

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO
DO RIO GRANDE DO NORTE
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**REPRODUÇÃO VIRTUAL DA IGREJA DE SANTO ANTÔNIO: Um monumento
histórico na cidade do Natal**

Ana Lygia Bezerra Rodrigues

**NATAL/RN
2005**

ANA LYGIA BEZERRA RODRIGUES

**REPRODUÇÃO VIRTUAL DA IGREJA DE SANTO ANTÔNIO: Um monumento
histórico na cidade do Natal**

Monografia apresentado ao curso de Graduação em Bacharelado em Sistemas de Informação da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador:

Luiz Gonzaga Damasceno, MSC.

**NATAL/RN
2005**

RB-26.556

FARN
BIBLIOTECA
Nº Reg. <u>1000436</u>
<u>17 / 05 / 06</u>

M
004.4'275
R63J7
2005

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Rodrigues, Ana Lygia Bezerra

Reprodução virtual da igreja de Santo Antônio: um momento histórico na cidade do Natal / Ana Lygia Bezerra Rodrigues. – Natal: FARN, 2005.

49 f.

Orientador: Luiz Gonzaga Damasceno (M.Sc.)

Monografia (Graduação do Bacharelado em Sistemas de Informação) – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

I. Computação Gráfica – Monografia. 2. Realidade Virtual – Monografia. I. Título.

RN/FARN/BC

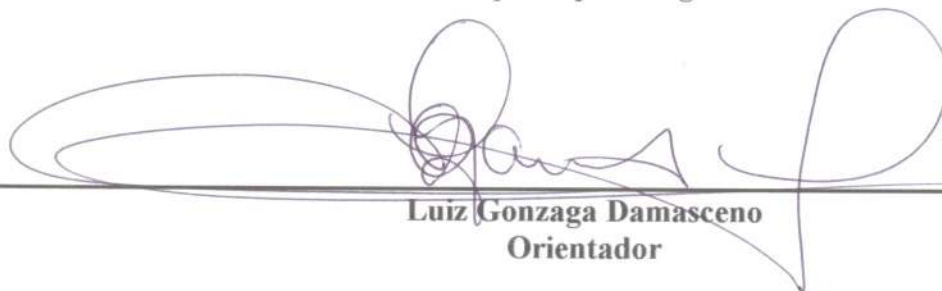
CDU 004.4'275

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RIO GRANDE DO
NORTE
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**REPRODUÇÃO VIRTUAL DA IGREJA DE SANTO ANTÔNIO: Um monumento
histórico na cidade do Natal**

Ana Lygia Bezerra Rodrigues

Monografia apresentada e aprovada em 17/12/05 de Dezembro de 2005 pela banca
examinadora composta pelos seguintes membros:



Luiz Gonzaga Damasceno
Orientador



Nilda Maria Teixeira de Cerrqueira
Membro Examinador



Ytalo Rosendo do Amaral
Membro Examinador

Natal, 17 de Dezembro de 2005.

AGRADECIMENTOS

Em especial a Deus, que me propiciou força e sabedoria em momentos de incertezas, tristezas, desânimo, o senhor me deu forças para continuar e superar todos os obstáculos.

Aos meus pais, José Rodrigues e Teresinha, que sempre estiveram ao meu lado em todos os momentos da minha vida, me apoiando, sendo meu porto seguro, e é com profundo amor e admiração que agradeço a vocês e lhes dedico mais essa conquista.

A minha irmã Daniela, por sempre me estimular nessa árdua busca dos meus objetivos, e ao meu irmão Carlos Eduardo, que mesmo distante, sempre me deu forças e incentivo.

Ao meu namorado Flamaliel, pelo apoio, pelo estímulo, pela presença constante, e as palavras de incentivo em momentos de dúvidas.

Ao meu tão querido orientador Damasceno, pela sua dedicação, carinho, em especial esse último ano, por ter se tornado referência não só pela sua experiência profissional, mas também pela pessoa que é um ser humano sempre disposto a ajudar em todos os momentos, minha eterna gratidão.

Aos professores que nos transmitiram seus conhecimentos e experiências profissionais e de vida, com muita dedicação, e aos meus colegas, companheiros dessa jornada, pela ajuda mútua, que tantas vezes suavizou os obstáculos da caminhada.

SUMÁRIO

1. INTRODUÇÃO	11
1.1 – OBJETIVOS	13
1.1.1 Objetivo Geral	13
1.1.2 Objetivos Específicos	13
1.2 – PROBLEMA DE PESQUISA	14
1.3 – JUSTIFICATIVA	15
1.4 – METODOLOGIA	16
1.4.2 Classificação da Pesquisa	16
2. REFERENCIAL TEÓRICO	17
2.1 – O QUE É E PARA QUE SERVE COMPUTAÇÃO GRÁFICA	17
2.2 – CORES E LUZ	20
2.3 – ANIMAÇÃO	22
2.4 – SENSORES	24
2.5 – O USO DA FERRAMENTA VRML	25
2.5.1 Arquivo VRML	29
2.5.2 Os Objetos	30
2.5.3 As transformações	31
2.6 – NODOS OU NÓS.....	32
2.7 – MATERIAL E APPEARANCE.....	33
2.8 – USO DO DEF E USE.....	34
2.9 – GRUPOS.....	35
2.10 – TEXTURAS	36
2.10.1 Nó image texture	37

2.10.2	Nó texture transform	37
2.11	– VIEWPOINT	38
3.	ANÁLISE DE RESULTADOS	39
3.1	– DESENVOLVIMENTO	39
4.	CONCLUSÃO	45
	REFERÊNCIAS	46
	ANEXOS	49

LISTA DE TABELAS

Tabela 1 – Sensores	23
---------------------------	----

LISTA DE FIGURAS

Figura 1 – Cubo	28
Figura 2 – Cone	28
Figura 3 – Esfera	28
Figura 4 – Cilindro	28
Figura 5 – Regra da mão direita para eixos 3D	29
Figura 6 – Criação da esfera em branco	31
Figura 7 – Utilização do DEF e USE	33
Figura 8 – Exemplo de agrupamento	34
Figura 9 – Visão do interior da igreja através do viewpoint	36
Figura 10 – Utilização das formas básicas do VRML	38
Figura 11 – Uso de textura retirada diretamente da fotografia do local	39
Figura 12 – Utilização de textura no teto da parte interna da Igreja	39
Figura 13 – Textura do teto da parte externa da igreja	40
Figura 14 – Varandas construídas com texturas retiradas das fotografias.....	40
Figura 15 – Parte da Frente da Igreja	41
Figura 16 – Utilização de sensores	42

RESUMO

Esse trabalho tem por objetivo descrever no mundo virtual, um ambiente já existente no mundo real, que é a Igreja do Galo, reproduzida por técnicas da Computação Gráfica, dando o maior nível de realidade possível nessa reprodução. A ferramenta a ser utilizada é o VRML (Linguagem para Modelagem em Realidade Virtual), onde esta irá dispor de todos os componentes necessários para que se possa atingir o objetivo esperado, através de recursos como animação, texturas, escala e outras funções que esta linguagem dispõe. Primeiramente serão dados alguns conceitos essenciais para um melhor entendimento desse trabalho, como características da linguagem usada, estruturas dos arquivos, etc. e ao fim o resultado da criação da Igreja no mundo tridimensional.

Palavras-Chaves: Computação Gráfica. VRML. Realidade Virtual.

ABSTRACT

That work has for objective to describe in the virtual world, an atmosphere already existent in the real world, that it is the Church of the Rooster, reproduced by techniques of the Graphic Computation, giving the largest level of possible reality in that reproduction. The tool to be used is VRML (Language for Modelling in Virtual Reality), where this will have all the necessary components so that she can reach the expected objective, through resources as animation, textures, it climbs and other functions that this language disposes. Firstly some essential concepts will be given for a better understanding of that work, as characteristics of the used language, structures of the files, etc. and to the end the result of the creation of the Church in the three-dimensional world.

Word-Key: Graphic computation. VRML. Virtual Reality.

1. INTRODUÇÃO:

A Computação Gráfica é um veículo que faz tornar possível o mundo virtual, ou seja, ela traz para o computador não só as imagens, mas também dá vida a esses objetos, como movimentos, cores, texturas. Essa tecnologia hoje em dia está ao nosso redor o tempo todo, até mesmo para aquelas pessoas que não trabalham com computador ou que não tem acesso, pois ela é muito presente em comerciais de televisão, efeitos especiais dos filmes. No mercado atual ela também tem um importante papel em diversos segmentos, entre eles na área da meteorologia, como a previsão do tempo, no turismo, através das visitas virtuais entre várias outras áreas.

Uma ferramenta muito importante usada nesses tipos de trabalho, é a linguagem VRML, que significa Linguagem para Modelagem em Realidade Virtual, que permite uma interação maior com o usuário, pois além de permitir a criação de ambientes virtuais, pode-se além de passear por todos os ambientes que o local dispõe também visualizá-los por ângulos diferentes. A Linguagem VRML tem como objetivo dar suporte necessário para o desenvolvimento de mundos virtuais multi-usuários na Internet, sem precisar de redes de alta velocidade. Ela trabalha em geometria 3-D e suporta transformações (translação, rotação, escala), luz, sombreamento. Outra característica muito importante dessa linguagem é o nível de detalhe (LOD, level of detail) que disponibiliza a quantidade certa de dados para um objeto na sua importância na cena. As versões mais recentes dessa linguagem possuem características como animação, movimentos de corpos e interação entre usuários, essa última versão é chamada Moving Worlds VRML 2.0.

Devido a essas tecnologias estarem crescendo bastante o número de pessoas que buscam o turismo através da Internet. Levados pelas facilidades que são encontradas por esses visitantes, que têm como principal atrativo o tempo, pois pode conhecer vários lugares do mundo, obter várias informações sem precisar ir até o local para conhecê-lo, como consequência está o custo, pois pelo fato de não haver deslocamento a pessoa não

gastará nada nessa “viagem”.

