



Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

**Reconstrução de ambientes históricos utilizando
VRML: o caso do
Fórum Flaviano de Conimbriga**

Alexandrino José Marques Gonçalves

Coimbra
Dezembro de 2002

Dissertação submetida à
Universidade de Coimbra
para obtenção do grau de
Mestre em Engenharia Informática
Área Científica de Sistemas Multimédia na
Educação e Formação Profissional

**Reconstrução de ambientes históricos utilizando
VRML: o caso do
Fórum Flaviano de Conimbriga**

Alexandrino José Marques Gonçalves

Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática
Dezembro de 2002

Dissertação realizada sob a orientação do
Professor Doutor António José Nunes Mendes
Professor Auxiliar
do Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia
da Universidade de Coimbra

À Memória do meu pai

*“These reconstructions open the door not just to the buildings,
but to the life that people lived back then.”*

Jeffrey Jacobson

Resumo

Cada vez mais, a utilização das denominadas “novas tecnologias” na preservação e divulgação do património histórico e cultural da humanidade é uma realidade. No entanto, o acesso a alguns desses projectos, nomeadamente os que envolvem a utilização de técnicas de Realidade Virtual, é por norma bastante restrito e limitado, em virtude das especificidades técnicas usadas no seu desenvolvimento e/ou na sua visualização, que tornam inviável a disponibilização dos mesmos ao público em geral, por exemplo através da Internet.

O Fórum Flaviano de Conimbriga, um dos monumentos mais emblemáticos da antiga cidade romana de Conimbriga, serviu de base a este nosso trabalho. O nosso objectivo foi disponibilizar uma reconstrução virtual deste monumento, mas fazê-lo de forma a torná-la acessível a qualquer pessoa através da Web, contribuindo para uma verdadeira democratização do saber. Assim, qualquer pessoa tem a possibilidade de efectuar uma visita virtual interactiva a um espaço grandioso e imponente, que já não se pode contemplar por se encontrar destruído quase na sua totalidade.

Normalmente na Realidade Virtual disponível na Web, existem três conceitos aparentemente incompatíveis entre si: realismo, desempenho e tamanho do(s) ficheiro(s). No entanto, através da linguagem utilizada no desenvolvimento deste trabalho, o VRML, e potencializando as suas capacidades técnicas, principalmente ao nível da optimização, conseguiu-se aliar esses três conceitos num só trabalho:

- ✓ Realismo: aprovado pelo Director do Museu Monográfico de Conimbriga;
- ✓ Desempenho: funciona perfeitamente num qualquer equipamento informático doméstico adquirido nos nossos dias;
- ✓ Dimensão dos ficheiros: o ficheiro principal (único) VRML, ocupa apenas 22Kb, possibilitando assim a disponibilização deste trabalho sem dificuldades através da Web (<http://ism.dei.uc.pt/forum>).

Abstract

The use of new technologies in the preservation and publicity of the historical and cultural heritage of Humanity has become an increasing reality. However, access to some of these projects, namely the ones involving the use of Virtual Reality techniques, is normally rather restricted and limited on account of technical specificities used in its development and/or visualization. Availability to the general public, for instance through the Internet, becomes, then, impracticable.

The prospective element under study is the Flavian Forum, one of the most outstanding monuments in the ancient roman city of Conimbriga. The aim of this study is to develop a virtual reconstruction of this monument in the most accessible form to the general public through the Web, thus contributing to a true democratisation of knowledge. Therefore, anyone can interactively proceed in a virtual visit to a vast and grand space that is no longer available to us, as it is practically destroyed.

There are normally three apparently incompatible concepts in Virtual Reality in the Web: realism, performance and file size. However, we have managed to conciliate these three concepts in this study, through the use of specific language used in this project, the VRML, thus enhancing its technical capacities at the level of optimisation:

- ✓ Realism: approved by the Director of the Monographic Museum of Conimbriga;
- ✓ Performance: functions perfectly in any given domestic information system equipment available nowadays;
- ✓ File size: the main (and only) VRML file, occupies just 22Kb, so this project is easily accessed through the Web (<http://lsm.dei.uc.pt/forum>).

