0.36,
0.35 0.00 0.65,
0.00 0.00 1.00,
0.35 0.00 0.65,
0.64 0.00 0.36,
0.82 0.00 0.18,
0.86 0.00 0.14,
0.75 0.00 0.25,
0.51 0.00 0.49,
0.18 0.00 0.82,
0.18 0.00 0.82,
0.51 0.00 0.49,
0.75 0.00 0.25,
0.86 0.00 0.14,
0.82 0.00 0.18,
0.64 0.00 0.36,
0.35 0.00 0.65,
# sample_x 5
0.00 0.00 1.00,
0.24 0.00 0.76,
```

```
0.44 0.00 0.56,
0.56 0.00 0.44,
0.58 0.00 0.42,
0.51 0.00 0.49,
0.35 0.00 0.65,
0.12 0.00 0.88,
0.12 0.00 0.88,
0.35 0.00 0.65,
0.51 0.00 0.49,
0.58 0.00 0.42,
0.56 0.00 0.44,
0.44 0.00 0.56,
0.24 0.00 0.76,
0.00 0.00 1.00,
0.24 0.00 0.76,
0.44 0.00 0.56,
0.56 0.00 0.44,
0.58 0.00 0.42,
0.51 0.00 0.49,
0.35 0.00 0.65,
0.12 0.00 0.88,
0.12 0.00 0.88,
0.35 0.00 0.65,
0.51 0.00 0.49,
0.58 0.00 0.42,
0.56 0.00 0.44,
0.44 0.00 0.56,
0.24 0.00 0.76,
# sample_x 6
0.00 0.00 1.00,
0.08 0.00 0.92,
0.15 0.00 0.85,
0.20 0.00 0.80,
0.21 0.00 0.79,
0.18 0.00 0.82,
0.12 0.00 0.88,
0.04 0.00 0.96,
0.04 0.00 0.96,
0.12 0.00 0.88,
0.18 0.00 0.82,
0.21 0.00 0.79,
0.20 0.00 0.80,
0.15 0.00 0.85,
0.08 0.00 0.92,
0.00 0.00 1.00,
0.08 0.00 0.92,
0.15 0.00 0.85,
0.20 0.00 0.80,
0.21 0.00 0.79,
0.18 0.00 0.82,
0.12 0.00 0.88,
0.04 0.00 0.96,
0.04 0.00 0.96,
0.12 0.00 0.88,
0.18 0.00 0.82,
0.21 0.00 0.79,
0.20 0.00 0.80,
0.15 0.00 0.85,
0.08 0.00 0.92,
# sample_x 7
0.00 0.00 1.00,
```

```
0.08 0.00 0.92,
0.15 0.00 0.85,
0.20 0.00 0.80,
0.21 0.00 0.79,
0.18 0.00 0.82,
0.12 0.00 0.88,
0.04 0.00 0.96,
0.04 0.00 0.96,
0.12 0.00 0.88,
0.18 0.00 0.82,
0.21 0.00 0.79,
0.20 0.00 0.80,
0.15 0.00 0.85,
0.08 0.00 0.92,
0.00 0.00 1.00,
0.08 0.00 0.92,
0.15 0.00 0.85,
0.20 0.00 0.80,
0.21 0.00 0.79,
0.18 0.00 0.82,
0.12 0.00 0.88,
0.04 0.00 0.96,
0.04 0.00 0.96,
0.12 0.00 0.88,
0.18 0.00 0.82,
0.21 0.00 0.79,
0.20 0.00 0.80,
0.15 0.00 0.85,
0.08 0.00 0.92,
# sample_x 8
0.00 0.00 1.00,
0.24 0.00 0.76,
0.44 0.00 0.56,
0.56 0.00 0.44,
0.58 0.00 0.42,
0.51 0.00 0.49,
0.35 0.00 0.65,
0.12 0.00 0.88,
0.12 0.00 0.88,
0.35 0.00 0.65,
0.51 0.00 0.49,
0.58 0.00 0.42,
0.56 0.00 0.44,
0.44 0.00 0.56,
0.24 0.00 0.76,
0.00 0.00 1.00,
0.24 0.00 0.76,
0.44 0.00 0.56,
0.56 0.00 0.44,
0.58 0.00 0.42,
0.51 0.00 0.49,
0.35 0.00 0.65,
0.12 0.00 0.88,
0.12 0.00 0.88,
0.35 0.00 0.65,
0.51 0.00 0.49,
0.58 0.00 0.42,
0.56 0.00 0.44,
0.44 0.00 0.56,
0.24 0.00 0.76,
```

```
# sample_x 9
0.00 0.00 1.00,
0.35 0.00 0.65,
0.64 0.00 0.36,
0.82 0.00 0.18,
0.86 0.00 0.14,
0.75 0.00 0.25,
0.51 0.00 0.49,
0.18 0.00 0.82,
0.18 0.00 0.82,
0.51 0.00 0.49,
0.75 0.00 0.25,
0.86 0.00 0.14,
0.82 0.00 0.18,
0.64 0.00 0.36,
0.35 0.00 0.65,
0.00 0.00 1.00,
0.35 0.00 0.65,
0.64 0.00 0.36,
0.82 0.00 0.18,
0.86 0.00 0.14,
0.75 0.00 0.25,
0.51 0.00 0.49,
0.18 0.00 0.82,
0.18 0.00 0.82,
0.51 0.00 0.49,
0.75 0.00 0.25,
0.86 0.00 0.14,
0.82 0.00 0.18,
0.64 0.00 0.36,
0.35 0.00 0.65,
# sample_x 10
0.00 0.00 1.00,
0.40 0.00 0.60,
0.74 0.00 0.26,
0.95 0.00 0.05,
0.99 0.00 0.01,
0.86 0.00 0.14,
0.58 0.00 0.42,
0.21 0.00 0.79,
0.21 0.00 0.79,
0.58 0.00 0.42,
0.86 0.00 0.14,
0.99 0.00 0.01,
0.95 0.00 0.05,
0.74 0.00 0.26,
0.40 0.00 0.60,
0.00 0.00 1.00,
0.40 0.00 0.60,
0.74 0.00 0.26,
0.95 0.00 0.05,
0.99 0.00 0.01,
0.86 0.00 0.14,
0.58 0.00 0.42,
0.21 0.00 0.79,
0.21 0.00 0.79,
0.58 0.00 0.42,
0.86 0.00 0.14,
0.99 0.00 0.01,
0.95 0.00 0.05,
0.74 0.00 0.26,
```

```
0.40 0.00 0.60,
# sample_x 11
0.00 0.00 1.00,
0.39 0.00 0.61,
0.71 0.00 0.29,
0.90 0.00 0.10,
0.95 0.00 0.05,
0.82 0.00 0.18,
0.56 0.00 0.44,
0.20 0.00 0.80,
0.20 0.00 0.80,
0.56 0.00 0.44,
0.82 0.00 0.18,
0.95 0.00 0.05,
0.90 0.00 0.10,
0.71 0.00 0.29,
0.39 0.00 0.61,
0.00 0.00 1.00,
0.39 0.00 0.61,
0.71 0.00 0.29,
0.90 0.00 0.10,
0.95 0.00 0.05,
0.82 0.00 0.18,
0.56 0.00 0.44,
0.20 0.00 0.80,
0.20 0.00 0.80,
0.56 0.00 0.44,
0.82 0.00 0.18,
0.95 0.00 0.05,
0.90 0.00 0.10,
0.71 0.00 0.29,
0.39 0.00 0.61,
# sample_x 12
0.00 0.00 1.00,
0.30 0.00 0.70,
0.55 0.00 0.45,
0.71 0.00 0.29,
0.74 0.00 0.26,
0.64 0.00 0.36,
0.44 0.00 0.56,
0.15 0.00 0.85,
0.15 0.00 0.85,
0.44 0.00 0.56,
0.64 0.00 0.36,
0.74 0.00 0.26,
0.71 0.00 0.29,
0.55 0.00 0.45,
0.30 0.00 0.70,
0.00 0.00 1.00,
0.30 0.00 0.70,
0.55 0.00 0.45,
0.71 0.00 0.29,
0.74 0.00 0.26,
0.64 0.00 0.36,
0.44 0.00 0.56,
0.15 0.00 0.85,
0.15 0.00 0.85,
0.44 0.00 0.56,
0.64 0.00 0.36,
0.74 0.00 0.26,
```

```
0.71 0.00 0.29,
0.55 0.00 0.45,
0.30 0.00 0.70,
# sample_x 13
0.00 0.00 1.00,
0.17 0.00 0.83,
0.30 0.00 0.70,
0.39 0.00 0.61,
0.40 0.00 0.60,
0.35 0.00 0.65,
0.24 0.00 0.76,
0.08 0.00 0.92,
0.08 0.00 0.92,
0.24 0.00 0.76,
0.35 0.00 0.65,
0.40 0.00 0.60,
0.39 0.00 0.61,
0.30 0.00 0.70,
0.17 0.00 0.83,
0.00 0.00 1.00,
0.17 0.00 0.83,
0.30 0.00 0.70,
0.39 0.00 0.61,
0.40 0.00 0.60,
0.35 0.00 0.65,
0.24 0.00 0.76,
0.08 0.00 0.92,
0.08 0.00 0.92,
0.24 0.00 0.76,
0.35 0.00 0.65,
0.40 0.00 0.60,
0.39 0.00 0.61,
0.30 0.00 0.70,
0.17 0.00 0.83,
# sample_x 14
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
```

```
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
0.00 0.00 1.00,
# sample_x 15
0.00 0.00 1.00,
0.17 0.00 0.83,
0.30 0.00 0.70,
0.39 0.00 0.61,
0.40 0.00 0.60,
0.35 0.00 0.65,
0.24 0.00 0.76,
0.08 0.00 0.92,
0.08 0.00 0.92,
0.24 0.00 0.76,
0.35 0.00 0.65,
0.40 0.00 0.60,
0.39 0.00 0.61,
0.30 0.00 0.70,
0.17 0.00 0.83,
0.00 0.00 1.00,
0.17 0.00 0.83,
0.30 0.00 0.70,
0.39 0.00 0.61,
0.40 0.00 0.60,
0.35 0.00 0.65,
0.24 0.00 0.76,
0.08 0.00 0.92,
0.08 0.00 0.92,
0.24 0.00 0.76,
0.35 0.00 0.65,
0.40 0.00 0.60,
0.39 0.00 0.61,
0.30 0.00 0.70,
0.17 0.00 0.83,
# sample_x 16
0.00 0.00 1.00,
0.30 0.00 0.70,
0.55 0.00 0.45,
0.71 0.00 0.29,
0.74 0.00 0.26,
0.64 0.00 0.36,
0.44 0.00 0.56,
0.15 0.00 0.85,
0.15 0.00 0.85,
0.44 0.00 0.56,
0.64 0.00 0.36,
0.74 0.00 0.26,
0.71 0.00 0.29,
0.55 0.00 0.45,
0.30 0.00 0.70,
0.00 0.00 1.00,
0.30 0.00 0.70,
0.55 0.00 0.45,
0.71 0.00 0.29,
0.74 0.00 0.26,
0.64 0.00 0.36,
0.44 0.00 0.56,
0.15 0.00 0.85,
0.15 0.00 0.85,
```

```
0.44 0.00 0.56,
0.64 0.00 0.36,
0.74 0.00 0.26,
0.71 0.00 0.29,
0.55 0.00 0.45,
0.30 0.00 0.70,
# sample_x 17
0.00 0.00 1.00,
0.39 0.00 0.61,
0.71 0.00 0.29,
0.90 0.00 0.10,
0.95 0.00 0.05,
0.82 0.00 0.18,
0.56 0.00 0.44,
0.20 0.00 0.80,
0.20 0.00 0.80,
0.56 0.00 0.44,
0.82 0.00 0.18,
0.95 0.00 0.05,
0.90 0.00 0.10,
0.71 0.00 0.29,
0.39 0.00 0.61,
0.00 0.00 1.00,
0.39 0.00 0.61,
0.71 0.00 0.29,
0.90 0.00 0.10,
0.95 0.00 0.05,
0.82 0.00 0.18,
0.56 0.00 0.44,
0.20 0.00 0.80,
0.20 0.00 0.80,
0.56 0.00 0.44,
0.82 0.00 0.18,
0.95 0.00 0.05,
0.90 0.00 0.10,
0.71 0.00 0.29,
0.39 0.00 0.61,
# sample_x 18
0.00 0.00 1.00,
0.40 0.00 0.60,
0.74 0.00 0.26,
0.95 0.00 0.05,
0.99 0.00 0.01,
0.86 0.00 0.14,
0.58 0.00 0.42,
0.21 0.00 0.79,
0.21 0.00 0.79,
0.58 0.00 0.42,
0.86 0.00 0.14,
0.99 0.00 0.01,
0.95 0.00 0.05,
0.74 0.00 0.26,
0.40 0.00 0.60,
0.00 0.00 1.00,
0.40 0.00 0.60,
0.74 0.00 0.26,
0.95 0.00 0.05,
0.99 0.00 0.01,
0.86 0.00 0.14,
0.58 0.00 0.42,
0.21 0.00 0.79,
```

```
0.21 0.00 0.79,
0.58 0.00 0.42,
0.86 0.00 0.14,
0.99 0.00 0.01,
0.95 0.00 0.05,
0.74 0.00 0.26,
0.40 0.00 0.60,
# sample_x 19
0.00 0.00 1.00,
0.35 0.00 0.65,
0.64 0.00 0.36,
0.82 0.00 0.18,
0.86 0.00 0.14,
0.75 0.00 0.25,
0.51 0.00 0.49,
0.18 0.00 0.82,
0.18 0.00 0.82,
0.51 0.00 0.49,
0.75 0.00 0.25,
0.86 0.00 0.14,
0.82 0.00 0.18,
0.64 0.00 0.36,
0.35 0.00 0.65,
0.00 0.00 1.00,
0.35 0.00 0.65,
0.64 0.00 0.36,
0.82 0.00 0.18,
0.86 0.00 0.14,
0.75 0.00 0.25,
0.51 0.00 0.49,
0.18 0.00 0.82,
0.18 0.00 0.82,
0.51 0.00 0.49,
0.75 0.00 0.25,
0.86 0.00 0.14,
0.82 0.00 0.18,
0.64 0.00 0.36,
0.35 0.00 0.65,
# sample_x 20
0.00 0.00 1.00,
0.24 0.00 0.76,
0.44 0.00 0.56,
0.56 0.00 0.44,
0.58 0.00 0.42,
0.51 0.00 0.49,
0.35 0.00 0.65,
0.12 0.00 0.88,
0.12 0.00 0.88,
0.35 0.00 0.65,
0.51 0.00 0.49,
0.58 0.00 0.42,
0.56 0.00 0.44,
0.44 0.00 0.56,
0.24 0.00 0.76,
0.00 0.00 1.00,
0.24 0.00 0.76,
0.44 0.00 0.56,
0.56 0.00 0.44,
0.58 0.00 0.42,
0.51 0.00 0.49,
```

```
0.35 0.00 0.65,
0.12 0.00 0.88,
0.12 0.00 0.88,
0.35 0.00 0.65,
0.51 0.00 0.49,
0.58 0.00 0.42,
0.56 0.00 0.44,
0.44 0.00 0.56,
0.24 0.00 0.76,
# sample_x 21
0.00 0.00 1.00,
0.08 0.00 0.92,
0.15 0.00 0.85,
0.20 0.00 0.80,
0.21 0.00 0.79,
0.18 0.00 0.82,
0.12 0.00 0.88,
0.04 0.00 0.96,
0.04 0.00 0.96,
0.12 0.00 0.88,
0.18 0.00 0.82,
0.21 0.00 0.79,
0.20 0.00 0.80,
0.15 0.00 0.85,
0.08 0.00 0.92,
0.00 0.00 1.00,
0.08 0.00 0.92,
0.15 0.00 0.85,
0.20 0.00 0.80,
0.21 0.00 0.79,
0.18 0.00 0.82,
0.12 0.00 0.88,
0.04 0.00 0.96,
0.04 0.00 0.96,
0.12 0.00 0.88,
0.18 0.00 0.82,
0.21 0.00 0.79,
0.20 0.00 0.80,
0.15 0.00 0.85,
0.08 0.00 0.92,
# sample_x 22
0.00 0.00 1.00,
0.08 0.00 0.92,
0.15 0.00 0.85,
0.20 0.00 0.80,
0.21 0.00 0.79,
0.18 0.00 0.82,
0.12 0.00 0.88,
0.04 0.00 0.96,
0.04 0.00 0.96,
0.12 0.00 0.88,
0.18 0.00 0.82,
0.21 0.00 0.79,
0.20 0.00 0.80,
0.15 0.00 0.85,
0.08 0.00 0.92,
0.00 0.00 1.00,
0.08 0.00 0.92,
0.15 0.00 0.85,
0.20 0.00 0.80,
0.21 0.00 0.79,
```

```
0.18 0.00 0.82,
0.12 0.00 0.88,
0.04 0.00 0.96,
0.04 0.00 0.96,
0.12 0.00 0.88,
0.18 0.00 0.82,
0.21 0.00 0.79,
0.20 0.00 0.80,
0.15 0.00 0.85,
0.08 0.00 0.92,
# sample_x 23
0.00 0.00 1.00,
0.24 0.00 0.76,
0.44 0.00 0.56,
0.56 0.00 0.44,
0.58 0.00 0.42,
0.51 0.00 0.49,
0.35 0.00 0.65,
0.12 0.00 0.88,
0.12 0.00 0.88,
0.35 0.00 0.65,
0.51 0.00 0.49,
0.58 0.00 0.42,
0.56 0.00 0.44,
0.44 0.00 0.56,
0.24 0.00 0.76,
0.00 0.00 1.00,
0.24 0.00 0.76,
0.44 0.00 0.56,
0.56 0.00 0.44,
0.58 0.00 0.42,
0.51 0.00 0.49,
0.35 0.00 0.65,
0.12 0.00 0.88,
0.12 0.00 0.88,
0.35 0.00 0.65,
0.51 0.00 0.49,
0.58 0.00 0.42,
0.56 0.00 0.44,
0.44 0.00 0.56,
0.24 0.00 0.76,
# sample_x 24
0.00 0.00 1.00,
0.35 0.00 0.65,
0.64 0.00 0.36,
0.82 0.00 0.18,
0.86 0.00 0.14,
0.75 0.00 0.25,
0.51 0.00 0.49,
0.18 0.00 0.82,
0.18 0.00 0.82,
0.51 0.00 0.49,
0.75 0.00 0.25,
0.86 0.00 0.14,
0.82 0.00 0.18,
0.64 0.00 0.36,
0.35 0.00 0.65,
0.00 0.00 1.00,
0.35 0.00 0.65,
0.64 0.00 0.36,
```

```
0.82 0.00 0.18,
0.86 0.00 0.14,
0.75 0.00 0.25,
0.51 0.00 0.49,
0.18 0.00 0.82,
0.18 0.00 0.82,
0.51 0.00 0.49,
0.75 0.00 0.25,
0.86 0.00 0.14,
0.82 0.00 0.18,
0.64 0.00 0.36,
0.35 0.00 0.65,
# sample_x 25
0.00 0.00 1.00,
0.40 0.00 0.60,
0.74 0.00 0.26,
0.95 0.00 0.05,
0.99 0.00 0.01,
0.86 0.00 0.14,
0.58 0.00 0.42,
0.21 0.00 0.79,
0.21 0.00 0.79,
0.58 0.00 0.42,
0.86 0.00 0.14,
0.99 0.00 0.01,
0.95 0.00 0.05,
0.74 0.00 0.26,
0.40 0.00 0.60,
0.00 0.00 1.00,
0.40 0.00 0.60,
0.74 0.00 0.26,
0.95 0.00 0.05,
0.99 0.00 0.01,
0.86 0.00 0.14,
0.58 0.00 0.42,
0.21 0.00 0.79,
0.21 0.00 0.79,
0.58 0.00 0.42,
0.86 0.00 0.14,
0.99 0.00 0.01,
0.95 0.00 0.05,
0.74 0.00 0.26,
0.40 0.00 0.60,
# sample_x 26
0.00 0.00 1.00,
0.39 0.00 0.61,
0.71 0.00 0.29,
0.90 0.00 0.10,
0.95 0.00 0.05,
0.82 0.00 0.18,
0.56 0.00 0.44,
0.20 0.00 0.80,
0.20 0.00 0.80,
0.56 0.00 0.44,
0.82 0.00 0.18,
0.95 0.00 0.05,
0.90 0.00 0.10,
0.71 0.00 0.29,
0.39 0.00 0.61,
0.00 0.00 1.00,
0.39 0.00 0.61,
```

```
0.71 0.00 0.29,
0.90 0.00 0.10,
0.95 0.00 0.05,
0.82 0.00 0.18,
0.56 0.00 0.44,
0.20 0.00 0.80,
0.20 0.00 0.80,
0.56 0.00 0.44,
0.82 0.00 0.18,
0.95 0.00 0.05,
0.90 0.00 0.10,
0.71 0.00 0.29,
0.39 0.00 0.61,
# sample_x 27
0.00 0.00 1.00,
0.30 0.00 0.70,
0.55 0.00 0.45,
0.71 0.00 0.29,
0.74 0.00 0.26,
0.64 0.00 0.36,
0.44 0.00 0.56,
0.15 0.00 0.85,
0.15 0.00 0.85,
0.44 0.00 0.56,
0.64 0.00 0.36,
0.74 0.00 0.26,
0.71 0.00 0.29,
0.55 0.00 0.45,
0.30 0.00 0.70,
0.00 0.00 1.00,
0.30 0.00 0.70,
0.55 0.00 0.45,
0.71 0.00 0.29,
0.74 0.00 0.26,
0.64 0.00 0.36,
0.44 0.00 0.56,
0.15 0.00 0.85,
0.15 0.00 0.85,
0.44 0.00 0.56,
0.64 0.00 0.36,
0.74 0.00 0.26,
0.71 0.00 0.29,
0.55 0.00 0.45,
0.30 0.00 0.70,
# sample_x 28
0.00 0.00 1.00,
0.17 0.00 0.83,
0.30 0.00 0.70,
0.39 0.00 0.61,
0.40 0.00 0.60,
0.35 0.00 0.65,
0.24 0.00 0.76,
0.08 0.00 0.92,
0.08 0.00 0.92,
0.24 0.00 0.76,
0.35 0.00 0.65,
0.40 0.00 0.60,
0.39 0.00 0.61,
0.30 0.00 0.70,
0.17 0.00 0.83,
```

```
                                        0.00 0.00 1.00,
                                        0.17 0.00 0.83,
                                        0.30 0.00 0.70,
                                        0.39 0.00 0.61,
                                        0.40 0.00 0.60,
                                        0.35 0.00 0.65,
                                        0.24 0.00 0.76,
                                        0.08 0.00 0.92,
                                        0.08 0.00 0.92,
                                        0.24 0.00 0.76,
                                        0.35 0.00 0.65,
                                        0.40 0.00 0.60,
                                        0.39 0.00 0.61,
                                        0.30 0.00 0.70,
                                        0.17 0.00 0.83,
                                        # sample_x 29
                                ]
                        }
                }
}
```

```
#VRML V2.0 utf8

# WHO:    John L. Moreland
# WHAT:   Ribbon
# WHY:    Demonstrates Extrusion (orientation and scale)
# WHERE:  San Diego Supercomputer Center
# WHEN:   Tue Apr 29 09:10:45 PDT 1997
# HOW:    VRML 2.0

Shape {
        appearance Appearance {
                material Material {}
        }
        geometry Extrusion {
                creaseAngle 0.7853975
                endCap   TRUE
                beginCap TRUE
                solid    TRUE
                crossSection [
                        -0.5 -0.1,
                        -0.5  0.1,
                         0.5  0.1,
                         0.5 -0.1,
                        -0.5 -0.1,
                ]
                spine [
                        0.000000 0.000000 2.000000,
                        0.415823 0.066667 1.956295,
                        0.813473 0.133333 1.827091,
                        1.175570 0.200000 1.618035,
                        1.486289 0.266667 1.338262,
                        1.732050 0.333333 1.000002,
                        1.902112 0.400000 0.618036,
                        1.989044 0.466667 0.209059,
                        1.989044 0.533333 -0.209054,
                        1.902114 0.600000 -0.618031,
                        1.732053 0.666667 -0.999997,
                        1.486292 0.733333 -1.338259,
                        1.175574 0.800000 -1.618032,
                        0.813478 0.866667 -1.827089,
                        0.415828 0.933333 -1.956294,
                        0.000005 1.000000 -2.000000,
                        -0.415818 1.066667 -1.956296,
                        -0.813468 1.133333 -1.827093,
                        -1.175565 1.200000 -1.618038,
                        -1.486285 1.266667 -1.338266,
                        -1.732047 1.333333 -1.000006,
                        -1.902111 1.400000 -0.618041,
                        -1.989043 1.466667 -0.209064,
                        -1.989045 1.533333 0.209049,
                        -1.902116 1.600000 0.618026,
                        -1.732055 1.666667 0.999992,
                        -1.486296 1.733333 1.338254,
                        -1.175578 1.800000 1.618028,
                        -0.813482 1.866667 1.827087,
                        -0.415833 1.933333 1.956293,
                        -0.000010 2.000000 2.000000,
                        0.415813 2.066667 1.956297,
                        0.813463 2.133333 1.827095,
                        1.175561 2.200000 1.618041,
                        1.486282 2.266667 1.338270,
```

```
                1.732045 2.333333 1.000011,
                1.902109 2.400000 0.618046,
                1.989043 2.466667 0.209069,
                1.989045 2.533333 -0.209044,
                1.902117 2.600000 -0.618020,
                1.732058 2.666667 -0.999987,
                1.486299 2.733333 -1.338251,
                1.175582 2.800000 -1.618026,
                0.813486 2.866667 -1.827085,
                0.415838 2.933333 -1.956292,
                0.000015 3.000000 -2.000000,
                -0.415808 3.066667 -1.956298,
                -0.813459 3.133333 -1.827098,
                -1.175557 3.200000 -1.618044,
                -1.486278 3.266667 -1.338274,
                -1.732042 3.333333 -1.000015,
                -1.902107 3.400000 -0.618051,
                -1.989042 3.466667 -0.209075,
                -1.989046 3.533333 0.209038,
                -1.902119 3.600000 0.618016,
                -1.732061 3.666667 0.999983,
                -1.486302 3.733333 1.338248,
                -1.175586 3.800000 1.618023,
                -0.813491 3.866667 1.827083,
                -0.813491 3.866667 1.827083,
                -0.415843 3.933333 1.956291,
        ]
        orientation [
                0.0 1.0 0.0  0.000000,
                0.0 1.0 0.0  0.209439,
                0.0 1.0 0.0  0.418879,
                0.0 1.0 0.0  0.628318,
                0.0 1.0 0.0  0.837757,
                0.0 1.0 0.0  1.047197,
                0.0 1.0 0.0  1.256636,
                0.0 1.0 0.0  1.466075,
                0.0 1.0 0.0  1.675515,
                0.0 1.0 0.0  1.884954,
                0.0 1.0 0.0  2.094393,
                0.0 1.0 0.0  2.303833,
                0.0 1.0 0.0  2.513272,
                0.0 1.0 0.0  2.722711,
                0.0 1.0 0.0  2.932151,
                0.0 1.0 0.0  3.141590,
                0.0 1.0 0.0  3.351029,
                0.0 1.0 0.0  3.560469,
                0.0 1.0 0.0  3.769908,
                0.0 1.0 0.0  3.979347,
                0.0 1.0 0.0  4.188787,
                0.0 1.0 0.0  4.398226,
                0.0 1.0 0.0  4.607666,
                0.0 1.0 0.0  4.817105,
                0.0 1.0 0.0  5.026544,
                0.0 1.0 0.0  5.235983,
                0.0 1.0 0.0  5.445423,
                0.0 1.0 0.0  5.654862,
                0.0 1.0 0.0  5.864302,
                0.0 1.0 0.0  6.073741,
                0.0 1.0 0.0  6.283180,
                0.0 1.0 0.0  6.492620,
                0.0 1.0 0.0  6.702059,
```

```
                0.0 1.0 0.0  6.911498,
                0.0 1.0 0.0  7.120937,
                0.0 1.0 0.0  7.330377,
                0.0 1.0 0.0  7.539816,
                0.0 1.0 0.0  7.749256,
                0.0 1.0 0.0  7.958695,
                0.0 1.0 0.0  8.168134,
                0.0 1.0 0.0  8.377573,
                0.0 1.0 0.0  8.587013,
                0.0 1.0 0.0  8.796453,
                0.0 1.0 0.0  9.005892,
                0.0 1.0 0.0  9.215331,
                0.0 1.0 0.0  9.424770,
                0.0 1.0 0.0  9.634210,
                0.0 1.0 0.0  9.843649,
                0.0 1.0 0.0  10.053088,
                0.0 1.0 0.0  10.262527,
                0.0 1.0 0.0  10.471967,
                0.0 1.0 0.0  10.681406,
                0.0 1.0 0.0  10.890845,
                0.0 1.0 0.0  11.100285,
                0.0 1.0 0.0  11.309724,
                0.0 1.0 0.0  11.519163,
                0.0 1.0 0.0  11.728603,
                0.0 1.0 0.0  11.938043,
                0.0 1.0 0.0  12.147482,
                0.0 1.0 0.0  12.356921,
        ]
        scale [
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
                1.0 1.0,
```

```
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.0 1.0,
                        1.5 1.5,
                        0.0 0.0,
                ]
        }
}
```

```
#VRML V2.0 utf8
#
#  A stop-watch with automatically moving second, minute,
#  and hour hands, start and stop buttons, and a stop-watch sweep
#  second hand
#
DEF StopWatch Transform {
    # rotation animated
    # scale animated
    children [
    # Frame and face
        Transform {
            rotation 1.0 0.0 0.0  1.571
            children [
            # Frame
                Shape {
                    appearance Appearance {
                        material Material { diffuseColor 0.7 0.3 0.0 }
                    }
                    geometry Cylinder {
                        radius 4.8
                        height 0.8
                    }
                }
            # Face
                Shape {
                    appearance Appearance {
                        material Material { diffuseColor 1.0 1.0 1.0 }
                    }
                    geometry Cylinder {
                        radius 4.0
                        height 0.9
                    }
                }
            ]
        }
    # Start button
        Transform {
            translation 3.5 3.5 0.0
            rotation 0.0 0.0 1.0 -0.71
            children [
                DEF Start TouchSensor { }
                Shape {
                    appearance Appearance {
                        material Material { diffuseColor 0.0 1.0 0.0 }
                    }
                    geometry Cylinder {
                        radius 0.38
                        height 0.3
                    }
                }
            ]
        }
    # Stop button
        Transform {
            translation -3.5 3.5 0.0
            rotation 0.0 0.0 1.0 0.71
            children [
                DEF Stop TouchSensor { }
                Shape {
                    appearance Appearance {
```

```
                        material Material { diffuseColor 1.0 0.0 0.0 }
                    }
                    geometry Cylinder {
                        radius 0.38
                        height 0.3
                    }
                }
            ]
        }
    # Hands
        DEF MinuteHand Transform {
            translation 0.0 1.5 0.6
            center 0.0 -1.5 0.6
            # animated rotation
            children [
            # Arm
                DEF Arm Shape {
                    appearance DEF Black Appearance {
                        material Material { diffuseColor 0.2 0.2 0.2 }
                    }
                    geometry Cylinder {
                        radius 0.17
                        height 3.0
                    }
                }
            # Pointy end
                DEF ArrowHead Transform {
                    translation 0.0 1.9 0.0
                    children Shape {
                        appearance USE Black
                        geometry Cone {
                            bottomRadius 0.4
                            height 0.8
                        }
                    }
                }
            ]
        }
        DEF HourHand Transform {
            translation 0.0 1.5 0.6
            center 0.0 -1.5 0.6
            # animated rotation
            scale 1.0 0.7 1.0
            children [ USE Arm, USE ArrowHead ]
        }
        DEF SecondHand Transform {
            translation 0.0 1.5 0.6
            center 0.0 -1.5 0.6
            # animated rotation
            scale 0.6 1.0 0.6
            children [ USE Arm, USE ArrowHead ]
        }
        DEF SweepHand Transform {
            translation 0.0 1.9 0.6
            center 0.0 -1.9 0.6
            # animated rotation
            scale 0.6 1.0 0.6
            children Shape {
                appearance DEF Black Appearance {
                    material Material { diffuseColor 1.0 0.0 0.0 }
```

```
                }
                geometry Cylinder {
                    radius 0.17
                    height 3.8
                }
            }
        }
    ]
}

