Desenvolvendo AV com uso de X3D e VRML

Alexandre Cardoso
Luciano P. Soares

www.alexandre.eletrica.ufu.br

Estrutura do Curso

- Introdução
- Grafo de Cena
- Codificação VRML
  - Formas Primitivas
  - Transformações Geométricas
  - Reutilização
  - Iluminação e Cenários
  - Animações
  - Sons e Filmes
  - VRML e JavaScript

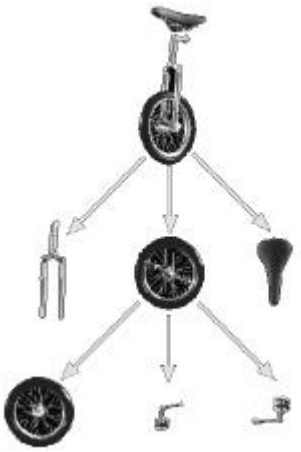
Estrutura do Curso

- X3D
  - Origem do X3D
  - Components e Profiles
  - Codificação X3D
- Transição do VRML para o X3D
  - Principais mudanças
  - Formas de converter os conteúdo

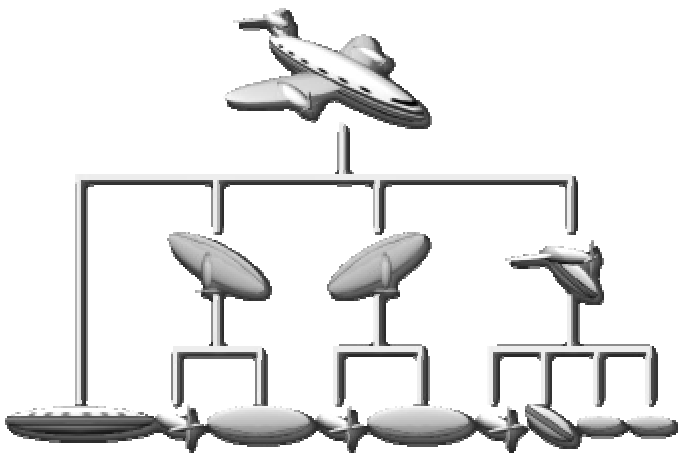
Grafos de Cena

- Definição hierárquica de componentes
- Definem-se as partes
- Definem-se transformadas para prover a união das partes
- Nomeia-se as partes e o todo.

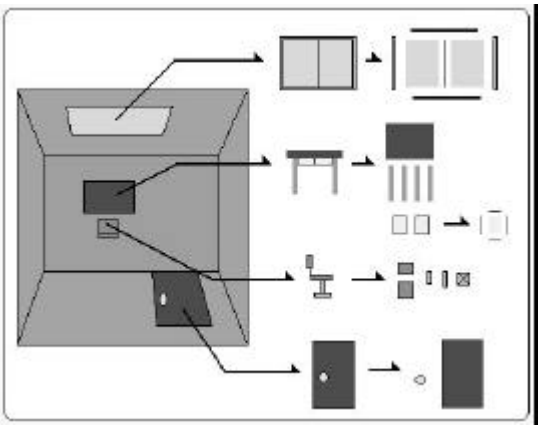
Grafos de Cena - Exemplos



Grafos de Cena - Exemplos



Grafos de Cena - Exemplos



Codificação VRML - introdução

- **Arquivo VRML é um arquivo texto - não tem o que Compilar!!!**
- **Para editar: qualquer editor de textos, indicado: *VRMLPad***
- **Instalar o *plug-in* para VRML, indicado: *CosmoPlayer* (ou Cortona)**

Dicas iniciais - estrutura básica

- **Abra o editor de textos...**
- **Insira o cabeçalho : #VRML v2.0 utf8**
- **Edite o seu mundo virtual, inserindo as formas que deseja**
- **Salve no formato ".wrl"**
- **Abra o *IExplorer* e carregue!**

Descrição de Formas

- **Descrição de uma forma:**
  - **Geometria**
  - **Aparência**
- **Geometria:**
  - **Box, Cone, Cylinder, Sphere**
  - **Text**
  - **Coordinate, IndexedLineSet, IndexedFaceSet**
  - **ElevationGrid, Extrusion**

Descrição de Formas

Estrutura da descrição:

```
Shape {
        appearance Appearance {}
        geometry Box {}
}
```

➉Aqui é descrita a geometria do nó

```
Box {
                size  0.2  0.2  0.2}
```