Agradecimentos

Momentos houve em que as vicissitudes da vida interferiram com a realização deste trabalho. Por tal motivo, a ajuda de algumas pessoas foi fundamental para neste momento estar a escrever estas palavras. Aproveito este espaço para lhes prestar o devido agradecimento.

Ao meu orientador, o Professor António Mendes, o meu muito obrigado pela disponibilidade, apoio, críticas e sugestões dadas ao longo do tempo, essenciais para que este trabalho se realizasse.

Aos responsáveis do Museu Monográfico de Conimbriga, principalmente na pessoa do seu Director o Dr. Virgílio Correia, não só as facilidades concedidas para a recolha de informação dentro de todo o recinto das Ruínas de Conimbriga, mas também pelas sugestões dadas durante a evolução do trabalho, sem as quais não seria possível realizar este trabalho nos moldes idealizados.

À Escola Superior de Tecnologia e Gestão (ESTG) do Instituto Politécnico de Leiria pelas facilidades concedidas para a realização deste trabalho, nomeadamente a concessão da dispensa de serviço docente durante um semestre. O meu agradecimento também aos colegas do Departamento de Engenharia Informática da ESTG pelo companheirismo, boa disposição e apoio dado.

Aos amigos da Granja do Ulmeiro agradeço a amizade e companhia tidas durante este tempo. Gostaria ainda de particularizar as ajudas do Adriano Esteves e o apoio da Gorete Marques.

E finalmente, aos meus pais por terem criado as condições para que possa ter chegado a este ponto. Além de toda a paciência tida durante o trabalho.

Índice

1. INTRODUÇÃO	1
2. VRML NA RECONSTRUÇÃO HISTÓRICA.....	11
2.1 VRML	11
2.1.1 Motivação	11
2.1.2 Génese.....	12
2.1.3 Evolução.....	13
2.1.4 Características	15
2.1.4.1 Nós e Campos	16
2.1.4.2 Sistema de coordenadas.....	17
2.1.4.3 Modelo de visualização.....	18
2.1.4.4 Representação do utilizador	18
2.1.5 O processo de desenvolvimento.....	18
2.2 EXEMPLOS	21
2.2.1 Introdução.....	21
2.2.2 Archeoguide.....	22
2.2.3 Viagem Virtual a Bracara Augusta	25
2.2.4 Projecto 3D Murale.....	27
2.2.5 Wat Ratchaburana.....	29
2.2.6 O Labirinto de Hawara	31
2.2.7 A Basílica de Fano	32
2.2.8 Acampamento Romano	34
2.2.9 Huys Hengelo	35
2.2.10 Santa Clara-a-Velha Virtual.....	37
3. RECONSTRUÇÃO DO FÓRUM FLAVIANO DE CONIMBRIGA.....	41
3.1 DESCRIÇÃO	41
3.2 O PROJECTO	44
3.2.1 Investigação.....	44
3.2.2 Ferramenta de desenvolvimento.....	45
3.2.3 Modelação	47
3.2.4 Assemblagem.....	49
3.2.5 Texturas	50
3.2.5.1 Coluna.....	51
3.2.5.2 Lajeado.....	54
3.2.5.3 Parede.....	55
3.2.5.4 Gradeamento	56
3.2.6 Coloração	57
3.2.7 Iluminação	57
3.2.8 Optimização.....	59
4. ASPECTOS DE OPTIMIZAÇÃO.....	61
4.1 VISUALIZAÇÃO 3D (EM VRML).....	61
4.1.1 Introdução.....	61
4.1.2 Detecção de colisões.....	62
4.1.3 Gravidade	64
4.1.4 <i>Culling</i>	66
4.1.5 Remoção de superfícies ocultas	68
4.1.6 Iluminação	68
4.1.7 Normais.....	70
4.1.8 Sensores	71
4.1.9 Áudio.....	73
4.2 OPTIMIZAÇÃO EM VRML	74
4.2.1 Introdução.....	74
4.2.2 Hierarquização	75
4.2.3 Colisões.....	78
4.2.4 Instanciação (Clonagem)	80