#
#  Timers and interpolators to spin hands
#
DEF SecondTimer TimeSensor {
    cycleInterval 60.0      # 60 seconds per sweep
    loop TRUE
    startTime 0.0
    stopTime -1.0
}
DEF MinuteTimer TimeSensor {
    cycleInterval 3600.0    # 60*60 seconds per sweep
    loop TRUE
    startTime 0.0
    stopTime -1.0
}
DEF HourTimer TimeSensor {
    cycleInterval 43200.0   # 60*60*12 seconds per sweep
    loop TRUE
    startTime 28800.0       # Adjust for Pacific Standard Time
        # start time of 0 is midnight Grenwich Mean Time (GMT)
        # Pacific Mean Time (PST) is 8 hours behind GMT
    stopTime -1.0
}

DEF SecondSpinner OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0,  0.0 0.0 1.0 -3.14,  0.0 0.0 1.0 -6.28 ]
}
DEF MinuteSpinner OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0,  0.0 0.0 1.0 -3.14,  0.0 0.0 1.0 -6.28 ]
}
DEF HourSpinner OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0,  0.0 0.0 1.0 -3.14,  0.0 0.0 1.0 -6.28 ]
}

ROUTE SecondTimer.fraction_changed TO SecondSpinner.set_fraction
ROUTE MinuteTimer.fraction_changed TO MinuteSpinner.set_fraction
ROUTE HourTimer.fraction_changed   TO HourSpinner.set_fraction
ROUTE SecondSpinner.value_changed  TO SecondHand.set_rotation
ROUTE MinuteSpinner.value_changed  TO MinuteHand.set_rotation
ROUTE HourSpinner.value_changed    TO HourHand.set_rotation

#
#  Timer and interpolators to spin stop watch hand
#
DEF SweepTimer TimeSensor {
    cycleInterval 60.0      # 60 seconds per sweep
    loop TRUE
    startTime 0.0
```

```
    # start time set on start button press
    stopTime 1.0
    # stop time set on stop button press
}
DEF SweepSpinner OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 0.0 0.0 1.0 0.0,  0.0 0.0 1.0 -3.14,  0.0 0.0 1.0 -6.28 ]
}

ROUTE Start.touchTime TO SweepTimer.set_startTime
ROUTE Stop.touchTime  TO SweepTimer.set_stopTime
ROUTE SweepTimer.fraction_changed TO SweepSpinner.set_fraction
ROUTE SweepSpinner.value_changed TO SweepHand.set_rotation

#
#  Timers and interpolators for quarter-hour animations
#
DEF QuarterHour TimeSensor {
    cycleInterval 900.0    # 60*15 seconds per action
    loop TRUE
    startTime 28800.0      # PST
    stopTime -1.0
}
DEF QuarterAnimation TimeSensor {
    cycleInterval 3.0
    loop FALSE
    startTime -1.0
    # start time set by quarter-hour clock
    stopTime 0.0
}

DEF QuarterSpinner OrientationInterpolator {
    key [ 0.0, 0.5, 1.0 ]
    keyValue [ 1.0 1.0 0.0 0.0,  1.0 1.0 0.0 -3.14,  1.0 1.0 0.0 -6.28 ]
}
DEF QuarterSquisher PositionInterpolator {
    key [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
    keyValue [
        1.0 1.0 1.0,  0.1 3.0 1.2,  3.0 0.1 1.0,  0.3 2.0 1.2,
        1.0 1.0 1.0,
    ]
}

ROUTE QuarterHour.cycleTime TO QuarterAnimation.set_startTime
ROUTE QuarterAnimation.fraction_changed TO QuarterSpinner.set_fraction
ROUTE QuarterAnimation.fraction_changed TO QuarterSquisher.set_fraction
ROUTE QuarterSpinner.value_changed TO StopWatch.set_rotation
ROUTE QuarterSquisher.value_changed TO StopWatch.set_scale
```

```
#VRML V2.0 utf8

# WHO:    John L. Moreland & David R. Nadeau
# WHAT:   Timed Timer
# WHY:    Demonstrates how to drive one TimeSensor with another
#         in order to cause periodic events to trigger animation
# WHERE:  San Diego Supercomputer Center
# WHEN:   Tue Apr 29 09:10:45 PDT 1997
# HOW:    VRML 2.0

# Red Boxes

DEF Red Transform {
        # animated scale
        children [
                Shape {
                        appearance Appearance {
                                material Material {
                                        diffuseColor 1.0 0.0 0.0
                                }
                        }
                        geometry Box { }
                }
        ]
}

Transform { translation  2.0  2.0  0.0 children USE Red }
Transform { translation -2.0  2.0  0.0 children USE Red }
Transform { translation  2.0 -2.0  0.0 children USE Red }
Transform { translation -2.0 -2.0  0.0 children USE Red }

Transform { translation  0.0 -2.0  2.0 children USE Red }
Transform { translation  0.0  2.0  2.0 children USE Red }
Transform { translation  2.0  0.0  2.0 children USE Red }
Transform { translation -2.0  0.0  2.0 children USE Red }

Transform { translation  0.0 -2.0 -2.0 children USE Red }
Transform { translation  0.0  2.0 -2.0 children USE Red }
Transform { translation  2.0  0.0 -2.0 children USE Red }
Transform { translation -2.0  0.0 -2.0 children USE Red }

# Blue Boxes

DEF Blue Transform {
        translation 2.0 0.0 0.0
        # animated scale
        children [
                Shape {
                        appearance Appearance {
                                material Material {
                                        diffuseColor 0.0 0.0 1.0
                                }
                        }
                        geometry Box { }
                }
        ]
}

Transform { translation -4.0  0.0  0.0 children USE Blue }
Transform { translation -2.0  0.0  2.0 children USE Blue }
Transform { translation -2.0  0.0 -2.0 children USE Blue }
Transform { translation -2.0  2.0  0.0 children USE Blue }
```

```
Transform { translation -2.0 -2.0  0.0 children USE Blue }

Transform { translation  0.0  2.0  2.0 children USE Blue }
Transform { translation  0.0  2.0 -2.0 children USE Blue }
Transform { translation -4.0  2.0  2.0 children USE Blue }
Transform { translation -4.0  2.0 -2.0 children USE Blue }

Transform { translation  0.0 -2.0  2.0 children USE Blue }
Transform { translation  0.0 -2.0 -2.0 children USE Blue }
Transform { translation -4.0 -2.0  2.0 children USE Blue }
Transform { translation -4.0 -2.0 -2.0 children USE Blue }

DEF Clock TimeSensor {
        cycleInterval 3.0
        loop FALSE
}

DEF Trigger TimeSensor {
        loop TRUE
        cycleInterval 5.0
}

DEF RedScale PositionInterpolator {
        key [ 0.0, 0.5, 1.0 ]
        keyValue [
                1.0 1.0 1.0,
                0.0001 0.0001 0.0001,
                1.0 1.0 1.0,
        ]
}

DEF BlueScale PositionInterpolator {
        key [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
        keyValue [
                1.0 1.0 1.0,
                0.0001 0.0001 0.0001,
                1.0 1.0 1.0,
                0.0001 0.0001 0.0001,
                1.0 1.0 1.0,
        ]
}

ROUTE Trigger.cycleTime       TO Clock.set_startTime
ROUTE Clock.fraction_changed  TO RedScale.set_fraction
ROUTE Clock.fraction_changed  TO BlueScale.set_fraction
ROUTE RedScale.value_changed  TO Red.set_scale
ROUTE BlueScale.value_changed TO Blue.set_scale
```

Mapeando texturas

Motivação

Exemplo

Exemplo de Texturas

Usandos tipos de textura

Sintaxe: Appearance

Usando materiais com texturas

Colorindo texturas

Sintaxe: ImageTexture

Sintaxe: PixelTexture

Sintaxe: MovieTexture

Usando texturas transparentes

Um exemplo de textura transparente

Um exemplo de textura transparente

Resumo

```
#VRML V2.0 utf8
#
#  snake.wrl
#  A morphing snake
#       by David R. Nadeau
#
#  This world creates a snake using an Extrusion node.  To make the snake
#  move, a CoordinateInterpolator sends the Extrusion node a series of
#  new spines.
#

# Ground

Shape {
        appearance Appearance {
                material Material {
                        diffuseColor 0.6 0.6 0.0
                }
        }
        geometry Box { size 20.0 0.01 20.0 }
}

# Snake shape

Transform {
        translation 0.0 0.3 0.0
        children Shape {
                appearance Appearance {
                        material Material {
                                diffuseColor 0.0 1.0 0.2
                        }
                }
                geometry DEF Snake Extrusion {
                        creaseAngle 1.57
                        crossSection [
                                # Circle
                                1.00  0.00,   0.92 -0.38,
                                0.71 -0.71,   0.38 -0.92,
                                0.00 -1.00,  -0.38 -0.92,
                                -0.71 -0.71,  -0.92 -0.38,
                                -1.00 -0.00,  -0.92  0.38,
                                -0.71  0.71,  -0.38  0.92,
                                0.00  1.00,   0.38  0.92,
                                0.71  0.71,   0.92  0.38,
                                1.00  0.00
                        ]
                        spine [
                                # Sine wave
                                -4.100 0.0  0.000,  -4.000 0.0  0.000,
                                -3.529 0.0  0.674,  -3.059 0.0  0.996,
                                -2.588 0.0  0.798,  -2.118 0.0  0.184,
                                -1.647 0.0 -0.526,  -1.176 0.0 -0.962,
                                -0.706 0.0 -0.895,  -0.235 0.0 -0.361,
                                0.235 0.0  0.361,   0.706 0.0  0.895,
                                1.176 0.0  0.962,   1.647 0.0  0.526,
                                2.118 0.0 -0.184,   2.588 0.0 -0.798,
                                3.059 0.0 -0.996,   3.529 0.0 -0.674,
                                4.000 0.0  0.000,
                        ]
                        scale [
                                0.050 0.020,  0.200 0.100,
```

```
                                0.400 0.150,  0.300 0.300,
                                0.300 0.300,  0.300 0.300,
                                0.300 0.300,  0.300 0.300,
                                0.300 0.300,  0.300 0.300,
                                0.290 0.290,  0.290 0.290,
                                0.290 0.290,  0.280 0.280,
                                0.280 0.280,  0.250 0.250,
                                0.200 0.200,  0.100 0.100,
                                0.050 0.050,
                        ]
                }
        }
}

# Animation clock

DEF Clock TimeSensor {
        cycleInterval 4.0
        loop TRUE
}

# Animation morph

DEF SnakeWiggle CoordinateInterpolator {
        key [ 0.0, 0.25, 0.50, 0.75, 1.0 ]
        keyValue [
                # time 0.0 position
                -4.100 0.0  0.000,  -4.000 0.0  0.000,
                -3.529 0.0  0.674,  -3.059 0.0  0.996,
                -2.588 0.0  0.798,  -2.118 0.0  0.184,
                -1.647 0.0 -0.526,  -1.176 0.0 -0.962,
                -0.706 0.0 -0.895,  -0.235 0.0 -0.361,
                0.235 0.0  0.361,   0.706 0.0  0.895,
                1.176 0.0  0.962,   1.647 0.0  0.526,
                2.118 0.0 -0.184,   2.588 0.0 -0.798,
                3.059 0.0 -0.996,   3.529 0.0 -0.674,
                4.000 0.0  0.000,
                # time 0.25 position
                -4.100 0.0 -1.000,  -4.000 0.0 -1.000,
                -3.529 0.0 -0.739,  -3.059 0.0 -0.092,
                -2.588 0.0  0.603,  -2.118 0.0  0.983,
                -1.647 0.0  0.850,  -1.176 0.0  0.274,
                -0.706 0.0 -0.446,  -0.235 0.0 -0.932,
                0.235 0.0 -0.932,   0.706 0.0 -0.446,
                1.176 0.0  0.274,   1.647 0.0  0.850,
                2.118 0.0  0.983,   2.588 0.0  0.603,
                3.059 0.0 -0.092,   3.529 0.0 -0.739,
                4.000 0.0 -1.000,
                # time 0.50 position
                -4.100 0.0  0.000,  -4.000 0.0  0.000,
                -3.529 0.0 -0.674,  -3.059 0.0 -0.996,
                -2.588 0.0 -0.798,  -2.118 0.0 -0.184,
                -1.647 0.0  0.526,  -1.176 0.0  0.962,
                -0.706 0.0  0.895,  -0.235 0.0  0.361,
                0.235 0.0 -0.361,   0.706 0.0 -0.895,
                1.176 0.0 -0.962,   1.647 0.0 -0.526,
                2.118 0.0  0.184,   2.588 0.0  0.798,
                3.059 0.0  0.996,   3.529 0.0  0.674,
                4.000 0.0  0.000,
                # time 0.75 position
                -4.100 0.0  1.000,  -4.000 0.0  1.000,
```

```
                -3.529 0.0  0.739,  -3.059 0.0  0.092,
                -2.588 0.0 -0.603,  -2.118 0.0 -0.983,
                -1.647 0.0 -0.850,  -1.176 0.0 -0.274,
                -0.706 0.0  0.446,  -0.235 0.0  0.932,
                0.235 0.0  0.932,   0.706 0.0  0.446,
                1.176 0.0 -0.274,   1.647 0.0 -0.850,
                2.118 0.0 -0.983,   2.588 0.0 -0.603,
                3.059 0.0  0.092,   3.529 0.0  0.739,
                4.000 0.0  1.000,
                # time 1.0 position
                -4.100 0.0  0.000,  -4.000 0.0  0.000,
                -3.529 0.0  0.674,  -3.059 0.0  0.996,
                -2.588 0.0  0.798,  -2.118 0.0  0.184,
                -1.647 0.0 -0.526,  -1.176 0.0 -0.962,
                -0.706 0.0 -0.895,  -0.235 0.0 -0.361,
                0.235 0.0  0.361,   0.706 0.0  0.895,
                1.176 0.0  0.962,   1.647 0.0  0.526,
                2.118 0.0 -0.184,   2.588 0.0 -0.798,
                3.059 0.0 -0.996,   3.529 0.0 -0.674,
                4.000 0.0  0.000,
        ]
}

ROUTE Clock.fraction_changed    TO SnakeWiggle.set_fraction
ROUTE SnakeWiggle.value_changed TO Snake.set_spine
```

Mapeando texturas

Motivação

. Você pode modelar qualquer tipo de detalhe de textura de um mundo usando um vasto número de superfícies coloridas

- **Leva muito tempo para escrever em VRML**
- **Leva muito tempo para desenhar**

. Usar um truque

- **Tirar uma foto de algo real**
- **Colar aquela foto na forma, tal qual um decalque**

. Esta técnica é chamada de *Mapeamento de Textura*

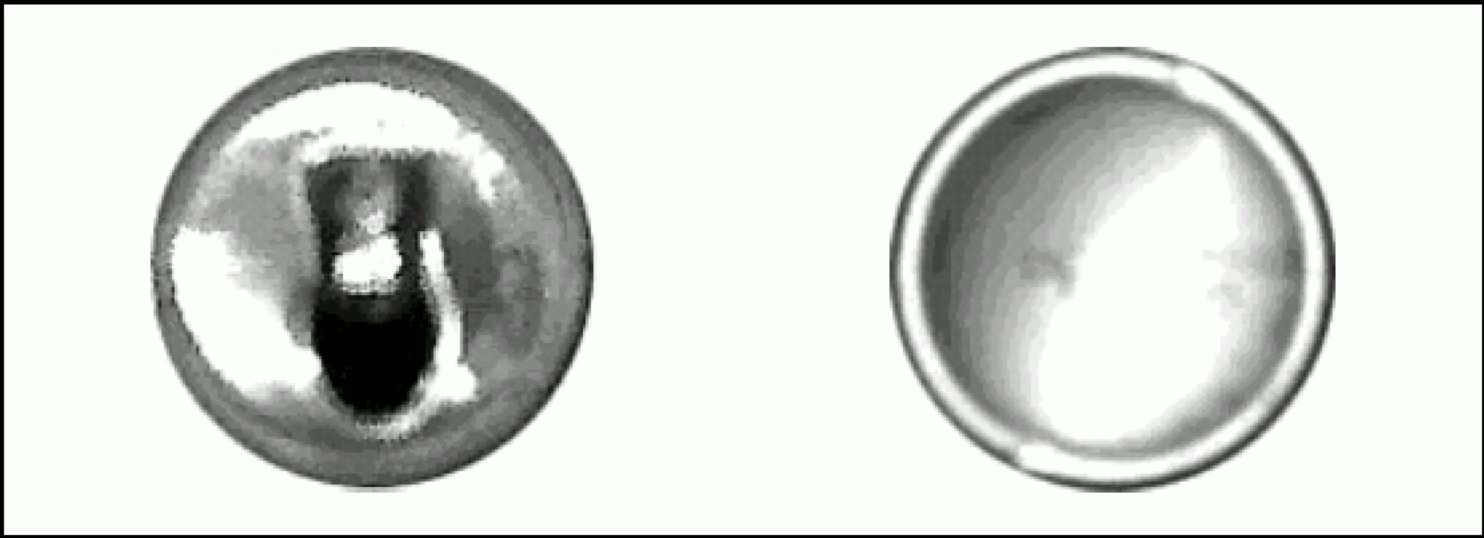
Mapeando texturas

Exemplo

[can.wrl]

Mapeando texturas

Exemplos de Texturas



Mapeando texturas

Usando tipos de texturas

- ***Texturas de imagens***
  - **Um simples arquivo de imagem**
  - **Formato JPEG, GIF ou PNG**
- ***Texturas de pixel***
  - **Uma simples imagem, dada do próprio arquivo VRML**
- ***Texturas de filmes***
  - **Um filme de arquivo**
  - **Formato MPEG**

Mapeando texturas

Sintaxe: Appearance

. **Um nó `Appearance` descreve a aparência completa de uma forma**

- ***texture*** **- textura fonte**

```
Appearance {
    material Material { . . . }
    texture ImageTexture { . . . }
}
```

Mapeando texturas

Usando materiais com texturas

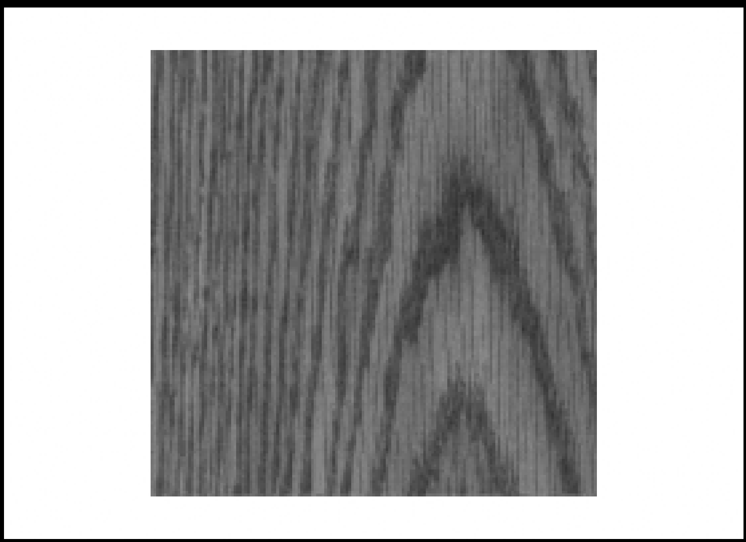
. Texturas coloridas sobrescrevem a cor no nó `Material`

. Texturas em tons de cinza se misturam com a cor do nó `Material`

- Bom para colorir texturas em tons de cinza

Mapeando texturas

Colorindo texturas



Mapeando texturas

Sintaxe: ImageTexture

- **Um nó `ImageTexture` seleciona uma textura de imagem para mapear textura**
  - ***url*** **- URL do arquivo da textura de imagem**

```
ImageTexture {
    url "wood.jpg"
}
```

Mapeando texturas

Sintaxe: PixelTexture

. Um nó `PixelTexture` especifica pixels em texturas de imagem para mapear texturas

- ***image*** **pixels - pixels de textura de imagem**
- **image data - comprimento, altura, bytes/pixel, valores de pixels**

```
PixelTexture {
    image 2 1 3 0xFFFF00 0xFF0000
}
```

Mapeando texturas

Sintaxe: MovieTexture

- **Um nó `MovieTexture` seleciona uma textura de filme para mapear textura**
  - ***url* - URL do arquivo de textura de filme**
  - **Quando começar o filme**
  - **Velocidade (como um nó `TimeSensor`)**

```
MovieTexture {
    url "movie.mpg"
    loop TRUE
    speed 1.0
}
```

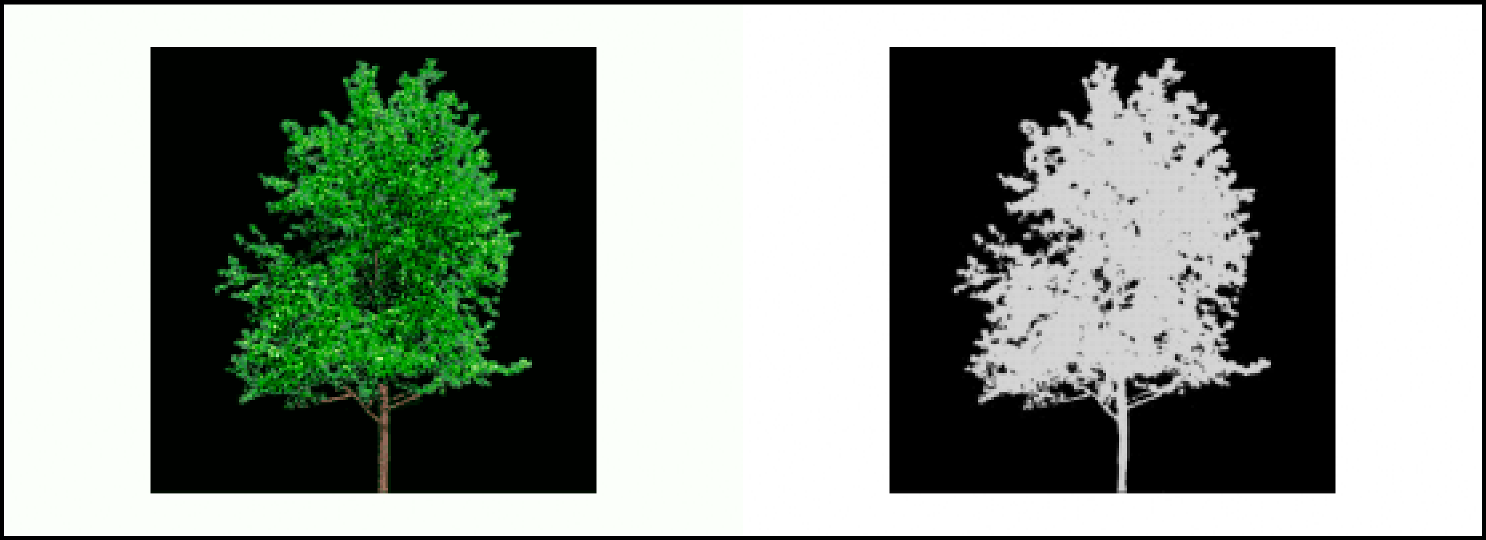
Mapeando texturas

Usando texturas transparentes

- Texturas de imagens podem incluir valores de cores e transparência para cada pixel
- Transparência de pixel permite que você faça partes de uma forma transparente
  - Janelas, redes, buracos
  - Árvores, sombras

Mapeando texturas

Um exemplo de textura transparente



Mapeando texturas

Um exemplo de textura transparente

[treewall.wrl]

Mapeando texturas

Resumo

. Uma textura é como um decalque colocado em uma forma

. Especificar a textura usando um nó `ImageTexture`, `PixelTexture`, ou `MovieTexture` no nó `Appearance`

. Cores de texturas sobrescrevem as cores do material, texturas em tons de cinza se misturam

. Texturas com transparências criam buracos

```
#VRML V2.0 utf8

Shape {
    appearance Appearance {
        material Material { }
        texture ImageTexture { url "cantop.jpg" }
    }
    geometry Cylinder {
        height  2.7
        side    FALSE
        bottom  FALSE
    }
}
Shape {
    appearance Appearance {
        material Material { }
        texture ImageTexture { url "canbot.jpg" }
    }
    geometry Cylinder {
        height  2.7
        side    FALSE
        top     FALSE
    }
}
Shape {
    appearance Appearance {
        material Material { }
        texture ImageTexture { url "canlabel.jpg" }
    }
    geometry Cylinder {
        height  2.8
        top     FALSE
        bottom  FALSE
    }
}
```

```
#VRML V2.0 utf8
#
#  treewall.wrl
#  A set of trees in front of a wall
#    by David R. Nadeau
#
#  This world illustrates the use of textures with a transparency
#  channel (alpha channel).  Such textures include pixel-by-pixel
#  info about what parts of the image are transparent.  A tree texture,
#  for instance, should be opaque where there are leaves, branches, or
#  the trun, and transparent everywhere else.  To illustrate this
#  use of transparency channel textures, this world places three trees
#  side-by-side in front of a wall.  The left 'tree' polygon has no
#  texture on it.  The middle polygon has a tree texture without
#  the transparency channel.  The right polygon has a tree texture with
#  a transparency texture.  All three trees are in Billboard nodes so
#  that they always turn to face the viewer.
#

Transform {    # Translate down so viewer is at "eye level"
    translation 0 -2 0
    children [

# Ground and lower wall
        Shape {
            appearance DEF Concrete Appearance {
                material Material { }
            }
            geometry IndexedFaceSet {
                coord Coordinate {
                    point [
                        -5.0 0.0   5.0,   5.0 0.0   5.0,
                         5.0 0.0   1.0,  -5.0 0.0   1.0,
                         5.0 0.05  1.0,  -5.0 0.05  1.0,
                         5.0 0.05 -2.0,  -5.0 0.05 -2.0,
                         5.0 0.25 -2.0,  -5.0 0.25 -2.0,
                         5.0 0.25 -2.05, -5.0 0.25 -2.05,
                    ]
                }
                coordIndex [
                    0, 1, 2, 3, -1,
                    3, 2, 4, 5, -1,
                    5, 4, 6, 7, -1,
                    7, 6, 8, 9, -1,
                    9, 8, 10, 11, -1,
                ]
                solid FALSE
            }
        }

    # Brick wall
        Shape {
            appearance DEF Brick Appearance {
                material Material { }
                texture ImageTexture { url "brick.jpg" }
                textureTransform TextureTransform {
                    scale 15.0 15.0
                }
            }
            geometry IndexedFaceSet {
                coord Coordinate {
```

```
                point [
                    -5.0 0.25 -2.05,  5.0 0.25 -2.05,
                     5.0 5.0  -2.05, -5.0 5.0  -2.05,
                ]
            }
            coordIndex [ 0, 1, 2, 3 ]
            texCoord TextureCoordinate {
                point [
                    0.0 0.0, 1.0 0.0,
                    1.0 1.0, 0.0 1.0,
                ]
            }
            texCoordIndex [ 0, 1, 2, 3 ]
            solid FALSE
        }
    }

# Planter
    DEF Planter Transform {
        scale 0.5 0.5 0.5
        children [
            Shape {
                appearance USE Concrete
                geometry IndexedFaceSet {
                    coord DEF PlanterCoordinates Coordinate {
                        point [
                            -1.0 0.5  1.0,
                             1.0 0.5  1.0,
                             1.0 0.5 -1.0,
                            -1.0 0.5 -1.0,
                            -0.8 0.5  0.9,
                             0.8 0.5  0.9,
                             0.8 0.5 -0.9,
                            -0.8 0.5 -0.9,
                            -1.0 0.0  1.0,
                             1.0 0.0  1.0,
                             1.0 0.0 -1.0,
                            -1.0 0.0 -1.0,
                            -0.8 0.3  0.9,
                             0.8 0.3  0.9,
                             0.8 0.3 -0.9,
                            -0.8 0.3 -0.9,
                        ]
                    }
                    coordIndex [
                        0, 1, 5, 4, -1,
                        1, 2, 6, 5, -1,
                        2, 3, 7, 6, -1,
                        3, 0, 4, 7, -1,
                        0, 8, 9, 1, -1,
                        9, 10, 2, 1, -1,
                        10, 11, 3, 2, -1,
                        11, 8, 0, 3, -1,
                        15, 14, 6, 7, -1,
                        14, 13, 5, 6, -1,
                        13, 12, 4, 5, -1,
                        12, 15, 7, 4, -1,
                    ]
                }
            }
            Shape {
```

```
                    appearance Appearance {
                        material Material {
                            diffuseColor 0.0 0.5 0.0
                        }
                    }
                    geometry IndexedFaceSet {
                        coord USE PlanterCoordinates
                        coordIndex [ 12, 13, 14, 15 ]
                    }
                }
            ]
        }

    # Tree faces

    # Right tree
        Transform {
            translation  3.5 0.0 0.0
            children [
                USE Planter
                Billboard {
                    axisOfRotation 0.0 1.0 0.0
                    children [
                        Shape {
                            appearance Appearance {
                                # No material, use emissive texturing
                                texture ImageTexture { url "tree1.png" }
                            }
                            geometry DEF TreeFace IndexedFaceSet {
                                coord Coordinate {
                                    point [
                                        -1.51 0.05 0.0,   1.51 0.05 0.0,
                                         1.51 3.05 0.0,  -1.51 3.05 0.0,
                                    ]
                                }
                                coordIndex [ 0, 1, 2, 3 ]
                                texCoord TextureCoordinate {
                                    point [
                                        0.0 0.0, 1.0 0.0,
                                        1.0 1.0, 0.0 1.0,
                                    ]
                                }
                                texCoordIndex [ 0, 1, 2, 3 ]
                                solid FALSE
                            }
                        }
                    ]
                }
            ]
        }
        Transform {
            children [
                DEF Tree Billboard {
                    axisOfRotation 0.0 1.0 0.0
                    children [
                        Shape {
                            appearance Appearance {
                                # No material, use emissive texturing
                                texture ImageTexture { url "tree1.jpg" }
                            }
                            geometry USE TreeFace
```

```
                        }
                    ]
                }
            ]
        }