Formas Básicas

- Sphere **{**
  radius 0.3}
- Cylinder**{**
  height 2.0
  bottom TRUE
  side TRUE
  radius 3.0}
- Cone {
  bottom TRUE
  bottomRadius 2.0
  height 3.0
  side TRUE }

Formas Básicas

- **Text {**

```
string "texto"
fontStyle FontStyle        {
justify "MIDDLE"
size 2.0
spacing 0.1
style "BOLD"}
length 1.0          }
```


Descrevendo a Aparência

- **Aparência:**
  - **Definição de Cores como propriedade da Forma**
  - **Aplicação de Texturas**
- **Cores:**

```
appearance Appearance {
        material Material {
                ambientIntensity 1.0
                diffuseColor 1.0 1.0 1.0
                specularColor 0.3 0.3 0.3
                emissiveColor 0.3 0.3 0.4
                transparency 1.0 }}
```

➓ ***exemplo***

Aparência - Texturas:

Texturas:

```
texture ImageTexture {url ""}
```

- **Endereço da imagem que será aplicada como textura**

```
textureTransform TextureTransform{
        scale 3.0 3.0
        center 0.0 0.0
        rotation 0.0
        translation 0.0 0.0}}
```

exemplo1 - exemplo2

- **Fatores de correção da textura**

Transformações Geométricas

Transformações:

- **Translação**
- **Rotação**
- **Escala**

Nó que suporta as transformações: *transform*

```
Transform {
        translation     0.0 0.0 0.0
        rotation 1.0 0.0 0.0 1.57
        scale 2.0 2.0 2.0
        children [ ]  }
```

- **Lista dos nós que sofrerão as transformações definidas**

Alterando Posição

Translação:

Pode ser nos eixos x, y e/ou z:

***translation x y z* (valores em real)**

Cuidado com números exagerados: risco de perder a forma no cenário definido

exemplo

Rotacionando a forma

rotação:

Pode ser em torno de qualquer eixo: x, y ou z:

***rotation x y z ang* (valor em rad)**

Cuidado com o eixo que escolhe

exemplo

Escala

escala:

- **deve ser usado um fator de escala: o mesmo para os três eixos**
- ***scale fat_x fat_y fat_z* (valores em real)**
- **Observe que a posição pode influir na escala**

exemplo

Reutilização de formas

Eventualmente, é necessário reutilizar uma forma ou uma aparência, já definida

Neste caso, a utilização dos termos DEF e USE permite que seja efetivada a reutilização

DEF *nome*

USE *nome*

exemplo

Reutilizando arquivos

Deseja-se reutilizar um dado arquivo, que define uma forma ou um conjunto de formas

Neste caso, é preciso indicar o caminho "url" do arquivo, que pode ser local ou arquivo da internet (necessita o caminho completo: http://...)

use o *inline*

exemplo

Fundos - cenários

Veja:

ex. anterior com fundo adequado

Nó que define fundos: background

permite usar imagens para composição ou a definição de cores, exclusivamente

é possível trocar fundos dinamicamente:

exemplo

Iluminação

Fontes de Luz em VRML

- **headlight**
- **Pointlight**
- **Directionallight**
- **Spotlight**

A importância de uma boa iluminação no realismo

ausência de sombras!!!!

Barreiras físicas X fontes de luz

Pointlight

Luz que emana de um ponto particular no espaço e se espalha igualmente em todas as direções

pincel de luz: divergente, sem que ocorra uma forma de direcionamento específica

variáveis importantes:

- **atenuação**
- **cor**
- **intensidade**
- **localização**
- **raio**

Exemplo1

Exemplo2

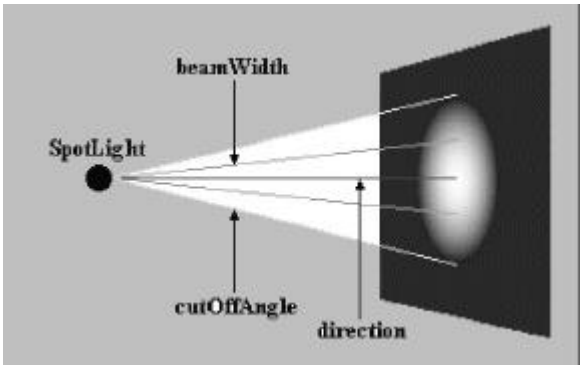
Spotlight

Luz que emana de um ponto particular no espaço e se espalha, na forma de um cone, em uma direção específica

variáveis importantes:

- **atenuação**
- **cor**
- **intensidade**
- **localização**
- **raio**
- **direção**
- **ângulo de abertura**

Spotlight



➓**exemplo**

Directionallight

Luz que parte de uma fonte de luz, segundo um pincel paralelo em uma única direção

Simula um projetor

variáveis importantes:

- **cor da luz**
- **direção**
- **intensidade**
- **interação com a iluminação ambiente**

exemplo

Exercícios

1. Simule um exemplo, onde, aplicando textos, seja apresentado um dipolo elétrico

2. Monte um poste de luz, que gere iluminação adequada

3. Aplicando texturas, simule um muro, onde foi feita uma pixação. Apresente uma abertura para este muro

4. Componha um sistema que, adequadamente, simule um sistema solar

Animando os mundos

Uma animação presssupõe:

- **tempo**
- **alteração de elementos no tempo**
- **disparo eventual (?)**

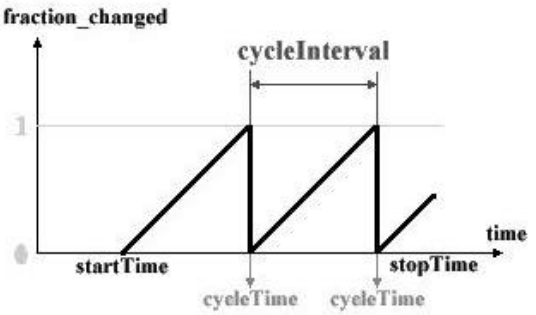
nós relacionados

- **PositionInterpolator**
- **OrientatorInterpolator**
- **timesensor**
- **sensores**

Sensores

TimeSensor: é basicamente, um timer

- **variáveis importantes:**



Sensores

VisibilitySensor: relativo a um volume no formato de uma caixa (invisível) que envia eventos quando o usuário entra ou sai dela

- **variáveis importantes:**
  - **centro da caixa**
  - **tamanho da caixa**
  - **se está ou não habilitado**

ProximitySensor: similar ao VisibilitySensor, exceto por parâmetros de saída

exemplo

Sensores

Collision

- **controla a colisão do navegante com as formas que estão definidas dentro de seu campo *children***
- **variáveis importantes:**
  - **elementos definidos dentro do nó**
  - **tamanho da região de colisão**
  - **collide - TRUE ou FALSE**

exemplo

Montando animações

O contador de tempo:

```
DEF     cont_tempo TimeSensor {
                startTime 0.0
                loop TRUE
                cycleInterval 5.0
        },
```

Este nó envia frações de tempo a outro nó, quando há uma definição de rota adequada
Sem a definição de rota, não há qualquer informação de saída deste nó
é preciso inserir o controlador da animação

Montando animações

O controlador de animação (posições, neste caso):

```
DEF     cont_pos PositionInterpolator {
                key     [ 0.0 0.2 0.4 0.6 0.8 1.0]
                keyValue [
                        0.0 0.0 0.0
                        1.0 0.0 0.0
                        1.0 1.0 0.0
                        0.0 1.0 0.0
                        0.0 2.0 0.0
                        1.0 2.0 0.0
                ]  }
```

Montando animações

Resta agora, definir a forma que será animada:

```
DEF   esfera Transform {
                        children Shape{
                        appearance Appearance {
                                material Material { }
                        }
                        geometry Sphere  {
                                radius 0.3}
                        }
                }
```


Montando animações

Finalmente, são feitas as rotas que controlam a animação, de forma que eventos de um nó sejam enviados a outro nó - troca de mensagens entre os objetos do cenário virtual:

ROUTE cont_tempo.fraction_changed TO cont_pos.set_fraction

ROUTE cont_pos.value_changed TO esfera.set_translation

exemplo1

exemplo2

Nó que recebe

Nó que envia

Montando animações

A animação não necessita ser de posição, pode ser, por exemplo, de escala do elemento
Neste caso, os fatores que são utilizados no elemento de controle devem ser mapeados para valores de escala:

ROUTE cont_esc.value_changed TO elemento.set_scale

exemplo

Montando animações

Para animar a rotação de uma forma, é necessário definir em torno de que eixo a mesma ocorrerá e será usado outro nó:

Orientator Interpolator

Neste caso, as rotas devem alterar a rotação de um nó transform:

exemplo

Sons e Filmes

A inserção de sons e de filmes pode tornar o ambiente virtual mais realístico

- é possível inserir som ativado pelo toque, aproximação ou movimentação do usuário

- é possível inserir um som ambiente - música de fundo

- permite-se: midi, wav, mpeg

- não se permite o mp3

Adicionando Som

Nós utilizados para sons e filmes:

AudioClip - descrição de uma fonte de som

MovieTexture - inserção de filmes

Sound - emissores de sons

```
AudioClip {
        url         ""
        startTime  0.0
        stopTime  0.0
        loop FALSE
        description          ""
        pitch 1.0 }
```

Adicionando Som

```
Sound {
        source AudioClip {}
        direction 0.0 0.0 0.0
        location 0.0 0.0 0.0
        spatialize TRUE
        maxBack        10.0
        maxFront 10.0
        minBack        0.0
        minFront 0.0
        intensity 1.0
        priority 0.0
}
```

➓**exemplo**

Links para outros AV ou sites

Dentro de um ambiente, podem ser adicionados links para outros ambientes virtuais ou outras páginas web (a critério do desenvolvedor)

Nó relacionado: Anchor

nó Anchor é um nó de agrupamento

formas que servem para ativar os links devem estar declaradas no campo 'children' deste nó

pode ser descrito o link para qual será levado o navegador - aparece na barra de status do browser

Adicionando Links

Anchor {

url ""
description ""
parameter []
children []

}

- **Permite, dinamicamente:**
  - **addChildren**
  - **removeChildren**

exemplo

VRML e JavaScript

Necessidades:

- **animações mais complexas**
- **adicionar e remover elementos de um ambiente virtual**
- **controle matemático de comportamento**
- **aplicações: games etc**
- **podem ser concebidas em JAVA**

Introdução a Scripts

O nó relacionado: Script

eventos de entrada

eventos de saída

campos

região de Script - funções ativadas pelos eventos de entrada

que geram os eventos de saída ou modificam parâmetros

relativos aos campo definidos

Script

```
Script{
        eventIn   MFInt32 entrada
        eventOut    SFFloat       saida
        field       MFString      texto     "valor inicial"
        url         "javascript:
                    //descrição das funções
                    entrada (p,s) {
                    // gera a saída ou modifica objetos (campos)
                    }
        "
}
```

Script

Característica básica de um *Script*:

```
Script {
# definição de campos, suas naturezas e valores iniciais:
field SFBool booleano1 TRUE
# definição de eventos de entrada e sua natureza:
eventIn  SFBool entrada
# definição de eventos de saída e sua natureza:
eventOut SFBool          saida
url      "javascript:
         function entrada(param){
         // processamento e determinação da saída
                  saida = param;} " }
```


Script - exemplo

```
DEF SCRIPT Script {
  eventIn       SFTime  touchTime
  field         SFInt32  touchCount     0
  field         SFInt32  eventCount     0
  field         MFString        texto           ""
  eventOut      MFString        outputString
                url "javascript:
                 function initialize() {
                outputString = new MFString(",");
                texto = new MFString(",");
                outputString[0] = 'Ready...';
    }
```

Script - exemplo

```
function touchTime (value, time) {
        touchCount++;
        texto[0] = 'touchCount: ' + touchCount.toString();
      }
      function eventsProcessed() {
        eventCount++;
        texto[1] = 'eventCount: ' + eventCount.toString();
        outputString = texto;
      }
  "
}
```

exemplo

Acesso a propriedades e métodos

Um objeto em VRML:

- **propriedades**
- **métodos**

acesso a propriedades:

myObject.x

acesso a métodos

myObject.x = 3;

result = myObject.multiply(4);

Script - exemplo 2

```
DEF controla Script{
        eventIn  SFTime  entrada_r
        eventIn  SFTime   entrada_t
        field SFNode  no          USE     forma
        url       "javascript:
                  //descrição das funções
                        function entrada_r (ts)     {
                                no.rotation[0] = 1.0;
                                no.rotation[1] = 0.0;
                                no.rotation[2] = 0.0;
                                no.rotation[3] = 1.57;
                        }
```


Script - exemplo2

```
function entrada_t (ts)     {
                no.translation[0]  = 1.0;
          no.translation[1] =  0.0;
                no.translation[2] =  0.0;
        }
        ”
}
```