4.2.5 <i>Inlining</i>	82
4.2.6 Iluminação.....	85
4.2.7 Número de Polígonos.....	88
4.2.8 Crease Angle.....	90
4.2.9 Texturas	91
4.2.10 <i>Bilboards</i>	94
4.2.11 LOD.....	95
4.2.12 Animações	96
4.2.13 Áudio.....	98
4.2.14 Primitivas simples	100
4.2.15 Propriedades do Avatar	102
4.2.16 Nevoeiro	104
4.2.17 Âncoras	105
4.2.18 Culling	106
4.2.19 <i>Viewpoints</i>	107
4.2.20 Organização espacial	108
4.2.21 Mudança de escala.....	109
4.2.22 Compressão.....	109
4.2.23 <i>Scripting</i>	111
4.2.24 Exportar para VRML	112
4.2.25 Memória	114
4.2.26 <i>Browser</i>	115
4.3 OPTIMIZAÇÕES EFECTUADAS NESTE TRABALHO	116
4.3.1 Introdução	116
4.3.2 Hierarquização	117
4.3.3 Colisões	118
4.3.4 Instanciação (Clonagem)	120
4.3.5 Iluminação.....	122
4.3.6 Número de Polígonos.....	122
4.3.7 Texturas.....	123
4.3.8 <i>Bilboards</i>	124
4.3.9 Animações	126
4.3.10 Primitivas simples.....	126
4.3.11 Propriedades do Avatar	128
4.3.12 Âncoras.....	128
4.3.13 <i>Viewpoints</i>	129
4.3.14 Organização espacial	129
4.3.15 Compressão	129
4.3.16 Exportar para VRML	130
4.3.17 <i>Browser</i>	130
5. CONCLUSÃO	133
6. BIBLIOGRAFIA.....	137

Índice de Figuras

Figura 1.1 Imagem representativa do Fórum Flaviano de Conimbriga	4
Figura 1.2 Fotografia aérea das ruínas do Fórum	5
Figura 2.1 Diagrama ilustrativo das etapas no processo de desenvolvimento	21
Figura 2.2 Imagem real a). Imagem gerada por realidade aumentada b)	23
Figura 2.3 Imagem representativa das termas romanas de Bracara Augusta	27
Figura 2.4 O teatro romano de Sagalassos no seu estado actual (esq.) e um modelo VRML de uma parte de Sagalassos (dir.)	29
Figura 2.5 Imagem inicial (esq). Imagem após a reconstrução (dir).....	30
Figura 2.6 Modelo de Petrie a) e modelo de Arnold b).....	32
Figura 2.7 Possíveis plantas de historiadores antigos: Frá Giocondo (esq.) e Andrea Palladio (dir.)	33
Figura 2.8 Imagem ilustrativa do acampamento romano.....	35
Figura 2.9 Imagem do modelo virtual do Castelo de Huys Hengelo.....	37
Figura 2.10 Imagens do Mosteiro de Santa Clara-a-Velha Virtual	38
Figura 3.1 Dois exemplos inseridos na categoria dos <i>Scene editors</i> . A aplicação da Sony (esq.) e o Spazz3D (dir.).....	45
Figura 3.2 Interface do Cosmo Worlds 2.0.....	46
Figura 3.3 A parede deste corredor segue um padrão geométrico repetitivo	48
Figura 3.4 Exemplo de peça modelada no 3Dstudio. O seu interior é visível, ou seja a peça é oca	49
Figura 3.5 Processo de montagem gradual, inicialmente com módulos adjacentes	50
Figura 3.6 Imagem do fórum com e sem texturas	51
Figura 3.7 Exemplo de um capitel coríntio.....	52
Figura 3.8 Imagem da foto adquirida. Não na sua dimensão original.....	53
Figura 3.9 Aspecto final da coluna	54
Figura 3.10 Textura usada no lajeado da praça.....	55
Figura 3.11 Textura usada como padrão das paredes.....	56
Figura 3.12 Textura do gradeamento	56
Figura 3.13 O antes e depois da coloração	57
Figura 3.14 Com o foco desligado (esq.). Com ele ligado (dir.).....	59
Figura 4.1 Esquema em árvore da estrutura hierárquica do braço humano	75
Figura 4.2 Modelo facetado (esq.). Mesmo modelo, apenas com alteração do <i>creaseAngle</i>	91
Figura 4.3 Exemplo construído apenas com recurso a primitivas simples	101
Figura 4.4 Estrutura hierárquica expressa no Cosmo Worlds.....	118
Figura 4.5 Zona, indicada a vermelho, sujeita à optimização das colisões	120
Figura 4.6 Bloco, único, modelado para usar nas paredes do pátio.....	121
Figura 4.7 A peça indicada foi obtida por clonagem.....	122
Figura 4.8 Textura usada no lajeado da praça.....	124
Figura 4.9 Como se pode ver todas as colunas estão de frente para o utilizador/visitante	126
Figura 4.10 Porta usada no Fórum	127
Figura 4.11 Exemplos de peças construídas por intersecção de primitivas simples. Um pódio (esq.) e um ornamento da entrada (dir.)	127
Figura 4.12 Posicionamento interno dos objectos no Cosmo Worlds.....	129