    # Left tree
        # Now do one face without the tree texture to show the difference
        Transform {
            translation -3.5 0.0 0.0
            children [
                USE Planter
                Billboard {
                    axisOfRotation 0.0 1.0 0.0
                    children [
                        Shape {
                            appearance Appearance {
                                material Material { }
                            }
                            geometry USE TreeFace
                        }
                    ]
                }
            ]
        }
    ]
}
```

Controlando como texturas são mapeadas

Motivação

Trabalhando direto com o processo de texturização

Usando o sistema de coordenadas de textura

Coordenadas de textura e transformações

Trabalhando direto com o processo de textura

Sintaxe: TextureCoordinate

Sintaxe: IndexedFaceSet

Sintaxe: ElevationGrid

Sintaxe: Appearance

Sintaxe: TextureTransform

Um exemplo não usando tensformação

Um exemplo usando translaão

Um exemplo usando rotação

Um exemplo usando escalonamento

[Um exemplo usando coordenadas de textura]

[Um exemplo usando escalonamento]

[Um exemplo usando escalonamento e rotação]

[Resumo]

Controlando como texturas são mapeadas

Motivação

- **Por default, uma textura de imagem é mapeada inteiramente ao redor de uma forma**
- **Você também pode:**
  - **Extrair pedaços interessantes**
  - **Criar padrões repetitivos**

Controlando como texturas são mapeadas

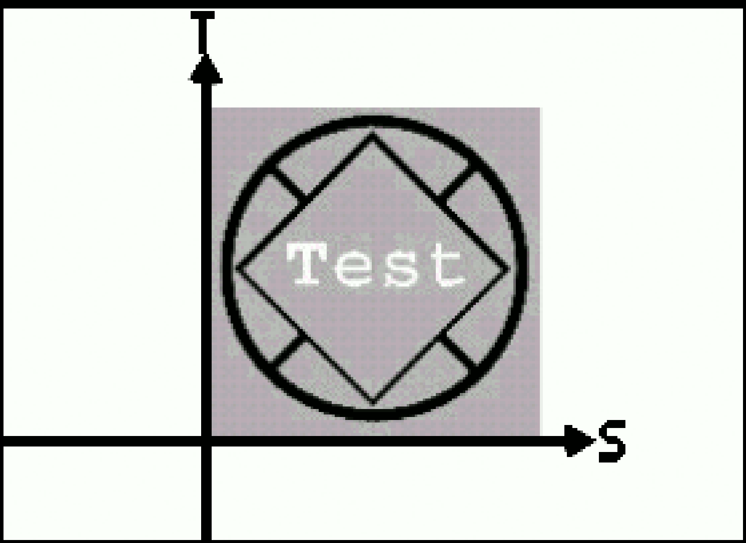
Trabalhando direto com o processo de texturização

- **Imagine que uma textura de imagem é um grande pedaço de um bolo elástico**
- **Selecione um pedaço de textura de imagem**
  - **Defina a forma do cortador de bolo**
  - **Posicione e oriente o cortador de bolo**
  - **Retire um pedaço da textura de farinha**
- **Estique a textura do bolo elástico para se a ajustar a uma superfície**

Controlando como texturas são mapeadas

Usando o sistema de coordenadas de textura

. Texturas de imagens (a massa de farinha) estão em um *sistema de coordenadas de textura*



. Direção *S* é horizontal

. Direção *T* é vertical

. (0,0) em inferior-esquerdo

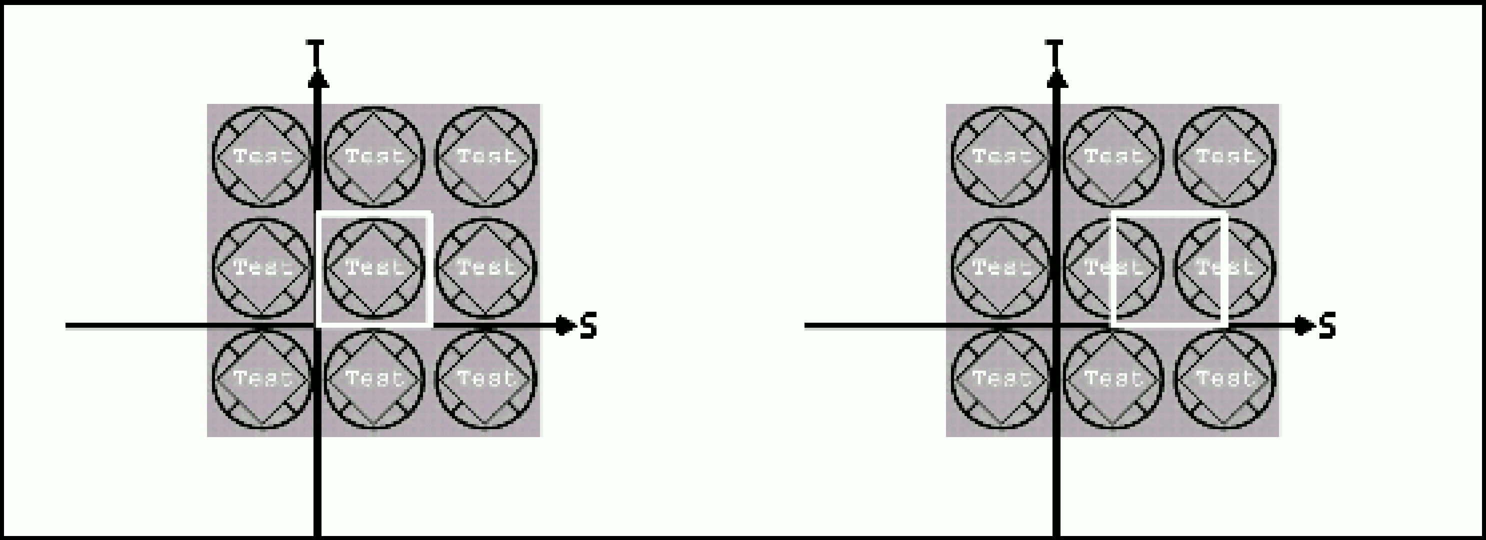
. (1,1) at superior-direito

Controlando como texturas são mapeadas

Coordenadas de textura e transformações

. ***Coordenadas de texturas*** **e** ***índices de coordenadas de textura*** **especificam um pedaço de textura de uma forma (o cortador de bolo da forma)**

. ***Transformações de texturas*** **transladam, rotacionam e escalonam as coordenadas de texturas (posicionam o cortador de bolo)**



Controlando como texturas são mapeadas

Trabalhando direto com o processo de textura

- **Selecione pedaços com coordenadas de textura e índices**
  - **Crie um cortador de bolo**
- **Transforme as coordenadas de textura**
  - **Posicione e oriente o cortador de bolo**
- **Amarre a textura à superfície**
  - **Retire e fixe a textura sobre uma superfície**
- **O processo é bastante similar ao cria superfícies!**

Controlando como texturas são mapeadas

Sintaxe: TextureCoordinate

. O nó `TextureCoordinate` contém uma lista de coordenadas de textura

```
TextureCoordinate {
    point [ 0.2 0.2, 0.8 0.2, . . . ]
}
```

. Usado como o valor do campo `texCoord` dos nós `IndexedFaceSet` ou `ElevationGrid`

Controlando como texturas são mapeadas

Sintaxe: IndexedFaceSet

- **Um nó geométrico IndexedFaceSet cria geometrias dentre superfícies**
  - ***Coordenadas de textura* e *índices* - especificam pedaços de textura**

```
IndexedFaceSet {
    coord Coordinate { . . . }
    coordIndex [ . . . ]
    texCoord TextureCoordinate { . . . }
    texCoordIndex [ . . . ]
}
```

Controlando como texturas são mapeadas

Sintaxe: ElevationGrid

- **Um nó geométrico `ElevationGrid` cria terrenos**
  - ***Texture coordinates* - especifica pedaços de textura**
  - **Gera automaticamente índices de coordenadas de textura**

```
ElevationGrid {
    height [ . . . ]
    texCoord TextureCoordinate { . . . }
}
```

Controlando como texturas são mapeadas

Sintaxe: Appearance

. Um nó `Appearance` descreve toda a aparência de uma forma

- ***textureTransform*** **- descreve a transformação**

```
Appearance {
    material Material { . . . }
    textureTransform TextureTransform {
. . . }
}
```

Controlando como texturas são mapeadas

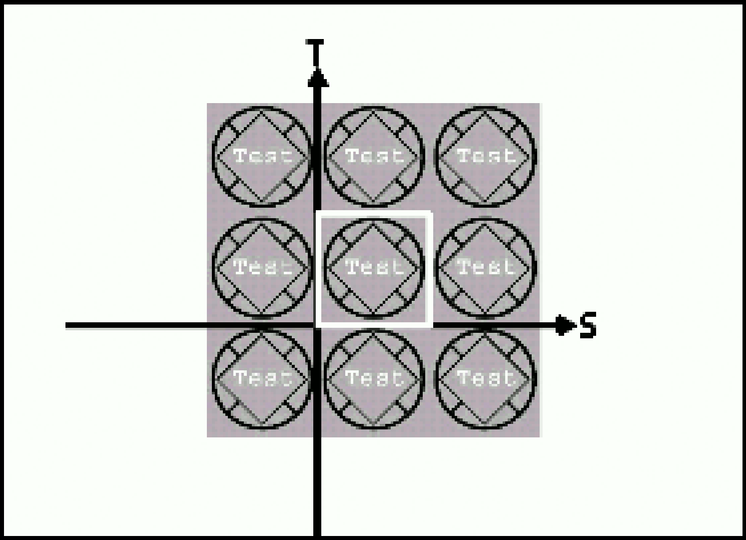
Sintaxe: TextureTransform

- **Um nó `TextureTransform` transforma coordenadas de textura**
  - ***translation* - posição**
  - ***rotation* - orientação**
  - ***scale* - tamanho**

```
TextureTransform {
    translation . . .
    rotation    . . .
    scale       . . .
}
```

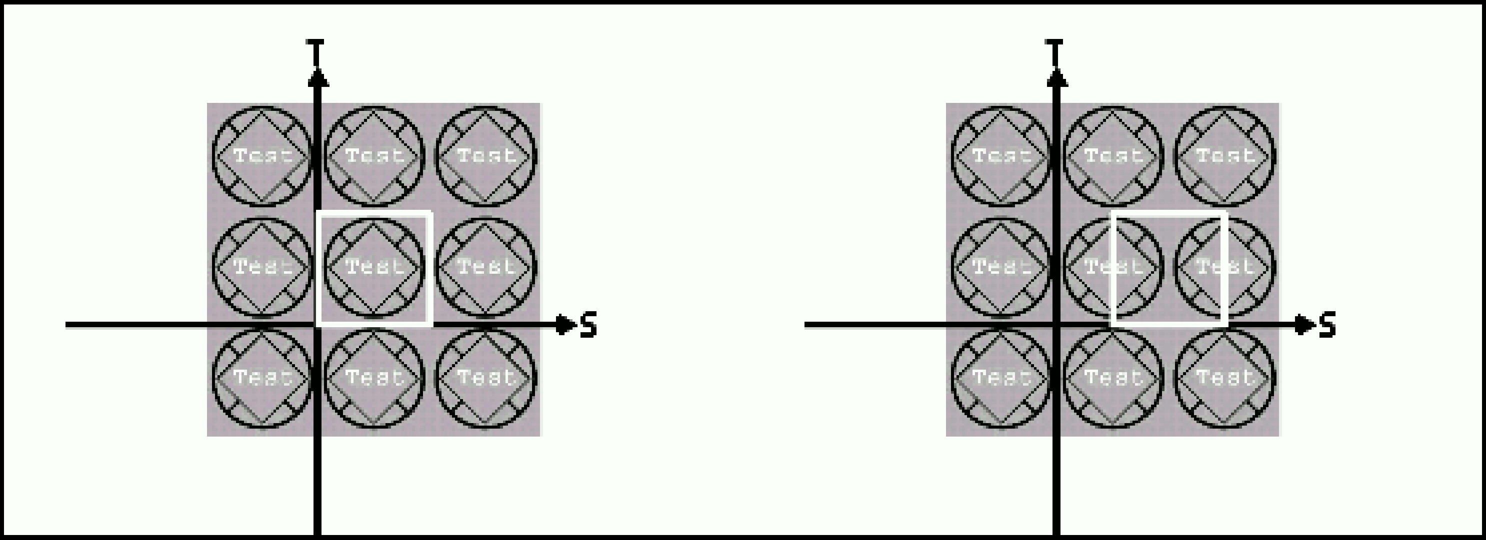
Controlando como texturas são mapeadas

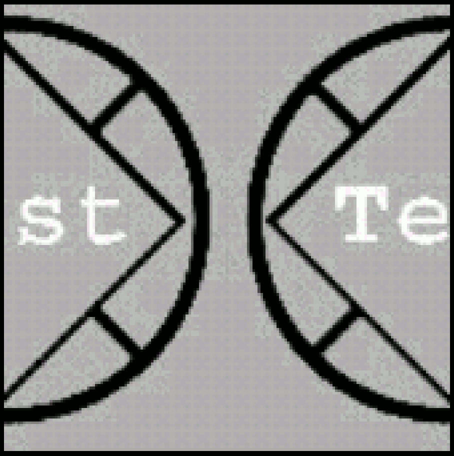
Um exemplo sem usar transformação



Controlando como texturas são mapeadas

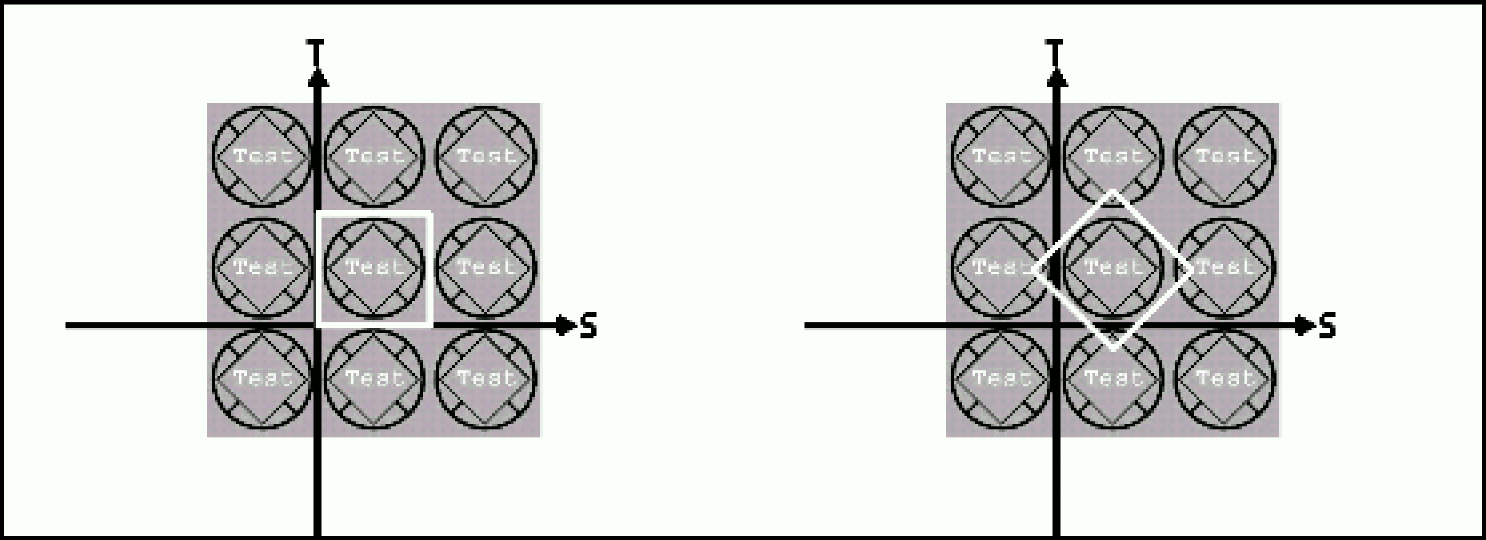
Um exemplo usando translação

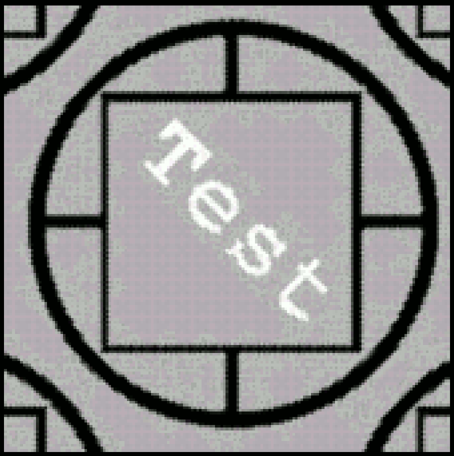




Controlando como texturas são mapeadas

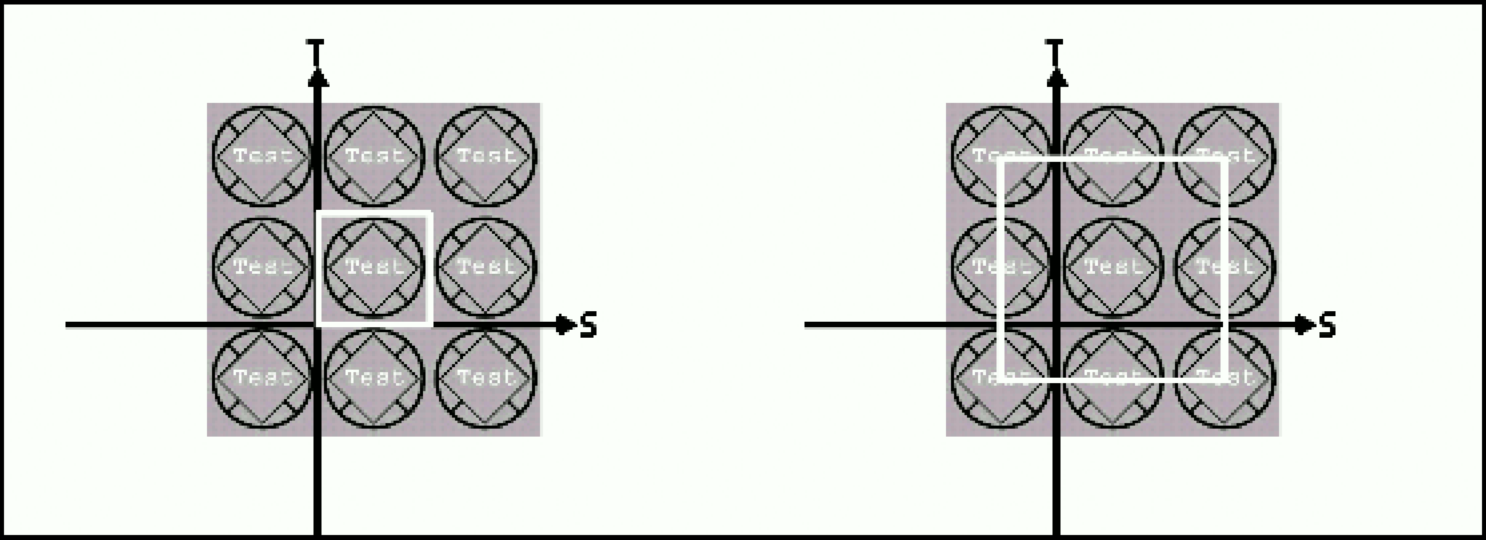
Um exemplo usando rotação





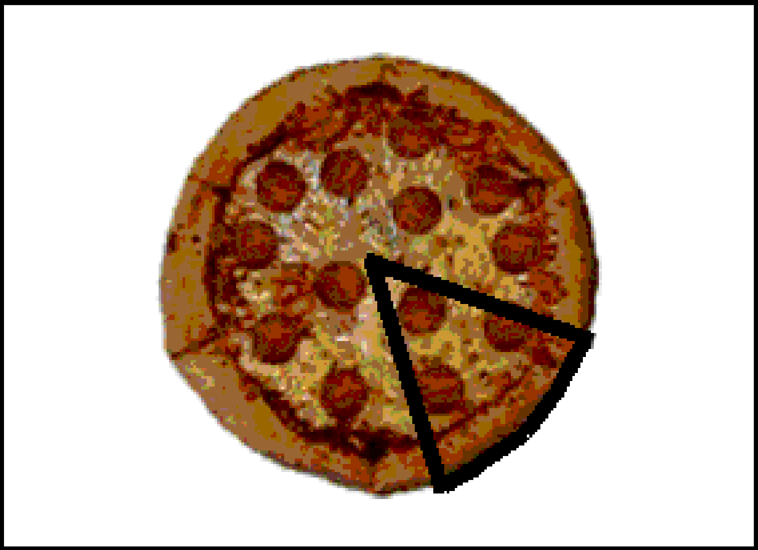
Controlando como texturas são mapeadas

Um exemplo usando escalonamento



Controlando como texturas são mapeadas

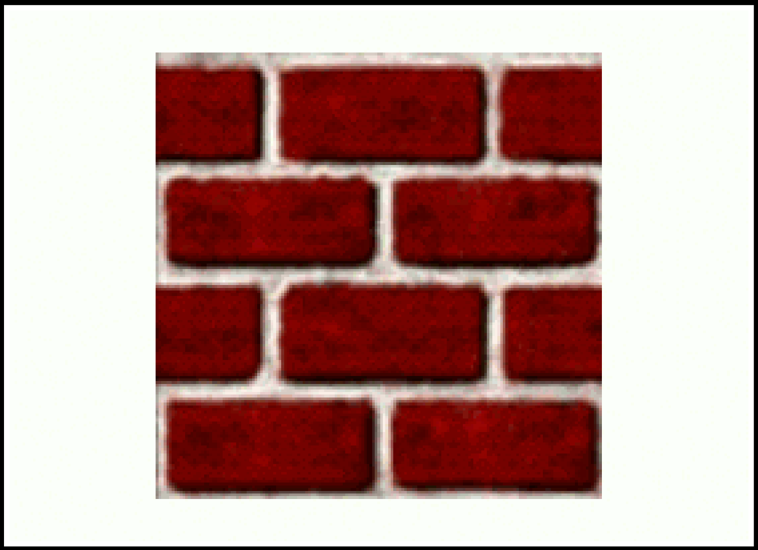
Um exemplo usando coordenadas de textura

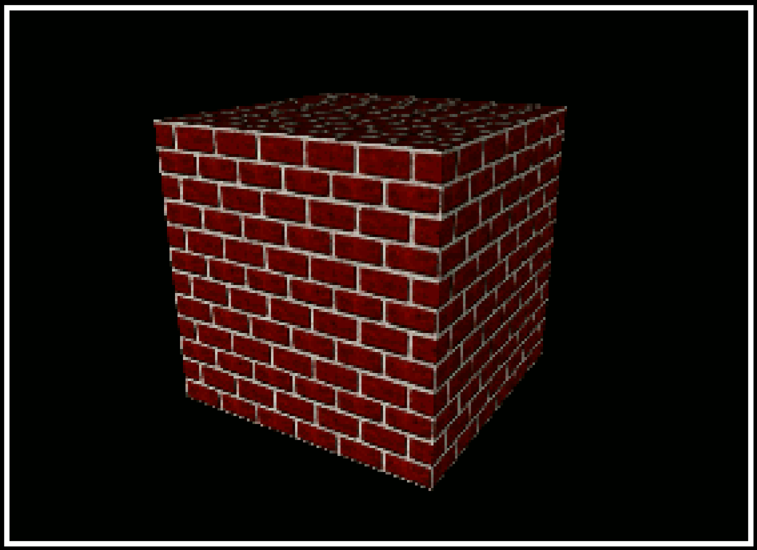


[pizza.wrl]

Controlando como texturas são mapeadas

Um exemplo usando escalonamento

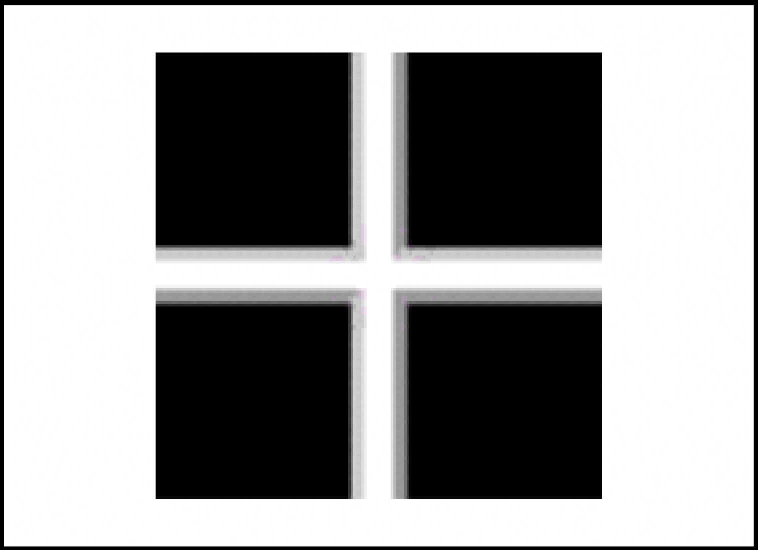


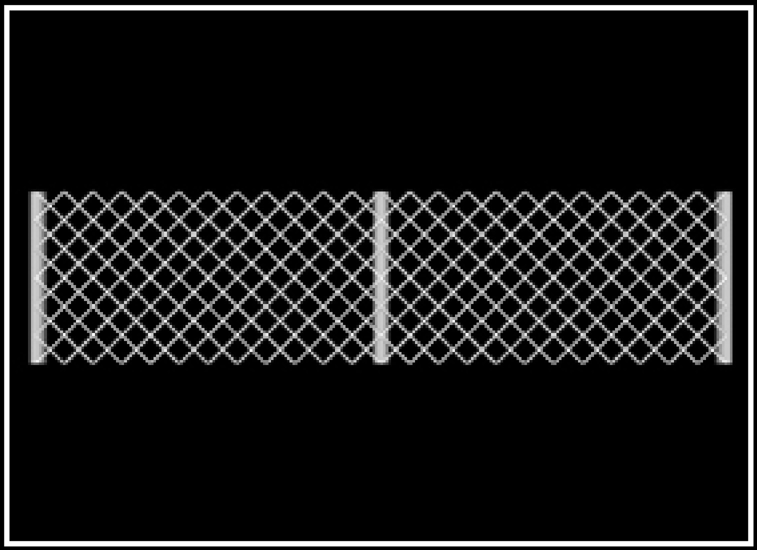


[brickb.wrl]

Controlando como as texturas são mapeadas

Um exemplo usando escalonamento e rotação





[fence.wrl]

Controlando como texturas são mapeadas

Resumo

. Textura de imagens estão em um sistema de coordenadas de textura
. Coordenadas de textura e índices descrevem um pedaço de textura de uma forma
. Transformações de texturas transladam, rotacionam e alteram a escala das coordenadas de textura
. Use um ou ambos ajustar uma textura para a geometria e aparência desejada

```
#VRML V2.0 utf8
#
#  Pizza texture mapping, with and without texture coordinates
#
Transform {
  rotation 1 0 0 1.57   # Rotate pizzas to be initially visible
  children [
    Transform {
      translation 3 0 0
      children [
        Shape {
              appearance DEF PizzaApp Appearance {
                    material Material {
                    }
                    texture ImageTexture {
                          url "pizza.jpg"
                    }
              }
              geometry IndexedFaceSet {
                    coord DEF PizzaCoords Coordinate {
                          point [
                                # Slice, pulled out of pizza
                                0.5  0.0  0.5,    0.88 0.0  1.42,
                                1.06 0.0  1.33,   1.21 0.0  1.21,
                                1.33 0.0  1.06,   1.42 0.0  0.88,
                                # Rest of pizza
                                0.00 0.0  0.00,   0.92 0.0 0.38,
                                0.98 0.0  0.20,   1.00 0.0 0.00,
                                0.98 0.0 -0.20,   0.92 0.0 -0.38,
                                0.83 0.0 -0.56,   0.71 0.0 -0.71,
                                0.56 0.0 -0.83,   0.38 0.0 -0.92,
                                0.20 0.0 -0.98,   0.00 0.0 -1.00,
                                -0.20 0.0 -0.98,  -0.38 0.0 -0.92,
                                -0.56 0.0 -0.83,  -0.71 0.0 -0.71,
                                -0.83 0.0 -0.56,  -0.92 0.0 -0.38,
                                -0.98 0.0 -0.20,  -1.00 0.0 0.00,
                                -0.98 0.0  0.20,  -0.92 0.0 0.38,
                                -0.83 0.0  0.56,  -0.71 0.0 0.71,
                                -0.56 0.0  0.83,  -0.38 0.0 0.92,
                                -0.20 0.0  0.98,   0.00 0.0 1.00,
                                0.20 0.0  0.98,   0.38 0.0 0.92
                          ]
                    }
                    coordIndex [
                          # Slice
                          0,  1,  2,  3,  4,  5, -1,
                          # Rest of pizza
                          6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                          17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
                          28, 29, 30, 31, 32, 33, 34, 35
                    ]
                    texCoord TextureCoordinate {
                          point [
                                # Center point of pizza image
                                0.5  0.5,
                                # Pizza perimeter
                                0.96 0.41,  0.97 0.50,
                                0.96 0.59,  0.93 0.68,
                                0.89 0.76,  0.83 0.83,
                                0.76 0.89,  0.68 0.93,
                                0.59 0.96,  0.50 0.97,
```

```
                                0.41 0.96,  0.32 0.93,
                                0.24 0.89,  0.17 0.83,
                                0.11 0.76,  0.07 0.68,
                                0.04 0.59,  0.03 0.50,
                                0.04 0.41,  0.07 0.32,
                                0.11 0.24,  0.17 0.17,
                                0.24 0.11,  0.32 0.07,
                                0.41 0.04,  0.50 0.03,
                                0.59 0.04,
                                # Slice perimeter
                                0.68 0.07,  0.76 0.11,
                                0.83 0.17,  0.89 0.24,
                                0.93 0.32
                          ]
                    }
                    texCoordIndex [
                          # Slice
                          0, 28, 29, 30, 31, 32, -1,
                          # Rest of pizza
                          0,  32,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10,
              11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
                          23, 24, 25, 26, 27, 28
                    ]
                    ccw    TRUE
                    solid  FALSE
                    convex FALSE
              }
        }   # end of Shape node
      ]
    }
    Transform {
      translation -3 0 0
      children [
        Shape {
              appearance USE PizzaApp
              geometry IndexedFaceSet {
                    coord USE PizzaCoords
                    coordIndex [
                          # Slice
                          0,  1,  2,  3,  4,  5, -1,
                          # Rest of pizza
                          6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                          17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
                          28, 29, 30, 31, 32, 33, 34, 35
                    ]
            texCoord NULL
                    ccw    TRUE
                    solid  FALSE
                    convex FALSE
          }
        }
      ]
    }
  ]
}
```

```
#VRML V2.0 utf8

# Texture without textureTransform
Transform {
  translation -1.5 0 0
  children [
    Shape {
      appearance Appearance {
        material Material { }
        texture DEF BrickTexture ImageTexture {
            url "brick.jpg"
        }
      }
      geometry Box { }
    }
  ]
}

# Texture scaled with textureTransform
Transform {
  translation 1.5 0 0
  children [
    Shape {
      appearance Appearance {
        material Material { }
        texture  USE BrickTexture
        textureTransform TextureTransform {
            scale 3.0 3.0
        }
      }
      geometry Box { }
    }
  ]
}
```

```
#VRML V2.0 utf8

# Chain-link fence
Shape {
    appearance Appearance {
        material Material { }
        texture ImageTexture { url "grill.png" }
        textureTransform TextureTransform {
            rotation .7853981750
            scale 8.5 8.5
            center 0.5 0.5
        }
    }
    geometry IndexedFaceSet {
        solid FALSE
        coord Coordinate {
            point [
                -4.0 -1.0 0.1,  -2.0 -1.0 0.1,
                -2.0  1.0 0.1,  -4.0  1.0 0.1,

                -2.0 -1.0 0.1,   0.0 -1.0 0.1,
                 0.0  1.0 0.1,  -2.0  1.0 0.1,

                 0.0 -1.0 0.1,   2.0 -1.0 0.1,
                 2.0  1.0 0.1,   0.0  1.0 0.1,

                 2.0 -1.0 0.1,   4.0 -1.0 0.1,
                 4.0  1.0 0.1,   2.0  1.0 0.1,
            ]
        }
        coordIndex [
             0,  1,  2,  3, -1
             4,  5,  6,  7, -1,
             8,  9, 10, 11, -1,
            12, 13, 14, 15,
        ]
        texCoord TextureCoordinate {
            point [
                0.0 0.0,  1.0 0.0,
                1.0 1.0,  0.0 1.0,
            ]
        }
        texCoordIndex [
            0, 1, 2, 3, -1,
            0, 1, 2, 3, -1,
            0, 1, 2, 3, -1,
            0, 1, 2, 3,
        ]
    }
}

# Fence posts
DEF Post Shape {
    appearance Appearance {
        material Material { }
    }
    geometry Cylinder {
        height 2.0
        radius 0.1
    }
}
```

```
Transform { translation -4.0 0.0 0.0  children USE Post }
Transform { translation  4.0 0.0 0.0  children USE Post }
```

Iluminando seu mundo

Motivação

Exemplo

Usando tipos de luzes

Usando características de iluminação comum

Usando características de iluminação comum

Sintaxe: PointLight

Sintaxe: DirectionalLight

Sintaxe: SpotLight

Sintaxe: SpotLight

Exemplo

Resumo

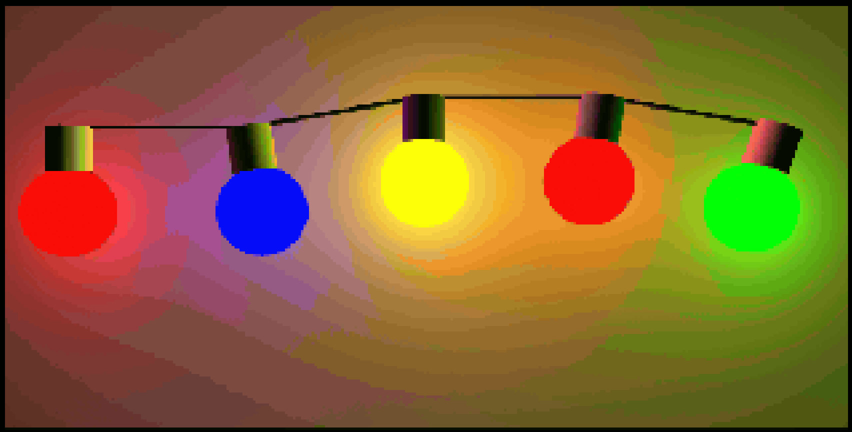
Iluminando seu mundo

Motivação

. **Por default, você tem uma luz na cena, colocada em sua cabeça**
. **Para maior realismo, você pode adicionar várias luzes**
  - **Luz solar, lâmpadas elétricas, velas**
  - **Lanternas, refletor, luz de fogo**
. **Luzes podem ser posicionadas, orientadas e colorizadas**
. **Luzes não podem espalhar sombras**

Iluminando seu mundo

Exemplo



Iluminando seu mundo

Usando tipos de luzes

. **Existem três tipos de luzes VRML**

- ***Point lights*** **- irradiam em todas as direções a partir de um ponto**
- ***Directional lights*** **- iluminam em uma única direção, até o infinito**
- ***Spot lights*** **- iluminam em uma direção a partir de um ponto, radiando em um cone**

Iluminando seu mundo

Usando características de iluminação comum

. Todas as luzes possuem vários campos em comum:

- **`on` - liga e desliga**
- **`intensity` - controla o brilho ou intensidade**
- **`ambientIntensity` - controla o efeito do ambiente**
- **`color` - seleciona a cor**

Iluminando seu mundo

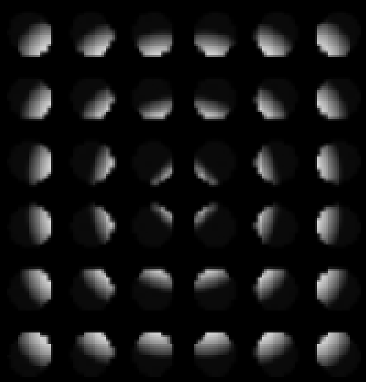
Usando características de iluminação comuns

- **Luzes de ponto e refletores de luzes também têm:**
  - **`location` - posição**
  - **`radius` - máxima distância de iluminação**
  - **`attenuation` - atenua com a distância**
- **Luzes direcionadas e refletores também têm:**
  - **`direction` - direção**

Iluminando seu mundo

Sintaxe: PointLight

. Um nó `PointLight` ilumina radialmente de um ponto

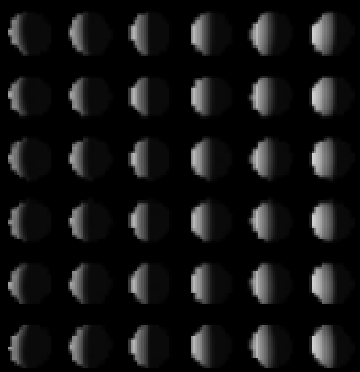