Diante de todos os recursos disponíveis no VRML, irá se fazer uma reprodução da Igreja de Santo Antônio, que é mais conhecida como Igreja do Galo por possuir um galo de bronze na sua torre. O visitante poderá através da web, conhecer todos os ambientes, seus detalhes, o mais próximo da realidade possível e também saber um pouco de sua história e de sua importância na cidade.

1.1 OBEJETIVOS:

1.1.1 Objetivo Geral:

Reproduzir em um ambiente tridimensional utilizando a linguagem VRML, a Igreja do Galo, permitindo assim que pessoas de qualquer parte do mundo possam visitá-la via Web, levando o máximo de realidade possível a esses visitantes.

1.1.2 Específicos:

O projeto está dividido em três objetivos específicos, que estão relacionados a seguir:

- Colher informações históricas sobre a Igreja do Galo
- Fotografar as partes internas e externas da Igreja para facilitar sua reprodução
- Modelar a Igreja em 3D para apresentação via Internet.

1.2 PROBLEMA DE PESQUISA:

Construção da Igreja de Santo Antônio segundo uma modelação tridimensional, criando ambientes onde as pessoas mediante a utilização de recursos da Internet possam tomar conhecimento dessa cultura, sem precisar ir até o local. Pelo fato que esse tipo de trabalho não é muito comum ainda, ou seja, essa tecnologia não é muito usada, é mais fácil o uso de fotos para o conhecimento do visitante sobre o local desejado para conhecer. Então o principal problema nesse projeto é levar a sensação de tridimensionalidade para a pessoa que estará visitando a Igreja, para que ela possa se sentir como se estivesse no ambiente fisicamente. **Como representar um monumento histórico de forma que pessoas de qualquer parte do mundo possam vê-lo como se estivessem no mesmo ambiente pessoalmente?**

1.3 JUSTIFICATIVA:

Diante do crescimento nos últimos tempos do turismo virtual, pelas diversas vantagens que são encontradas pelos visitantes, permitindo que em questão de minutos se possa conhecer várias lugares do mundo sem sair do frente de seu computador, o que atrai pessoas do mundo todo nesse tipo de turismo, que alguns autores até citam como um tipo de turismo que não se faz, mas que se assiste.

Essa tecnologia também é uma grande aliada do setor turístico, pois o uso do 3D, faz com que as pessoas não só saibam quais os principais pontos histórico-culturais da cidade, mas também permite que esses “turistas virtuais” conheçam esses lugares, faz com que eles possam passear por todos os ambientes que o local dispõem, conhecendo os detalhes de sua arquitetura, permitindo total interação com o visitante, tornando a visita muito mais interessante e havendo assim uma maior divulgação desses pontos.

A linguagem VRML, é uma linguagem que dispõem de recursos suficientes para que se possa levar o máximo de realismo possível na reprodução da Igreja de Santo Antônio, fazendo com que esse tipo de trabalho de reprodução virtual seja um dos raros que a cidade possui, havendo uma inovação através de recursos da Internet, nas visitas feitas pelas pessoas que se interessam por pontos históricos.

1.4 METODOLOGIA:

O trabalho é dividido basicamente em 2 pontos importantes. Construção do referencial teórico para servir como base de todo estudo; e o desenvolvimento prático, onde foi feita toda parte de criação do ambiente tridimensional, utilizando a ferramenta VRML.

Foi necessário fazer uso da Pesquisa Documental, pois foi preciso coletar informações importantes do local, para se obter um maior conhecimento do ambiente escolhido para a reprodução virtual, como sua história, ano de construção, etc.

A Pesquisa de Campo foi de fundamental importância, pois foi preciso fazer algumas visitas a Igreja para fotografar de vários ângulos os detalhes de sua arquitetura, para que se pudesse dar início a sua reprodução. Através das fotografias se pôde ter uma exatidão de alguns pontos da Igreja. Para assim poder reproduzir muito de seus detalhes, levando assim ao visitante o maior nível de realidade possível, fazendo com que este se sinta como se estivesse no ambiente pessoalmente.

A Pesquisa Bibliográfica foi bastante usada para obter informações através de livros, textos, documentos que abordassem este tema, para um melhor conhecimento do assunto e esclarecimento em alguns casos de seu desenvolvimento.

1.4.2 – Classificação da Pesquisa

Essa pesquisa pode ser classificada da seguinte maneira:

De natureza Aplicada, pois objetiva gerar conhecimentos para aplicação prática dirigida à solução de problemas específicos.

Abordagem Qualitativa, pois será analisada cada situação, a partir de dados descritivos.

E por fim quanto ao objetivo, é exploratória, pois visa proporcionar familiaridade com o problema com vistas a torná-lo explícito.

2. REFERENCIAL TEÓRICO:

2.1 - O que é, e para que serve computação gráfica:

Existem algumas definições para computação gráfica, dentre algumas estão: Para ISO apud Azevedo e Conci (2003, p.3), "Computação Gráfica é um conjunto de ferramentas e técnicas para converter dados para ou de um dispositivo gráfico através do computador", outra definição é dada por Computação (2003), que a define como, "computação gráfica é o veículo de comunicação homem/máquina mais adequado à percepção humana", outra definição é a de Gomes e Velho(1998), onde diz que, "é um conjunto de métodos e técnicas para transformar dados em imagem através de um dispositivo gráfico". Ela surgiu em 1959 com o termo Computer Graphics, criado por Verne Hudson, enquanto ele coordenava um projeto para o Boeing de simulação de fatores humanos em aviões(AZEVEDO E CONCI).

Essa tecnologia atualmente engloba, para melhor descrição didática, pelo menos três sub-áreas: Síntese de Imagens, processamento de imagens e análise de imagens (Azevedo e Conci), onde cada uma desempenha um importante papel. Para Gomes e Velho (1998) as funções de cada uma dessas sub-áreas são as seguintes: na síntese de imagens, os dados gerados pelo sistema de modelagem geométrica são processados e o produto final é uma imagem que pode ser vista através de algum dispositivo de saída gráfica, já o processamento de imagens, o sistema admite como entrada uma imagem que, após ser processada, produz outra imagem na saída. Essa imagem pode ser observada num dispositivo gráfico, e por fim a análise de imagens, que tem por finalidade obter, a partir de uma imagem (entrada), informações geométricas, topológicas ou físicas sobre dados que deram origem a imagem.

Conforme Azevedo e Conci (2003):

Quando a imagem real não é suficiente ou mesmo inviável, a imagem virtual assume seu lugar. Um exemplo é a correção de imperfeições causadas por ruídos, falta de luz, visualizar uma nuvem radioativa como a de Chernobyl, esse tipo de "visão" só será possível com o auxílio da computação gráfica.

As áreas que a computação gráfica atua são praticamente todas, onde em qualquer de suas aplicações, para Gomes e Velho (1998), existem três características que são:

- A computação gráfica permite visualizar objetos que ainda se encontram em fase de projeto;
- A computação gráfica permite visualizar objetos que estão fora do alcance de nossa percepção visual;
- A computação gráfica permite visualizar objetos que fogem de nossa realidade tridimensional.

Uma imagem pode ser vista como a capacidade do ser humano ter de distinguir a forma, cores, textura e a relação espacial existente entre objetos de uma parte do mundo real.

Segundo Azevedo e Conci, há três categorias de estímulos visuais usadas pelo cérebro para formar uma imagem 3D: informações monoculares, informações óculo-motoras e informações estereoscópicas. As informações monoculares são conhecidas também como static depth cues (informações estáticas de profundidade) ou pictorial depth cues (informações de profundidade na imagem), entre as informações monoculares temos como exemplo a variação de reflexão da luz, sombras etc. As óculos-motoras são produzidas pelos dois conjuntos de músculos do globo ocular, já as informações visuais-estereoscópicas é bastante útil na noção da distância de objetos colocados até 10 metros do observador.

As projeções geométricas permitem a visualização bidimensional de objetos

tridimensionais. Para ser gerada uma imagem em 3D, primeiramente as coordenadas precisam ser convertidas para 2D, que correspondam a uma visão do objeto de uma posição específica. Esse processo é chamado de projeção.

A fotografia que se obtém de uma máquina fotográfica real é uma projeção da cena em um plano, que corresponde ao filme. Da mesma forma que no mundo real, a imagem que se obtém da cena sintética depende de vários fatores que determinam como esta é projetada em um plano para formar a imagem 2D exibida em algum dispositivo, como o vídeo.

Azevedo e Conci diz que:

Ao gerar imagens de cenas 3D em computação gráfica, é comum fazermos uma analogia com a máquina fotográfica. Nessa analogia, imaginamos um observador que, posicionado em um ponto de observação, vê a cena através das lentes de uma câmera virtual que pode ser posicionada de forma a obter imagem na cena, e onde se pode definir, além da posição da câmera, sua orientação e foco, o tipo de projeção usada e a posição dos planos que limitam a visibilidade da cena, os chamados clipping planes.

Nesse projeto o uso da fotografia será de grande importância, pois será através de fotos, que se poderá ser tirado os detalhes da Igreja, podendo assim levar o maior nível de realidade possível nessa reprodução para o visitante.

O que despertará mais ainda o interesse do visitante no seu passeio virtual, ou melhor dizendo o turismo virtual que pode ser definido segundo Turismo(2005) como: Uso da Internet para obter todo tipo de informações sobre diferentes locais, permitindo a partir do espaço doméstico a visualização de fotos, filmes e informação textual sobre diversos locais, por vezes de forma interativa.

Sendo a interação como um fator de grande importância nesse projeto, pois é bastante difícil encontrar sites onde se possa passear utilizando recursos tridimensionais,

onde o visitante possa conhecer todos os ambientes, fazendo com que haja uma interação do usuário com o ambiente, tornando os passeios muito mais interessantes, fazendo assim um grande diferencial, já que o uso dessa tecnologia nesses tipos de trabalhos são bastante raros.