1. Introdução

Em 1948, o Boletim da Direcção Geral dos Edifícios e Monumentos Nacionais escrevia: “...ao cabo de alguns séculos de obscura existência...os romanos e velhos lusitanos largamente fizeram prosperar com o nome de Conimbriga...foi ali, naquelas esquecidas terras de Condeixa-a-Velha, que floresceu até ao fim do século V da nossa era uma das mais civilizadas cidades da Lusitânia romanizada...e o mais importante agregado social de todo território (hoje português) compreendido entre o Douro e o Tejo...É desse período que data...aquelas ruínas que tão evocadas como as de Pompeia, se erguem no rijo solo beirão, ao sul de Coimbra, já longe do Mondego...foram descobertos, ao longo das escavações iniciadas em 1930, numerosos vestígios, cuja importância excedem então as mais confiantes expectativas...Tudo isto é suficiente para demonstrar que...a velha Conimbriga foi deveras uma cidade de prazer e de trabalho...podemos conceber sem dificuldade o sobressalto ali causado, nos primeiros anos do século V da nossa era, pela aproximação dos chamados «bárbaros do Norte»... Foi por esse tempo e por tal motivo que se construíram aodadamente muitos dos lanços de muralha...essa obra formidável...não logrou impedir a catástrofe que todos procuravam conjurar. Na segunda metade do século V, os suevos de além Douro, acelerando a sua marcha de conquistadores para sul...e após uma *circumsessio* implacável...a cidade caiu finalmente em seu poder... fixa no ano de 468 esse acontecimento memorável, de que resultou a destruição do belo *oppidum* luso-romano e dos seus habitantes – pois estes, mortos ou cativos, desapareceram então para sempre.”.

É desta forma um pouco dramática que, em 1948, se dava a conhecer o nascimento, ascensão e queda daquela que foi uma das maiores e mais prósperas cidades em território “Lusitano” construídas na época do glorioso, ou será impiedoso, Império Romano. De qualquer maneira, este passado não pode ser ignorado, pois também nele se alicerçaram alguns dos valores, hábitos e costumes da nossa sociedade. Dado o crescente interesse que têm suscitado, quer a organizações internacionais, quer à população em geral, as questões relacionadas com o passado cultural e arquitectónico da humanidade, o trabalho

desenvolvido nesta dissertação tenta dar uma contribuição para um melhor conhecimento de uma realidade da qual não podemos ficar indiferentes.