```
PointLight {
    location 0.0 0.0 0.0
    intensity 1.0
    color 1.0 1.0 1.0
}
```

Iluminando seu mundo

Sintaxe: DirectionalLight

. Um nó `DirectionalLight` ilumina em uma direção até o infinito

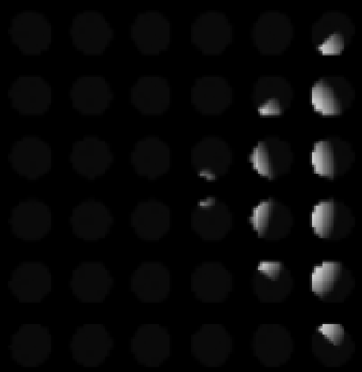


```
DirectionalLight {
    direction 1.0 0.0 0.0
    intensity 1.0
    color 1.0 1.0 1.0
}
```

Iluminando seu mundo

Sintaxe: SpotLight

. Um nó `SpotLight` ilumina de um ponto, em uma direção, como um cone



```
SpotLight {
    location  0.0 0.0 0.0
    direction 1.0 0.0 0.0
    intensity 1.0
    color 1.0 1.0 1.0
}
```

Iluminando seu mundo

Sintaxe: SpotLight

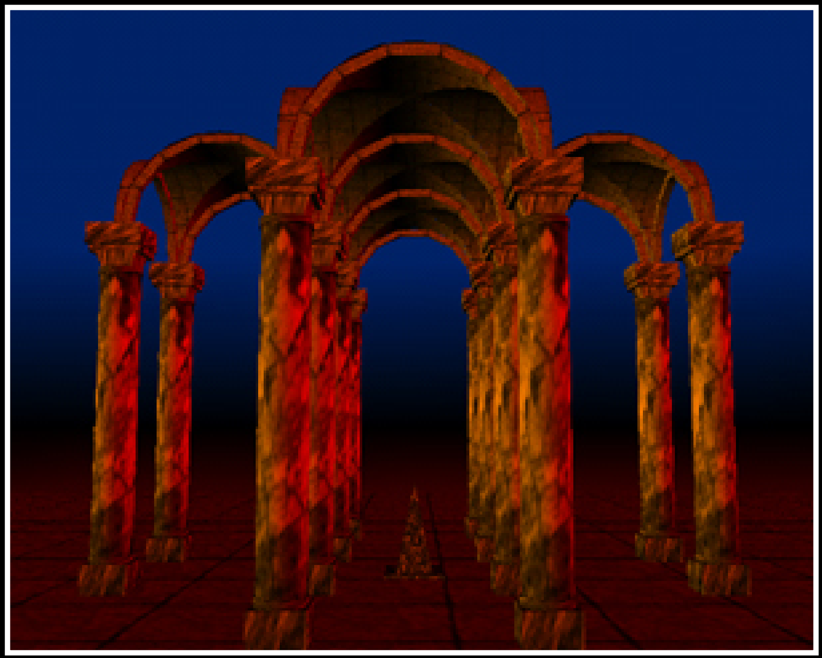
. A largura máxima de um refletor de cone de luz é controlado pelo campo `cutOffAngle`

. Uma região interior de um cone com uma constante de reflexão é controlada pelo campo `beamWidth`

```
SpotLight {
    . . .
    cutOffAngle 0.785
    beamWidth   1.571
}
```

Iluminando seu mundo

Exemplo



[temple.wrl]

Iluminando seu mundo

Resumo

. Existem três tipos de luzes: ponto, direcional, and refletor
. Todas as luzes têm on/off, intensidade, efeitos ambiente, e cor
. Pontos e refletores de luzes possuem uma localização, raio e atenuação
. Luzes direcionais e refletores têm uma direção

Adicionando fundos

Motivação

Usando os componentes de fundo

Usando os componentes de fundo

Sintaxe: Background

Um exemplo de fundo

Sintaxe: Background

Um exemplo de imagem de fundo

Um exemplo de fundo

Resumo

Adicionando fundos

Motivação

. Formas formam a parte mais próxima da cena

. Você pode adicionar um fundo para fornecer um contexto a cena

. Fundos descrevem:

- **Cores do céu e da terra**
- **Imagens panorâmicas de montanhas, cidades, etc.**

. Fundos são rápidos de desenhar, se você usar formas para construí-los

Sumário Índice do Capítulo Slide Anterior Slide Seguinte 281

Adicionando fundos

Usandos os componentes de fundo

- **Um fundo cria três formas especiais:**
  - **Uma *esfera celeste***
  - **Uma *esfera terrestre* dentro da esfera celeste**
  - **Uma *caixa panorâmica* dentro da esfera terrestre**
- **As esferas celeste e terrestre são escurecidas com um gradiente de cor**
- **A caixa panorâmica é uma textura mapeada com seis imagens**

Adicionando fundos

Usando os componentes de fundo

. Partes transparentes da esfera terrestre revelam a esfera celeste
. Partes transparentes da caixa panorâmica revelam as esferas terrestre e celeste
. O observador pode levantar, baixar, e ir de um lado para outro para ver diferentes partes do fundo
. O observador não pode obter uma vista do fundo

Adicionando fundos

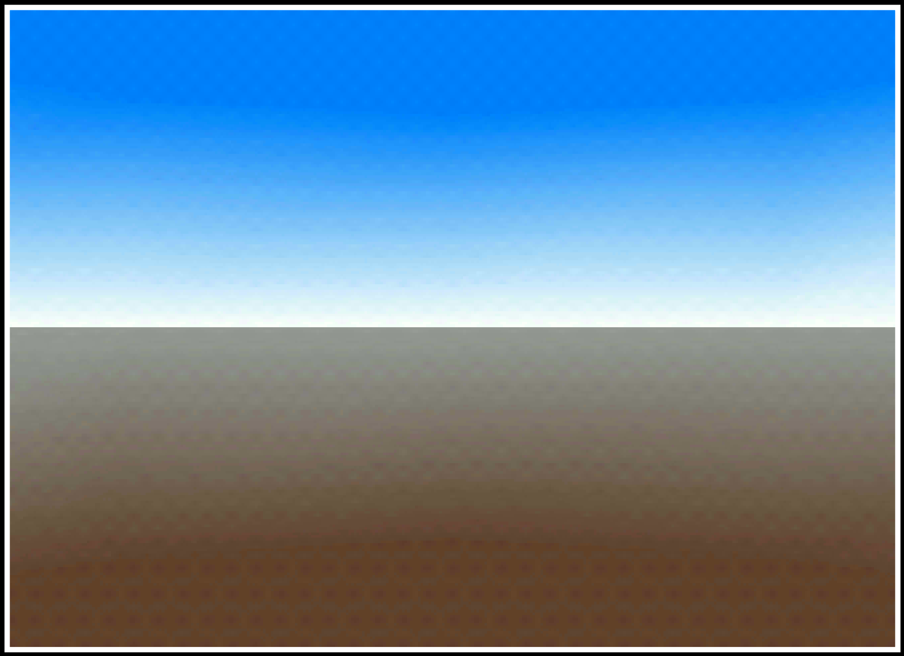
Sintaxe: Background

- **Um nó `Background` descreve cores de fundo**
  - ***ground colors* e *angles* - graduação da terra**
  - ***sky colors* e *angles* - graduação do céu**
  - **mais . . .**

```
Background {
    groundColor [ 0.0 0.2 0.7, . . . ]
    groundAngle [ 1.309, 1.571 ]
    skyColor    [ 0.1 0.1 0.0, . . . ]
    skyAngle    [ 1.309, 1.571 ]
}
```

Adicionando fundos

Um exemplo de fundo



[back.wrl]

Adicionando fundos

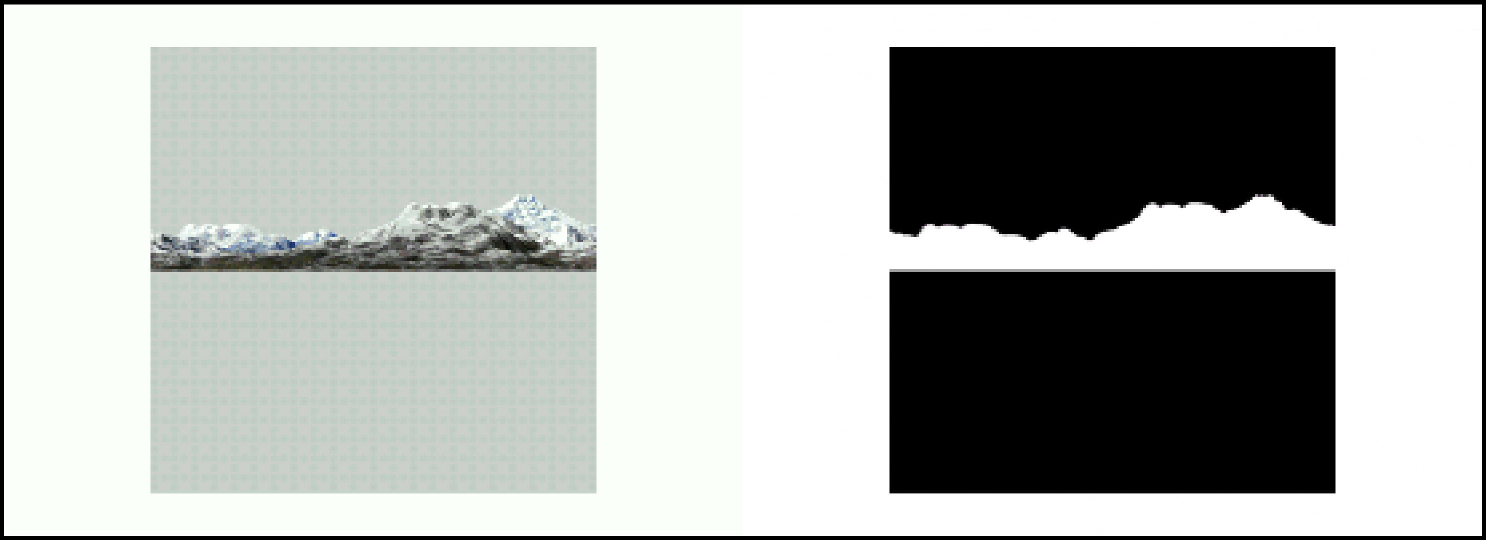
Sintaxe: Background

- Um nó **Background** descreve imagens
  - *frontUrl* - URL da textura de imagem para a frente da caixa
  - etc .

```
Background {
    . .
    frontUrl  "mountns.png"
    backUrl   "mountns.png"
    leftUrl   "mountns.png"
    rightUrl  "mountns.png"
    topUrl    "clouds.png"
    bottomUrl "ground.png"
}
```

Adicionando fundos

Um exemplo de imagem de fundo



Adicionando fundos

Um exemplo de fundo

[back2.wrl]

Adicionando fundos

Resumo

- **Fundos descrevem:**
  - **Gradiente de cores do céu e da terra nas esferas celeste e terrestre**
  - **Imagens panorâmicas em uma caixa panorâmica**
- **O observador pode observar em toda as direções, mas nunca obter um close do fundo**

```
#VRML V2.0 utf8
Background {
    skyColor [
        0.0 0.2 0.7,
        0.0 0.5 1.0,
        1.0 1.0 1.0
    ]
    skyAngle [ 1.309, 1.571 ]
    groundColor [
        0.1 0.10 0.0,
        0.4 0.25 0.2,
        0.6 0.60 0.6,
    ]
    groundAngle [ 1.309, 1.571 ]
}
```

```
#VRML V2.0 utf8
Background {
    skyColor [
        0.0 0.2 0.7,
        0.0 0.5 1.0,
        1.0 1.0 1.0
    ]
    skyAngle [ 1.309, 1.571 ]
    groundColor [
        0.1 0.10 0.0,
        0.4 0.25 0.2,
        0.6 0.60 0.6,
    ]
    groundAngle [ 1.309, 1.571 ]
    frontUrl "mountns.png"
    backUrl  "mountns.png"
    leftUrl  "mountns.png"
    rightUrl "mountns.png"
}
```

Adicionando nevoeiro

Motivação

Exemplos

Usando os controles de visibilidade de nevoeiro

Selecionando uma cor de nevoeiro

Sintaxe: Fog

Vários exemplos de nevoeiro

Resumo

Adicionando nevoeiro

Motivação

. Nevoeiro aumenta o realismo:

○ Adicionar nevoeiro externo para criar mundos confusos

○ Adicionar nevoeiro internamente para criar calabouços

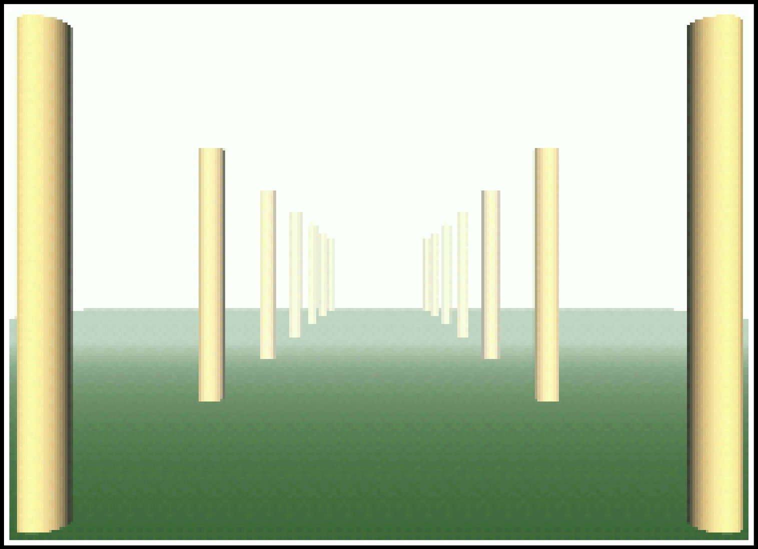
○ Use nevoeiro para ativar um capricho

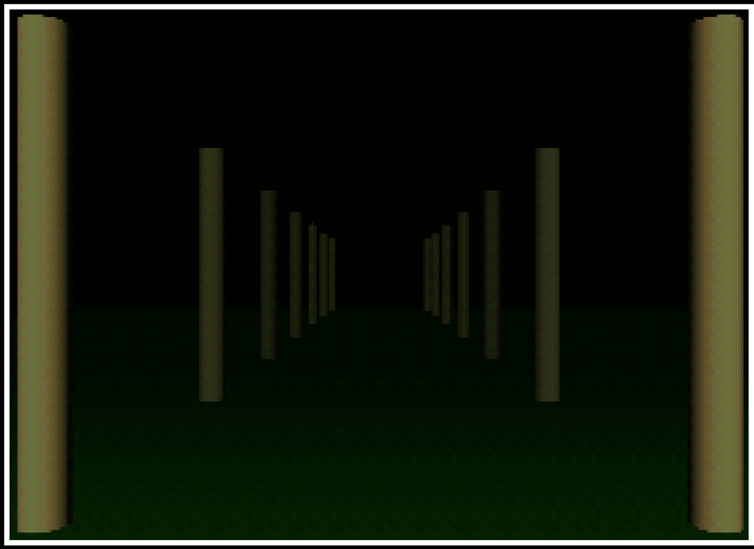
. Quanto mais distante o observador puder ver, mais dificuldade para modelar e desenhar

. Para reduzir o tempo de desenvolvimento e de desenho, limitar a observação usando fog

Adicionando nevoeiro

Exemplos





[fog2.wrl] [fog4.wrl]

Adicionando nevoeiro

Usando os controles de visibilidade do nevoeiro

. O *tipo de nevoeiro* seleciona o fator de redução de visibilidade com a distância para linear ou exponencial

- **Linear é mais fácil de controlar**
- **Exponential é mais realístico e pesado**

. O *range de visibilidade* seleciona a distância em que o nevoeiro atinge o máximo de densidade

- **Nevoeiro é "vazio" no observador, e gradualmente se reduz a visibilidade**

Adicionando nevoeiro

Selecionando a cor do nevoeiro

. O nevoeiro possui uma *cor de nevoeiro*

○ Branco é a típica, mas preta, vermelha, etc. também são possíveis

. *Formas* são are desaparecem na cor do neoveiro com a distância

. O fundo não é afetado

○ Para melhor efeito, faça o fundo com a cor do nevoeiro

Adicionando nevoeiro

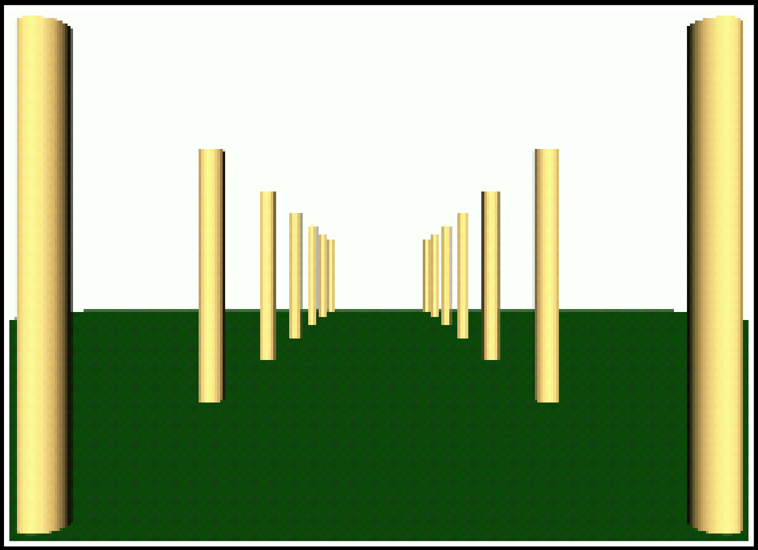
Sintaxe: Fog

- Um nó `Fog` cria nevoeiros colorizados
  - *color* - cor do nevoeiro
  - *type* - tipo do nevoeiro
  - *visibility range* - limite máximo de visibilidade

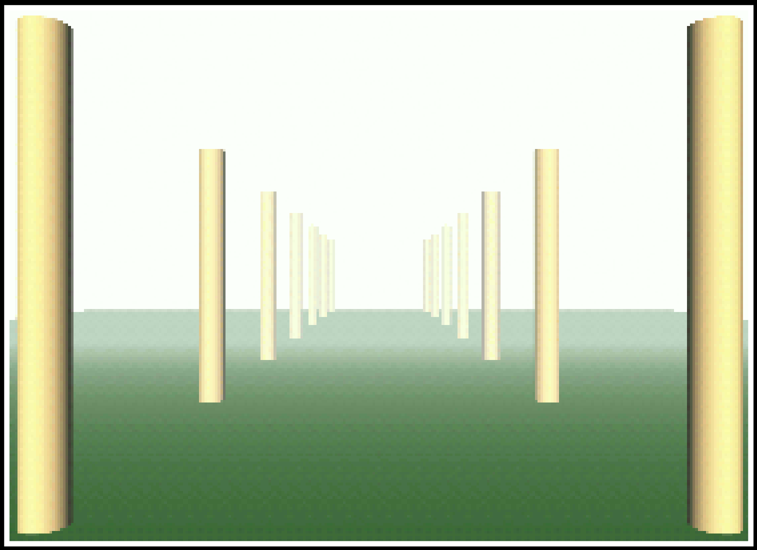
```
Fog {
    color 1.0 1.0 1.0
    fogType "LINEAR"
    visibilityRange 0.0
}
```

Adicionando nevoeiro

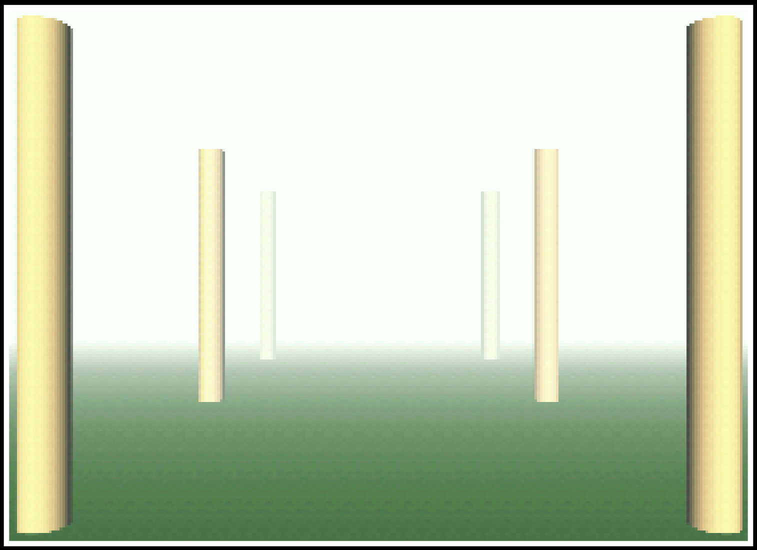
Vários exemplos de nevoeiro



[fog1.wrl]



[fog2.wrl]



[fog3.wrl]

Adicionando nevoeiro

Resumo

. Nevoeiro possui uma cor, um tipo e um lacance de visibilidade

. Nevoeiro pode ser usado para capricho, dentro de ambientes

. Nevoeiros limitam a visão do observador:

- **Reduzem o tamanho do mundo que foi construído**
- **Reduzem o tamanho do mundo que deve ser desenhado**

```
#VRML V2.0 utf8
Fog {
    color 1.0 1.0 1.0
    fogType "LINEAR"
    visibilityRange 30.0
}
Background { skyColor 1.0 1.0 1.0 }
Inline { url "fogworld.wrl" }
```

```
#VRML V2.0 utf8
Fog {
    color 0.0 0.0 0.0
    fogType "EXPONENTIAL"
    visibilityRange 30.0
}
Background { skyColor 0.0 0.0 0.0 }
Inline { url "fogworld.wrl" }
```

```
#VRML V2.0 utf8
# No fog
Background { skyColor 1.0 1.0 1.0 }
Inline { url "fogworld.wrl" }
```

```
#VRML V2.0 utf8
Fog {
    color 1.0 1.0 1.0
    fogType "EXPONENTIAL"
    visibilityRange 40.0
}
Background { skyColor 1.0 1.0 1.0 }
Inline { url "fogworld.wrl" }
```

Adicionando sons

Motivação

Criando sons

Sintaxe: AudioClip

Sintaxe: MovieTexture

Selecionando tipos de fontes de sonm

Sintaxe: Sound

Sintaxe: Sound

Sintaxe: Sound

Ativando o range do som

Criando sons ativados

Um exemplo usando som ativado

Um exemplo usando som ativado

Criando sons localizados contínuos

Criando sons de fundo contínuos

Um exemplo usando som localizado contínuo

Um exemplo usando som localizado contínuo

Resumo

Adicionando som

Motivação

- **Sons podem ser ativados pelas ações do observador**
  - **Cliques, buzinas, barulho de trinco de portas**
- **Sons podem ser contínuos no fundo**
  - **Vento, barulhos de multidão, música de elevador**
- **Sons são emitidos de uma localização, em uma direção, dentro de uma área**

Adicionando som

Criando sons

- **Sons tem dois componentes**
  - **Uma *fonte de som* fornecendo um sinal de som**
- **Como um componente stereo**
  - **Um *emissor de som* converte um sinal para um som virtual**
- **Como um alto-falante stereo**

Adicionando som

Sintaxe: AudioClip

- Um nó `AudioClip` cria uma fonte de som digital
  - *url* - uma URL de arquivo de som
  - *pitch* - velocidade de reprodução
  - Controles de reprodução, como um nó `TimeSensor`

```
AudioClip {
    url "myfile.wav"
    pitch 1.0
    startTime 0.0
    stopTime  0.0
    loop FALSE
}
```

Adicionando som

Sintaxe: MovieTexture

. Um nó `MovieTexture` cria uma fonte de som de filme

- ***url* - uma URL de arquivo de filme de textura**
- ***speed* - velocidade de reprodução**
- **Controles de reprodução, como um nó `TimeSensor`**

```
MovieTexture {
    startTime 0.0
    stopTime  0.0
    loop FALSE
    speed 1.0
    url "movie.mpg"
}
```

Adicionando som

Selecinando tipos de fonte de som

- **Suportado pelo nó `AudioClip`:**
  - ***WAV* - arquivos de sons digitais**
    - **Bom para efeitos de sons**
  - ***MIDI* - arquivos musicais MIDI**
    - **Arquivos MIDI são bons para músicas de fundo**
- **Suportado pelo nó `MovieTexture`:**
  - ***MPEG* - arquivo de filme com sons**
    - **Bom para TVs virtuais**

Adicionando som

Sintaxe: Sound

- **Um nó `Sound` descreve um som emitido**
  - ***source* - nó `AudioClip` ou `MovieTexture`**
  - ***location* e *direction* - lugar de emissão**
  - **mais . . .**

```
Sound {
    source AudioClip { . . . }
    location  0.0 0.0 0.0
    direction 0.0 0.0 1.0
}
```

Adicionando som

Sintaxe: Sound

- **Um nó Sound descreve um som emitido**
  - ***intensity* - volume**
  - ***spatialize* - usa o processamento spatialize**
  - ***priority* - priorizar o som**
  - **mais . . .**

```
Sound {
    . . .
    intensity 1.0
    spatialize TRUE
    priority 0.0
}
```

Adicionando som

Sintaxe: Sound

- **Um nó Sound descreve um som emitido**
  - ***minimum* e *maximum* range - área na qual o som pode ser ouvido**

```
Sound {
    . . .
    minFront 1.0
    minBack  1.0
    maxFront 10.0
    maxBack  10.0
}
```

Adicionando som

Ajustando o range do som

. Os campos do range do som especificam duas *elipsóides*

- **`minFront` e `minFront` controlam uma elipsóide interna**
- **`maxFront` e `maxFront` controlam uma elipsóide externa**

. O som possui um volume constante dentro da elipsóide interna

. O som desce para o volume zero da elipsóide enterna para a externa

Adicionando som

Criando sons ativados

- **Nó `AudioClip`:**
  - **`loop FALSE`**
  - **Ajusta `startTime` de um nó com sensor**
- **Nó `Sound`:**
  - **`spatialize TRUE`**
  - **`minFront` etc. com valores pequenos**
  - **`priority 1.0`**

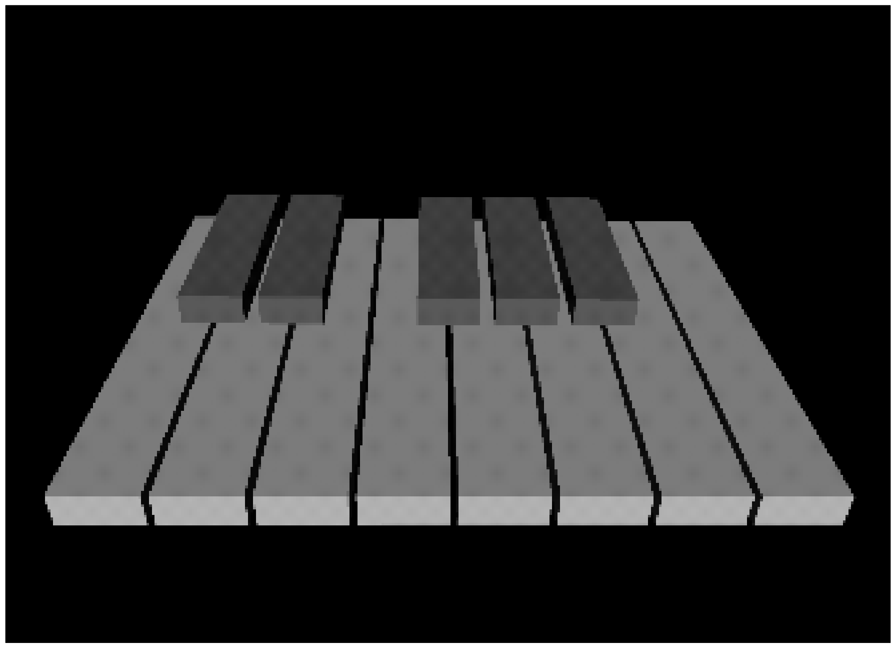
Adicionando som

Um exemplo usando som ativado

```
Sound {
    source DEF C4 AudioClip {
        url "tone1.wav"
        pitch 1.0
    }
}
ROUTE Touch.touchTime
   TO C4.set_startTime
```

Adicionando som

Um exemplo usando som ativado



[kbd.wrl]

Adicionando som

Criando sons localizados contínuos

- **Nó `AudioClip`:**
  - `loop TRUE`
  - `startTime 0.0` **(default)**
  - `stopTime 0.0` **(default)**
- **Nó `Sound`:**
  - `spatialize TRUE` **(default)**
  - `minFront` **etc. com valores médios**
  - `priority 0.0` **(default)**

Adicionando som

Criando sons de fundo contínuos

- **Nó `AudioClip`:**
  - `loop TRUE`
  - `startTime 0.0` **(default)**
  - `stopTime 0.0` **(default)**
- **Nó `Sound` :**
  - `spatialize FALSE` **(default)**
  - `minFront` **etc. com valores grandes**
  - `priority 0.0` **(default)**

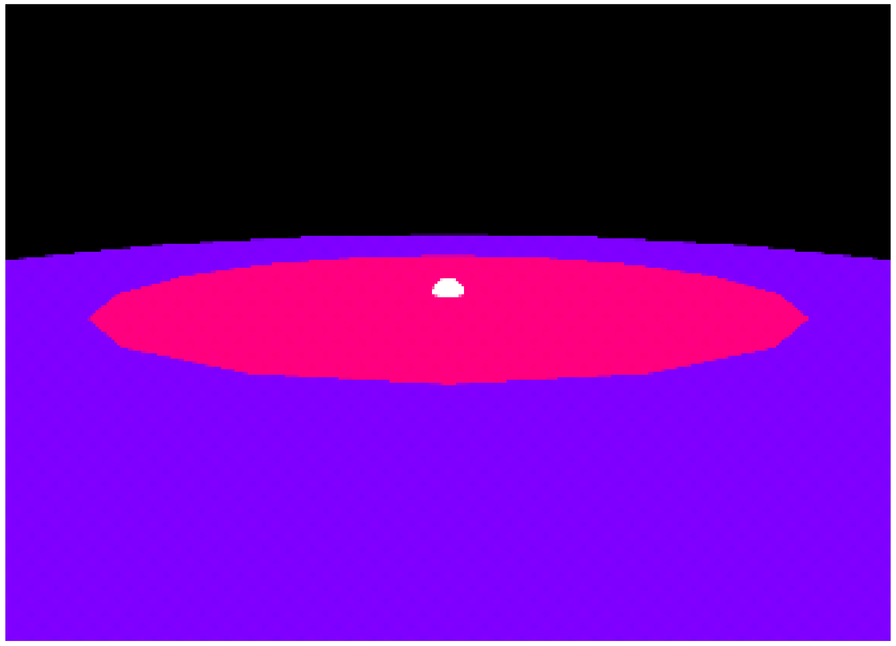
Adicionando som

Um exemplo usando som localizado contínuo

```
Sound {
    source AudioClip {
        url "willow1.wav"
        loop TRUE
    }
}
```

Adicionando som

Um exemplo usando som localizado contínuo



[ambient.wrl]

Adicionando som

Resumo

- **Um nó `AudioClip` ou um nó `MovieTexture` descrevem uma fonte de som**
  - **Uma URL fornece um arquio de som**
  - **Looping, start time, e stop time controlam a reprodução**
- **Um nó `Sound` descreve um som emitido**
  - **Um nó fonte fornece o som**
  - **Campos limites descrevem o volume do som**