2.2 - Cores e Luz:

Para Azevedo e Conci (2003):

O uso da cor na computação gráfica apresenta vários aspectos interessantes: melhora a legibilidade da informação, possibilita gerar imagens realistas, permite indicar mecanismos de segurança, permite focar a atenção do observador, permite passar emoções e muito mais. Enfim, o uso de cores torna o processo de comunicação mais eficiente.

De acordo com Azevedo e Conci (2003):

A retina do olho humano contém dois tipos de células que detectam a luz e a transformam em impulsos nervosos, são elas: os cones e os bastonetes. Onde os cones são sensíveis ao alto nível de iluminação e responsáveis pela percepção das cores. Já os bastonetes, são sensíveis ao baixo nível de iluminação, distinguem os tons de cinza e são responsáveis pela visão periférica.

Segundo Azevedo e Conci (2003):

A área que enxergamos é chamada de campo visual. Em cada olho existem campos visuais separados para os vários tipos de cones e bastonetes, pois essas células têm disposição diferente na retina. O campo visual dos bastonetes é maior, ainda que fraco, no centro da visão, pois no centro da retina existem poucas células desse tipo. O campo visual dos cones verdes é o menor de todos, porque eles se localizam muito próximos e no centro da retina. Assim só temos visão exata das cores quando olhamos de frente para um objeto e sua imagem incide na região central da retina (fóvea).

A luz segundo Azevedo e Conci pode ser definida como, “a luz é uma onda eletromagnética que permite ver o mundo ao redor”. As diferentes sensações que uma luz pura produz no olho humano, e que produzem as cores, dependem da frequência ou onda dessa luz. Onde se determina frequência de uma onda o número de ciclos da onda por segundo, e o comprimento da onda é igual a sua velocidade de propagação em um meio dividido pela sua frequência.

O olho humano é sensível à radiação eletromagnética na faixa de 400 a 700 nanômetros, chamada de espectro visual, onde estão localizadas as sete cores visíveis, distinguidas por seus respectivos comprimentos de onda.

Violeta	Azul	Ciano	Verde	Amarelo	Laranja	Vermelho
380-450	540-480	480-490	490-560	560-580	580-600	600-700

Para colocar uma luz no cenário VRML usa-se o comando *PointLight* que permite especificar a localização, a cor da luz e a intensidade da luz. Em um mesmo arquivo VRML pode haver várias luzes.

Essas são as cores que aparecem no arco-íris. A cor vermelha é a que possui o maior comprimento de onda e a violeta a menor. Elas sempre aparecem na mesma ordem também quando uma luz branca é decomposta ao atravessar um prisma de cristal. Esse fenômeno é conhecido como dispersão da luz e foi descoberto por Isaac Newton em 1666. Onde ele mostrou que a luz branca é a mistura de todas as luzes coloridas do espectro. A cor dos objetos que não emitem luz é percebida pela energia que ela reflete, ou seja, quando a luz incide sobre um objeto, uma parte é absorvida pela própria superfície e a outra é refletida chegando aos nossos olhos. Onde para o melhor entendimento se faz necessário à definição de reflexão e refração.

De acordo com Azevedo e Conci (2003):

A maioria dos objetos ao nosso redor não emite luz própria, pelo contrário, reflete a radiação neles incidente em diferentes comprimentos de onda. A reflexão se deve à interação molecular entre a radiação incidente e o material que compõe a superfície dos objetos.

Já para a definição de refração Azevedo e Conci dizem que, “Um feixe de luz incidente sobre uma superfície líquida é refletido e desviado ao penetrar na água. Esse desvio é chamado refração. Os raios luminosos, refletidos e refratados estão contidos em um mesmo plano”.

2.3 - Animação

A animação na computação gráfica está presente em quase todos os segmentos e não apenas como estamos acostumados a associar a filmes ou propagandas. Ela vai muito, além disso, e pode ser definida através do dicionário como do Latim: *Animare*, que significa dar vida, movimento, coragem, entusiasmo, alma, o que descreve também para a produção de animação em computação gráfica.

Um recurso que foi introduzido a essa última versão do VRML 2.0 são os chamados de interpoladores, que segundo Pollo (1997), “permitem incorporar animações quadro-a-quadro à cena. Basta prover descrições do objeto em alguns pontos críticos para que o interpolador faça todos os cálculos necessários para as descrições intermediárias, garantindo uma maneira compacta e eficiente de criar as animações”.

Para Reis (2004)

Nodes do tipo **Interpolator** (interpoladores) são essencialmente scripts já embutidos no VRML que executam cálculos simples para uso em animação. São normalmente combinados com um node do tipo **TimeSensor** e algum outro tipo de node existente no cenário que deva ser movimentado.

Outro grande avanço em termos de interação foi à introdução do conceito de sensores. Para Pollo (1997), “Nós sensores são nós que esperam por um evento em particular para realizar alguma ação em resposta”.

“A animação por computador é dividida em duas categorias: computer-assisted animation (animação assistida por computador) e computer generated animation (animação gerada por computador)” (AZEVEDO e CONCI, 2003).

No caso da Animação assistida por computador, o animador faz os quadro-chave e o computador se encarrega de gerar os quadros intermediários. Já no caso da animação gerada por computador, esta apresenta um grupo maior de técnicas subdivididas nos seguintes grupos: técnicas de baixo nível (low level techniques – técnicas que ajudam o animador na especificação precisa do movimento), e técnicas de alto nível (high level techniques – técnicas usadas para descrever como se comporta o personagem durante a animação). Nesse caso, altos e baixos níveis estão associados ao nível de abstração.

“As formas de animação variam de acordo com os aspectos considerados” (AZEVEDO e CONCI, 2003), existem várias como: Animação por Quadro-Chave (Keyframe), por Script, Procedural, Representacional, Estocástica, Straight Ahead, Pose-to-Pose, Comportamental e Animação Track Based.

A animação de cena pode ser definida como: “Uma cena é um evento ou interação única entre personagens, acontecendo durante um período de tempo e em um lugar único, que move a história adiante na direção de um clímax” (AZEVEDO e CONCI).

A adição de informação de comportamento aos objetos 3D nos arquivos VRML abre várias novas possibilidades para a criação de simulações de tempo real para a web.

Uma animação em VRML pode ocorrer em três formas: durante um certo período de tempo, a partir de um determinado instante e a partir da interação do usuário com o mundo virtual.

2.4 – Sensores

De acordo com Pollo “sensores são nós que têm a capacidade de gerar eventos respondendo à ações do usuário”.

Nodos sensores habilitam eventos quando o objeto se move para certas áreas do mundo virtual e quando certos objetos são clicados. Estes sensores permitem que o usuário arraste ou controle objetos de um lugar para o outro. Outro tipo de sensor permite que se faça animações repetitivas, baseadas em um certo período de tempo pré-definido. Eles permitem ao usuário interagir com o mundo na cena. Os sensores que são utilizados nesse trabalho são o *TimeSensor* e *TouchSensor*.

O *TimeSensor* gera automaticamente um evento, e cada tique do relógio, representa este, uma fração de tempo do ciclo de animação, definido no campo *cycleInterval*, sendo normalmente usado para coordenar animação. Já o *TouchSensor*, quando o usuário clica em um dado objeto ele detecta essa interação e dispara o processo.

Mas existem outros tipos de sensores como é mostrado na tabela abaixo.

Tabela1. Sensores

Sensor	Reações de cada um
ProximitySensor	Posição do observador na cena
VisibilitySensor	Posição do observador na cena
CylinderSensor	“arrasto” do dispositivo de entrada (<i>dragging</i>) sobre o(s) objeto(s) mapeado(s)
PlaneSensor	
SphereSensor	
Collision	Colisão do observador com o(s) objeto(s) determinado(s)

Fonte: Pollo (1997)

Os sensores servem para determinar a posição do usuário no mundo, bem como a entrada e saída do mesmo.

2.5 - O Uso da ferramenta VRML:

Para Kay e Muder (1997), “VRML trata-se de uma linguagem de computador para descrever cenas em 3-D”. Como complemento Pollo (1997) define VRML como, “uma linguagem padrão usada para descrever objetos 3D e combiná-los em cenas ou mundos virtuais”.

Já para Ivan (2005) ele conceitua VRML da seguinte maneira:

A Virtual Reality Modeling Language é uma linguagem que permite que se apresentem objetos e mundos tridimensionais através da World Wide Web. E mais do que simplesmente mostrar essas cenas estáticas. Possibilitar que visitante de um mundo virtual interaja com seus objetos.

Para Pesce apud Barros (2004), VRML pode ser descrita da seguinte forma:

O poder do VRML reside na sua capacidade de fazer uma ponte entre dois métodos básicos de comunicação humana: a imaginação cerebral e a sensação visceral.

Outra definição encontrada é a de Bramorski e Marques (1998) que a define da seguinte forma:

VRML é uma Linguagem para Modelação de Realidade Virtual usada para criar mundo de três dimensões. Trata-se de um formato de arquivo objeto de 3D, análogo ao HTML (para objetos tridimensionais). Usando-se formas pré-definidas e se especificando novas formas a partir dessas, pode-se atribuir cor, forma, luz e outros a um documento VRML.

De acordo com Reis (2004) VRML pode ser assim definido:

Numa visão mais ampla, o VRML fornece a tecnologia que permite integrar objetos tridimensionais, bidimensionais, texto e multimídia em um único e consistente modelo. Quando se acresce a isto a combinação com linguagens Script e recursos de Internet, um novo gênero de aplicações interativas torna-se possível.

VRML é uma abreviação da palavra Linguagem para Modelagem em Realidade Virtual, onde se fará uso dessa ferramenta durante a reprodução da Igreja, título desse projeto, essa tecnologia foi criada com o objetivo de construir um padrão que não apenas funcionasse, mas que fosse acessível a todos e não dominado ou controlado por uma empresa de software.