Durante a era de conquista e expansão, o Império Romano consolidava a sua presença nos territórios ocupados, através da construção de cidades habilmente localizadas e inteligentemente interligadas por um bem estruturado conjunto de vias de comunicação, sempre com o ensejo de aglomerar as populações, para um mais eficaz “enraizamento” dos valores e desígnios de Roma. Na concepção das estruturas físicas dessas cidades, era comum existir uma que, pelo seu papel e importância na vida social, jurídica, política e religiosa da cidade estaria localizado em local de destaque, e seria sempre uma das estruturas mais imponentes e representativas dessa cidade. O Fórum, tal como era conhecido, seria, até pelas suas dimensões consideráveis, um local privilegiado de encontro, comércio e culto. Apesar da sua terminologia singular, o Fórum era invariavelmente constituído por vários monumentos e áreas distintas, de entre os quais se destacam o templo, os pórticos, a praça e umas colunas magistrais que, pelo seu número e localização, lhe davam um cariz imponente e imperial. As próprias dimensões dos monumentos eram concebidas de tal forma que vistas à altura de um homem comum tomam um relevo impressionante.

É esta importância histórica, cultural e arquitectónica atribuída aos Fóruns das antigas cidades romanas que motiva o trabalho que nos propusemos efectuar.

As doze campanhas de escavações, levadas a cabo entre 1964 e 1971 nas ruínas de Conimbriga pela Missão Arqueológica Francesa em Portugal e o Museu Monográfico de Conimbriga, conduziram, em particular, à descoberta das fundações do Fórum outrora existente nesta secular cidade romana. O Fórum Flaviano de Conimbriga foi mandado edificar entre os anos 75 e 80 da nossa era, pelo município da cidade de Conimbriga no reinado de *Flavius Vespasianus* para dedicar ao Culto Imperial. Com efeito, a revolução arquitectónica levada a cabo na altura, com a destruição de alguns monumentos da época Augustana¹ existentes naquele local, teve como fundamento a criação de um fórum “Imperial” onde só a majestade do soberano triunfa. Para tal, nada foi deixado ao acaso: desde a subtileza do jogo matemático das dimensões à relação entre espaços cobertos e espaços livres, em nada ocasional (Alarcão et al., 1994).

¹ Relativo ao antigo imperador romano

O Fórum Flaviano de Conimbriga era composto por 3 monumentos: O templo, os pórticos do templo e os pórticos do fórum (praça). O templo do culto imperial ergue-se no ponto mais alto da cidade, bem no centro de uma área sagrada, pelo que a sua localização e características, lhe conferem dimensões imponentes, tal como foi desejado. Toda esta área sagrada, está ela própria delimitada por um pórtico em U cuja função é essencialmente arquitetónica, onde mais uma vez a imponência e harmonia com as restantes volumetrias são factores primordiais. A praça, único monumento que se manteve da época Augustana, era o espaço cívico por excelência. Os seus dois pórticos laterais situam-se a um nível inferior relativamente aos pórticos do templo e o pavimento é agora revestido por um lajeado distribuído por fiadas. Para romper um pouco com a monotonia das lajes meticulosamente alinhadas, foram implantados novos plintos² honoríficos. Esses plintos têm geralmente uma forma rectangular e “não estão implantados em perfeita simetria, reflectindo uma certa desordem que é a imagem da vida, da riqueza dos doadores e da posição política e social daqueles que se homenageava” (Alarcão et al., 1994). Estes estão sempre dispostos a delimitarem espaço mais que suficiente para a circulação e, sobretudo, de modo nenhum mascaram a visão do templo do culto imperial. Ao fundo da praça existia um patamar intermédio de acesso ao templo que, com uma dupla escadaria nos extremos, permitia formas de circular distintas: dois cortejos que podiam encontrar-se e abordar a grande escadaria do templo em simultâneo, ou então permitir uma fila ascendente e outra descendente para canalizar a multidão de fiéis, aquando da realização de procissões em honra dos seus deuses.