```
#VRML V2.0 utf8
#
#  kbd.wrl
#  Synthesizer keyboard
#       by David R. Nadeau
#
#  This world creates a 'playable' synthesizer keyboard.  Each key on
#  the keyboard has a sound and touch sensor associated with it.
#  Touching a key triggers the touch sensor, which triggers the sound.
#
#  Things to experiment with...
#       Use a PROTO to create a piano key that plays a note of a given
#       pitch, supplied as a PROTO argument.  Make the sound file an
#       argument as well.
#

# Middle C (C4)
Transform {
    children [
        DEF WhiteKey Shape {
            appearance Appearance {
                material Material { }
            }
            geometry Box { size 0.23 0.1 1.5 }
        }
        DEF C4 TouchSensor { }
        Sound {
            source DEF PitchC4 AudioClip {
                url "tone1.wav"
                pitch 1.0
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# C# above middle C (Cs4)
Transform { translation 0.125 0.1 -0.375
    children [
        DEF BlackKey Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.4 0.4 0.4
                }
            }
            geometry Box { size 0.2 0.1 0.75 }
        }
        DEF Cs4 TouchSensor { }
        Sound {
            source DEF PitchCs4 AudioClip {
                url "tone1.wav"
                pitch 1.059
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# D above middle C (D4)
Transform { translation 0.25 0.0 0.0
    children [
```

```
        USE WhiteKey,
        DEF D4 TouchSensor { }
        Sound {
            source DEF PitchD4 AudioClip {
                url "tone1.wav"
                pitch 1.122
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# D# above middle C (Ds4)
Transform { translation 0.375 0.1 -0.375
    children [
        USE BlackKey,
        DEF Ds4 TouchSensor { }
        Sound {
            source DEF PitchDs4 AudioClip {
                url "tone1.wav"
                pitch 1.189
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# E above middle C (E4)
Transform { translation 0.5 0.0 0.0
    children [
        USE WhiteKey,
        DEF E4 TouchSensor { }
        Sound {
            source DEF PitchE4 AudioClip {
                url "tone1.wav"
                pitch 1.260
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# F above middle C (F4)
Transform { translation 0.75 0.0 0.0
    children [
        USE WhiteKey,
        DEF F4 TouchSensor { }
        Sound {
            source DEF PitchF4 AudioClip {
                url "tone1.wav"
                pitch 1.335
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# F# above middle C (Fs4)
Transform { translation 0.875 0.1 -0.375
    children [
        USE BlackKey,
```

```
        DEF Fs4 TouchSensor { }
        Sound {
            source DEF PitchFs4 AudioClip {
                url "tone1.wav"
                pitch 1.414
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# G above middle C (G4)
Transform { translation 1.0 0.0 0.0
    children [
        USE WhiteKey,
        DEF G4 TouchSensor { }
        Sound {
            source DEF PitchG4 AudioClip {
                url "tone1.wav"
                pitch 1.498
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# G# above middle C (Gs4)
Transform { translation 1.125 0.1 -0.375
    children [
        USE BlackKey,
        DEF Gs4 TouchSensor { }
        Sound {
            source DEF PitchGs4 AudioClip {
                url "tone1.wav"
                pitch 1.587
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# A above middle C (A5)
Transform { translation 1.25 0.0 0.0
    children [
        USE WhiteKey,
        DEF A5 TouchSensor { }
        Sound {
            source DEF PitchA5 AudioClip {
                url "tone1.wav"
                pitch 1.682
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# A# above middle C (As5)
Transform { translation 1.375 0.1 -0.375
    children [
        USE BlackKey,
```

```
        DEF As5 TouchSensor { }
        Sound {
            source DEF PitchAs5 AudioClip {
                url "tone1.wav"
                pitch 1.782
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# B above middle C (B5)
Transform { translation 1.5 0.0 0.0
    children [
        USE WhiteKey,
        DEF B5 TouchSensor { }
        Sound {
            source DEF PitchB5 AudioClip {
                url "tone1.wav"
                pitch 1.888
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
# C above middle C (C5)
Transform { translation 1.75 0.0 0.0
    children [
        USE WhiteKey,
        DEF C5 TouchSensor { }
        Sound {
            source DEF PitchC5 AudioClip {
                url "tone1.wav"
                pitch 2.0
            }
            maxFront 100.0
            maxBack 100.0
        }
    ]
}
ROUTE C4.touchTime  TO PitchC4.set_startTime
ROUTE Cs4.touchTime TO PitchCs4.set_startTime
ROUTE D4.touchTime  TO PitchD4.set_startTime
ROUTE Ds4.touchTime TO PitchDs4.set_startTime
ROUTE E4.touchTime  TO PitchE4.set_startTime
ROUTE F4.touchTime  TO PitchF4.set_startTime
ROUTE Fs4.touchTime TO PitchFs4.set_startTime
ROUTE G4.touchTime  TO PitchG4.set_startTime
ROUTE Gs4.touchTime TO PitchGs4.set_startTime
ROUTE A5.touchTime  TO PitchA5.set_startTime
ROUTE As5.touchTime TO PitchAs5.set_startTime
ROUTE B5.touchTime  TO PitchB5.set_startTime
ROUTE C5.touchTime  TO PitchC5.set_startTime
```

```
#VRML V2.0 utf8
# Sound emitter
Sound {
    source AudioClip {
        url "willow1.wav"
        loop TRUE
    }
    minFront 5.0
    minBack  5.0
    maxFront 10.0
    maxBack  10.0
}
Transform {
  translation 0 -2 0    #Move down so we can see everything
  children [
    # Sound emitter markers
    Inline { url "sndmark.wrl" }
  ]
}
```

Controlando o ponto de vista

Motivação

Criando pontos de vista

Sintaxe: Viewpoint

Resumo

Controlando o ponto de vista

Motivação

. Por default, o observador entra no mundo em (0.0, 0.0, 10.0)

. Voce pode fornecer o seu próprio ponto de observação

- **Selecione a posição do ponto de entrada**
- **Selecione vistas favoritas para o observador**
- **Nomeie as vistas para um menu**

Controlando o ponto de vista

Criando pontos de vista

. Pontos de vista especificam uma localização desejada, uma orientação e um campo de câmera de vista

. Pontos de vista podem ser transformados usando o nó `Transform`

. O primeiro ponto de vista encontrado em um arquivo é o ponto de entrada

Controlando o ponto de vista

Sintaxe: Viewpoint

. Um nó `Viewpoint` especifica uma localização de vista

- ***position* e *orientation* - localização da vista**
- ***fieldOfView* - ângulos das lentes da câmera**
- ***description* - descrição do menu do ponto de vista**

```
Viewpoint {
    position    0.0 0.0 10.0
    orientation 0.0 0.0 1.0 0.0
    fieldOfView 0.785
    description "Entry View"
}
```

Controlando o ponto de vista

Resumo

. Especifica o ponto de vista favorito no nó `Viewpoint`

. O primeiro ponto de vista no arquivo é o ponto de vista de entrada

Controlando a navegação

Motivação

Selecionando tipos de navegações

Especificando um avatar

Controlando a luz na cabeça

Sintaxe: NavigationInfo

Resumo

Controlando a nevegação

Motivação

- **Diferentes tipos de mundos requerem diferentes estilos de navegação**
    - **Caminhando através de um calabouço**
    - **Voando através de um mundo de sombras**
    - **Examinando formas em uma aplicação CAD**
- **Você pode selecionar o tipo de navegação**
- **Você pode descrever o tamanho e velocidade do *avatar* do observador**

Controlando a nevegação

Selecionando tipos de navegação

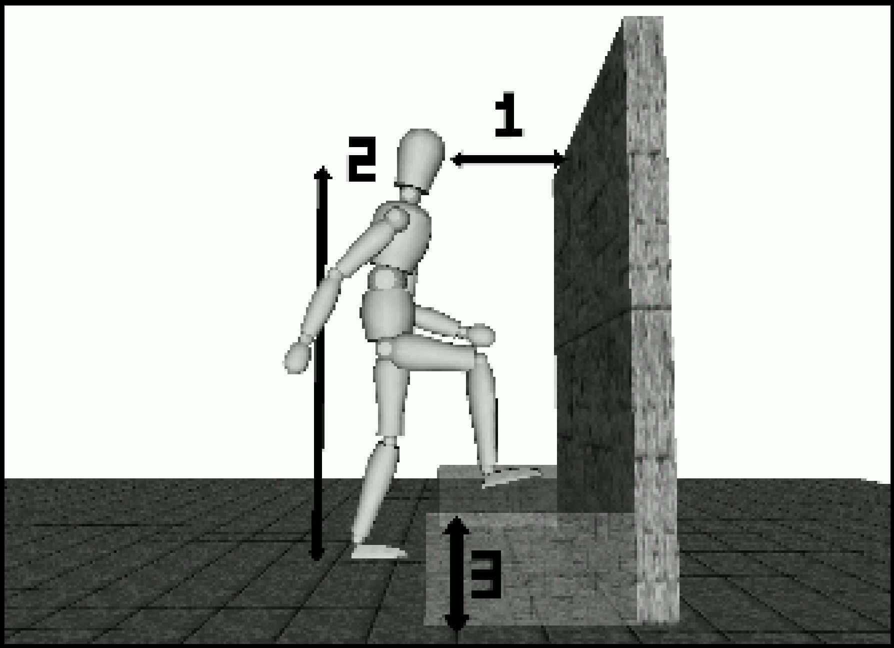
. Existem cinco tipos padrões de palavras chave em navegação:

- **`WALK` - andar, puxado para baixo pela gravidade**
- **`FLY` - voando, não é afetado pela gravidade**
- **`EXAMINE` - examinando um objeto como "comprimento dos braços"**
- **`NONE` - nenhuma navegação, movimento controlado por um observador fora do mundo!**
- **`ANY` - permite ao usuário mudar o tipo de navegação**

. Alguns browsers suportam tipos adicionais de navegação

Controlando a nevegação

Especificando um avatar



. O tamanho (comprimento, altura, altura do passo) e velocidade do avatar podem ser especificados

Controlando a nevegação

Controlando a luz da cabeça

- **Por default, um luz de cabeça é colocada na cabeça do avatar e é dirigida na direção da cabeça**
- **Você pode ligar e desligar esta luz**
  - **A maioria dos browsers fornecem um menu de opções para controla esta luz**
  - **Você também pode controlar esta luz com o nó `NavigationInfo`**

Controlando a nevegação

Sintaxe: NavigationInfo

- **Um nó `NavigationInfo` seleciona o tipo de navegação e as características do avatar**
  - ***type*** **- estilo de navegação**
  - ***avatarSize*** **e** ***speed*** **- características do avatar**
  - ***headlight*** **- liga e desliga a luz na cabeça**

```
NavigationInfo {
    type          [ "WALK", "ANY" ]
    avatarSize [ 0.25, 1.6, 0.75 ]
    speed         1.0
    headlight   TRUE
}
```

Controlando a nevegação

Resumo

. O tipo de navegação especifica como o observador pode se mover em um mundo

○ (andando, voando, examinando, ou nenhum destes)

. O tamanho completo do avatar e sua velocidade especificam as características do avatar do observador

Sentindo o observador

Motivação

Sentindo o observador

Usando sensores de visibilidade e proximidade

Sintaxe: ProximitySensor

Sintaxe: ProximitySensor

Sintaxe: VisibilitySensor

Um exemplo usando um sensor de proximidade

Detectando colisão do observador com uma forma

Criando grupos de colisão

Sintaxe: Collision

Um examplo do uso de um grupo de colisão

Otimizando a detecção de colisão

Usando múltiplos sensores

Resumo

Resumo

Resumo

Sentindo o observador

Motivação

. Sentir o observador permite que você ative animações

❍ quando uma região está visível ao observador

❍ quando o observador está dentro de uma região

❍ quando o observador colide com uma forma

. Os nós `LOD` e `Billboard` são sensores do observador de propósito especial com suas próprias respostas

Sentindo o observador

Sentindo o observador

. Existem três tipos de sensores do observador:

- **Um nó `VisibilitySensor` sente se o observador pode ver uma região**
- **Um nó `ProximitySensor` sente se o observador está dentro de uma região**
- **Um nó `Collision` sente se o observador colidiu com uma forma**

Sentindo o observador

Usando sensores de visibilidade e proximidade

- **Nós `VisibilitySensor` e `ProximitySensor` sentem uma região em forma de caixa**
  - **center - centro da região**
  - **size - dimensões da região**
- **Ambos os nós tem saída similar:**
  - **`enterTime` - envia o tempo em que se entra na região**
  - **`exitTime` - envia o tempo em que se sai da região**
  - **`isActive` - envia *true* na entrada e *false* na saída**

Sentindo o observador

Sintaxe: ProximitySensor

- **Um nó `ProximitySensor` sente se o observador entra ou deixa uma região**
  - ***center* e *size* - a localização e o tamanho da região**
  - ***enterTime* e *exitTime* - envia o tempo de entrada e saída**
  - ***isActive* - envia true/false na entrada/saída**
  - **mais . . .**

```
DEF DoorSense ProximitySensor {
    center 0.0 1.75 0.0
    size   6.0 3.5 8.0
}
ROUTE DoorSense.enterTime
   TO OpenSound.set_startTime
```

Sentindo o observador

Sintaxe: ProximitySensor

- **Um nó `ProximitySensor` sente o observador enquanto ele está dentro de uma região**
  - ***position* e *orientation* - envia a posição e orientação enquanto o observador está dentro da região**

```
DEF DoorSense ProximitySensor {
    . . .
}
ROUTE DoorSense.position_changed
   TO PetRobotFollower.set_translation
```

Sentindo o observador

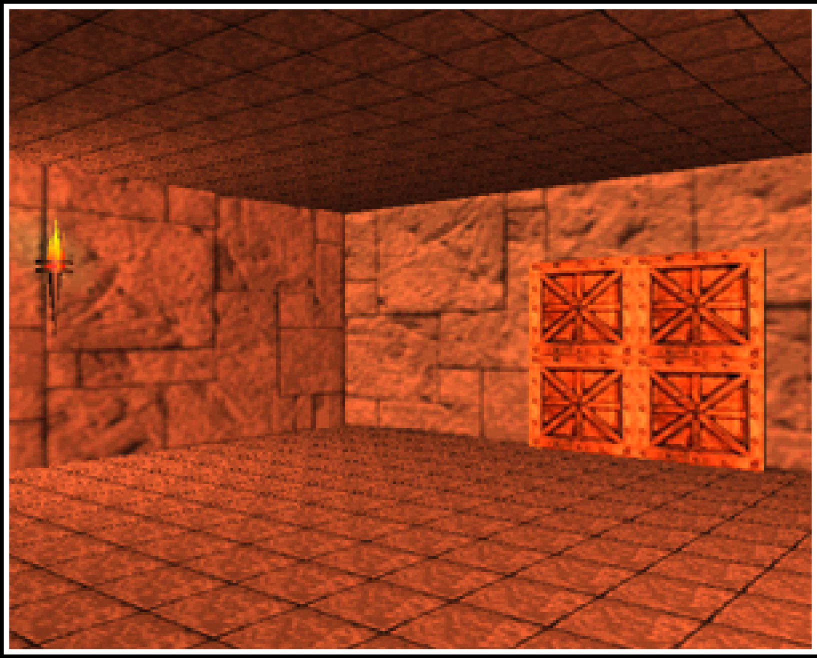
Sintaxe: VisibilitySensor

- Um nó `VisibilitySensor` sente se o observador pode ver uma região
  - *center* e *size* - localização e tamanho da região
  - *enterTime* e *exitTime* - envia o tempo de entrada/saída
  - *isActive* - envia true/false na entrada/saída

```
DEF DoorSense VisibilitySensor {
    center 0.0 1.75 0.0
    size 3.0 2.5 1.0
}
ROUTE DoorSense.enterTime
   TO OpenSound.set_startTime
```

Sentindo o observador

Um exemplo usando umsensor de proximidade



[prox1.wrl]

Sentindo o observador

Detectando colisão entre observador e um forma

- **Um nó de grupo `Collision` sente formasdentro do grupo**
  - **Detecta se o observador colidiu com alguma forma no grupo**
  - **Automaticamente para o observador para não atravesar a forma**
- **Colisão ocorre quando o avatar do observador consegue se aproximar muito de uma forma**
  - **A distância de colisão é controlada pelo tamanho do avatar no nó `NavigationInfo`**

Sentindo o observador

Criando grupos de colisão

- **A checagem de colisão é bastante pesada, checar colisão com formas aproximadas**
  - **Formas aproximadas são versões extremamente simplificadas das formas atuais**
  - **Formas aproximadas nunca são desenhadas**
- **Um grupo de colisão com um forma aproximada cria uma forma colidível invisível**
  - **Janelas e cercas invisíveis**
  - **Limites do mundo invisíveis**

Sentindo o observador

Sintaxe: Collision

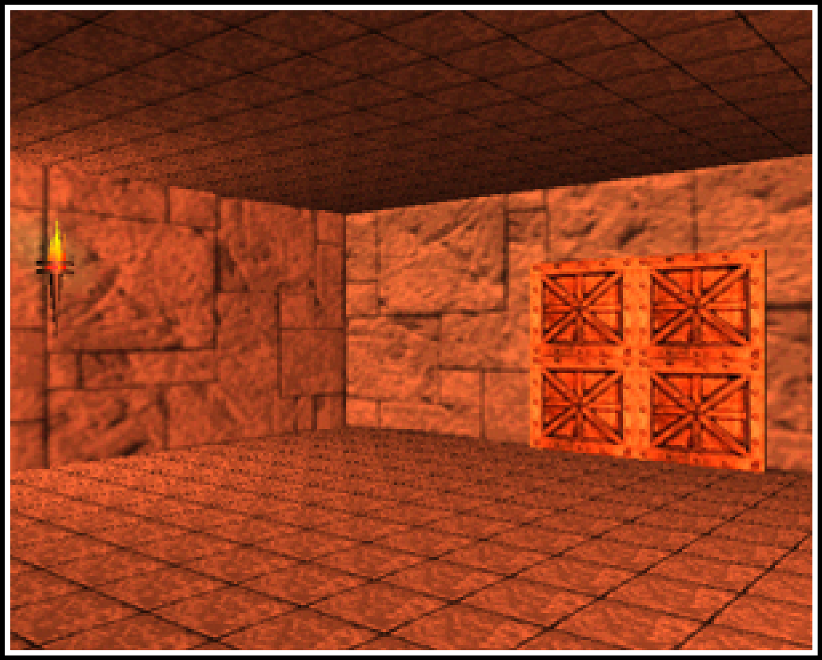
. Um nó de grupo `Collision` sente o observador colidir com um grupo de formas

- *collide* - habilita/desabilita o sensor
- *children* - filho sensível
- *proxy* - forma simples sensível ao invés do filho (children)

```
DEF DoorCollide Collision {
    proxy . . .
    children [ . . . ]
}
ROUTE DoorCollide.collideTime
   TO OpenSound.set_startTime
```

Sentindo o observador

Um exemplo do uso de um grupo de colisão



[collide1.wrl]

Sentindo o observador

Otimizando a detecção de colisão

. Colisão esta ativada por default

○ Pode-se desligá-la quando se desejar!

. Entretanto, quando um pai (parent) desliga um filho (children) não pode desligá-la!

. A colisão resulta do observador colidir com uma forma, mas não de uma forma colidir com o observador

Sentindo o observador

Usando múltiplos sensores

. Qualquer número de sensores podem sentir ao mesmo tempo

- **Você pode ter múltiplos sensores de visibilidade, proximidade e de colisão**
- **Áreas de sensores podem ser sobrepostas**
- **Se múltiplos sensores puderem ser ativados, eles serão**

Sentindo o observador

Resumo

. Um nó `VisibilitySensor` checa se uma região está visível ao observador

- **A região é descrita por um centro e um tamanho**
- **O tempo é sentido na entrada e saída da visibilidade**
- **True/false é enviado na entrada e saída da visibilidade**

Sentindo o observador

Resumo

- **Um nó `ProximitySensor` checa se o observador está dentro de uma região**
  - **A região é descrita por um centro e um tamanho**
  - **O tempo é enviado na entrada e saída do observador**
  - **True/false é enviado na entrada e saída do observador**
  - **Posição e orientação do observador são enviadas enquanto o observador está dentro da região**

Sentindo o observador

Resumo

- **Um nó de grupo `Collision` checa se o observador colidiu com uma forma**
  - **As formas são definidas pelos grupos de children ou de proxy**
  - **O tempo de colisão é enviado no contato**

```
#VRML V2.0 utf8
#
#  prox1.wrl
#  Proximity door
#       by David R. Nadeau
#
#  This world illustrates the use of a ProximitySensor node around a door.
#  When the traveler gets near that door, an animation is triggered
#  that opens the door and plays a sound.  When the traveler
#  leaves the area, the door closes and another sound is played.
#

Viewpoint {
    position 0.0 1.6 7.0
    description "Entry view"
}

NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
    avatarSize [ 1.0, 1.6, 0.75 ]
}

# Room
Inline { url "droom.wrl" }

# Wall
Transform {
        translation 0.0 0.0 -5.0
        children Inline { url "dwall2.wrl" }
}

# Left and right door panels
Transform {
        translation 0.0 0.0 -4.95
        children [
                DEF LeftDoor Transform {
                        children Transform {
                                translation -0.75 0.0 0.0
                                children DEF Door Inline { url "ddoor.wrl" }
                        }
                }
                DEF RightDoor Transform {
                        children Transform {
                                translation 0.75 0.0 0.0
                                children USE Door
                        }
                }
        # Proximity sensor
                DEF DoorSense ProximitySensor {
                        center 0.0 1.75 0.0
                        size 6.0 3.5 8.0
                }
        ]
}

# Sounds
Sound {
        source DEF OpenSound AudioClip { url "clunk2.wav" }
        minFront 20.0 minBack 20.0
        maxFront 60.0 maxBack 60.0
}
```

```
Sound {
        source DEF CloseSound AudioClip { url "clunk1.wav" }
        minFront 20.0 minBack 20.0
        maxFront 60.0 maxBack 60.0
}

# Animation clocks
DEF OpenClock TimeSensor {
        cycleInterval 0.5
        loop FALSE
        startTime 0.0
        stopTime 1.0
}
DEF CloseClock TimeSensor {
        cycleInterval 0.5
        loop FALSE
        startTime 0.0
        stopTime 1.0
}

# Animation paths for the left and right doors
DEF LeftOpen PositionInterpolator {
        key [ 0.0, 1.0 ]
        keyValue [ 0.0 0.0 0.0, -1.3 0.0 0.0 ]
}
DEF LeftClose PositionInterpolator {
        key [ 0.0, 1.0 ]
        keyValue [ -1.3 0.0 0.0, 0.0 0.0 0.0 ]
}
DEF RightOpen PositionInterpolator {
        key [ 0.0, 1.0 ]
        keyValue [ 0.0 0.0 0.0, 1.3 0.0 0.0 ]
}
DEF RightClose PositionInterpolator {
        key [ 0.0, 1.0 ]
        keyValue [ 1.3 0.0 0.0, 0.0 0.0 0.0 ]
}

ROUTE DoorSense.enterTime          TO OpenSound.set_startTime
ROUTE DoorSense.exitTime           TO OpenSound.set_stopTime
ROUTE DoorSense.enterTime          TO OpenClock.set_startTime
ROUTE DoorSense.exitTime           TO OpenClock.set_stopTime

ROUTE DoorSense.exitTime           TO CloseSound.set_startTime
ROUTE DoorSense.enterTime          TO CloseSound.set_stopTime
ROUTE DoorSense.exitTime           TO CloseClock.set_startTime
ROUTE DoorSense.enterTime          TO CloseClock.set_stopTime

ROUTE OpenClock.fraction_changed  TO LeftOpen.set_fraction
ROUTE OpenClock.fraction_changed  TO RightOpen.set_fraction
ROUTE CloseClock.fraction_changed TO LeftClose.set_fraction
ROUTE CloseClock.fraction_changed TO RightClose.set_fraction

ROUTE LeftOpen.value_changed      TO LeftDoor.set_translation
ROUTE LeftClose.value_changed     TO LeftDoor.set_translation
ROUTE RightOpen.value_changed     TO RightDoor.set_translation
ROUTE RightClose.value_changed    TO RightDoor.set_translation
```

```
#VRML V2.0 utf8
#
#  collide1.wrl
#  Collidable door
#       by David R. Nadeau
#
#  This world illustrates the use of a Collision node around a door.
#  When the traveler collides with that door, an animation is triggered
#  that opens the door and plays a sound.  A ProximitySensor watches for
#  the traveler leaving the area, then closes the door and plays a sound.
#

Viewpoint {
    position 0.0 1.6 7.0
    description "Entry view"
}

NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
    avatarSize [ 1.0, 1.6, 0.75 ]
}


# Floor (two strips)
    Transform {
        translation 0.0 0.0 2.5
        children DEF Floor Inline { url "dfloor.wrl" }
    }
    Transform { translation 0.0 0.0 -2.5 children USE Floor }

# Ceiling (reuse the floor)
    Transform { translation 0.0 3.5  2.5 children USE Floor }
    Transform { translation 0.0 3.5 -2.5 children USE Floor }


DEF WallCollide Collision {
    children [
    # Room
        Inline { url "dwalls.wrl" }

    # Wall
        Transform {
            translation 0.0 0.0 -5.0
            children Inline { url "dwall2.wrl" }
        }
    ]
}

# Left and right door panels in a collision group
DEF DoorCollide Collision {
    children Transform {
        translation 0.0 0.0 -4.95
        children [
            DEF LeftDoor Transform {
                # animated translation
                children Transform {
                    translation -0.75 0.0 0.0
                    children DEF Door Inline { url "ddoor.wrl" }
                }
            }
            DEF RightDoor Transform {
```

```
                # animated translation
                children Transform {
                    translation 0.75 0.0 0.0
                    children USE Door
                }
            }
            DEF DoorSense ProximitySensor {
                center 0.0 1.75 0.0
                size 6.0 3.5 8.0
            }
        ]
    }
}

# Sounds
Sound {
    source DEF OpenSound AudioClip { url "clunk2.wav" }
    minFront 20.0 minBack 20.0
    maxFront 60.0 maxBack 60.0
}
Sound {
    source DEF CloseSound AudioClip { url "clunk1.wav" }
    minFront 20.0 minBack 20.0
    maxFront 60.0 maxBack 60.0
}
Sound {
    source DEF UhSound AudioClip { url "uh3.wav" }
    minFront 20.0 minBack 20.0
    maxFront 60.0 maxBack 60.0
}

# Animation clocks
DEF OpenClock TimeSensor {
    cycleInterval 0.5
    loop FALSE
    startTime 0.0
    stopTime 1.0
}
DEF CloseClock TimeSensor {
    cycleInterval 0.5
    loop FALSE
    startTime 0.0
    stopTime 1.0
}

# Animation paths for the left and right doors
DEF LeftOpen PositionInterpolator {
    key [ 0.0, 1.0 ]
    keyValue [ 0.0 0.0 0.0, -1.3 0.0 0.0 ]
}
DEF LeftClose PositionInterpolator {
    key [ 0.0, 1.0 ]
    keyValue [ -1.3 0.0 0.0, 0.0 0.0 0.0 ]
}
DEF RightOpen PositionInterpolator {
    key [ 0.0, 1.0 ]
    keyValue [ 0.0 0.0 0.0, 1.3 0.0 0.0 ]
}
DEF RightClose PositionInterpolator {
    key [ 0.0, 1.0 ]
    keyValue [ 1.3 0.0 0.0, 0.0 0.0 0.0 ]
```

```
}

ROUTE DoorCollide.collideTime     TO OpenSound.set_startTime
ROUTE DoorCollide.collideTime     TO OpenClock.set_startTime

ROUTE DoorSense.exitTime          TO CloseSound.set_startTime
ROUTE DoorSense.exitTime          TO CloseClock.set_startTime

ROUTE WallCollide.collideTime     TO UhSound.set_startTime

ROUTE OpenClock.fraction_changed  TO LeftOpen.set_fraction
ROUTE OpenClock.fraction_changed  TO RightOpen.set_fraction
ROUTE CloseClock.fraction_changed TO LeftClose.set_fraction
ROUTE CloseClock.fraction_changed TO RightClose.set_fraction

ROUTE LeftOpen.value_changed      TO LeftDoor.set_translation
ROUTE LeftClose.value_changed     TO LeftDoor.set_translation
ROUTE RightOpen.value_changed     TO RightDoor.set_translation
ROUTE RightClose.value_changed    TO RightDoor.set_translation
```

Exemplos

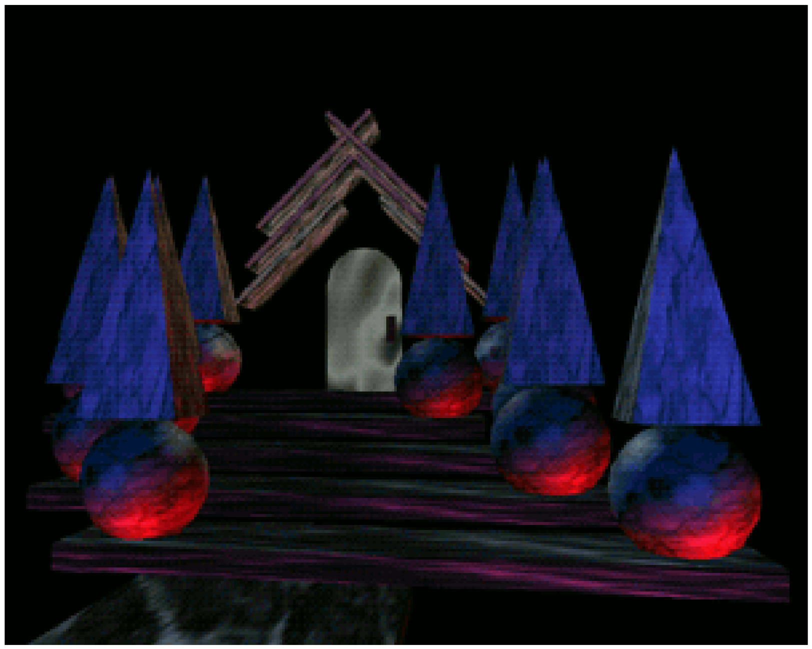
A doorway

A mysterious temple

Exemplos

A doorway

. Um conjunto de nós `ImageTexture` adicionam texturas de mármores
. Nós de luzes criam luzes dramáticas
. Um nó `Fog` faz desaparecer formas distantes
. Um nó `ProximitySensor` controla a animação



[doorway.wrl]

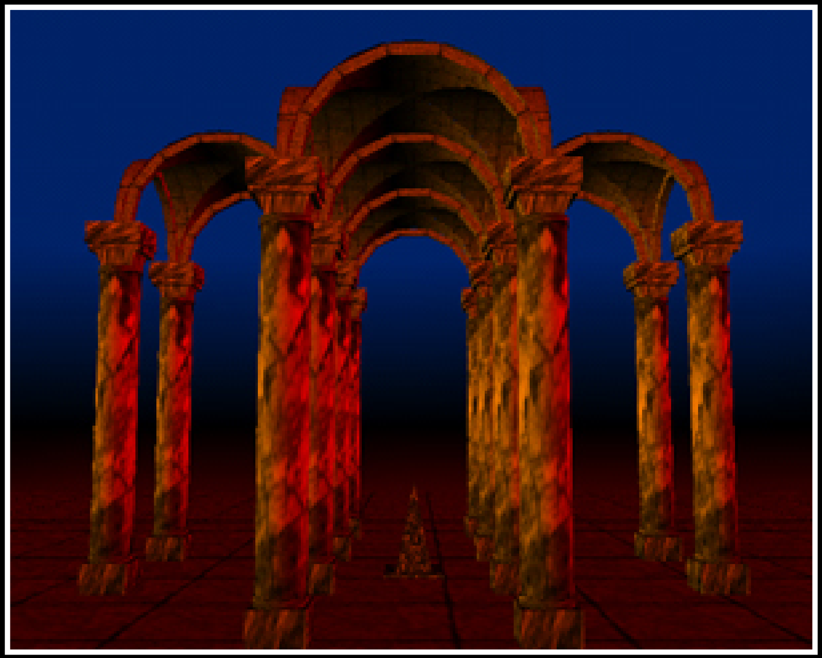
Exemplos

A mysterious temple

. Un nó `Background` cria um gradiente no céu

. Um nó `Sound` cria um efeito de som spatialized

. Um conjunto de nós `Viewpoint` fornecem vistas padrões



[temple.wrl]