E as rotas:

```
        ROUTE  toque_r.touchTime TO controla.entrada_r
        ROUTE  toque_t.touchTime          TO  controla.entrada_t
```

exemplo

Links para visitar

Repositório de formas :

3Dcafe

VRML na Educação

WebTOP
http://www.ecoguildcom/sgraves/aace/aace_99/
http://id.mind.net/~zona/mstm/physics/mechanics/mechanics.html
http://www.hitl.washington.edu/projects/knowledge_base/edapps.html
http://physics.syr.edu/courses/vrml/electromagnetism/

Links para visitar

- **VRML/3D Multi-User Browser Resources**
- **http://www.realism.com/vrml/example.html#Create Geometry**
- **http://vrmlworks.crispen.org/**
- VRML, X3D e XML - recomendado:
  - **http://www.realism.com/x3d/**
- **Tutoriais:**
  - **Floppy´s**
- **Site do autor**

O X3D

- X3D - Extensible 3D
  - Formato Universal de Transferência de dados 3D
  - Um padrão aberto
  - Fácil conversão de arquivos VRML
  - Fácil de entender e modelar
  - Portável entre plataformas
  - Fácil de ensinar e programar

X3D & ISO

- Formato definido pela ISO
- VRML 2.0 ISO/IEC 14772-1:1997 (aka VRML97)
- X3D ISO/IEC FDIS 19775:200x
- Não tem royalties associados
- A ISO publicará a especificação para o público

Desenvolvimento do X3D

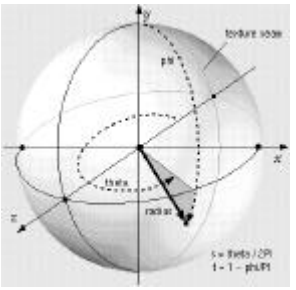
- Conjunto de modelos para conformidade
  - http://www.web3d.org/x3d/content/examples/Conformance/index.html
- Conjunto de exemplos na WEB
  - http://www.web3d.org/x3d/content/examples/
- Navegadores X3D em desenvolvimento
  - http://www.web3d.org/applications/tools/

Abordagem do X3D (exemplo)

Sphere : X3DGeometryNode {

SFNode	[in,out]	metadata	NULL [X3DMetadataObject]	
SFFloat	[ ]	radius	1	(0,8)
SFBool	[ ]	solid	TRUE	

}



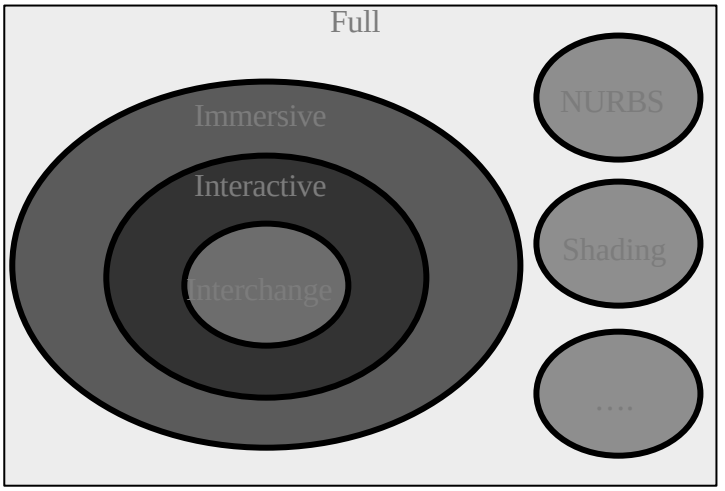
Components e Profiles

- Definem, extensibilidade e o conjunto de serviços que o conteúdo dos usuários necessita.
- Um componente (*component*) define uma específica coleção de nós e um conjunto de níveis.
- Um nível define os nós, seleção de campos do componente.
- Um perfil (*profile)* é uma coleção de componentes para um específico nível de suporte.

Components e Profiles (na prática)

- Todos os arquivos X3D requerem a definição do perfil
- Podem ser criados novos Componentes
- Uma nova versão implica mais funcionalidade que as versões anteriores.
- Um navegador não precisa suportar todos os Profiles, Components e Levels.