² Ornamentos existentes na praça

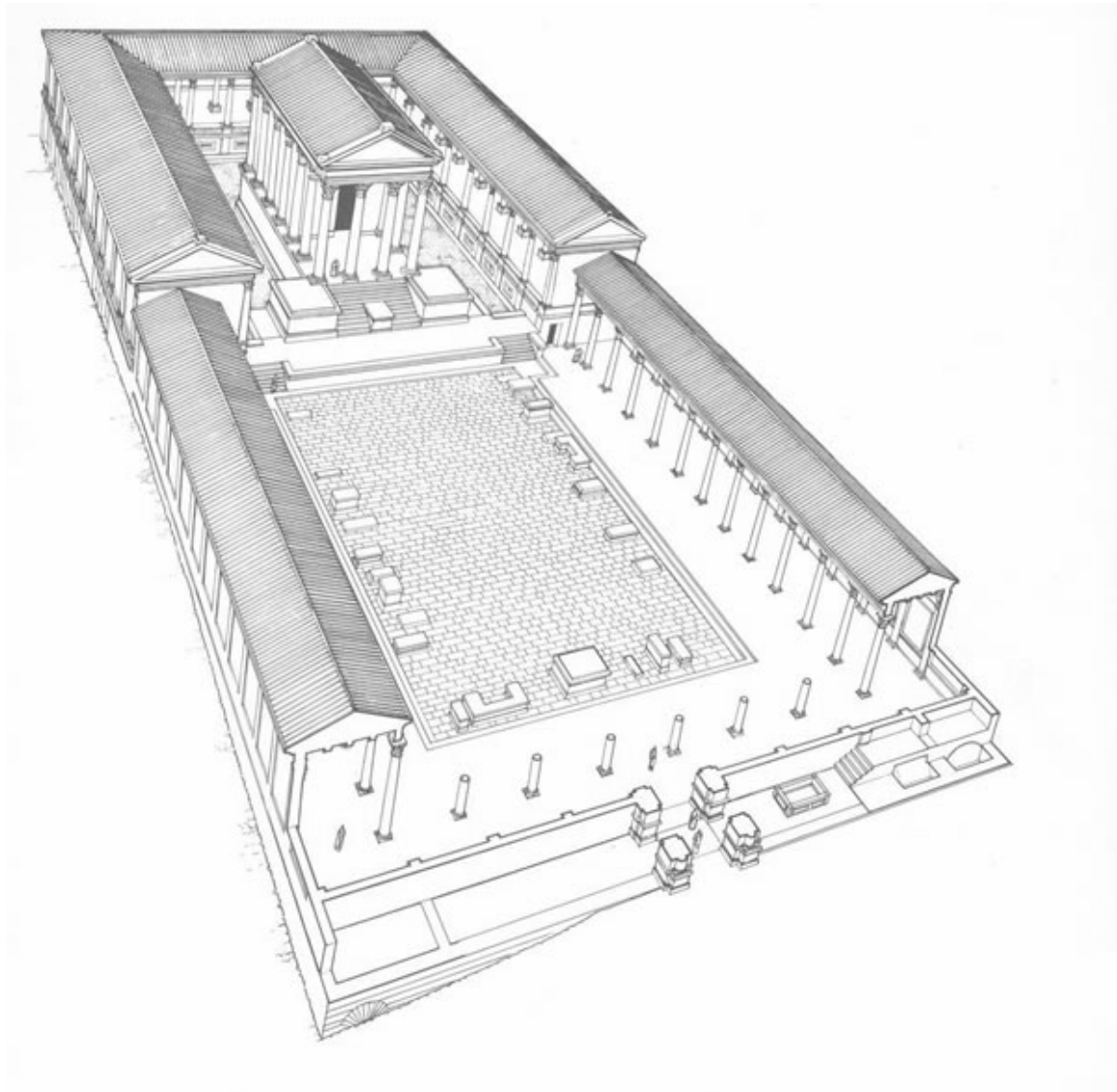


Figura 1.1 Imagem representativa do Fórum Flaviano de Conimbriga

Em 1899, a Rainha D. Amélia deu ordens para iniciar umas escavações arqueológicas no sítio de Conimbriga (Direcção Geral dos Edifícios e Monumentos Nacionais, 1948) mas foi a partir de 1930 que essas escavações foram complementadas e aprofundadas, de tal modo que acabaram por “colocar a nu” vestígios inegáveis de uma cidade construída pela civilização romana. A quantidade e qualidade dos achados arqueológicos apresentavam uma riqueza arquitectónica e cultural de tal ordem que se considerou por bem disponibilizar ao público em geral o contacto com tal realidade, consideravelmente distinta da dos nossos dias. De onde resultaram as agora bastante apreciadas e famosas, mesmo além fronteiras, Ruínas de Conimbriga.