```
#VRML V2.0 utf8
#
#  doorway.wrl
#  The Doorway
#      by David R. Nadeau
#
#  Four sliding stairs lead forward along a path to a glowing white
#  doorway.
#
#  A ProximitySensor routes to four TimeSensors, one per sliding
#  stair.  Each stair's timer routes to a PositionInterpolator that
#  outputs varying positions side-to-side along the X axis.
#  The PositionInterpolator's positions are routed into translation
#  fields for Transform nodes surrounding each stair.
#
#  By using different starting positions, the four stairs are made to
#  slide back and forth in a pattern.
#
#  Things to experiment with:
#        Add sound!  Perhaps the crackling of fire below and the grinding
#        sliding sounds of the stone stairs.
#
#        Use a ScalarInterpolator to flicker the red light as if there is
#        fire below.
#
#        Surround the doorway with an Anchor node that leads somewhere!
#
#        Place a ProximitySensor around the doorway.  As you approach it,
#        the ProximitySensor triggers an OrientationInterpolator which
#        slowly rotates the door open.  The open door reveals ... !
#

WorldInfo {
    title "The Doorway"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 3.1 20.0
    description "Entry View"
}
NavigationInfo {
    type [ "WALK", "ANY" ]
    avatarSize [ 0.5, 3.0, 1.5 ]
    headlight FALSE
}

DEF Proximity ProximitySensor {
    size 30.0 30.0 30.0
}

Background {
    skyColor [ 0.0 0.0 0.0 ]
}

Fog {
    color 0.0 0.0 0.0
    visibilityRange 40.0
    fogType "EXPONENTIAL"
}
```

```
#
#  Generic light sources used to give the world ambience
#
PointLight {
    ambientIntensity 0.0
    color 0.6 0.6 0.6
    location 0.0 6.0 -8.0
}
DirectionalLight {
    ambientIntensity 0.0
    color 0.0 0.15 0.5
    direction 1.0 -1.0 -4.0
}
DirectionalLight {
    ambientIntensity 0.0
    intensity 1.5  # extra bright
    color 1.0 0.0 0.0
    direction -1.0 2.5 -1.0
}


#
#  Path to stairs
#
Transform {
    translation 0.0 -0.5 8.0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.8 0.8 0.8
                }
                texture DEF Granit ImageTexture {
                    url "granit_g.jpg"
                }
                textureTransform TextureTransform {
                    scale 0.5 0.5
                }
            }
            geometry Box {
                size 4.0 0.4 16.0
            }
        }
    ]
}


#
#  Moving stairs
#
DEF Stair1 Transform {
    translation 0.0 0.0 0.0
    children [
        DEF Stair Inline {
            url "tread.wrl"
        }
    ]
}

Transform {
    translation -1.0 0.5 -2.0
```

```
    children [
        DEF Stair2 Transform {
            translation 0.0 0.0 0.0
            children [ USE Stair ]
        }
    ]
}

Transform {
    translation 1.5 1.0 -4.0
    children [
        DEF Stair3 Transform {
            translation 0.0 0.0 0.0
            children [ USE Stair ]
        }
    ]
}

Transform {
    translation 0.0 1.5 -6.0
    children [
        DEF Stair4 Transform {
            translation 0.0 0.0 0.0
            children [ USE Stair ]
        }
    ]
}


#
#  The last stair and a door
#
Transform {
    translation 0.0 2.0 -8.0
    children [ USE Stair ]
}

Transform {
    translation 0.0 2.0 -10.0
    scale 0.7 0.7 0.7
    children [
        Inline {
            url "glowdoor.wrl"
        }
    ]
}


#
#  An archway above the door
#
Transform {
    translation -2.0 6.0 -10.0
    rotation 0.0 0.0 1.0 0.785
    children [
        DEF ArchBox Shape {
            appearance Appearance {
                material Material {
                    ambientIntensity 0.0
                    diffuseColor 1.0 1.0 1.0
                    emissiveColor 0.4 0.4 0.4
                }
```

```
                texture ImageTexture {
                    url "marble_g.jpg"
                }
            }
            geometry Box {
                size 4.0 0.2 2.0
            }
        }
        Transform {
            translation 1.0 0.2 1.0
            children [ USE ArchBox ]
        }
        Transform {
            translation 2.0 0.4 2.0
            children [ USE ArchBox ]
        }
    ]
}

Transform {
    translation 2.0 6.0 -10.0
    rotation 0.0 0.0 1.0 -0.785
    children [
        USE ArchBox,
        Transform {
            translation -1.0 0.2 1.0
            children [ USE ArchBox ]
        }
        Transform {
            translation -2.0 0.4 2.0
            children [ USE ArchBox ]
        }
    ]
}





#
#  Animation control
#
#  For each stair, a TimeSensor is triggered by a world ProximitySensor.
#  Once triggered, the time sensor outputs fractions forever.  The fractions
#  are routed in to PositionInterpolators that smoothly interpolate a
#  sliding position back and forth along the X axis.  These positions are
#  routed in to translations for the individual stairs.
#
DEF Stair1Timer TimeSensor {
    cycleInterval 4.0
    loop TRUE
    startTime -1.0
}
DEF Stair1Path PositionInterpolator {
    key      [ 0.0,         0.25,        0.5,         0.75,         1.0         ]
    keyValue [ 0.0 0.0 0.0, 3.0 0.0 0.0, 0.0 0.0 0.0, -3.0 0.0 0.0, 0.0 0.0 0.0 ]
}
ROUTE Proximity.enterTime           TO Stair1Timer.set_startTime
ROUTE Proximity.exitTime            TO Stair1Timer.set_stopTime
ROUTE Stair1Timer.fraction_changed TO Stair1Path.set_fraction
ROUTE Stair1Path.value_changed     TO Stair1.set_translation
```

```
DEF Stair2Timer TimeSensor {
    cycleInterval 4.0
    loop TRUE
    startTime -1.0
}
DEF Stair2Path PositionInterpolator {
    key      [ 0.0,          0.25,          0.5,            0.75,          1.0           ]
    keyValue [ 3.0 0.0 0.0, 0.0 0.0 0.0, -3.0 0.0 0.0, 0.0 0.0 0.0, 3.0 0.0 0.0 ]
}
ROUTE Proximity.enterTime          TO Stair2Timer.set_startTime
ROUTE Proximity.exitTime           TO Stair2Timer.set_stopTime
ROUTE Stair2Timer.fraction_changed TO Stair2Path.set_fraction
ROUTE Stair2Path.value_changed     TO Stair2.set_translation


DEF Stair3Timer TimeSensor {
    cycleInterval 4.0
    loop TRUE
    startTime -1.0
}
DEF Stair3Path PositionInterpolator {
    key      [ 0.0,           0.25,          0.5,          0.75,           1.0           ]
    keyValue [ -3.0 0.0 0.0, 0.0 0.0 0.0, 3.0 0.0 0.0, 0.0 0.0 0.0, -3.0 0.0 0.0 ]
}
ROUTE Proximity.enterTime          TO Stair3Timer.set_startTime
ROUTE Proximity.exitTime           TO Stair3Timer.set_stopTime
ROUTE Stair3Timer.fraction_changed TO Stair3Path.set_fraction
ROUTE Stair3Path.value_changed     TO Stair3.set_translation


DEF Stair4Timer TimeSensor {
    cycleInterval 4.0
    loop TRUE
    startTime -1.0
}
DEF Stair4Path PositionInterpolator {
    key      [ 0.0,          0.25,           0.5,          0.75,          1.0           ]
    keyValue [ 0.0 0.0 0.0, -3.0 0.0 0.0, 0.0 0.0 0.0, 3.0 0.0 0.0, 0.0 0.0 0.0 ]
}
ROUTE Proximity.enterTime          TO Stair4Timer.set_startTime
ROUTE Proximity.exitTime           TO Stair4Timer.set_stopTime
ROUTE Stair4Timer.fraction_changed TO Stair4Path.set_fraction
ROUTE Stair4Path.value_changed     TO Stair4.set_translation
```

Controlando detalhes

Motivação

Exemplo

Criando múltiplas versões de forma

Controlando nível detalhe

Escolhendo a distância dos detalhes

Sintaxe: LOD

Otimizando a forma

Uma amostra de versões de detalhes

Uma amostra LOD

Uma amostra LOD

Resumo

Controlando detalhes

Motivação

. Quanto mais distante o observador pode ver, existirá mais coisas a desenhar

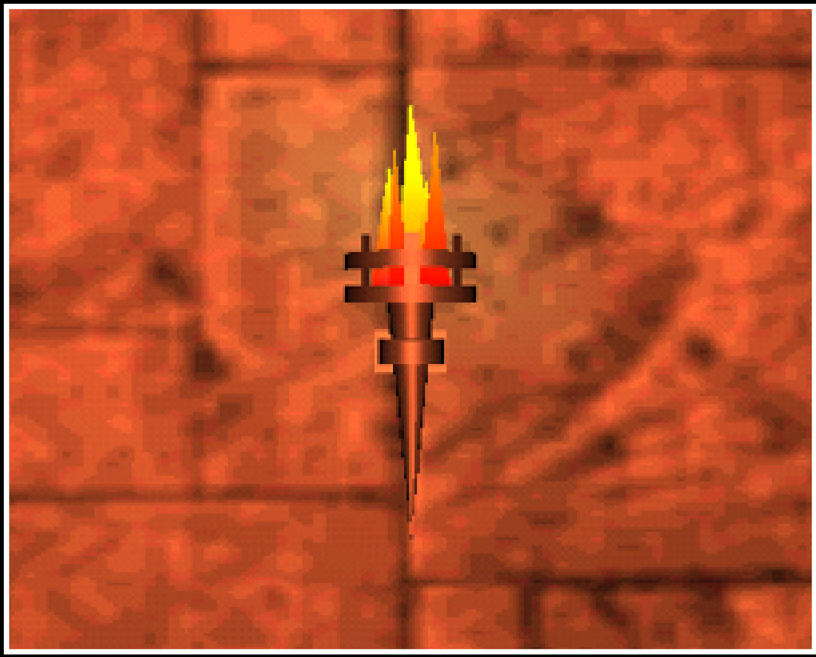
. Se a forma está distante:

- **A forma é pequena**
- **O observador não pode vê-la com muito detalhe**
- **Então... desenhe comp pouco detalhe**

. Variando o detalhe com a distância, reduz o tempo de download e aumenta a velocidade de desenho

Controlando detalhes

Exemplo



[prox1.wrl]

Controlando detalhes

Criando múltiplas versões de forma

. Para controlar o detalhe, o modelo deve ter a mesma forma várias vezes

- **muitos detalhes quando o observador estiver em close**
- **detalhes médios quando o observador estive mais ou menos perto**
- **poucos detalhes quando o observador estiver distante**

. Usualmente, duas ou três versões são suficientes, mas você pode ter mais, como você desejar

Controlando detalhes

Controlando nível de detalhes

- **Versões de forma Group como *níveis* em um nó grupo `LOD`**
  - ***LOD* é uma abreviação de *Level of Detail***
  - **São listados do mais detalhado para o menos detalhado**
- **Posicione o grupo inteiro no ponto *central***

Controlando detalhes

Escolhendo a distância dos detalhes

. Use uma lista de distâncias para versões dos pontos chave

- **Se você tem 3 versões, você necessita de duas distâncias**
- **As distâncias são sugestões para o *browser***

```
range [ 7.5, 12.0 ]
```

observador < 7.5	primeiro nó filho é usedo
7.5 <= observador < 12.0	segundo nó filho é usado
12.0 < observador	terceiro nó filho é usado

Controlando detalhes

Sintaxe: LOD

. Um nó grupo `LOD` cria um grupo de formas descrevendo diferentes versões da mesma forma

- ***center*** **- o centro da forma**
- ***range*** **- uma lista de distâncias chaves**
- ***level*** **- uma lista de versões de forma**

```
LOD {
    center 0.0 0.0 0.0
    range [ . . . ]
    level [ . . . ]
}
```

Controlando detalhes

Otimizando a forma

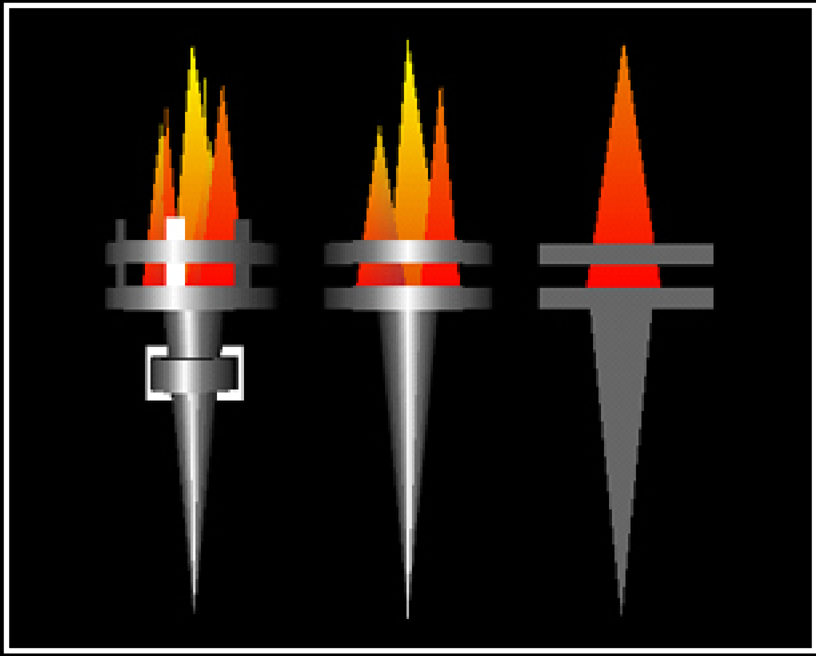
. Procedimento sugerido para fazer diferentes versões:

- **Faça a forma com mais detalhe primeiro**
- **Copie na versão de detalhe médio**
- **Mova a forma com detalhes médios para distância chave desejada**
- **Apague as partes que não são dominantes**
- **Repita para a versão de poucos detalhes**

. Versões de poucos detalhes podem ter geometria simplificada, poucas texturas e sem texto

Controlando detalhes

Uma amostra de versões de detalhes



[torches3.wrl]

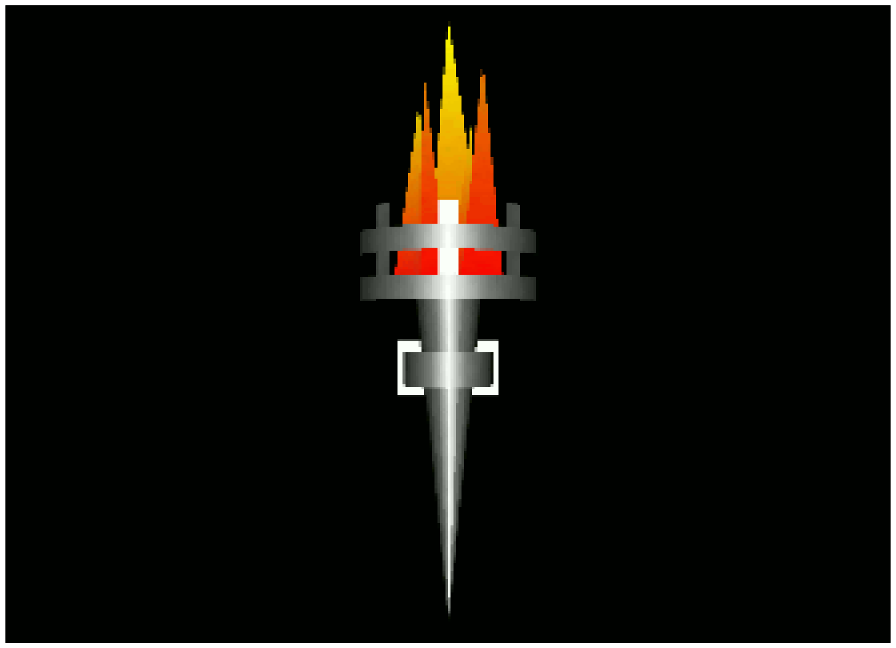
Controlando detalhes

Uma amostra LOD

```
LOD {
    center 0.0 0.0 0.0
    range [ 7.5, 12.0 ]
    level [
        Inline { url "torch1.wrl" }
        Inline { url "torch2.wrl" }
        Inline { url "torch3.wrl" }
    ]
}
```

Controlando detalhes

Uma amostra LOD



[torches.wrl]

Controlando detalhes

Resumo

- **Aumenta a performance fazendo múltiplas versões das formas**
  - **Muitos detalhes para a forma em close para observador**
  - **Poucos detalhes para o observador mais distante**
- **Versões de nó Group em um nó `LOD`**
  - **Ordenados de muitos detalhes para poucos detalhes**
  - **Limites são selecionados para trocar distâncias**

```
#VRML V2.0 utf8

Viewpoint {
    position 0.0 0.0 2.5
    description "Entry view"
}

Transform {
    translation -0.5 0.0 0.0
    children Inline { url "torch1.wrl" }
}

Transform {
    translation 0.0 0.0 0.0
    children Inline { url "torch2.wrl" }
}
Transform {
    translation 0.5 0.0 0.0
    children Inline { url "torch3.wrl" }
}
```

```
#VRML V2.0 utf8
#
#  torches.wrl
#  A torch
#    by David R. Nadeau
#
#  This file illustrates the use of an LOD node to automatically
#  switch among different versions of a shape based upon the
#  distance the viewer is from that shape.
#

LOD {
    center 0.0 0.0 0.0
    range [ 7.0, 10.0 ]
    level [
    # High-detail
        Inline { url "torch1.wrl" },
    # Medium-detail
        Inline { url "torch2.wrl" },
    # Low-detail
        Inline { url "torch3.wrl" }
    ]
}
```

Introduzindo o uso de script

Motivação

Sintaxe: Script

Definindo a interface script do programa

Uma amostra usando um programa script

Uma amostra usando um programa script

Resumo

Introduzindo o uso de script

Motivação

. Muitas ações são muito complexas para nós animados

- **Calcular animações (exemplo gravidade)**
- **Formas Algoritmicas (exemplo fractais)**
- **Ambientes Colaborativos (exemplo jogos)**

. Você pode criar novos sensores, interpoladores, etc., usando programas scripts escritos em:

- ***Java* - poderosa linguagem de propósito geral**
- ***JavaScript* - linguagem de fácil aprendizado**
- ***VRMLscript* - mesmo que JavaScript**

Introduzindo o uso de script

Sintaxe: Script

. Um nó `Script` seleciona um programa script a ser executado:

- ***url* - programa script escolhido**

```
DEF MyScript Script {
    url "myscript.class"
ou...
    url "myscript.js"
ou...
    url "javascript: ..."
ou...
    url "vrmlscript: ..."
}
```

Introduzindo o uso de script

Definindo a interface script do programa

- **Um nó `Script` também declara a interface de programa script**
  - ***fields* e *events* - ins e outs**
- **Cada um tem o nome e o tipo de dados**
- **Os campos tem um valor inicial**

```
DEF Bouncer Script {
    field    SFFloat bounceHeight 3.0
    eventIn  SFFloat set_fraction
    eventOut SFVec3f value_changed
}
```

Introduzindo o uso de script

Uma amostra usando um programa script

```
DEF Bouncer Script { . . .  }

ROUTE Clock.fraction_changed
   TO Bouncer.set_fraction

ROUTE Bouncer.value_changed
   TO Ball.set_translation
```

Introduzindo o uso de script

Uma amostra usando um programa script

[bounce1.wrl]

Introduzindo o uso de script

Resumo

- **O nó `Script` seleciona um programa script especificado por uma URL**
- **Os programas scripts possuem declarações de campos e interfaces de eventos, cada um com:**
  - **O tipo de dado**
  - **Um nome**
  - **Um valor inicial (para campos somente)**

```
#VRML V2.0 utf8
#
#  bounce1.wrl
#  Bouncing beachball (JavaScript/VRMLscript version)
#        by David R. Nadeau
#
#  This world illustrates the use of a Script node to create a computed
#  animation path.  In particular, the Script node uses a JavaScript
#  (or VRMLScript) program script to compute translation values for a
#  vertically bouncing beach ball.
#
#  The bounce path is based upon the projectile motion equation of
#  physics, constrained to create a cyclic bouncing path with a
#  user-selected maximum bounce height.  Also, there is no friction,
#  drag, or damping.
#
#  The equation is derived as follows:
#
#        (1)  Projectile motion computes a y(t) value (height as a function
#             of time) based upon the gravitation constant, g, an initial
#             y-direction velocity, v0, an initial y position, y0, and the
#             current time, t:
#
#                y(t) = g * t * t + v0 * t + y0
#
#        (2)  At time t=0, the ball should be on the ground with y=0.
#             So, y0 = y(0) = 0.  The equation in (1) simplifies to:
#
#                y(t) = g * t * t + v0 * t
#
#        (3)  At time t=1, at the end of the TimeSensor's fractional time
#             cycle, the ball should again be on the ground with y=0.
#             So, y(1) = 0.  Plugging this in to the equation in (2),
#             we get:
#
#                y(t) = g * t * t + v0 * t
#                y(1) = g * 1 * 1 + v0 * 1
#                   0 = g + v0
#
#             So
#
#                   v0 = -g
#
#        (4)  At time t=0.5, the ball should be at the peak of its bounce
#             at a user-selected maximum height, h.  So, y(0.5) = h.
#             Plugging this in to the equation in (2), we get:
#
#                y(t)   = g * t * t + v0 * t
#                y(0.5) = g * 0.5 * 0.5 + v0 * 0.5
#                     h = g * 0.25 + v0 * 0.5
#
#            And v0 = -g from equation (3), so
#
#                     h = g * 0.25 - g * 0.5
#                     h = -g * 0.25
#
#            So
#
#                     g = -4.0 * h
#
#        (5)  We can now simplify the equation in (2) using the results
```

```
#               from (3) and (4) to get an equation that computes the
#               ball height y(t) parameterized only by the maximum height, h,
#               giving us:
#
#                  y(t) = g * t * t + v0 * t
#                       = (-4.0 * h) * t * t + (-g) * t
#                       = (-4.0 * h) * t * t + (4.0 * h) * t
#                       = 4.0 * h * (-t * t + t)
#                       = 4.0 * h * t * (1.0 - t)
#
#               In the program script, the maximum height, h, is given in
#               the 'bounceHeight' field.  The current time, t, is given in
#               the 'set_fraction' eventIn and passed to the eventIn function
#               as the 'frac' parameter.  Using these names, the above
#               equation becomes:
#
#                   y = 4.0 * bounceHeight * frac * (1.0 - frac)
#
#  Things to experiment with
#        -  Encapsulate the ball, script, timer, and sensors within a
#           PROTO for a new node named "BouncingBall".  Then use that
#           new BouncingBall node multiple times to create multiple
#           bouncing balls.  Your PROTO interface might look like this:
#
#                PROTO BouncingBall [
#                        field SFFloat bounceHeight     2.0
#                        field SFTime  cycleInterval 2.0
#                ] { . . . }
#
#           See 'bounce3.wrl', which implements such a PROTO.
#
#        -  Add a shadow under the bouncing ball.  To do this, add a
#           circular, semi-transparent, black shape that doesn't bounce.
#           To make the shadow more realistic, scale the shadow in the X
#           and Z directions, shrinking it as the ball goes up, and
#           increasing it as the ball comes down.  You'll need to add
#           another eventOut for the Script node and send an XYZ scaling
#           factor triple out that eventOut.  Try the following values
#           for the XYZ scale values:
#
#                xzscale = 1.0 - 0.5 * y / bounceHeight;
#                shadowScale_changed[0] = xzscale;
#                shadowScale_changed[1] = 1.0;
#                shadowScale_changed[2] = xzscale;
#
#           See 'bounce4.wrl', which implements shadows using the above
#           scale values.
#
#        -  Add a sound to the PROTO so that each time the ball touches
#           the ground, it makes a 'boing' sound.
#
#        -  When the ball hits the ground, scale the ball slightly so that
#           it appears to squish.
#

WorldInfo {
        title "Bouncing beachball (JavaScript)"
        info [ "Copyright (c) 1997, David R. Nadeau" ]
}

Viewpoint {
```

```
    position 0.0 0.6 8.0
    orientation 1.0 0.0 0.0 0.1
}

NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
    speed 2.0
}

DirectionalLight {
    ambientIntensity 0.5
    direction 0.0 -1.0 -0.5
}

#
#  Sky
#
Background {
    skyColor [
        0.0 0.0 1.0,
        0.0 0.5 1.0,
        0.7 0.7 1.0,
    ]
    skyAngle [
        1.371,
        1.571,
    ]
}


#
#  Beach
#
Shape {
    appearance Appearance {
        material Material { }
        texture ImageTexture { url "sand.jpg" }
        textureTransform TextureTransform { scale 10.0 10.0 }
    }
    geometry IndexedFaceSet {
        coord Coordinate {
            point [
                -50.0 -1.0  50.0,
                 50.0 -1.0  50.0,
                 50.0 -1.0 -50.0,
                -50.0 -1.0 -50.0,
            ]
        }
        coordIndex [ 0, 1, 2, 3 ]
        solid FALSE
    }
}


#
#  Palm trees
#
Transform {
    translation -3.0 -1.0 -10.0
    children [
        DEF Palm Group {
```

```
          children [
          # Palm tree - in a billboard so it is never edge-on
              Billboard {
                  children [
                      Shape {
                          appearance Appearance {
                              material NULL  # emissive texturing
                              texture ImageTexture { url "palm.png" }
                          }
                          geometry IndexedFaceSet {
                              coord Coordinate {
                                  point [
                                      -2.5  0.0 0.0,
                                       2.5  0.0 0.0,
                                       2.5 11.25 0.0,
                                      -2.5 11.25 0.0,
                                  ]
                              }
                              coordIndex [ 0, 1, 2, 3 ]
                              texCoord TextureCoordinate {
                                      point [
                                              0.0 0.0,
                                              1.0 0.0,
                                              1.0 1.0,
                                              0.0 1.0,
                                      ]
                              }
                              texCoordIndex [ 0, 1, 2, 3 ]
                              solid FALSE
                          }
                      }
                  ]
              }
          # Fake tree shadow - a black semi-transparent rectangle with
          # a texture map to give it the right shape
              Shape {
                  appearance Appearance {
                      material Material {
                              diffuseColor 0.0 0.0 0.0
                              transparency 0.5
                      }
                      texture ImageTexture { url "palmsh.png" }
                  }
                  geometry IndexedFaceSet {
                      coord Coordinate {
                          point [
                              -2.5  0.05  2.5,
                               2.5  0.05  2.5,
                               2.5  0.05 -2.5,
                              -2.5  0.05 -2.5,
                          ]
                      }
                      coordIndex [ 0, 1, 2, 3 ]
                      texCoord TextureCoordinate {
                              point [
                                      0.0 0.0,
                                      1.0 0.0,
                                      1.0 1.0,
                                      0.0 1.0,
                              ]
                      }
```

```
                        texCoordIndex [ 0, 1, 2, 3 ]
                        solid FALSE
                    }
                }
            ]
        }
    ]
}
Transform { translation -5.0 -1.0  -6.0 scale 0.6 0.6 0.6 children USE Palm }
Transform { translation  5.0 -1.0  -9.0 children USE Palm }
Transform { translation 10.0 -1.0 -15.0 children USE Palm }


#
#  Bouncing beach ball
#
DEF Ball Transform {
    # animated translation
    children [
        Shape {
            appearance Appearance {
                material Material {
                    ambientIntensity 0.5
                    diffuseColor 1.0 1.0 1.0
                    specularColor 0.7 0.7 0.7
                    shininess 0.4
                }
                texture ImageTexture { url "beach.jpg" }
                textureTransform TextureTransform { scale 2.0 1.0 }
            }
            geometry Sphere { }
        }
    ]
}

DEF Clock TimeSensor {
    cycleInterval 2.0
    startTime 1.0
    stopTime 0.0
    loop TRUE
}

DEF Bouncer Script {
    field    SFFloat bounceHeight 3.0
    eventIn  SFFloat set_fraction
    eventOut SFVec3f value_changed

    # change 'vrmlscript' to 'javascript' for newer browsers
    url "vrmlscript:
        function set_fraction( frac, tm ) {
            y = 4.0 * bounceHeight * frac * (1.0 - frac);
            value_changed[0] = 0.0;
            value_changed[1] = y;
            value_changed[2] = 0.0;
        }"
}

ROUTE Clock.fraction_changed TO Bouncer.set_fraction
ROUTE Bouncer.value_changed  TO Ball.set_translation
```

Escrevendo programas scripts com JavaScript

Motivação

Declarando uma interface programa script

Initializando um programa script

Encerrando um programa script

Respondendo aos eventos

Processando eventos em JavaScript

Acessando campos do JavaScript

Accessando eventOuts do JavaScript

Uma amostra de script JavaScript

Uma amostra de script JavaScript

Uma amostra de script JavaScript

Uma amostra de script JavaScript

Uma amostra de script JavaScript

Uma amostra de script JavaScript

Uma amostra de script JavaScript

Uma amostra de script JavaScript

Construindo interfaces do usuário

Construindo uma chave de toque

Usando um botão de toque

Usando um botão de toque

Construindo um seletor de cores

Usando um seletor de cores

Usando um seletor de cores

Resumo

Escrevendo programas scripts com JavaScript

Motivação

. Um programa script implementa um nó `Script` usando valores da interface

- **O script responde as entradas e dispara saídas**

. Um programa script pode ser escrito em *Java*, *JavaScript*, e outras linguagens

- **JavaScript é mais fácil de programar**
- **Java é mais poderoso**

Escrevendo programas script com JavaScript

Declarando uma interface programa script

. Para um programa script JavaScript, usualmente fornecemos o script no campo `url` do nó `Script`

```
DEF Bouncer Script {
    field     SFFloat bounceHeight 3.0
    eventIn   SFFloat set_fraction
    eventOut SFVec3f value_changed
    url "javascript: . . ."
}
```

Escrevendo programas script com JavaScript

Inicializando um programa script

- A função opcional `initialize` é chamada quando o script é carregado

```
function initialize ( ) {
    . . .
}
```

- A inicialização ocorre quando:
  - o nó `Script` é criado (tipicamente quando o browser carrega o mundo)

Escrevendo programas script com JavaScript

Encerrando um programa script

- **A função opcional `shutdown` é chamada quando o script é descarregado**

```
function shutdown ( ) {
     . . .
}
```

- **O encerramento pode ocorrer quando:**
  - **o nó `Script` é deletado**
  - **o browser carrega um novo mundo**

Escrevendo programas script com JavaScript

Respondendo aos eventos

- **Uma função *eventIn* deve ser declarada para cada eventIn**
- **A função eventIn é chamada cada vez que o evento é recebido passando**
  - **valor, e**
  - **time stamp; do evento**

```
function set_fraction( value, timestamp ) {
    . . .
}
```

Escrevendo programas script com JavaScript

Processando eventos em JavaScript

. Se múltiplos eventos chegam ao mesmo tempo, então múltiplas funções eventIn são chamadas

. A função opcional `eventsProcessed` é chamada depois de todas (ou algumas) funções eventIn functions terem sido chamadas

```
function eventsProcessed ( ) {
    . . .
}
```

Escrevendo programas script com JavaScript

Acessando campos do JavaScript

- Cada campo de interface é uma variável JavaScript
  - Leia a variável para acessar o valor do campo
  - Escreva a variável para trocar o valor do campo

```
lastval = bounceHeight;     # get field
bounceHeight = newval;      # set field
```

Escrevendo programas script com JavaScript

Acessando eventOuts do JavaScript

- **Cada interface eventOut é uma variável JavaScript**
  - **Leia a variável para accessar o último valor eventOut**
  - **Escreva a variável para mandar um evento sobre o eventOut**

```
lastval = value_changed[0];  # get last event
value_changed[0] = newval;   # send new event
```

Escrevendo programas script com JavaScript

Uma amostra de script JavaScript

- **Crie um interpolador de bola saltando (*Bouncing ball interpolator)* que calcula a gravidade no movimento de salto vertical de uma entrada de fração de tempo**
- **Campos necessários:**
  - **Altura do salto**

```
DEF Bouncer Script {
    field SFFloat bounceHeight 3.0
    . . .
}
```

Escrevendo programas script com JavaScript

Construindo interfaces do usuário

- **Programas scripts podem ser usados para ajudar a criar interfaces 3D para o usuário**
  - **Botões de toque**
  - **Botões Radio**
  - **Botão rotativo**
  - **Scrollbars**
  - **Linhas de comando**
  - **Mensagens de texto de debug**

Escrevendo programas script com JavaScript

Construindo um botão de toque

- Um script botão de toque liga no primeiro toque, e deliga no segundo toque
    - Um nó `TouchSensor` pode suportar os eventos de toque