Essa Linguagem não é compilada e sim um simples arquivo de texto que pode ser interpretado. Esses programas interpretadores são os chamados VRML browsers, no entanto, linguagens de programação como Java, Java Script, podem ser usados internamente para executar procedimentos complexos.

Esses browsers eles interpretam as informações contidas no arquivo como descrições geométricas de objetos no mundo VRML.

Segundo VRML:

Como HTML (Hypertext Markup Language) é um formato de arquivo que define o formato e o conteúdo em uma página em 2D com links para mais informações, VRML também é um formato de arquivo que define o formato e o conteúdo de um mundo 3D com links para mais informações. Diferente de HTML, porém mundos em VRML são espaçosos e interativos, cheios de objetos que agradam ao usuário.

Em VRML 2.0 que é a versão mais atual, os objetos podem se mover pelo mundo virtual e responder a eventos temporais ou disparados pelo usuário, e é possível adicionar som espacial e filmes de vídeo à cena, tornando a experiência do observador muito mais realista.

O princípio básico de funcionamento de VRML é deixar de lado a complexidade matemática exigida pelas operações de renderização dos objetos 3D, provendo única e exclusivamente uma base consistente, apesar de simples, da qual todos possam se beneficiar, desde desenvolvedores de browsers até usuários finais.

Existem outras ferramentas, onde uma bastante conhecida é a OpenGL (Open Graphical Library).

Conforme Azevedo e Conci (2003):

Pode ser definida como uma linguagem de software (API – Application Program Interface) para aceleração da programação de dispositivos gráficos, com aproximadamente 120 comandos para especificação de objetos e operações necessárias para a produção de aplicações gráficas interativas 3D.

Para Manssour (2003), “OpenGL é uma biblioteca de rotinas gráficas e de modelagem, bi(2D) e tridimensionalidade (3D), extremamente portátil e rápida”.

Outra explicação de Manssour (2003) é a seguinte:

Ela possibilita a criação de gráficos 3D com excelente qualidade visual e rapidez. Justamente pela sua portabilidade, não possui funções para gerenciamento de janelas, interação com o usuário, arquivos de entrada/saída, gerenciar eventos provenientes de mouse e teclado e apresentação de menus. Cada ambiente, como, por exemplo, o Microsoft Windows, possui suas próprias funções para esses propósitos.

Como definida por Azevedo e Conci (2003):

OpenGL não é uma linguagem de programação como C, C++, Java, é uma poderosa e sofisticada biblioteca de códigos, para desenvolvimento de aplicações gráficas 3D em tempo real, seguindo a convenção de chamada de bibliotecas da linguagem de programação C.

Para Azevedo e Conci (2003):

Programas criados com OpenGL podem se tornar bastante complicados, diante do volume de operações que podem ser realizadas. Entretanto, a estrutura básica de vários programas é relativamente simples.

2.5.1 – Arquivo VRML

A extensão normal para os arquivos VRML é .wrl, ainda que ocasionalmente seja usado .wrz para arquivos compactados, existem outra extensão usada também para arquivos compactados como .wrz.gz.

A estrutura de um arquivo VRML basicamente contém:

- Cabeçalho: especifica a versão e caracteres usados
- Comentários, que são opcionais: caracter #
- Nodes ou Nós: descrevem objetos
- Fields: atributos dos nodos (nodes)
- Values: valores dos campos (fields)
- Routes: associam eventos entre nodos.

Cada arquivo VRML executa as seguintes funções:

- Implicitamente, estabelece um Espaço Global de Coordenadas para todos os Objetos definidos no arquivo, assim como, para os Objetos incluídos;
- Explicitamente, define e compõe um conjunto de Objetos 3D e de multimídia;
- Pode especificar Hyperlinks para outros arquivos e aplicações;
- Pode definir comportamentos e inter-relacionamentos entre os objetos.

Uma importante característica dos arquivos VRML é a sua habilidade de concatenar múltiplos arquivos, seja através da Inclusão, seja através do Hyperlink entre arquivos relacionados.

2.5.2 – Os objetos

A linguagem VRML define as formas básicas como objetos primitivos. No entanto podem ser usados para definir formas mais complexa. Essas formas podem ser descritas como: Cubo (box), Cone (cone), Cilindro (cylinder), Esfera (sphere) que são chamadas de *shapes primitivos*, como mostrado abaixo:



Figura 1. Cubo



Figura2. Cone

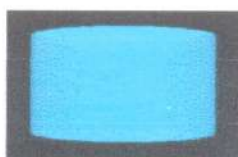


Figura 3. Esfera

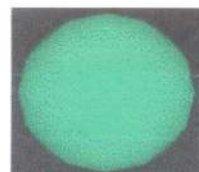


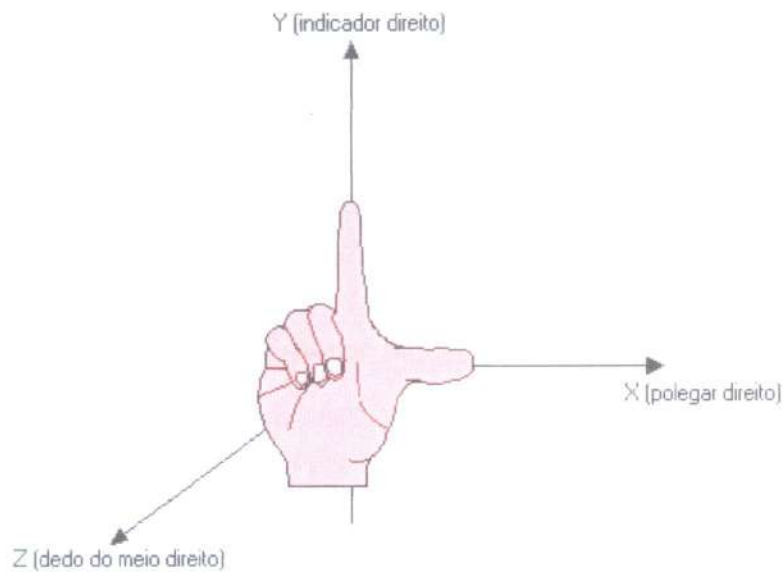
Figura 4. Cilindro

Os nós do tipo *shape* descrevem a geometria (estrutura ou forma) e aparência (cor ou textura). Segundo Damasceno (2005), “nós são os blocos básicos de construção do mundo VRM. É um conjunto de especificações que determinam as características dos objetos contidos no cenário”.

As dimensões do objeto são controladas pelos campos (fields) do tipo geometry. Dessa forma ao se desenhar um objeto (shape) primitivo deve-se informar suas dimensões.

Todas as distâncias em VRML são mensuradas em metros. O sistema de coordenadas da linguagem trabalha como mostra a figura abaixo.

Figura 5. Regra da Mão Direita para eixos 3D



Fonte: Damasceno (2005)

O eixo X é horizontal, Y é vertical, e Z dá a impressão que vem da tela até o usuário. Todas as transformações são realizadas sobre estes eixos.

Outro ponto importante a ressaltar é que todas as rotações, realizadas sobre o objeto são mensuradas em radianos e não em graus.

2.5.3 – As Transformações

Conforme Damasceno (2005):

Essa linguagem possui três tipos de transformações que podem ser aplicadas sobre os objetos. São elas: *translação*, *rotação* e *escala*. Para realizar estas transformações é utilizado o nodo *Transform*. Onde ele não necessariamente deve conter as três ao mesmo tempo, podendo conter

simplesmente uma ou outra transformação desejada, esse nodo controla a translação onde esta posiciona o novo sistema de coordenadas nos eixos X, Y e Z. A rotação como o próprio campo já indica *rotation*, permite rodar o grupo de nós em torno de um eixo designado pelos três primeiros valores do campo, ela orienta o sistema de coordenadas sobre o eixo da rotação em um dado ângulo. E por fim a função da escala é aumentar ou diminuir um sistema de coordenadas por fatores de escala em X, Y e Z que é definida pelo campo *scale*. Uma escala unitária, em qualquer direção, implica na não alteração de tamanho naquela direção e, portanto uma escala igual a (1 1 1) não implicaria em nenhuma modificação nas dimensões do objeto.

2.6 – Nodos ou Nós:

De acordo com Castro e Magalhães:

Conjuntos de objetos, ou melhor, abstração de objetos e de certas entidades do mundo real, tais como: esferas, cubos, luz, som, etc. São definidos como sendo os "componentes fundamentais" de uma cena em VRML, pois esta é construída a partir da disposição, combinação e interação entre os nodos.

Cada nodo apresenta, em geral, as seguintes características:

- a) **Tipo:** citam-se como exemplos: Box, Color, Group, Sphere, Sound, etc;
- b) **zero ou mais atributos para *fields*** que definem como um nodo difere de outro nodo do mesmo tipo;
- c) **conjunto de eventos que um nodo pode receber e enviar:** cada nodo pode receber zero ou mais tipos diferentes de eventos, os quais geram alterações no estado do nodo. Analogamente, podem gerar tipos distintos de eventos para comunicar alterações em seu estado;
- d) **implementação:** a implementação de cada nodo define como ele reage a eventos que pode receber à geração de eventos e às suas características visuais no mundo virtual;

e) **nome**: pode-se atribuir um nome a um nodo. Isso permite que possamos fazer referências a uma instância específica de um nodo. Isso é feito através das palavras DEF/USE.

2.7 - Material e Appearance

Para Manssour (2000), podem ser utilizadas as definições para os seguintes nós: “o valor do campo *appearance* especifica um nó que define a aparência do *shape*, incluindo sua cor e textura”. Já para definição de Material, “especifica um nó que define os atributos do material. Normalmente é utilizado o nó *Material* neste campo”.