Profiles



Codificação X3D

- Clássica – Muito semelhante ao VRML97
- XML – Técnica atual

A sintaxe do VRML 97 é baseada no Open Inventor

O XML (Extensible Markup Language), é asintaxe dominante na Web

Uso de ferramentas XML para edição

Document Type Definition (DTD) para X3D

Também existe um Schema para X3D aceito

Binária – Em desenvolvimento

X3D Clássico

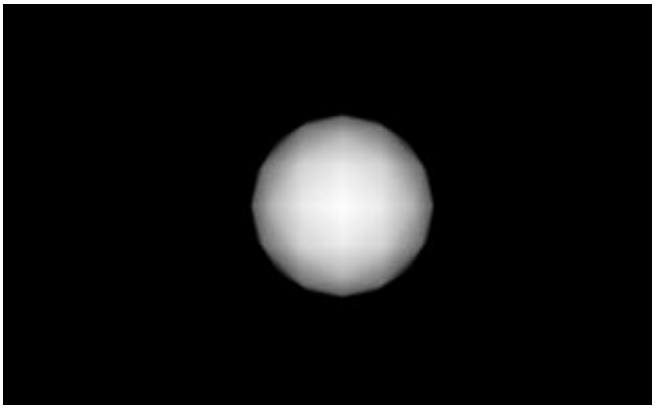
```
#X3D V3.0 utf8
Profile Immersive

NavigationInfo {
type [ "ANY" ]
}
Transform {
children [
   Shape {
        appearance Appearance {
               material Material {
                      diffuseColor 1.0 1.0 1.0
               }
        }
        geometry Sphere {}
   }
]
}
```