```
DEF Toggle Script {
    field     SFBool on TRUE
    eventIn   SFTime set_active
    eventOut SFBool on_changed
    url "vrmlscript:
        function set_active( b, tm ) {
            if ( b == FALSE ) return;
            if ( on == TRUE ) on =
FALSE;
            else              on = TRUE;
            on_changed = on;
        }"
}
```

Escrevendo programas script com JavaScript

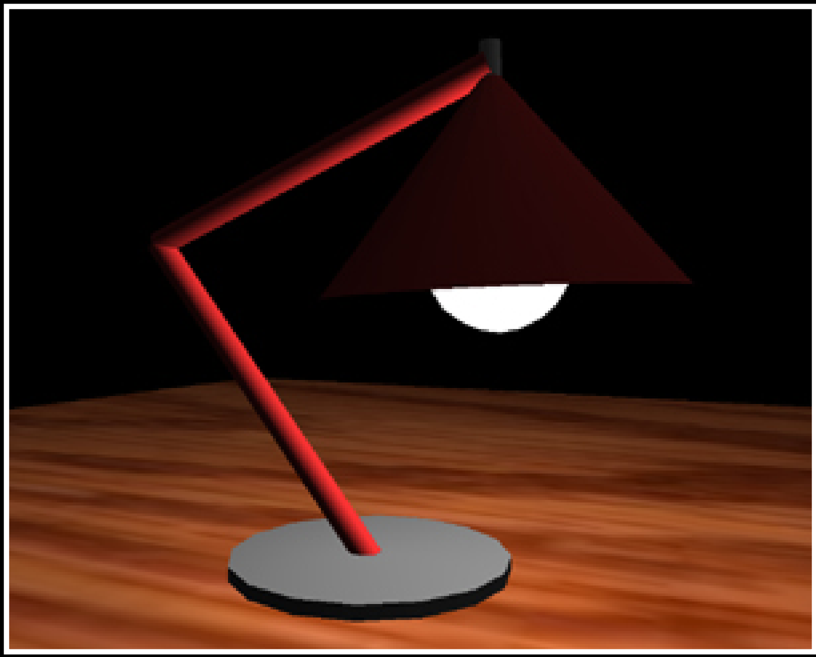
Usando um botão de toque

. Use o botão de toque para fazer a lâmpada ligar e desligar

- **Use o nó `TouchSensor` para sentir a forma trocada**
- **Envie o nó sensor eventOut `isActive` para o nó script eventIn `set_active`**
- **Envie o nó script eventOut `on_changed` para o nó de luz eventIn `set_on`**

Escrevendo programas script com JavaScript

Usando botão de toque



[lamp2a.wrl]

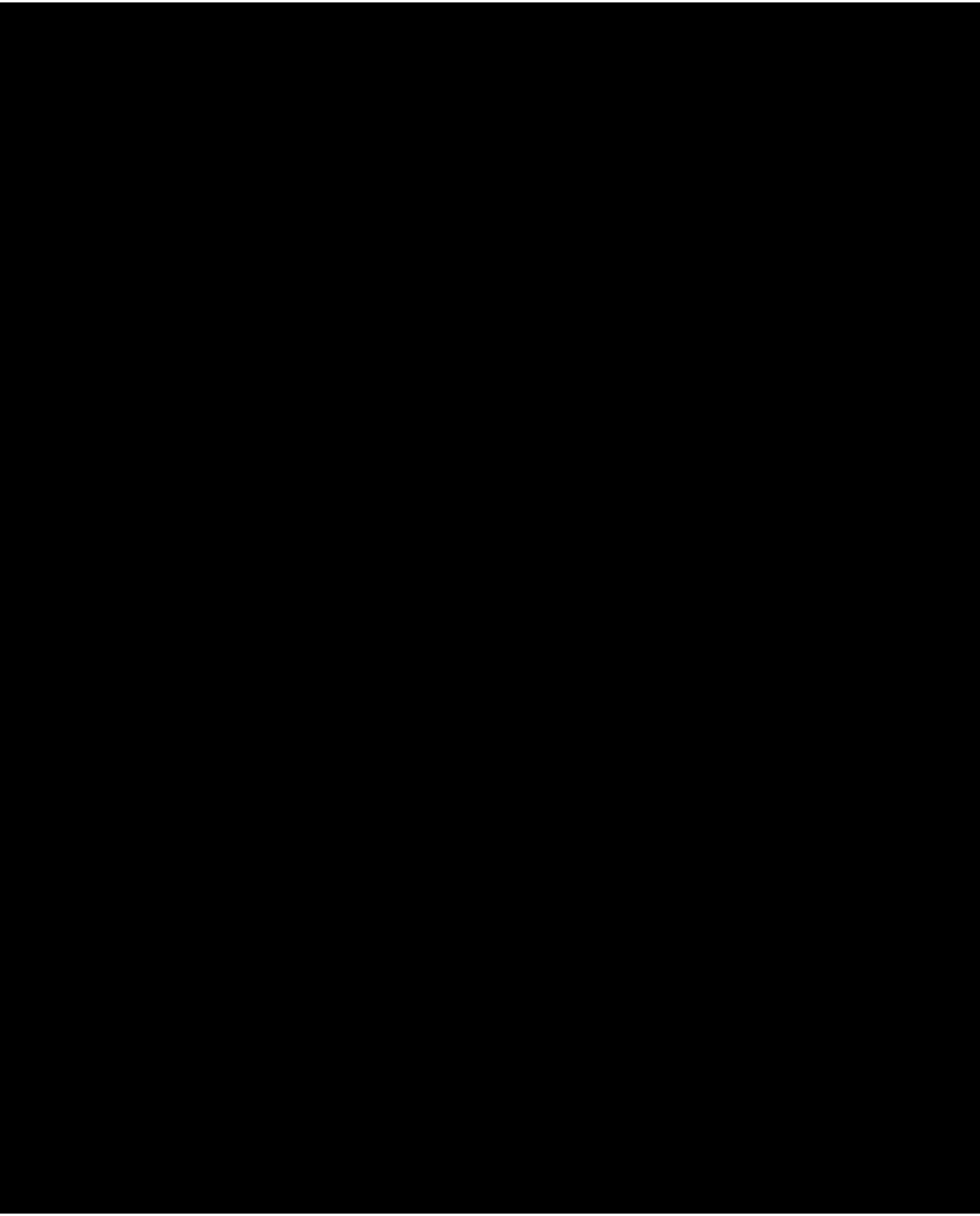
Escrevendo programas script com JavaScript

Construindo um seletor de cores

. A lâmpada do exemplo anterior liga e desliga, mas o bulbo da luz não troca de cor!

. Um script seletor de cores envia cor *on* para uma entrada `TRUE` , e um cor *off* para uma entrada `FALSE`

```
DEF ColorSelector Script {
    field    SFColor onColor  1.0 1.0
1.0
    field    SFColor offColor 0.0 0.0
0.0
    eventIn  SFBool  set_selection
    eventOut SFColor color_changed
    url "vrmlscript:
        function set_selection( b, tm )
{
            if ( b == TRUE )
color_changed  = onColor;
            else
color_changed = offColor;
        }"
}
```



Escrevendo programas script com JavaScript

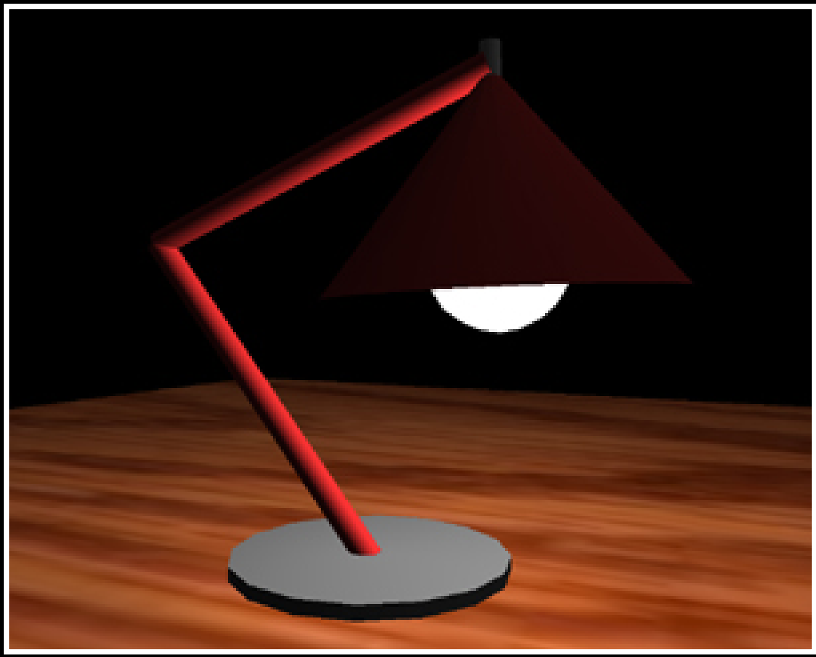
Usando um seletor de cores

. Use o seletor de cores para troca a cor do bulbo da lâmpada

- **Envie o eventOut do nó script toque `on_changed` para o eventIn do nó seletor script `set_selection`**
- **Envie o eventOut do nó script selector `color_changed` para o eventIn do nó do bulbo `Material` eventIn `set_emissiveColor`**

Escrevendo programas script com JavaScript

Usando um seletor de cores



[lamp2.wrl]

Escrevendo programas script com JavaScript

Resumo

. As funções `initialize` e `shutdown` são chamadas no carregamento e descarregamento

. Uma função eventIn é chamada quando o evento é recebido

. A função `eventsProcessed` é chamada depois de todos (ou alguns) eventos terem sido recebidos

. As funções podem receber valores de campos e enviarem eventos de saída

Escrevendo programas script com JavaScript

Uma amostra de script JavaScript

- **Entradas e saídas necessitam:**
  - **Fração de tempo de entrada**
  - **Valor de posição de saída**

```
DEF Bouncer Script {
    . . .
    eventIn  SFFloat set_fraction
    eventOut SFVec3f value_changed
    . . .
}
```

Escrevendo programas script com JavaScript

Uma amostra de script JavaScript

[bounce1.wrl]

```
#VRML V2.0 utf8
#
#  lamp2a.wrl
#  Desk lamp
#       by David R. Nadeau
#
#  This file builds a desk lamp that can be moved about.  Dragging
#  on the lamp base slides the lamp back and forth.  Dragging on the
#  lamp arms turns the lamp arm.  Dragging on the lamp shade turns
#  the lamp shade.  Touching the lamp's light switch (on the top of the
#  lamp shade) turns the lamp on and off.
#
#  To make the lamp go on and off, a TouchSensor is routed into a
#  'toggle' script that alternates between TRUE and FALSE outputs,
#  sending TRUE on the first TRUE input, then FALSE on the next TRUE
#  input.  These TRUE/FALSE values are routed into a SpotLight node
#  within the lamp, turning it on and off.
#

WorldInfo {
    title "Desk lamp"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 0.1 1.0
    description "Entry View"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight FALSE
}

Background {
    skyColor [ 0.0 0.0 0.0 ]
}

DirectionalLight {
  direction 0.0 -1.0 -1.0
  intensity 0.3
}

Transform {
  translation -1.0 0.0 -1.0
  children [
    Shape {
      appearance Appearance {
        material Material { }
        texture ImageTexture { url "wood.jpg" }
      }
      geometry ElevationGrid {
        xDimension 10
        zDimension 10
        xSpacing 0.25
        zSpacing 0.25
        creaseAngle 3.14
        height [
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        ]
      }
    }
  ]
}


Transform {
  translation 0.0 0.01 0.0
  rotation 1.0 0.0 0.0 1.571
  children [
  # Lamp
    DEF MoveLamp PlaneSensor { }
    DEF Lamp Transform {
      rotation 1.0 0.0 0.0 -1.571
      children [
      # Lamp base
        Shape {
          appearance Appearance {
            material Material { diffuseColor 0.5 0.5 0.5 }
          }
          geometry Cylinder {
            radius 0.1
            height 0.01
          }
        }
      # Base - First arm joint
        Group {
          children [
            DEF MoveFirstArm CylinderSensor { }
            DEF FirstArm Transform {
              children [
Transform {
translation 0.0 0.15 0.0
rotation    1.0 0.0 0.0  -0.7
center      0.0 -0.15 0.0
children [
              # Lower arm
                DEF LampArm Shape {
                  appearance DEF Red Appearance {
                    material Material { diffuseColor 1.0 0.2 0.2 }
                  }
                  geometry Cylinder {
                    radius 0.01
                    height 0.3
                  }
                }
              # First arm - second arm joint
                Group {
                  children [
                    DEF MoveSecondArm CylinderSensor { }
                    DEF SecondArm Transform {
                      children [
Transform {
translation 0.0 0.3 0.0
```

```
rotation  1.0 0.0 0.0  1.9
center    0.0 -0.15 0.0
children [
                        # Second arm
                          USE LampArm,
                        # Second arm - shade joint
                          Group {
                            children [
                              DEF MoveLampShade SphereSensor {
                                offset 1.0 0.0 0.0 -1.25
                              }
                              DEF LampShade Transform {
                                translation 0.0 0.075 0.0
                                rotation  1.0 0.0 0.0  -1.25
                                center    0.0 0.075 0.0
                                children [
                                # Shade
                                  Shape {
                                    appearance USE Red
                                    geometry Cone {
                                      height 0.15
                                      bottomRadius 0.12
                                      bottom FALSE
                                    }
                                  }
                                # Switch
                                  Transform {
                                    translation 0.0 0.075 0.0
                                    children [
                                      DEF LightSwitch TouchSensor { }
                                      Shape {
                                        appearance Appearance {
                                          material Material {
                                            diffuseColor 1.0 1.0 1.0
                                          }
                                        }
                                        geometry Cylinder {
                                          radius 0.007
                                          height 0.03
                                        }
                                      }
                                    ]
                                  }
                                # Light bulb
                                  Transform {
                                    translation 0.0 -0.05 0.0
                                    children [
                                      DEF LampLight SpotLight {
                                        ambientIntensity 0.2
                                        intensity 1.0
                                        direction 0.0 -1.0 0.0
                                        location 0.0 0.0 0.0
                                        beamWidth 0.6
                                        cutOffAngle 1.571
                                      }
                                      Shape {
                                        appearance Appearance {
                                          material DEF BulbMaterial Material {
                                             diffuseColor 0.0 0.0 0.0
                                             emissiveColor 1.0 1.0 1.0
                                          }
```

```
                                       }
                                       geometry Sphere {
                                         radius 0.05
                                       }
                                     }
                                   ]
                                 }
                               ]
                             }
                           ]
                         }
] }
                       ]
                     }
                   ]
                 }
] }
               ]
             }
           ]
         }
       ]
     }
   ]
 }
ROUTE MoveLamp.translation_changed   TO Lamp.set_translation
ROUTE MoveFirstArm.rotation_changed  TO FirstArm.set_rotation
ROUTE MoveSecondArm.rotation_changed TO SecondArm.set_rotation
ROUTE MoveLampShade.rotation_changed TO LampShade.set_rotation


DEF Toggle Script {
        field    SFBool on TRUE
        eventIn  SFBool set_active
        eventOut SFBool on_changed

        # use 'javascript' on newer browsers
        url "vrmlscript:
                function set_active( b, tm ) {
                        if ( b == FALSE ) return;
                        if ( on == TRUE ) on = FALSE;
                        else              on = TRUE;
                        on_changed = on;
                }"
}

ROUTE LightSwitch.isActive TO Toggle.set_active
ROUTE Toggle.on_changed    TO LampLight.set_on
```

```
#VRML V2.0 utf8
#
#  lamp2.wrl
#  Desk lamp
#       by David R. Nadeau
#
#  This file builds a desk lamp that can be moved about.  Dragging
#  on the lamp base slides the lamp back and forth.  Dragging on the
#  lamp arms turns the lamp arm.  Dragging on the lamp shade turns
#  the lamp shade.  Touching the lamp's light switch (on the top of the
#  lamp shade) turns the lamp on and off.
#
#  To make the lamp go on and off, a TouchSensor is routed into a
#  'toggle' script that alternates between TRUE and FALSE outputs,
#  sending TRUE on the first TRUE input, then FALSE on the next TRUE
#  input.  These TRUE/FALSE values are routed into a SpotLight node
#  within the lamp, turning it on and off.
#
#  A second script converts an incoming TRUE or FALSE in to one of
#  two colors.  Those colors are routed into the lamp's bulb to change
#  it from an on color to an off color.
#

WorldInfo {
    title "Desk lamp"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 0.1 1.0
    description "Entry View"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight FALSE
}

Background {
    skyColor [ 0.0 0.0 0.0 ]
}

DirectionalLight {
  direction 0.0 -1.0 -1.0
  intensity 0.3
}

Transform {
  translation -1.0 0.0 -1.0
  children [
    Shape {
      appearance Appearance {
        material Material { }
        texture ImageTexture { url "wood.jpg" }
      }
      geometry ElevationGrid {
        xDimension 10
        zDimension 10
        xSpacing 0.25
        zSpacing 0.25
        creaseAngle 3.14
        height [
```

```
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        ]
      }
    }
  ]
}


Transform {
  translation 0.0 0.01 0.0
  rotation 1.0 0.0 0.0 1.571
  children [
  # Lamp
    DEF MoveLamp PlaneSensor { }
    DEF Lamp Transform {
      rotation 1.0 0.0 0.0 -1.571
      children [
      # Lamp base
        Shape {
          appearance Appearance {
            material Material { diffuseColor 0.5 0.5 0.5 }
          }
          geometry Cylinder {
            radius 0.1
            height 0.01
          }
        }
      # Base - First arm joint
        Group {
          children [
            DEF MoveFirstArm CylinderSensor { }
            DEF FirstArm Transform {
              children [
Transform {
translation 0.0 0.15 0.0
rotation    1.0 0.0 0.0  -0.7
center      0.0 -0.15 0.0
children [
              # Lower arm
                DEF LampArm Shape {
                  appearance DEF Red Appearance {
                    material Material { diffuseColor 1.0 0.2 0.2 }
                  }
                  geometry Cylinder {
                    radius 0.01
                    height 0.3
                  }
                }
              # First arm - second arm joint
                Group {
                  children [
                    DEF MoveSecondArm CylinderSensor { }
```

```
                    DEF SecondArm Transform {
                      children [
Transform {
translation 0.0 0.3 0.0
rotation  1.0 0.0 0.0  1.9
center    0.0 -0.15 0.0
children [
                      # Second arm
                        USE LampArm,
                      # Second arm - shade joint
                        Group {
                          children [
                            DEF MoveLampShade SphereSensor {
                              offset 1.0 0.0 0.0 -1.25
                            }
                            DEF LampShade Transform {
                              translation 0.0 0.075 0.0
                              rotation  1.0 0.0 0.0  -1.25
                              center    0.0 0.075 0.0
                              children [
                              # Shade
                                Shape {
                                  appearance USE Red
                                  geometry Cone {
                                    height 0.15
                                    bottomRadius 0.12
                                    bottom FALSE
                                  }
                                }
                              # Switch
                                Transform {
                                  translation 0.0 0.075 0.0
                                  children [
                                    DEF LightSwitch TouchSensor { }
                                    Shape {
                                      appearance Appearance {
                                        material Material {
                                          diffuseColor 1.0 1.0 1.0
                                        }
                                      }
                                      geometry Cylinder {
                                        radius 0.007
                                        height 0.03
                                      }
                                    }
                                  ]
                                }
                              # Light bulb
                                Transform {
                                  translation 0.0 -0.05 0.0
                                  children [
                                    DEF LampLight SpotLight {
                                      ambientIntensity 0.2
                                      intensity 1.0
                                      direction 0.0 -1.0 0.0
                                      location 0.0 0.0 0.0
                                      beamWidth 0.6
                                      cutOffAngle 1.571
                                    }
                                    Shape {
                                      appearance Appearance {
```

```
                                        material DEF BulbMaterial Material {
                                           diffuseColor 0.0 0.0 0.0
                                           emissiveColor 1.0 1.0 1.0
                                        }
                                      }
                                      geometry Sphere {
                                        radius 0.05
                                      }
                                    }
                                  ]
                                }
                              ]
                            }
                          ]
                        }
] }
                      ]
                    }
                  ]
                }
] }
              ]
            }
          ]
        }
      ]
    }
  ]
}
ROUTE MoveLamp.translation_changed   TO Lamp.set_translation
ROUTE MoveFirstArm.rotation_changed  TO FirstArm.set_rotation
ROUTE MoveSecondArm.rotation_changed TO SecondArm.set_rotation
ROUTE MoveLampShade.rotation_changed TO LampShade.set_rotation


DEF Toggle Script {
        field    SFBool on TRUE
        eventIn  SFBool set_active
        eventOut SFBool on_changed

        # use 'javascript' on newer browsers
        url "vrmlscript:
                function set_active( b, tm ) {
                        if ( b == FALSE ) return;
                        if ( on == TRUE ) on = FALSE;
                        else              on = TRUE;
                        on_changed = on;
                }"
}

DEF ColorSelector Script {
        field    SFColor onColor  1.0 1.0 1.0
        field    SFColor offColor 0.0 0.0 0.0
        eventIn  SFBool  set_selection
        eventOut SFColor color_changed

        # use 'javascript' on newer browsers
        url "vrmlscript:
                function set_selection( b, tm ) {
                        if ( b == TRUE )
                                color_changed = onColor;
```

```
                        else
                                color_changed = offColor;
                }"
}
ROUTE LightSwitch.isActive TO Toggle.set_active
ROUTE Toggle.on_changed    TO LampLight.set_on
ROUTE Toggle.on_changed    TO ColorSelector.set_selection
ROUTE ColorSelector.color_changed TO BulbMaterial.set_emissiveColor
```

Escrevendo programas scritps com Java

Motivação

Declarando uma interface programa script

Criando uma classe Java

Inicializando um programa script

Encerrando um programa script

Respondendo aos eventos

Processando eventos em Java

Acessando campos do Java

Acessando eventos de saída do Java

Uma amostra script Java

Uma amostra script Javat

Uma amostra script Java

Uma amostra script Java

Uma amostra script Java

Uma amostra script Java

Uma amostra script Java

Uma amostra script Java

Uma amostra script Java

Uma amostra script Java

Resumo

Escrevendo programas script com JavaScript

Uma amostra de script JavaScript

- **As ações de inicialização e encerramento necessitam:**
  - **Nada - Todo trabalho é feito com a função eventIn**

Escrevendo programas script com JavaScript

Uma amostra de script JavaScript

- **As rotas necessitam:**
  - **Relógio (Clock) dentro do script `set_fraction`**
  - **O `value_changed` do script dentro do transform**

```
ROUTE Clock.fraction_changed
   TO Bouncer.set_fraction

ROUTE Bouncer.value_changed
   TO Ball.set_translation
```

Escrevendo programas scritps com Java

Motivação

- **Comparado ao JavaScript, Java habilita:**
  - **Melhor modularity**
  - **Melhores estruturas de dados**
  - **Potencial para uma execução mais rápida**
  - **Acesso a rede**
- **Para tarefas simples, use JavaScript**
- **Para tarefas complexas, use Java**

Escrevendo programas scritps com Java

Declarando uma interface programa script

. Para um programa script Java, forneça o arquivo class no campo `url` do nó `Script`

○ O arquivo class é um programa escrito em Java compilado

```
DEF Bouncer Script {
    field    SFFloat bounceHeight 3.0
    eventIn  SFFloat set_fraction
    eventOut SFVec3f value_changed
    url "bounce2.class"
}
```

Escrevendo programas scritps com Java

Criando uma classe Java

. O arquivo do programa escrito deve importar os pacotes VRML:

```
import vrml.*;
import vrml.field.*;
import vrml.node.*;
```

. O programa escrito deve definir uma classe *public* extendendo a classe `Script`

```
public class bounce2
    extends Script
{
    . . .
}
```

Escrevendo programas scritps com Java

Inicializando um programa script

. O método opcional `initialize` é chamado quando o script é carregado

```
public void initialize ( ) {
    . . .
}
```

. A inicialização ocorre quando:

◦ o nó `Script` é criado (normalmente quando o browser carrega o mundo)

Escrevendo programas scritps com Java

Encerrando um programa script

. O método opcional `shutdown` é chamado quando o script é descarregado

```
public void shutdown ( ) {
    . . .
}
```

. O encerramento ocorre quando:

- o nó `Script` é deletado
- o browser carrega um novo mundo

Escrevendo programas scritps com Java

Respondendo aos eventos

. O método `processEvent` é chamado a cada tempo que um evento é recebido, passando um objeto `Event` contido no evento

- **valor**
- **time stamp**

```
public void processEvent( Event event )
{
      . . .
}
```

Sumário Índice do Capítulo Slide Anterior Slide Seguinte 398

Escrevendo programas scritps com Java

Processando eventos em Java

. Se múltiplos eventos chegam ao mesmo tempo, então o método `processEvent` é chamado múltipas vezes

. O método opcional `eventsProcessed` é chamado depois de todos (ou alguns) eventos terem sido tratados

```
public void eventsProcessed ( ) {
    . . .
}
```

Escrevendo programas scritps com Java

Acessando campos do Java

- **Cada campo interface pode ser lido ou escrito**
  - **Chame `getField` para pegar um campo objeto**

```
obj = (SFFloat) getField( "bounceHeight" );
```

  - **Chame `getValue` para pegar um valor de campo**

```
lastval = obj.getValue( );
```

  - **Chame `setValue` para atribuir um valor de campo**

```
obj.setValue( newval );
```

Escrevendo programas scritps com Java

Acessando eventos de saída do Java

- **Cada interface eventOut pode ser lida ou escrita**
  - **Chame `getEventOut` Para pegar um objeto eventOut**

```
obj = (SFVec3f) getEventOut(
"value_changed" );
```

  - **Chame `getValue` para pegar o último evento enviado**

```
lastval = obj.getValue( );
```

  - **Chame `setValue` para mandar um evento**

```
obj.setValue( newval );
```

Escrevendo programas scritps com Java

Uma amostra script Java

. Crie um interpolador (*Bouncing ball interpolator)* que calcula a gravidade em um movimento de salto de uma bola, em uma fração de um tempo de entrada

. Forneça a mesma interface do exemplo do JavaScript

```
DEF Bouncer Script {
    field    SFFloat bounceHeight 3.0
    eventIn  SFFloat set_fraction
    eventOut SFVec3f value_changed
    url "bounce2.class"
}
```

Escrevendo programas scritps com Java

Uma amostra script Java

. É necessario importar e definir uma classe:

```
import vrml.*;
import vrml.field.*;
import vrml.node.*;

public class bounce2
    extends Script
{
    . . .
}
```

Escrevendo programas scritps com Java

Resumo

. Os métodos `initialize` e`shutdown` são chamados no carregamento e descarregamento

. O método `processEvent` é chamado quando o evento é recebido

. O método `eventsProcessed` é chamado depois de todos (ou alguns) eventos terem sido recebidos

. Os métodos podem fornecer valores de campo e enviarem eventos de saída

Escrevendo programas script com JavaScript

Uma amostra de script JavaScript

- **As ações de processar eventos necessitam:**
  - **a função eventIn** `set_fraction`
  - **Não precisa da função** `eventsProcessed`

```
DEF Bouncer Script {
    . . .
    url "javascript:
        function set_fraction( frac, tm
) {
            . . .
        }"
}
```

Escrevendo programas script com JavaScript

Uma amostra de script JavaScript

- **São necessários cálculos:**
  - **Calcular nova posição da bola**
  - **Enviar evento de nova posição**
- **Use a equação de posição de bola baseado aproximadamente sobre regras de Física**
  - **Veja comentários no arquivo VRML para a derivação da equação**

Escrevendo programas script com JavaScript

Uma amostra de script JavaScript

```
function set_fraction( frac, tm ) {
    y = 4.0 * bounceHeight * frac * (1.0
- frac);
    value_changed[0] = 0.0;
    value_changed[1] = y;
    value_changed[2] = 0.0;
}
```

Escrevendo programas scritps com Java

Uma amostra script Java

- **São necessárias as variáveis para a classes:**
  - **Uma para o campo `bounceHeight`**
  - **Outra para o objeto eventOut `value_changed`**

```
private float   bounceHeight;
private SFVec3f value_changedObj;
```

Escrevendo programas scritps com Java

Uma amostra script Java

[bounce2.wrl]

Criando novos tipos de nós

Motivação

Sintaxe: PROTO

Definindo o corpo do protótipo

Sintaxe: IS

Usando IS

Usando nós prototipados

Controlando regras de uso

Controlando regras de uso

Uma amostra usando protótipos

Uma amostra usando protótipos

Uma amostra usando protótipos

Uma amostra usando protótipos

Uma amostra usando protótipos

Trocando o protótipo

Uma amostra usando protótipos

Sintaxe: EXTERNPROTO

Resumo

Escrevendo programas scritps com Java

Uma amostra script Java

- **São necessárias ações de inicialização:**
  - **Pegue o valor do campo `bounceHeight`**
  - **Pegue o objeto eventOut `value_changedObj`**

```
public void initialize( )
{
    SFFloat obj  = (SFFloat) getField(
"bounceHeight" );
    bounceHeight = (float)
obj.getValue( );
    value_changedObj = (SFVec3f)
getEventOut( "value_changed" );
}
```

Escrevendo programas scritps com Java

Uma amostra script Java

- **As rotas necessitam:**
  - **Relógio (Clock) dentro do script `set_fraction`**
  - **O `value_changed` do script dentro do transform**

```
ROUTE Clock.fraction_changed
   TO Bouncer.set_fraction

ROUTE Bouncer.value_changed
   TO Ball.set_translation
```

```
#VRML V2.0 utf8
#
#  bounce2.wrl
#  Bouncing beachball (Java version)
#       by David R. Nadeau
#
#  This world illustrates the use of a Script node to create a computed
#  animation path.  In particular, the Script node uses a Java
#  program script to compute translation values for a
#  vertically bouncing beach ball.
#
#  The bounce path is based upon the projectile motion equation of
#  physics, constrained to create a cyclic bouncing path with a
#  user-selected maximum bounce height.  Also, there is no friction,
#  drag, or damping.  For an explanation of the script, see 'bounce1.wrl'.
#
#  Things to experiment with
#       -  Encapsulate the ball, script, timer, and sensors within a
#          PROTO for a new node named "BouncingBall".  Then use that
#          new BouncingBall node multiple times to create multiple
#          bouncing balls.  Your PROTO interface might look like this:
#
#               PROTO BouncingBall [
#                       field SFFloat bounceHeight     2.0
#                       field SFTime  cycleInterval 2.0
#               ] { . . . }
#
#          See 'bounce3.wrl', which implements such a PROTO.
#
#       -  Add a shadow under the bouncing ball.  To do this, add a
#          circular, semi-transparent, black shape that doesn't bounce.
#          To make the shadow more realistic, scale the shadow in the X
#          and Z directions, shrinking it as the ball goes up, and
#          increasing it as the ball comes down.  You'll need to add
#          another eventOut for the Script node and send an XYZ scaling
#          factor triple out that eventOut.  Try the following values
#          for the XYZ scale values:
#
#               xzscale = 1.0 - 0.5 * y / bounceHeight;
#               shadowScale_changed[0] = xzscale;
#               shadowScale_changed[1] = 1.0;
#               shadowScale_changed[2] = xzscale;
#
#          See 'bounce4.wrl', which implements shadows using the above
#
#       -  Add a sound to the PROTO so that each time the ball touches
#          the ground, it makes a 'boing' sound.
#
#       -  When the ball hits the ground, scale the ball slightly so that
#          it appears to squish.
#

WorldInfo {
        title "Bouncing beachball (Java)"
        info [ "Copyright (c) 1997, David R. Nadeau" ]
}

Viewpoint {
    position 0.0 0.6 8.0
    orientation 1.0 0.0 0.0 0.1
}
```

```
NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
    speed 2.0
}

DirectionalLight {
    ambientIntensity 0.5
    direction 0.0 -1.0 -0.5
}

#
#  Sky
#
Background {
    skyColor [
        0.0 0.0 1.0,
        0.0 0.5 1.0,
        0.7 0.7 1.0,
    ]
    skyAngle [
        1.371,
        1.571,
    ]
}


#
#  Beach
#
Shape {
    appearance Appearance {
        material Material { }
        texture ImageTexture { url "sand.jpg" }
        textureTransform TextureTransform { scale 10.0 10.0 }
    }
    geometry IndexedFaceSet {
        coord Coordinate {
            point [
                -50.0 -1.0  50.0,
                 50.0 -1.0  50.0,
                 50.0 -1.0 -50.0,
                -50.0 -1.0 -50.0,
            ]
        }
        coordIndex [ 0, 1, 2, 3 ]
        solid FALSE
    }
}


#
#  Palm trees
#
Transform {
    translation -3.0 -1.0 -10.0
    children [
        DEF Palm Group {
            children [
            # Palm tree - in a billboard so it is never edge-on
                Billboard {
```