O valor *default NULL* nos dois casos determina um material e o objeto fique com a cor branca.

```
#VRML V2.0 utf8
# Desenho de uma esfera branca
Shape {
  appearance Appearance {
    material Material {

    }
  }
  geometry Sphere {
    radius 4.0
  }
}
```



Figura 6. Criação da Esfera em branco

2.8 – Uso do DEF e USE

Para definir um nó para ser instanciado, basta colocar a palavra **"DEF"** antes do nome escolhido. E para usar este nó novamente dentro do arquivo basta colocar a palavra **"USE"** na frente do nome. Uma vez que um nó tem um nome, este nó pode ser usado posteriormente no arquivo.

A declaração de USE não cria uma nova cópia do nó, em vez disso, o mesmo nó é inserido na cena uma segunda vez. A utilização de múltiplos exemplos (ou instâncias) de um mesmo nó dá-se o nome de "instanciação". Sendo assim, caso seja necessária a reutilização de um mesmo nó, várias vezes numa cena, é mais eficiente atribuir a ele um nome, quando da primeira vez que o descrevemos e, a partir daí, referenciá-lo pelo nome cada vez que for necessário usá-lo novamente. Essa técnica, além de tornar o arquivo VRML menor e mais fácil de ler, diminui o tempo que se leva para carregar a cena, o que indica uma significativa melhoria na performance geral.

A seguir um exemplo da utilização do DEF e USE:

```

Transform {
  translation 0.4 -0.7 0 #0 0 -0 7 0
  rotation 0 0 1 1.57
  children [
    Transform {
      translation 0.6 0 0
      children [
        DEF Bloco Shape {
          appearance Appearance {
            material Material {
            }
            texture ImageTexture {
              url ["ardosialamin.jpg"]
            }
          }
          geometry Box {
            size 0.2 0.4 0.2 # 0 2 0 4 0 2
          }
        }
      ]
    }
  ]
}
Transform {
  translation 0.6 0.4 0
  children USE Bloco
}
Transform {
  translation 0.6 0.8 0
  children USE Bloco
}
]

```

Como resultado desse código se tem o seguinte objeto:



Figura 7. Utilização do DEF e USE

2.9 – Grupos

Uma das grandes vantagens do VRML é a forma de estruturar as cenas através da composição de múltiplos nós. Para conseguir tal estruturação torna-se evidente a necessidade de agrupar vários nós e passar a tratar esse novo grupo como um outro nó.

Estes nós são designados por nós de Agrupamento. São utilizados para criar grafos hierárquicos de objetos, e são constituídos através de um campo children (filho) que determina uma lista de nós.

Segundo Kay e Muder “O uso de grupos permite juntar formas para criar objetos mais complexos, permitindo que qualquer ação afete todos os elementos de grupo simultaneamente”.

A figura abaixo mostra um exemplo da utilização do nó grupo, onde uniu-se a parede, as janelas, a porta, o rodapé e a coluna azul.

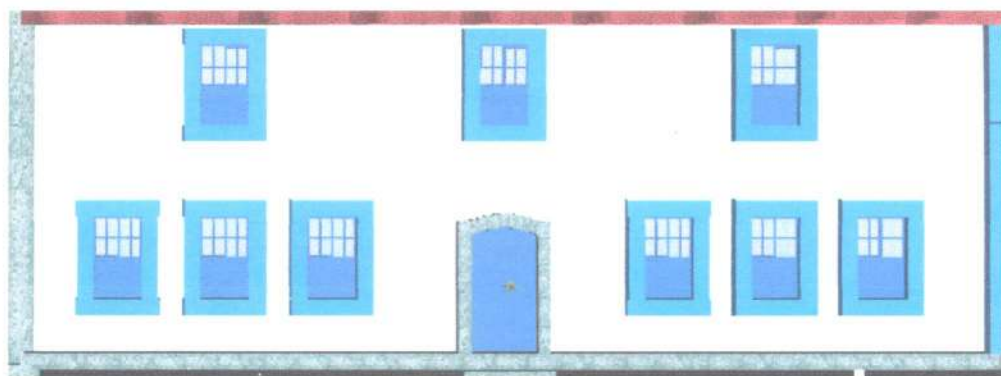


Figura 8. Exemplo de agrupamento

2.10 – Texturas:

Segundo Ivan “Textura é uma imagem bidimensional que é aplicada (mapeada) a objetos de um mundo virtual a fim de aumentar a qualidade de detalhes”. Outra definição encontrada é a de Kay e Muder que diz que “Texturas são imagens. Mais especificamente, elas são imagens em mapa de bits (bitmap). As imagens que você costuma ver na web também são desse tipo, geralmente usando o formato GIF ou JPEG”.

Usualmente os visualizadores permitem mapeamento de texturas nos formatos GIF, JPEG e MPEG. Alguns já suportam o uso do PNG.

O uso da textura nesse trabalho foi de bastante importância, pois dá uma maior semelhança com o ambiente real, levando um maior realismo ao ambiente reproduzido.

2.10.1 - Nó Image Texture

Segundo Damasceno (2005), “Este nó especifica a posição da imagem a ser usada para texturizar a forma, como também se a imagem deve ser repetida verticalmente, ou horizontalmente, ao longo de cada uma das faces da forma.”

Um campo que é bastante usado é o url, que diz onde está o que se deseja extrair

para o programa.

2.10.2 – Nó textureTransform:

Que irá definir as mudanças que irão ser feitas na textura do material

Sintaxe do nó:

```
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 0 1 0
    }
  }
  texture ImageTexture {
    url ["C:/AnaLygia/ardosina.jpg"]
  }
  textureTransform textureTransform {
    scale 4 4
  }
}
```

2.11 – Viewpoint

Segundo Reis (2002) o viewpoint pode ser definido da seguinte maneira:

Descreve uma Posição e uma Orientação para visualização do cenário. Este é um recurso que o autor VRML pode utilizar para guiar o usuário, segundo uma interação pretendida entre usuário e cenário.

Nem todos cenários contém viewpoints, mas quando existirem, podem ou não ser usados pelo usuário.

Conforme Barros Neto (2001), “Um nó viewpoint em VRML determina uma posição no espaço de onde o usuário pode visualizar a cena”.

A seguir um exemplo do uso do viewpoint:



Figura 9. Visão do Interior da Igreja através do viewpoint

3. ANÁLISE DE RESULTADOS:

Através da implementação desenvolvida para a reprodução da Igreja de Santo Antônio, é possível levar ao visitante um nível de realidade virtual bem acentuado. O uso da textura nesse trabalho foi de bastante importância, pois dá uma maior semelhança com o ambiente real, levando um maior realismo ao ambiente reproduzido.

A ferramenta utilizada para reprodução desse ambiente real em virtual, foi de grande importância, pois ela possui vários comandos que faz com que se possa levar ao visitante uma sensação de tridimensionalidade, além de animação, texturas, iluminação, etc. o que prende bastante a atenção do visitante.

3.1 Desenvolvimento:

No início do desenvolvimento, foram criados arquivos separados contendo o código de cada objeto criado, como o piso, as janelas laterais, as da frente da igreja, os bancos, as portas, para que posteriormente fosse utilizada em outras posições. Depois foi se agrupando as peças, como a parede lateral da igreja, suas janelas e a porta; a parede da frente com as portas e as janelas; e os bancos com o chão da parte de dentro da igreja. Quando cada conjunto era concluído unia-se a uma única peça. Pois as portas possuem sensores, e se não fossem feitos em cada arquivo separado, ao clicar em uma das portas todas iriam se abrir, o que não era interessante.

Outro ponto de fundamental importância foram as fotografias, que serviu de base durante todo desenvolvimento desse trabalho. Outro motivo que pode ser destacado, é o fato de ter se usado texturas nessa reprodução, e em alguns casos essas texturas foram extraídas dessas fotos que foram tiradas do local, tendo em vista a grande quantidade de detalhes a serem reproduzidos e que não se conseguiria no tempo dado para conclusão do trabalho, e também por talvez não ficar tão próximo da realidade.

Foram utilizadas as formas básicas do VRML na construção de alguns objetos. Como exemplo, na figura abaixo, foi utilizado o cilindro, esfera e cubo. Cada um construído separadamente para depois fazer a união numa só peça. No box, foi aplicado uma textura, para que se fosse colocado o famoso galo da Igreja na sua torre, onde esta é a parte mais alta da igreja.



Figura 10. Utilização das formas básicas do VRML

Em determinadas partes da construção foi necessária a utilização de texturas para que se pudesse ter uma maior exatidão da reprodução de seus detalhes, onde algumas dessas foram retiradas das fotografias que foram tiradas do local, como o exemplo abaixo:



Figura 11. Uso de textura retirada diretamente da fotografia do local

Houve também a utilização de texturas que foram extraídas de alguns sites da web, o que fazia com que se assemelhasse cada vez mais com a realidade, como demonstra as figuras abaixo.