X3D XML

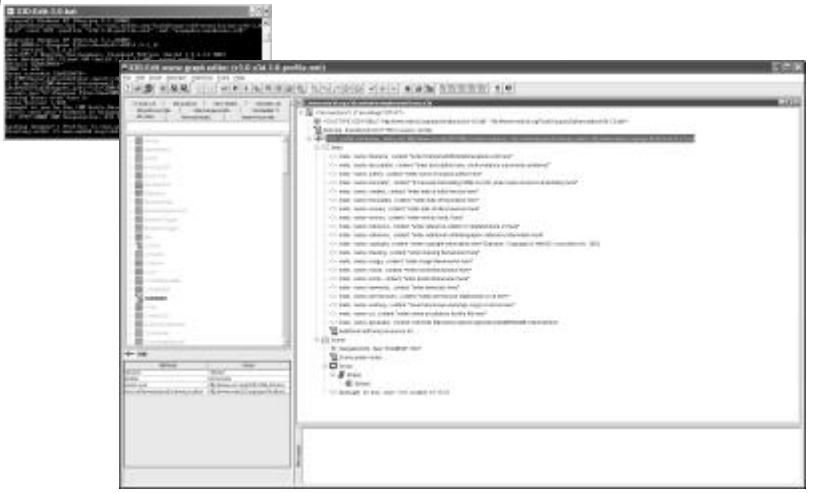
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "http://www.web3d.org/specifications/x3d-
   3.0.dtd">
<X3D profile="Immersive">
<Scene>
   <NavigationInfo type="ANY"/>
   <Transform>
        <Shape>
               <Appearance>
                      <Material diffuseColor=”1 1
   1”/>
               </Appearance>
               <Sphere/>
        </Shape>
   </Transform>
</Scene>
</X3D>
```

Exemplo

- Esfera Branca



Edição de arquivos X3D (X3D-Edit)

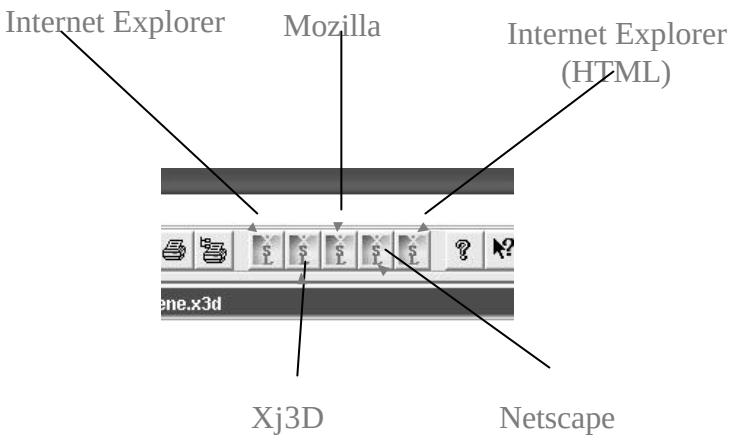


IBM Xeena

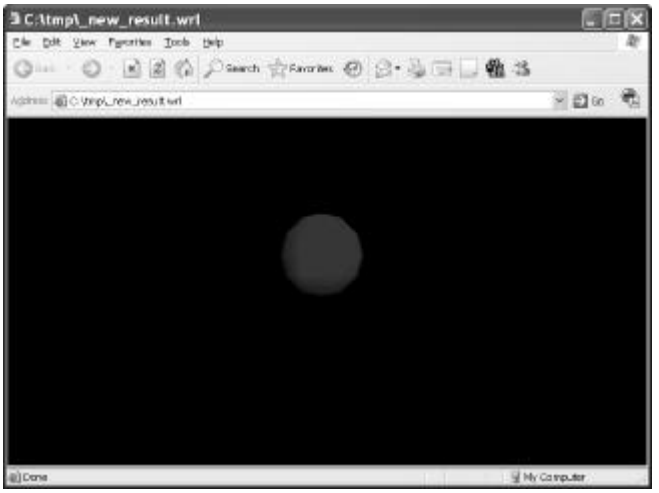
Exercício

- Criando uma esfera saltitante.

Visualizando conteúdo

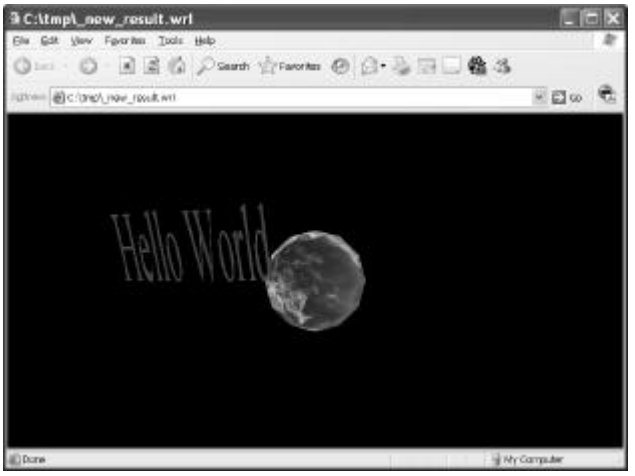


Resultado



Hello World X3D

Resultado



Working Groups (avançando os recursos)

- X3D Specification
- X3D Shaders
- GeoSpatial
- H-Anim
- X3D Source
- CAD
- Medical
- VizSim

Transição do VRML para o X3D

Principais mudanças:

- Maior precisão com a iluminação e modelo de eventos e a troca de certos nomes de campos para uma maior consistência
- Capacidades do grafo de cena expandidas
- Modelo de programação de aplicações revisado e unificado
- Multiplos formatos de codificação, descrevendo o mesmo modelo abstrato, incluindo XML
- Arquitetura modular permitindo uma faixa de níveis para serem adotados e suportados por diversos tipos de mercados
- Estrutura da especificação expandida

Transição do VRML para o X3D

- Os arquivos estão agora estruturados para definir as capacidades necessárias como parte do cabeçalho
- Externprotos somente para definir conteúdo externo ao arquivo X3D.
- Nomes de acesso para campos mudaram de eventIn, eventOut, field e exposedField, para inputOnly, outputOnly, initializeOnly e inputOutput, respectivamente.
- Um nome DEF não pode ser multidefinido mais.
- Todo conteúdo de leitura é desacoplado. (scripts, texturas, sons, inlines, externprotos)
- Rigorosa definição de conjunto de definições de tipos abstratos para nós.

Tabela de Extensões

Os arquivos X3D devem seguir as seguintes regras:

Codificação X3D	Extensão de arquivo	Extensões Compactadas	Tipo MIME
Clássico VRML	.x3dv	.x3dvz, .x3dv.gz	model/x3d+vrml
XML	.x3d	.x3dz, .x3d.gz	model/x3d+xml
Binário	.x3db	.x3dbz, .x3db.gz	model/x3d+binary

* os tipos MIME ainda estão sendo aprovados

Conversão

Conversão entre VRML97 e o “X3D clássico”:

#VRML V2.0 utf8

para

#X3D V3.0 utf8
Profile Immersive

API

- O X3D tem uma única interface de programação de aplicações.
- ECMAScript na verdade é um sistema Java interpretado.

Mais Informações

- http://www.web3d.org
- http://www.lsi.usp.br/~lsoares/x3d/faq.html
- http://www.alexandre.eletrica.ufu.br