```
            children [
                Shape {
                    appearance Appearance {
                        material NULL  # emissive texturing
                        texture ImageTexture { url "palm.png" }
                    }
                    geometry IndexedFaceSet {
                        coord Coordinate {
                            point [
                                -2.5  0.0 0.0,
                                 2.5  0.0 0.0,
                                 2.5 11.25 0.0,
                                -2.5 11.25 0.0,
                            ]
                        }
                        coordIndex [ 0, 1, 2, 3 ]
                        texCoord TextureCoordinate {
                                point [
                                        0.0 0.0,
                                        1.0 0.0,
                                        1.0 1.0,
                                        0.0 1.0,
                                ]
                        }
                        texCoordIndex [ 0, 1, 2, 3 ]
                        solid FALSE
                    }
                }
            ]
        }
    # Fake tree shadow - a black semi-transparent rectangle with
    # a texture map to give it the right shape
        Shape {
            appearance Appearance {
                material Material {
                        diffuseColor 0.0 0.0 0.0
                        transparency 0.5
                }
                texture ImageTexture { url "palmsh.png" }
            }
            geometry IndexedFaceSet {
                coord Coordinate {
                    point [
                        -2.5  0.05  2.5,
                         2.5  0.05  2.5,
                         2.5  0.05 -2.5,
                        -2.5  0.05 -2.5,
                    ]
                }
                coordIndex [ 0, 1, 2, 3 ]
                texCoord TextureCoordinate {
                        point [
                                0.0 0.0,
                                1.0 0.0,
                                1.0 1.0,
                                0.0 1.0,
                        ]
                }
                texCoordIndex [ 0, 1, 2, 3 ]
                solid FALSE
            }
```

```
                }
            ]
        }
    ]
}
Transform { translation -5.0 -1.0  -6.0 scale 0.6 0.6 0.6 children USE Palm }
Transform { translation  5.0 -1.0  -9.0 children USE Palm }
Transform { translation 10.0 -1.0 -15.0 children USE Palm }


#
#  Bouncing beach ball
#
DEF Ball Transform {
    # animated translation
    children [
        Shape {
            appearance Appearance {
                material Material {
                    ambientIntensity 0.5
                    diffuseColor 1.0 1.0 1.0
                    specularColor 0.7 0.7 0.7
                    shininess 0.4
                }
                texture ImageTexture { url "beach.jpg" }
                textureTransform TextureTransform { scale 2.0 1.0 }
            }
            geometry Sphere { }
        }
    ]
}

DEF Clock TimeSensor {
    cycleInterval 2.0
    startTime 1.0
    stopTime 0.0
    loop TRUE
}

DEF Bouncer Script {
    field    SFFloat bounceHeight 3.0
    eventIn  SFFloat set_fraction
    eventOut SFVec3f value_changed

    url "bounce2.class"
}

ROUTE Clock.fraction_changed TO Bouncer.set_fraction
ROUTE Bouncer.value_changed  TO Ball.set_translation
```

Criando novos tipos de nós

Motivação

- **Você pode criar novos tipos de nós que encapsulam:**
  - **Formas**
  - **Sensores**
  - **Interpoladores**
  - **Scripts**
  - **qualquer coisa . . .**
- **Isto cria nós de alto nível**
  - **Robôs, menus, novas formas, etc.**

Criando novos tipos de nós

Sintaxe: PROTO

- **O comando `PROTO` declara um novo tipo de nó**
  - ***name* - o nome do novo tipo de nó**
  - ***fields* e *events* - interface para o protótipo**

```
PROTO BouncingBall [
    field SFFloat bounceHeight  1.0
    field SFTime  cycleInterval 1.0
] { . . . }
```

Criando novos tipos de nós

Definindo o corpo do protótipo

- definição de **PROTO** :
  - *body* - nós e rotas para o novo tipo de nó

```
PROTO BouncingBall [ . . .  ] {
    Transform {
        children [ . . . ]
    }
    ROUTE . . .
}
```

Criando novos tipos de nós

Sintaxe: IS

. A sintaxe IS conecta o campo interface do protótipo, eventIn, ou eventOut para o corpo

```
PROTO BouncingBall [
    field SFFloat bounceHeight  1.0
    field SFTime  cycleInterval 1.0
] {
    . . .
    DEF Clock TimeSensor {
        cycleInterval IS cycleInterval
        . . .
    }
    . . .
}
```

Criando novos tipos de nós

Usando IS

Interface	Pode IS para . . .			
	Fields	Exposed fields	EventIns	EventOuts
Fields	yes	yes	no	no
Exposed fields	no	yes	no	no
EventIns	no	yes	yes	no
EventOuts	no	yes	no	yes

Criando novos tipos de nós

Usando nós prototipados

. O novo tipo de nó pode ser usados como qualquer outro tipo

```
BouncingBall {
    bounceHeight  3.0
    cycleInterval 2.0
}
```

Criando novos tipos de nós

Controlando regras de uso

- **Lembre-se que o uso do nó deve ser apropriado para o contexto**
  - **Um nó `Shape` especifica forma, não cor**
  - **Um nó `Material` especifica cor, não forma**
  - **Um nó `Box` especifica geometria, nem forma ou cor**

Sumário Índice do Capítulo Slide Anterior Slide Seguinte 421

Criando novos tipos de nós

Uma amostra usando protótipos

- Crie um novo tipo `BouncingBall` que:
  - Constrói uma bola de praia
  - Cria um relógio de animação
- Usa o campo `PROTO` para selecionar o intervalo de ciclo
  - Salte a bola de praia
- Usando o programa script de bola saltando
- Usando o campo `PROTO` para selecionar a altura de salto

Criando novos tipos de nós

Trocando o protótipo

. Se você trocar o protótipo, toda utilização daquele protótipo será trocada

◦ Os prototipos habilitam a modularidade do mundo

◦ Grandes mundos fazem uso pesado de protótipos

. Para o protótipo `BouncingBall` , adicionando a sombra para o protótipo faz com que todas as bolas tenham sombra

Criando novos tipos de nós

Sintaxe: EXTERNPROTO

. Os Prototipos estão tipicamente em um arquivo separado *external*

. Uma `EXTERNPROTO` declara um novo tipo de nó em um arquivo externo

- ***name, fields, events* - como em `PROTO`**
- ***url* - a URL do arquivo protótipo**

```
EXTERNPROTO BouncingBall [
    field SFFloat bounceHeight 1.0
    field SFTime  cycleInterval 1.0
] "bounce.wrl#BouncingBall"
```

Criando novos tipos de nós

Resumo

- **`PROTO` declara um novo tipo de nó e define o seu corpo nó**
- **`EXTERNPROTO` declara um novo tipo de nó especificado pela URL**
- **O novo nó pode ser usado em qualquer lugar, do primeiro nó, no corpo do protótipo**
-

Escrevendo programas scritps com Java

Uma amostra script Java

- **As ações de encerramento necessitam:**
  - **Nada - todo trabalho é feito no método `processEvent`**

Escrevendo programas scritps com Java

Uma amostra script Java

- **As ações de processar eventos necessitam:**
  - **Método evento `processEvent`**
  - **Não necessárias para o método `eventsProcessed`**

```
public void processEvent( Event event )
{
    . . .
}
```

Escrevendo programas scritps com Java

Uma amostra script Java

```
public void processEvent( Event event )
{
    ConstSFFloat flt = (ConstSFFloat)
event.getValue( );
    float frac        = (float)
flt.getValue( );

    float y = (float)(4.0 * bounceHeight
* frac * (1.0 - frac));

    float[] changed = new float[3];
    changed[0] = (float)0.0;
    changed[1] = y;
    changed[2] = (float)0.0;
    value_changedObj.setValue( changed
);
}
```

Escrevendo programas scritps com Java

Uma amostra script Java

- **Os cálculos necessitam:**
  - **Calcular nova posição da bola**
  - **Enviar o evento nova posição**

Criando novos tipos de nós

Controlando regras de uso

- **O contexto para um novo tipo de nó depende do primeiro nó no corpo `PROTO`**
- **Por exemplo, se o primeiro nó é um nó de geometria (*geometry node*):**
  - **O protótipo cria um novo nó de geometria**
- **O novo tipo de nó pode ser usado onde quer que o primeiro nó do corpo de protótipo pode ser usado**

Criando novos tipos de nós

Uma amostra usando protótipos

- **Os campos necessitam:**
  - **Altura de salto**
  - **Intervalo de ciclo**

```
PROTO BouncingBall [
    field SFFloat bounceHeight 1.0
    field SFTime  cycleInterval 1.0
] { . . . }
```

Criando novos tipos de nós

Uma amostra usando protótipos

- **Entradas e Saídas necessitam:**
  - **Nada - um nó `TimeSensor` é construído para o novo nó**

Criando novos tipos de nós

Uma amostra usando protótipos

[bounce3.wrl]

Criando novos tipos de nós

Uma amostra usando protótipos

- **O corpo do protótipo necessita:**
  - **Uma forma bola dentro do transform**
  - **Um relógio de animação**
  - **Um programa script bola saltando (bouncing ball)**
  - **Rotas conectando-os todos juntos**

```
PROTO BouncingBall [ . . . ] {
    DEF Ball Transform {
        children [
            Shape { . . . }
        ]
    }
    DEF Clock   TimeSensor { . . . }
    DEF Bouncer Script { . . . }
    ROUTE . . .
}
```

```
#VRML V2.0 utf8
#
#  bounce3.wrl
#  Bouncing beachball (with PROTO and JavaScript)
#    by David R. Nadeau
#
#  This world illustrates the use of a PROTO to encapsulate a Script
#  node, timer, shape, and routes to create a vertically bouncing
#  beach ball.
#
#  The bounce path is based upon the projectile motion equation of
#  physics, constrained to create a cyclic bouncing path with a
#  user-selected maximum bounce height.  Also, there is no friction,
#  drag, or damping.  For an explanation of the script, see 'bounce1.wrl'.
#

WorldInfo {
        title "Bouncing beachball (PROTO)"
        info [ "Copyright (c) 1997, David R. Nadeau" ]
}

Viewpoint {
    position 0.0 0.6 8.0
    orientation 1.0 0.0 0.0 0.1
}

NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
    speed 2.0
}

DirectionalLight {
    ambientIntensity 0.5
    direction 0.0 -1.0 -0.5
}

#
#  Sky
#
Background {
    skyColor [
        0.0 0.0 1.0,
        0.0 0.5 1.0,
        0.7 0.7 1.0,
    ]
    skyAngle [
        1.371,
        1.571,
    ]
}


#
#  Beach
#
Shape {
    appearance Appearance {
        material Material { }
        texture ImageTexture { url "sand.jpg" }
        textureTransform TextureTransform { scale 10.0 10.0 }
    }
```

```
    geometry IndexedFaceSet {
        coord Coordinate {
            point [
                -50.0 -1.0  50.0,
                 50.0 -1.0  50.0,
                 50.0 -1.0 -50.0,
                -50.0 -1.0 -50.0,
            ]
        }
        coordIndex [ 0, 1, 2, 3 ]
        solid FALSE
    }
}


#
#  Palm trees
#
Transform {
    translation -3.0 -1.0 -10.0
    children [
        DEF Palm Group {
            children [
            # Palm tree - in a billboard so it is never edge-on
                Billboard {
                    children [
                        Shape {
                            appearance Appearance {
                                material NULL  # emissive texturing
                                texture ImageTexture { url "palm.png" }
                            }
                            geometry IndexedFaceSet {
                                coord Coordinate {
                                    point [
                                        -2.5  0.0 0.0,
                                         2.5  0.0 0.0,
                                         2.5 11.25 0.0,
                                        -2.5 11.25 0.0,
                                    ]
                                }
                                coordIndex [ 0, 1, 2, 3 ]
                                texCoord TextureCoordinate {
                                        point [
                                                0.0 0.0,
                                                1.0 0.0,
                                                1.0 1.0,
                                                0.0 1.0,
                                        ]
                                }
                                texCoordIndex [ 0, 1, 2, 3 ]
                                solid FALSE
                            }
                        }
                    ]
                }
            # Fake tree shadow - a black semi-transparent rectangle with
            # a texture map to give it the right shape
                Shape {
                    appearance Appearance {
                        material Material {
                                diffuseColor 0.0 0.0 0.0
```

```
                                transparency 0.5
                        }
                        texture ImageTexture { url "palmsh.png" }
                    }
                    geometry IndexedFaceSet {
                        coord Coordinate {
                            point [
                                -2.5  0.05  2.5,
                                 2.5  0.05  2.5,
                                 2.5  0.05 -2.5,
                                -2.5  0.05 -2.5,
                            ]
                        }
                        coordIndex [ 0, 1, 2, 3 ]
                        texCoord TextureCoordinate {
                                point [
                                        0.0 0.0,
                                        1.0 0.0,
                                        1.0 1.0,
                                        0.0 1.0,
                                ]
                        }
                        texCoordIndex [ 0, 1, 2, 3 ]
                        solid FALSE
                    }
                }
            ]
        }
    ]
}
Transform { translation -5.0 -1.0  -6.0 scale 0.6 0.6 0.6 children USE Palm }
Transform { translation  5.0 -1.0  -9.0 children USE Palm }
Transform { translation 10.0 -1.0 -15.0 children USE Palm }


#
#  Bouncing ball prototype
#
PROTO BouncingBall [
    field SFFloat bounceHeight  1.0
    field SFTime  cycleInterval 1.0
] {
    DEF Ball Transform {
        # animated translation
        children [
            Shape {
                appearance Appearance {
                    material Material {
                        ambientIntensity 0.5
                        diffuseColor 1.0 1.0 1.0
                        specularColor 0.7 0.7 0.7
                        shininess 0.4
                    }
                    texture ImageTexture { url "beach.jpg" }
                    textureTransform TextureTransform { scale 2.0 1.0 }
                }
                geometry Sphere { }
            }
        ]
    }
    DEF Clock TimeSensor {
```

```
        cycleInterval IS cycleInterval
        startTime 1.0
        stopTime 0.0
        loop TRUE
    }
    DEF Bouncer Script {
        field    SFFloat bounceHeight IS bounceHeight
        eventIn  SFFloat set_fraction
        eventOut SFVec3f value_changed

        # change 'vrmlscript' to 'javascript' for newer browsers
        url "vrmlscript:
            function set_fraction( frac, tm ) {
                y = 4.0 * bounceHeight * frac * (1.0 - frac);
                value_changed[0] = 0.0;
                value_changed[1] = y;
                value_changed[2] = 0.0;
            }"
    }
    ROUTE Clock.fraction_changed      TO Bouncer.set_fraction
    ROUTE Bouncer.value_changed       TO Ball.set_translation
}



#
#  Bouncing beachballs
#
BouncingBall { cycleInterval 2.0 bounceHeight 3.0 }
Transform {
    translation  2.0 0.0 -2.0
    children BouncingBall { cycleInterval 2.2 bounceHeight 2.5 }
}
Transform {
    translation -2.0 0.0 -2.0
    children BouncingBall { cycleInterval 2.4 bounceHeight 3.5 }
}
```

Criando novos tipos de nós

Uma amostra usando protótipos

[bounce4.wrl]

```
#VRML V2.0 utf8
#
#  bounce4.wrl
#  Bouncing beachball (with PROTO and JavaScript implementing shadows)
#    by David R. Nadeau
#
#  This world illustrates the use of a PROTO to encapsulate a Script
#  node, timer, shape, and routes to create a vertically bouncing
#  beach ball with a shadow.
#
#  The bounce path is based upon the projectile motion equation of
#  physics, constrained to create a cyclic bouncing path with a
#  user-selected maximum bounce height.  Also, there is no friction,
#  drag, or damping.  For an explanation of the script, see 'bounce1.wrl'.
#

WorldInfo {
        title "Bouncing beachball (PROTO and shadows)"
        info [ "Copyright (c) 1997, David R. Nadeau" ]
}

Viewpoint {
    position 0.0 0.6 8.0
    orientation 1.0 0.0 0.0 0.1
}

NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
    speed 2.0
}

DirectionalLight {
    ambientIntensity 0.5
    direction 0.0 -1.0 -0.5
}

#
#  Sky
#
Background {
    skyColor [
        0.0 0.0 1.0,
        0.0 0.5 1.0,
        0.7 0.7 1.0,
    ]
    skyAngle [
        1.371,
        1.571,
    ]
}


#
#  Beach
#
Shape {
    appearance Appearance {
        material Material { }
        texture ImageTexture { url "sand.jpg" }
        textureTransform TextureTransform { scale 10.0 10.0 }
    }
```

```
    geometry IndexedFaceSet {
        coord Coordinate {
            point [
                -50.0 -1.0  50.0,
                 50.0 -1.0  50.0,
                 50.0 -1.0 -50.0,
                -50.0 -1.0 -50.0,
            ]
        }
        coordIndex [ 0, 1, 2, 3 ]
        solid FALSE
    }
}


#
#  Palm trees
#
Transform {
    translation -3.0 -1.0 -10.0
    children [
        DEF Palm Group {
            children [
            # Palm tree - in a billboard so it is never edge-on
                Billboard {
                    children [
                        Shape {
                            appearance Appearance {
                                material NULL  # emissive texturing
                                texture ImageTexture { url "palm.png" }
                            }
                            geometry IndexedFaceSet {
                                coord Coordinate {
                                    point [
                                        -2.5  0.0 0.0,
                                         2.5  0.0 0.0,
                                         2.5 11.25 0.0,
                                        -2.5 11.25 0.0,
                                    ]
                                }
                                coordIndex [ 0, 1, 2, 3 ]
                                texCoord TextureCoordinate {
                                        point [
                                                0.0 0.0,
                                                1.0 0.0,
                                                1.0 1.0,
                                                0.0 1.0,
                                        ]
                                }
                                texCoordIndex [ 0, 1, 2, 3 ]
                                solid FALSE
                            }
                        }
                    ]
                }
            # Fake tree shadow - a black semi-transparent rectangle with
            # a texture map to give it the right shape
                Shape {
                    appearance Appearance {
                        material Material {
                                diffuseColor 0.0 0.0 0.0
```

```
                                transparency 0.5
                        }
                        texture ImageTexture { url "palmsh.png" }
                    }
                    geometry IndexedFaceSet {
                        coord Coordinate {
                            point [
                                -2.5  0.05  2.5,
                                 2.5  0.05  2.5,
                                 2.5  0.05 -2.5,
                                -2.5  0.05 -2.5,
                            ]
                        }
                        coordIndex [ 0, 1, 2, 3 ]
                        texCoord TextureCoordinate {
                                point [
                                        0.0 0.0,
                                        1.0 0.0,
                                        1.0 1.0,
                                        0.0 1.0,
                                ]
                        }
                        texCoordIndex [ 0, 1, 2, 3 ]
                        solid FALSE
                    }
                }
            ]
        }
    ]
}
Transform { translation -5.0 -1.0  -6.0 scale 0.6 0.6 0.6 children USE Palm }
Transform { translation  5.0 -1.0  -9.0 children USE Palm }
Transform { translation 10.0 -1.0 -15.0 children USE Palm }


#
#  Bouncing ball prototype
#
PROTO BouncingBall [
    field SFFloat bounceHeight  1.0
    field SFTime  cycleInterval 1.0
] {
    Group {
        children [
            DEF Ball Transform {
                # animated translation
                children [
                    Shape {
                        appearance Appearance {
                            material Material {
                                ambientIntensity 0.5
                                diffuseColor 1.0 1.0 1.0
                                specularColor 0.7 0.7 0.7
                                shininess 0.4
                            }
                            texture ImageTexture { url "beach.jpg" }
                            textureTransform TextureTransform { scale 2.0 1.0 }
                        }
                        geometry Sphere { }
                    }
                ]
```

```
            }
            DEF BallShadow Transform {
                translation 0.0 -1.9 0.0
                # animated scale
                children [
                    Shape {
                        appearance Appearance {
                            material Material {
                                diffuseColor 0.0 0.0 0.0
                                transparency 0.5
                            }
                        }
                        geometry Cylinder {
                            top TRUE
                            bottom FALSE
                            side FALSE
                        }
                    }
                ]
            }
        ]
    }
    DEF Clock TimeSensor {
        cycleInterval IS cycleInterval
        startTime 1.0
        stopTime 0.0
        loop TRUE
    }
    DEF Bouncer Script {
        field    SFFloat bounceHeight IS bounceHeight
        eventIn  SFFloat set_fraction
        eventOut SFVec3f value_changed
        eventOut SFVec3f shadowScale_changed

        # change 'vrmlscript' to 'javascript' for newer browsers
        url "vrmlscript:
            function set_fraction( frac, tm ) {
                y = 4.0 * bounceHeight * frac * (1.0 - frac);
                value_changed[0] = 0.0;
                value_changed[1] = y;
                value_changed[2] = 0.0;

                xzscale = 1.0 - 0.5 * y / bounceHeight;
                shadowScale_changed[0] = xzscale;
                shadowScale_changed[1] = 1.0;
                shadowScale_changed[2] = xzscale;
            }"
    }
    ROUTE Clock.fraction_changed      TO Bouncer.set_fraction
    ROUTE Bouncer.value_changed       TO Ball.set_translation
    ROUTE Bouncer.shadowScale_changed TO BallShadow.set_scale
}



#
#  Bouncing beachballs
#
BouncingBall { cycleInterval 2.0 bounceHeight 3.0 }
Transform {
    translation  2.0 0.0 -2.0
```

```
    children BouncingBall { cycleInterval 2.2 bounceHeight 2.5 }
}
Transform {
    translation -2.0 0.0 -2.0
    children BouncingBall { cycleInterval 2.4 bounceHeight 3.5 }
}
```

Fornecendo informações sobre o seu mundo

Motivação

Sintaxe: WorldInfo

Fornecendo informações sobre o seu mundo

Motivação

. Depois de ter criado um grande mundo, assine-o!

. Você pode fornecer um título e uma descrição inseridos dentro do arquivo

Fornecendo informações sobre o seu mundo

Sintaxe: WorldInfo

. Um nó `WorldInfo` fornece a informação do título e da descrição de seu mundo

- ***title* - o nome do seu mundo**
- ***info* - qualquer informação adicional**

```
WorldInfo {
    title "My Masterpiece"
    info  [ "Copyright (c) 1997 Me." ]
}
```

Exemplos resumo

Um nó chama animado

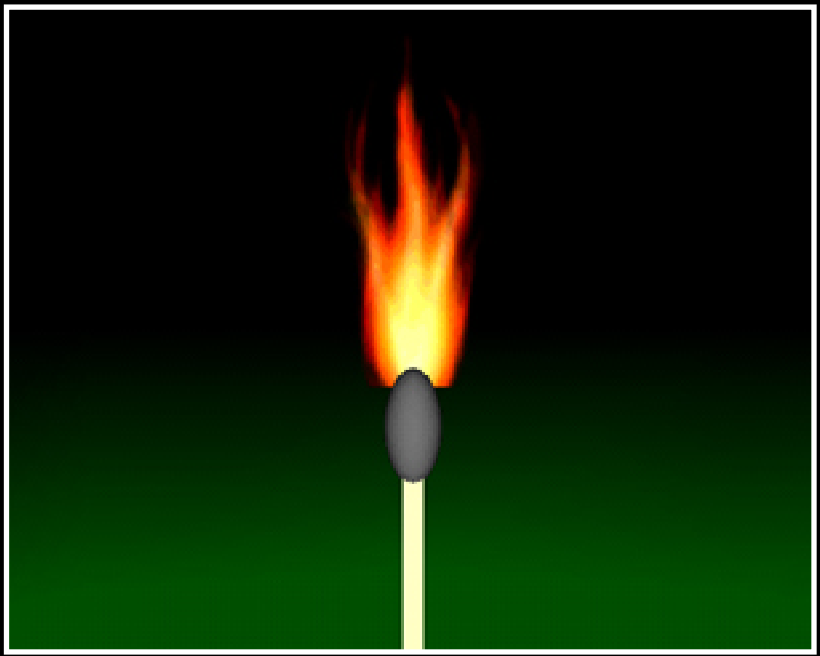
Um nó tocha

Exemplos resumo

Um nó chama animado

. Um nó `Script` troca as texturas entre chamas

. Um `PROTO` encapsula a chama, forma, script e rotas entre os nós `Flames`



[match.wrl]

Exemplos resumo

Um nó tocha

. Um nó `Flame` cria uma chama animada

. Um nó `LOD` seleciona entre as chamas usadas pelas tochas

. `PROTO` encapsula as tochas no nó `Torch`

[columns.wrl]

```
#VRML V2.0 utf8
#
#  match.wrl
#  A Match!
#    by David R. Nadeau
#
#  This world illustrates the use of the 'Flames' node defined in the
#  external file 'flames.wrl'.  That node creates an animated flame.
#  This world uses such flames on the head of a match.
#

Viewpoint {
    position 0.0 0.3 4.0
    description "Entry view"
}

NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight TRUE
}

Background {
    skyColor [
        0.0 0.0 0.0
        0.0 0.0 0.0
        0.0 0.2 0.0
    ]
    skyAngle [
        1.57,
        1.87,
    ]
}


EXTERNPROTO Flames [
    field SFTime cycleInterval
    field SFTime startTime
] "flames.wrl#Flames"

#
#  Match flames
#
Billboard {
    axisOfRotation 0.0 1.0 0.0
    children Flames { }
}


#
#  Match head and stick
#
Transform {
    translation 0.0 -0.2 0.0
    children [
        Transform {
            scale 0.5 1.0 0.5
            children Shape {
                appearance Appearance {
                    material Material {
                        diffuseColor 0.3 0.3 0.3
                    }
                }
```

```
                geometry Sphere { radius 0.3 }
            }
        }
        Transform {
            translation 0.0 -1.2 0.0
            children Shape {
                appearance Appearance {
                    material Material {
                        diffuseColor 0.8 0.7 0.5
                    }
                }
                geometry Box { size 0.1 2.0 0.1 }
            }
        }
    ]
}
```

Extensões misturadas

Extensões

Usando o arquivo de formato binário

Usando o arquivo de formato binário

Usando a interface externa de autoria - external authoring interface

Usando a interface externa de autoria - external authoring interface

Usando um framework multiusuário

```
#VRML V2.0 utf8
#
#  columns.wrl
#  Columns and torches
#    by David R. Nadeau
#
#  This world uses an EXTERNPROTO to reference an externally defined
#  'Torch' node #  with an animated flame.  That torch is then placed
#  on a column, which is in turn instanced multiple times to form a
#  double row of columns.  A glowing window at the end creates a goal
#  for the column row... whatever it is.
#

EXTERNPROTO Torch [ ] "atorch.wrl#Torch"

WorldInfo {
    title "Columns and torches"
    info [ "Copyright (c) 1997, David R. Nadeau" ]
}

DEF Entry Viewpoint {
    position 0.0 1.6 10.0
    description "Entry view"
}

NavigationInfo {
    type [ "WALK", "ANY" ]
    headlight FALSE
}


#
#  Background and fog (both black)
#    These forces the lighting to dim in the distance, making the
#    world seem more real
#
Background {
    skyColor [ 0.0 0.0 0.0 ]
}

Fog {
    color 0.0 0.0 0.0
    fogType "LINEAR"
    visibilityRange 20.0
}


#
#  Lights - roughly one for each pair of torches on columns
#
PointLight {
    color 1.0 1.0 1.0
    ambientIntensity 0.0
    intensity 1.0
    location 0.0 0.5 -6.0
}
PointLight {
    color 1.0 0.8 0.5
    ambientIntensity 0.0
    intensity 1.0
    location 0.0 0.5 0.0
}
```

```
PointLight {
    color 0.8 0.0 0.0
    ambientIntensity 0.0
    intensity 1.0
    location 0.0 0.5 6.0
}


#
#  Columns
#
DEF ColumnRow Transform {
    translation -1.5 0.0 0.0
    children [
        DEF ColumnAndTorch Transform {
            rotation 0.0 1.0 0.0  1.571
            children [
                DEF Column Transform {
                    scale 0.5 0.7 0.5
                    children Inline { url "column.wrl" }
                }
                Transform {
                    translation 0.0 2.0 0.15
                    scale 0.5 0.5 0.5
                    children Torch { }
                }
            ]
        }
        Transform { translation 0.0 0.0 -9.0 children USE Column }
        Transform { translation 0.0 0.0 -6.0 children USE ColumnAndTorch }
        Transform { translation 0.0 0.0 -3.0 children USE Column }
        Transform { translation 0.0 0.0  3.0 children USE Column }
        Transform { translation 0.0 0.0  6.0 children USE ColumnAndTorch }
        Transform { translation 0.0 0.0  9.0 children USE Column }
    ]
}
Transform {
    rotation 0.0 1.0 0.0 3.14159
    children USE ColumnRow
}


#
#  Window
#
Transform {
    translation 0.0 1.5 -9.0
    scale 0.5 0.5 0.5
    children [
        Shape {
            appearance Appearance {
                material NULL  # emissive texturing
                texture ImageTexture { url "window.jpg" }
            }
            geometry IndexedFaceSet {
                coord Coordinate {
                    point [
                        -1.5 0.0 0.0,
                         1.5 0.0 0.0,
                         1.5 3.0 0.0,
                        -1.5 3.0 0.0,
```

```
                    ]
                }
                solid FALSE
                coordIndex [ 0, 1, 2, 3 ]
                texCoord TextureCoordinate {
                        point [
                                0.0 0.0,
                                1.0 0.0,
                                1.0 1.0,
                                0.0 1.0,
                        ]
                }
                texCoordIndex [ 0, 1, 2, 3 ]
            }
        }
    ]
}


#
#  Floor
#    Use an elevation grid so that we get a grid of polygons that will
#    be individually lit.  This insures that we get bright spots where
#    the lights are, which adds realism and depth to the world.
#
Transform {
    translation -3.0 0.0 -10.0
    children Shape {
        appearance Appearance {
            material Material { ambientIntensity 0.5 }
            texture ImageTexture { url "stone2.jpg" }
            textureTransform TextureTransform { scale 4.0 15.0 }
        }
        geometry ElevationGrid {
            xDimension 8
            zDimension 15
            xSpacing 1.0
            zSpacing 2.0
            height [
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0,
            ]
            solid FALSE
        }
    }
}
```

Extensões misturadas

Extensões

- Várias extensões VRML estão em desenvolvimento
  - Arquivo de formato binário
  - Interface de autoria externa
  - Framework multiusuário

Extensões misturadas

Usando arquivo de formato binário

. O arquivo de formato binário habilita arquivos menores para downloads mais rápidos

. O arquivo de formato binário inclui:

- **Representação binário de nós e campos**
- **Suporte para protótipos**
- **Compressão da geometria**

Extensões misturadas

Usando o arquivo de formato binário

. A maioria dos autores trabalharão com construtores de mundo que exportam diretamente arquivos VRML binários

. Textos VRML dos autores serão compilados ao formato binário

. Conversores quando tornarem-se disponíveis traduzirão para VRML

- **Os comentáriors serão perdidos pela tradução**
- **Os nós `WorldInfo` serão mantidos**

Extensões misturadas

Usando uma interface de autoria externa - external authoring interface

- **Os programas escritos em nós `Script` são internos**
  - **Dentro do mundo**
  - **Conectado por rotas**
- **Programas scripts *externos* podem ser escritos em java usando uma interface de autoria externa - *External Authoring Interface* (EAI)**
  - **Fora do mundo, em uma página HTML**
  - **Não é necessário usar rotas!**

Extensões misturadas

Usando um framework multiusuário

. Várias extensões estão em desenvolvimento para criar um framework para mundos multiusuários

- **Compartilhando objetos e espaços**
- **Pilotando objetos (como avatares)**
- **Descrições de avatares comuns**

Extensões misturadas

Usando uma interface de autoria externa - external authoring interface

. Uma típica pagina Web contém:

- **Texto HTML**
- **Um plug-in VRML encaixado no browser**
- **Um applet Java**

. O EAI habilita o applet java "conversar" com o browser VRML

. O EAI não é parte do VRML standard (ainda), mas é largamente suportado

- **Verifique as notas da versão do seu browser para suporte ao EAI**

Conclusão

Cobrimos

Cobrimos

Onde encontrar mais

Introdução ao VRML 97

Conclusão

Cobrimos

- **Nós cobrimos:**
  - **Construimos formas primitivas**
  - **Construimos formas complexas**
  - **Transladamos, rotacionamos e mudamos a escala de formas shapes**
  - **Controlamos aparência**
  - **Agropamos formas**
  - **Animamos transformações**
  - **Interpolamos valores**
  - **Ações sensíveis ao observador**

Conclusão

Cobrimos

- **Nós cobrimos:**
  - **Controlamos textura**
  - **Controlamos graduação de cor**
  - **Adicionamos luzes**
  - **Adicionamos backgrounds e neblina**
  - **Controlamos detalhes**
  - **Controlamos a visão**
  - **Adicionamos sons**
  - **Detetamos o visor**
  - **Usamos e escrevemos programas scripts**
  - **Construímos novos tipos de nós**

Conclusão

Onde encontrar mais

- **Especificação VRML 2.0**
  http://vag.vrml.org/VRML2.0/FINAL
- **Especificação VRML 97**
  http://vrml.sgi.com/moving-worlds
- **O Repositório VRML**
  http://www.sdsc.edu/vrml

Conclusão

Introdução ao VRML 97

Até o Java 3D!