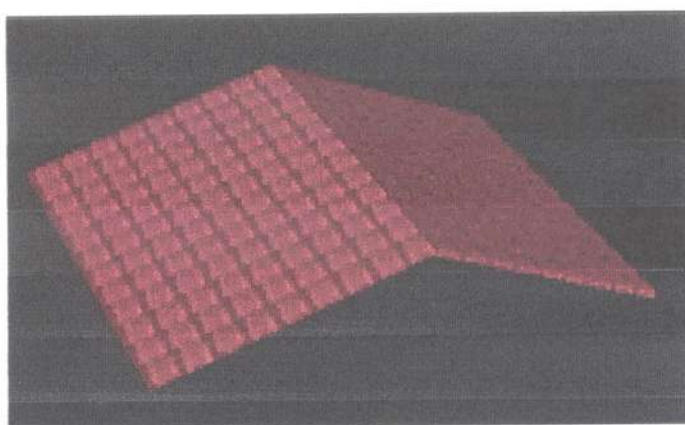


Figura 12. Utilização de textura no teto da parte interna da Igreja

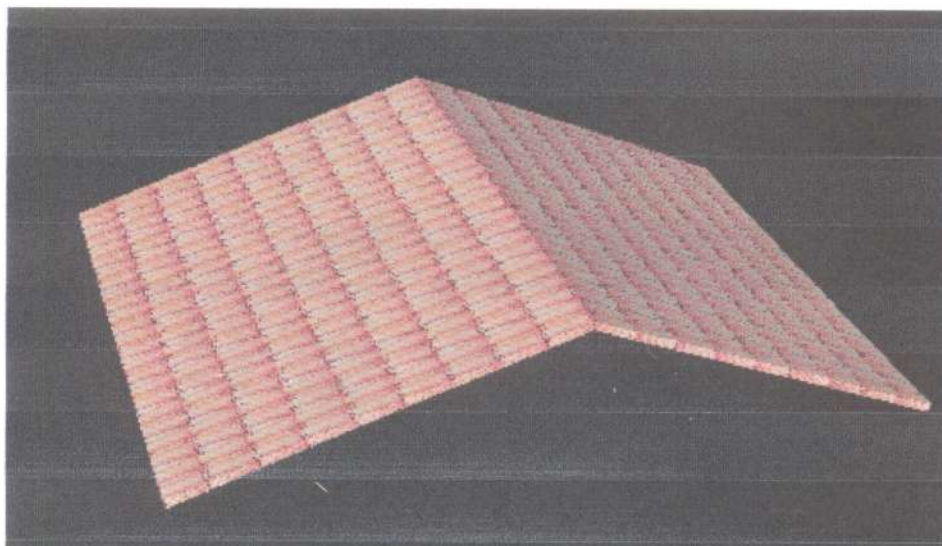


Figura 13. Textura do teto da parte externa da igreja

E como outra forma, foi preciso construir o objeto, depois aplicar uma textura extraída das fotografias tiradas durante as pesquisas de campo, para se conseguir uma melhor aproximação da realidade. Pode servir como exemplo, as janelas da parte interna da igreja, onde tirou-se da fotografia a parede de madeira que serve como uma varandinha da janela. Veja a figura abaixo:

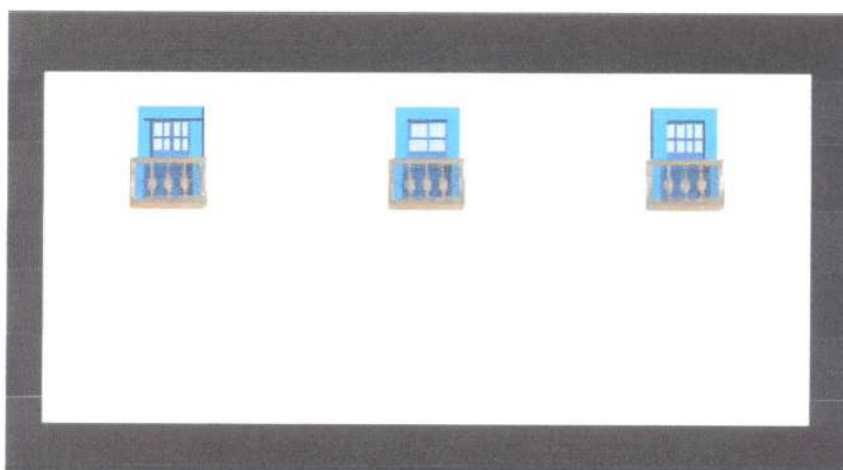


Figura 14. Varandas construídas com texturas retiradas das fotografias.

Já na parte externa da igreja foram utilizadas texturas que foram encontradas na internet, para que pudesse colocar nas janelas, as portas e as torres que se encontram na

parte da frente da igreja. As texturas das janelas foram construídas como um objeto a parte, já a parte azul foi construída no Paint, e depois colocado ao fundo da textura, para que se desse esse efeito mais real.

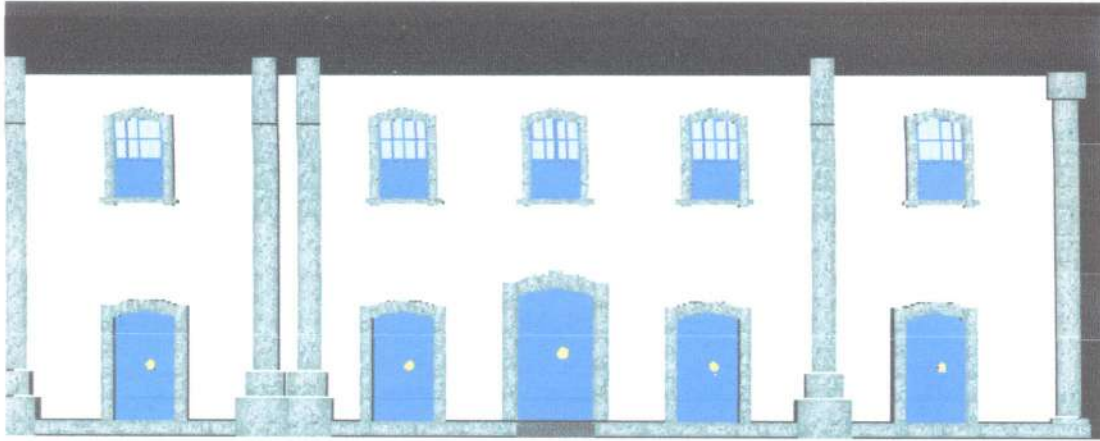


Figura 15. Parte da Frente da Igreja

Como elementos utilizados para construção das portas e janelas usou-se cubos (box) para se fazer a parte que fica ao redor da parte azul, usando textura, da mesma forma, se fez nas torres e no rodapé da igreja.

Em especial no caso das portas, foi preciso se construir uma a uma, e não como se foi feito na maioria dos objetos criados, onde se construía apenas um, e se fosse preciso utilizá-lo poderia ser usado quantas vezes necessária. Mas diferente dos outros, as portas por possuírem sensores, tiveram que ser construídas em arquivos separados. Se fossem feitas da mesma forma dos outros objetos, ao abrirmos uma porta todas seriam abertas ao mesmo tempo.

A seguir na figura abaixo, uma demonstração da utilização dos sensores nas portas, o que fazem com que elas abram e fechem.

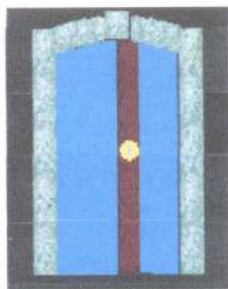


Figura 16. Utilização de sensores

Nesse exemplo, a porta está sendo fechada, e como se foi feito cada arquivo da porta separadamente, apenas a porta em que se está dando o clique irá se abrir ou fechar como nesse caso.

4. CONCLUSÃO:

Através desse trabalho foi possível se aprofundar as tecnologias que são usadas na construção de ambientes virtuais, obtendo um conhecimento bastante vasto do uso da tecnologia que foi usada no decorrer dessa implementação.

Foi possível concluir que o uso da computação gráfica tende a crescer cada vez mais, pois proporcionam grandes facilidades a diversos setores, facilitando assim a divulgação de determinados locais, etc. Através dessa técnica é possível a criação de ambientes, personagens, reproduções, enfim tem grandes fatores que fazem com que a computação gráfica venha a se expandir cada vez mais em todos os segmentos.

O início do trabalho foi um pouco difícil, pois a linguagem utilizada para a reprodução, o VRML, não tinha sido estudado, mas por saber que essa linguagem era simples e que através dela era possível levar um nível de realidade bastante acentuado ao visitante, ficou então decidido seu uso. Com o passar do tempo, as dificuldades com o VRML foram totalmente superadas, e todos os objetivos descritos anteriormente foram concretizados.

Com esse trabalho, foi possível notar que se pode construir, reproduzir, enfim, fazer trabalhos de alto nível usando técnicas como essa. E por isso vem sendo utilizado em todas as áreas para facilitar os trabalhos em diversos segmentos.

Como trabalho futuro sugere-se estudos de novas linguagens usadas na computação gráfica, que contribuem para a realização de trabalhos mais complexos, possibilitando uma interface de comunicação de alta qualidade com o usuário.

REFERÊNCIAS:

AZEVEDO, Eduardo; CONCI, Aura. **Computação Gráfica: Teoria e prática**. Rio de Janeiro: Campus, 2003.

BARROS, Paulo Gonçalves de. **Realidade Virtual e Multimídia**. 2004. Disponível em:

<http://www.di.ufpe.br/~ifl124/vrml/vrml11.htm>. Acesso em: 24/05/2005.

BARROS NETO, Antônio José de Barros. 2001. Disponível em:

http://www.di.ufpe.br/~ifl124/projetos/AnimacoesVolumetricasEmVRML/projeto_vrml.htm. Acesso em 24/05/2005.

BRAMORSKI, Mônica Maria; MARQUES, Everlin Fighera. **Apostila VRML Básico**. Disponível em:

<http://www.inf.ufsc.br/~awangenh/CG/apostilas/vrml.doc>. Acesso em 19/11/2005

Acesso em 22/11/2005.

CASTRO, Elisa Calhau de; MAGALHÃES, Léo Pini. **AM 533A: Computação Gráfica e Multimídia I**. 1999. Disponível em:

<http://www.dca.fee.unicamp.br/sibgrapi99/vrml/Aula1/aula1.html#grouping>

Acesso em: 13/10/2005.

CORRIGAN, John. **Computação Gráfica segredos e soluções**. Rio de Janeiro: Ciência moderna, 1994.

DAMASCENO, Luiz Gonzaga. **Matemática e Computação.com**: Computação Gráfica. 2005. Disponível em:

<http://clientes.interjato.com.br/~damasceno/MeuSite/index8.htm>. Acesso em 11/05/2005

DUARTE, Glaucius Décio. **Modelagem e visualização de projetos arquitetônicos na Internet**. 2000. Disponível em:

<http://atlas.ucpel.tche.br/~glaucius/pesq0.htm>. Acesso em: 22/04/2005.

GOMES, Jonas; VELHO, Luiz. **Computação Gráfica**: volume 1. Rio de Janeiro: IMPA/SBM. 1998.

IVAN. **rpa vrml.brasil**. Disponível em:

<http://members.tripod.com/~brasil3d/rpavrm.html#textura>. Acesso em 23/06/2005.

LABORATÓRIO de computação gráfica. Disponível em:

<http://www.cg.eletrica.ufu.br/>. Acesso em: 28/05/2005.

KAY, David; MUDER, Douglas. **VRML & 3D na Web para leigos**. São Paulo: Berkley Brasil, 1997.

KIRNER, Cláudio. **Grupo de pesquisa em realidade virtual**. Disponível em:

<http://www.inf.ufsc.br/~awangenh/CG/apostilas/vrml.doc>. Acesso em 19/11/2005

Acesso em 01/11/2005

MANSSOUR, Isabel Harb. **Introdução à VRML 2.0**. 2000. Disponível em:

<http://www.inf.pucrs.br/~manssour/VRML/ShapesPreDefinidos.html>

Acesso em 15/10/2005.

POLLO, Luís Fernando. **Software para Geração Automática de Modelos 3D em VRML**. 1997. Disponível em:

<http://www-usr.inf.ufsm.br/~pollo/TG/capitulo1.html>. Acesso em: 10/06/2005.

REIS, Adriano J. G. **Computação Gráfica 3D: Conceitos, fundamentos, geométricas e algoritmos**. Disponível em:

<http://grafica3d.eng.br/adrhpg.htm>. Acesso em: 19/06/2005.

TURISMO. Turismo e Internet. Disponível em:

http://www.geocities.com/cev_escite/cev.internet.html#TURISMO%20e%20INTERNET. Acesso em 25/05/2005.

VIANNA, Arlindo Cardarett. **Computação Gráfica**. 2003. Disponível em:

<http://www.inf.ufes.br/~thomas/graphics/www/apostilas/CIV280/AcvCompGraf.pdf>.

Acesso em: 24/05/2005.

VRML, O que é VRML 2.0. Disponível em:

www.geocities.com/MotorCity/4630/historia/oqueevrml.html. Acesso em: 23/06/2005.

ANEXOS

#VRML V2.0 utf8

Banco

```
DEF lado Transform {
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor .32 .16 .08
          specularColor .12 .04 .01
          emissiveColor .14 .14 .14
          ambientIntensity .0467
          shininess .03
        }
      }
    }
    geometry Box {size .05 2 .5}
  ]
}
```

```
Transform {
  children USE lado
  translation 3 0 0
}
```

```
DEF ladom Transform {
  translation 0 -.5 .5
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor .32 .16 .08
          specularColor .12 .04 .01
          emissiveColor .14 .14 .14
          ambientIntensity .0467
          shininess .03
        }
      }
    }
    geometry Box {size .05 1 .5}
  ]
}
```

```
Transform {
  children USE ladom
  translation 3 0 0
}
```

```
DEF banco Transform {
  translation 1.5 -0.2 .3
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor .32 .16 .08
          specularColor .12 .04 .01
          emissiveColor .14 .14 .14
          ambientIntensity .0467
          shininess .03
        }
      }
    }
    geometry Box {size 3 .1 .8}
  ]
}
```

```
Transform {
  children USE banco
  translation 0 .8 .1
  rotation 1 0 0 1.57
}
```

#VRML V2.0 utf8

#Igreja

```
Inline {
  url ["PisoInterior.wrl"]
}
Transform {
  translation 8.0 0 0
  children [
    Inline {
      url ["PisoInteriorMuseu.wrl"]
    }
  ]
}
Transform {
  translation 9.85 2.15 0
  rotation 0 1 0 1.57
  children [
    Inline {
      url ["ParedeLatJanelaCompleta.wrl"]
    }
  ]
}
Transform {
  translation -6.15 2.15 0
  rotation 0 1 0 1.57
  children [
    Inline {
      url ["ParedeLateralDireita.wrl"]
    }
  ]
}
Transform {
  translation 6.15 2.85 0
  rotation 0 1 0 1.57
  children [
    Inline {
      url ["ParedeLateralEsquerda2.wrl"]
    }
  ]
}
Transform {
  translation -2.30 2.85 0    # -2.30 2.15 0
  rotation 0 1 0 1.57
  children [
    Inline {
      url ["ParedeLateralEsquerda.wrl"]
    }
  ]
}
Transform {
  translation 2.1 2.15 6
  scale 1.135 1 1
  children [
    Inline {
      url ["ParedeFrentePrincipal.wrl"]
    }
  ]
}
Transform {
  translation 1.8 -0.1 0
  children [
    Inline {
      url ["Teto1.wrl"]
    }
  ]
}
Transform {
  translation 1.8 0 0
  children [
    Inline {
```

```

        url ["Teto2.wrl"]
    }
}
Transform {
    translation -4 6 4
    scale 2 2 2
    children [
        Inline {
            url ["TorreCompleta.wrl"]
        }
    ]
}
Transform {
    translation 2.3 6.8 6
    scale 1.25 0.85 1
    children [
        Inline {
            url ["TexturaParedPrincipal.wrl"]
        }
    ]
}
Transform {
    translation 7.5 5.6 5.8
    scale 0.8 0.8 0.8
    children [
        Inline {
            url ["TorreFrenteLat1.wrl"]
        }
    ]
}

```

```
#VRML V2.0 utf8
#Janela Principal Frente Menor
```

```
Transform {
  scale 0.80 0.80 1
  children [
    Inline {
      url ["PortalEntPrincipal.wrl"]
    }
    Inline {
      url ["TexturaBloco2.wrl"]
    }
    Transform {
      translation 0 0.8 0.08
      children [
        Shape {
          appearance Appearance {
            texture ImageTexture {
              url ["TexturaJanela.jpg"]
            }
          }
          geometry Box { size 1 1.9 0.02 }
        }
      ]
    }
  ]
}
```

#VRML V2.0 utf8

#Porta Azul Frente Principal

```
Transform {
  translation 0.5 0 0 # 0.5 0 0
  children [
    DEF TR1 Transform {
      children [
        Transform {
          translation -0.25 0.8 0 # -0.5 0.8 0
          children [
            Shape {
              appearance Appearance {
                material Material {
                  diffuseColor 0 0 1
                }
              }
            }
            geometry Box {
              size 0.5 2 0.1 # 1 2 0.1
            }
          ]
        }
      ]
    }

    Transform {
      translation -0.40 0.8 0 # -0.85 0.8 0
      children [
        Transform {
          rotation 1 0 0 1.57
          children [
            Shape {
              appearance Appearance {
                material Material {
                  diffuseColor .21 .21 0
                  specularColor .93 .58 0
                  emissiveColor .07 .05 0
                  ambientIntensity .00667
                  shininess .06
                }
              }
            }
            geometry Cylinder {
              radius 0.08
              height 0.15
            }
          ]
        }
      ]
    }
  ]
}

Shape {
  appearance Appearance {
    material Material {
      diffuseColor .1 .1 .1
      specularColor .74 .6 .74
      ambientIntensity .25
      shininess .11
    }
  }
  geometry Box {
    size 0.04 0.08 0.16
  }
}

DEF TouSen TouchSensor {
  enabled TRUE
}

DEF AbrePorta TimeSensor {
  cycleInterval 3
  loop FALSE
}

DEF AbrePortaOI OrientationInterpolator {
  key [ 0 1 ]
}
```

```

        keyValue [ 0 1 0 0, 0 1 0 -1.57 ]
    }
    DEF TouSen2 TouchSensor {
        enabled TRUE
    }
    DEF FechaPorta TimeSensor {
        cycleInterval 3
        loop FALSE
    }
    DEF FechaPortaOI OrientationInterpolator {
        key [ 0 1 ]
        keyValue [ 0 1 0 -1.57, 0 1 0 0 ]
    }
}
]
}

```

```

DEF Abrir Script { # ToggleDoor
    field SFBool state FALSE
    eventIn SFTIME toggle
    eventOut SFTIME startClose
    eventOut SFTIME startOpen
    url "vrmlscript:
        function toggle(curtime) {
            state = !state;
            if (state) {
                startOpen = curtime;
                startClose = -1;
            }
            else
            {
                startOpen = -1;
                startClose = curtime;
            }
        }
    "
}

```

```

ROUTE TouSen.touchTime TO Abrir.toggle
ROUTE Abrir.startClose TO FechaPorta.startTime
ROUTE Abrir.startOpen TO FechaPorta.startTime
ROUTE FechaPorta.fraction_changed TO FechaPortaOI.set_fraction
ROUTE FechaPortaOI.value_changed TO TR1.set_rotation
ROUTE FechaPorta.fraction_changed TO FechaPortaOI.set_fraction
ROUTE FechaPortaOI.value_changed TO TR1.set_rotation

```

```
#VRML V2.0 utf8
#Torre Completa
Viewpoint {
  position 0 5 10
  orientation 0 0 -1 0
}
```

FARN

```
Transform {
  translation -0.1 0 0
  children [
    Inline {
      url ["Torre.wrl"]
    }
    DEF JanTorre Transform {
      translation 0.02 0.5 1
      scale 0.6 0.6 0.6
      children [
        Inline {
          url ["JanelaTorre.wrl"]
        }
      ]
    }
  ]
}
```

```
Transform {
  rotation 0 1 0 1.57
  children USE JanTorre
}
```

```
Transform {
  rotation 0 1 0 3.14
  children USE JanTorre
}
```

```
Transform {
  rotation 0 1 0 4.71
  children USE JanTorre
}
```

```

]
}
Transform {
  translation 0.58 -1.1 0.97
  scale 0.5 1 0.5
  children [
    Inline {
      url ["BlocoTorre2.wrl"]
    }
  ]
}
```

```
Transform {
  translation 0.58 -1.1 -0.97
  scale 0.5 1 0.5
  children [
    Inline {
      url ["BlocoTorre2.wrl"]
    }
  ]
}
```

```
Transform {
  translation -1.30 -1.1 0.97
  scale 0.5 1 0.5
  children [
    Inline {
      url ["BlocoTorre2.wrl"]
    }
  ]
}
```

```
Transform {
  translation -1.30 -1.1 -0.97
  scale 0.5 1 0.5
  children [
    Inline {
      url ["BlocoTorre2.wrl"]
    }
  ]
}
```

```

Transform {
  translation -0.1 1.7 0
  children [
    Shape {
      appearance Appearance {
        material Material {
        }
      }
      geometry Sphere {
        radius 1.0
      }
    }
  ]
}

Transform {
  translation -0.1 2.9 0
  children [
    Shape {
      appearance Appearance {
        material Material {
        }
        texture ImageTexture {
          url ["TexturaTopo2.jpg"]
        }
      }
      geometry Cylinder {
        radius 0.7
        height 1.0
      }
    }
  ]
}

Transform {
  translation -0.1 3.5 0
  children [
    Shape {
      appearance Appearance {
        material Material {
        }
      }
      geometry Sphere {
        radius 0.7
      }
    }
  ]
}

Transform {
  translation -0.1 5.0 0
  children [
    Shape {
      appearance Appearance {
        material Material {
        }
      }
      geometry Cylinder {
        radius 0.02
        height 2
      }
    }
  ]
}

Transform {
  translation -0.1 6.25 0
  rotation 0 1 0 0
  children [
    Shape {
      appearance Appearance {
        material Material {
        }
        texture ImageTexture {
          url ["Galo.jpg"]
        }
      }
    }
  ]
}

```

```

    }
    geometry Box {
        size 0.5 0.5 0.5
    }
}
]
}
Transform {
    translation 0 -0.55 0.95
    children [
        Shape {
            appearance Appearance {
                material Material {
                }
                texture ImageTexture {
                    url ["TexturaTorreFrente02.JPG"]
                }
            }
            geometry Box {
                size 2 0.2 0.2
            }
        }
    ]
}

```

#VRML V2.0 utf8
#Porta Principal Frente Maior

FARN

```
Transform {  
  children [  
    Inline {  
      url ["C:/AnaLygia/PortalEntPrincipal.wrl"]  
    }  
    Inline {  
      url ["C:/AnaLygia/PortaAzulFrentePrincipal.wrl"]  
    }  
    Inline {  
      url ["C:/AnaLygia/PortaAzulFrentePrincipal2.wrl"]  
    }  
    Transform {  
      translation 0 0.8 -0.05  
      children [  
        Shape {  
          appearance Appearance {  
            material Material {  
              diffuseColor 0.2 0 0  
            }  
          }  
          geometry Box {  
            size 1 2 0.1  
          }  
        }  
      ]  
    }  
  ]  
}
```

#VRML V2.0 utf8

#Rampa

```
Shape {
  appearance Appearance {
    material Material {
    }
    texture ImageTexture {
      repeatS TRUE
      repeatT TRUE
      url ["piso3exterior.jpg"]
    }
    textureTransform TextureTransform {
    }
  }
  geometry Box {
    size 1 0.2 1.5
  }
}
```

#VRML V2.0 utf8
#RodaPé

FARN

```
Transform {
  translation 0.6 0 0
  rotation 0 0 1 1.57
  children [
    DEF Bloco Shape {
      appearance Appearance {
        material Material {
          diffuseColor 1 1 1
        }
        texture ImageTexture {
          url ["C:/AnaLygia/ardosialamin.jpg"]
        }
      }
      geometry Box {
        size 0.2 0.4 0.2
      }
    }
  ]
}

Transform {
  translation 1.0 0.0 0
  rotation 0 0 1 1.57
  children USE Bloco
}

Transform {
  translation 1.4 0.0 0
  rotation 0 0 1 1.57
  children USE Bloco
}

Transform {
  translation 1.8 0.0 0
  rotation 0 0 1 1.57
  children USE Bloco
}

Transform {
  translation 2.2 0.0 0
  rotation 0 0 1 1.57
  children USE Bloco
}

Transform {
  translation 2.6 0.0 0
  rotation 0 0 1 1.57
  children USE Bloco
}

Transform {
  translation 3.0 0.0 0
  rotation 0 0 1 1.57
  children USE Bloco
}
```

```
#VRML V2.0 utf8
#Textura Bloco
```

```
Transform {
  translation 0.6 0 0
  children [
    DEF Bloco Shape {
      appearance Appearance {
        material Material {
          diffuseColor 1 1 1
        }
        texture ImageTexture {
          url ["C:/AnaLygia/ardosialamin.jpg"]
        }
      }
      geometry Box {
        size 0.2 0.4 0.2
      }
    }
  ]
}
```

#VRML V2.0 utf8
Parede Lateral Frente

```
Transform {  
  children [  
    Inline {  
      url ["ParedeLateral2.wrl"]  
    }  
  ]  
}  
Transform {  
  translation 0 0 0.2  
  children [  
    Inline {  
      url ["JanelaFrenteLateral.wrl"]  
    }  
  ]  
}  
Transform {  
  translation -1.8 0 0.2  
  children [  
    Inline {  
      url ["JanelaFrenteLateral.wrl"]  
    }  
  ]  
}  
Transform {  
  translation 1.8 -1.4 0.2  
  children [  
    Inline {  
      url ["PortaCompleta3.wrl"]  
    }  
  ]  
}
```

FARN

#VRML V2.0 utf8

#Altar

```
Transform {
  translation -0.2 0 0
  scale 2 1 1
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 1 1 1
        }
      }
      geometry Box {
        size 5 8 0.2
      }
    }
    Transform {
      translation 0 -0.2 0
      children [
        Shape {
          appearance Appearance {
            material Material {
              diffuseColor 1 0 1
            }
          }
          texture ImageTexture {
            url ["TextAltar1.jpg"]
          }
          geometry Box {
            size 2.3 7.6 0.2
          }
        }
      ]
    }
  ]
}

Transform {
  translation -1.75 -0.2 0
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 1 0 0
        }
      }
      texture ImageTexture {
        url ["Altar9.jpg"]
      }
      geometry Box {
        size 1.5 4.5 0.2 # 1.5 5.06 0.2
      }
    }
  ]
}

Transform {
  translation 1.75 -0.2 0
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0 0 1
        }
      }
      texture ImageTexture {
        url ["Altar8.jpg"]
      }
      geometry Box {
        size 1.5 4.5 0.2
      }
    }
  ]
}
```

```

    }
    Transform {
        translation -1.75 -3.0 0
        children [
            Shape {
                appearance Appearance {
                    material Material {
                        diffuseColor 1 0 0
                    }
                    texture ImageTexture {
                        url ["Altar7.jpg"]
                    }
                }
                geometry Box {
                    size 1.5 2.0 0.2 # 1.5 5.06 0.2
                }
            }
        ]
    }
}

```

```

Transform {
    translation 1.75 -3.0 0
    children [
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0 0 1
                }
                texture ImageTexture {
                    url ["Altar7.jpg"]
                }
            }
            geometry Box {
                size 1.5 2.0 0.2
            }
        }
    ]
}

```

```

}
Transform {
    scale 3.5 3.5 3.5
    translation -0.2 -3.2 0.2
    children [
        Inline {
            url ["JanelaTextural.wrl"]
        }
        Inline {
            url ["JanelaRotTextural.wrl"]
        }
    ]
}

```

```

#VRML V2.0 utf8
#Piso Interior
Group {
  children [
    Shape {
      appearance Appearance {
        texture ImageTexture {
          repeatS TRUE
          repeatT TRUE
          url ["TexturaPisoInt.jpg"]
        }
        textureTransform TextureTransform {
          scale 12 15
        }
      }
      geometry Box {      size 12 0.2 12      }
    }
  ]
}
DEF Bancol Transform {
  translation 1.5 1 0.3    # -0.5 1 0.3
  rotation 0 1 0 3.14
  scale 1.0 1 1    # 1.5 1 1
  children [
    Inline {
      url ["banco.wrl"]
    }
  ]
}
DEF Banco2 Transform {
  translation 5.5 1 0.3
  rotation 0 1 0 3.14
  scale 1.0 1 1
  children [
    Inline {
      url ["banco.wrl"]
    }
  ]
}
Transform {
  translation 0 0 1.5
  children USE Bancol
}
Transform {
  translation 0 0 3.0
  children USE Bancol
}
Transform {
  translation 0 0 4.5
  children USE Bancol
}
Transform {
  translation 0 0 1.5
  children USE Banco2
}
Transform {
  translation 0 0 3.0
  children USE Banco2
}
Transform {
  translation 0 0 4.5
  children USE Banco2
}
Transform {
  translation 0 0 -1.5
  children USE Bancol
}
Transform {
  translation 0 0 -1.5
  children USE Banco2
}

```

```
Transform {  
  translation 0 0 -3.0  
  children USE Bancol  
}  
Transform {  
  translation 0 0 -3.0  
  children USE Banco2  
}  
Transform {  
  translation 0 0 -4.5  
  children USE Bancol  
}  
Transform {  
  translation 0 0 -4.5  
  children USE Banco2  
}  
  
]  
}
```

```
#VRML V2.0 utf8
#Piso Exterior Frente
Shape {
  appearance Appearance {
    texture ImageTexture {
      repeatS TRUE
      repeatT TRUE
      url ["Ceramica4.jpg"]
    }
    textureTransform TextureTransform {
      scale 8 12
    }
  }
  geometry Box {    size 8 0.2 12  }
}
```