Um qualquer visitante deste espaço, ao deparar-se com todo o cenário de edifícios em ruínas, pavimentos degradados ou colunas danificadas, acaba por,

inevitavelmente, ser inconscientemente “transportado” numa viagem no tempo imaginária, onde tenta recriar, através do conjunto de sensações assimiladas naquele momento, todo o ambiente urbanístico, social e arquitectónico vivido naquela época. Dada a distinta capacidade de abstracção de cada ser humano, ainda que fundamentada com desenhos e maquetas, essa tarefa de recriação imaginária estará sempre repleta de imprecisões próprias da interpretação individual de cada um, e que podem variar consoante o estado de espírito, idade ou até o nível cultural de cada pessoa.

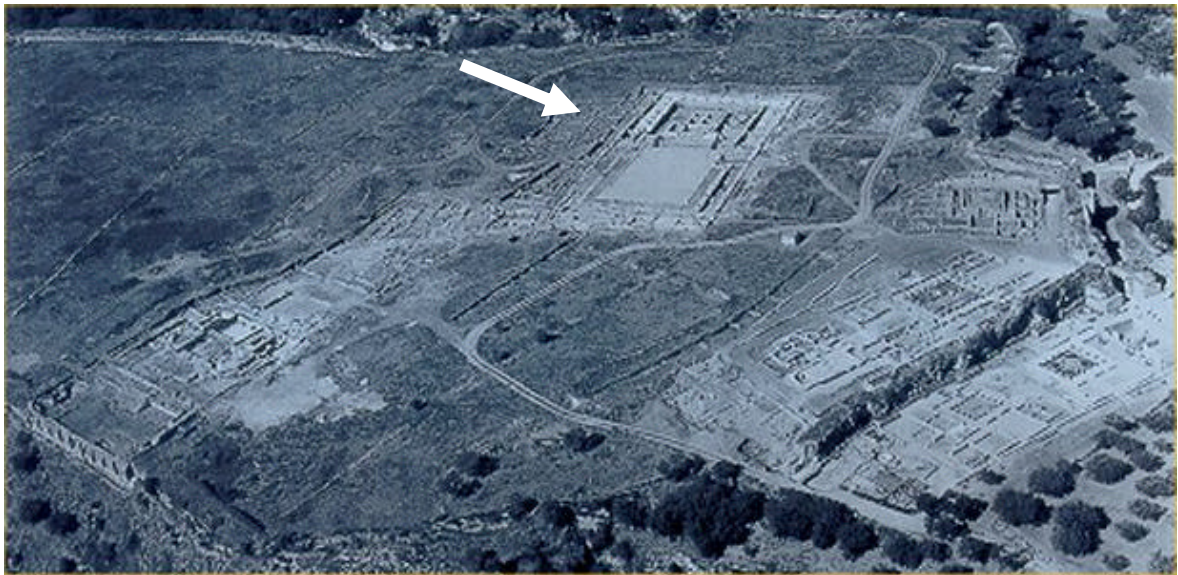


Figura 1.2 Fotografia aérea das ruínas do Fórum

Este trabalho tem por finalidade possibilitar a qualquer pessoa desfrutar de uma visita, ainda que virtual, a um dos locais mais emblemáticos de grande parte das cidades do antigo Império Romano, o Fórum. O que, neste caso, se traduziu na recriação tridimensional o mais fidedigna possível (número de degraus das escadarias, dimensões das lajes, número de colunas, padrões da época, cores, etc.) e à escala real, do Fórum Flaviano de Conimbriga, descrito anteriormente.

Nesta fase, torna-se pertinente a questão: porquê “apenas” o Fórum, quando as ruínas de Conimbriga revelam ao público que as visita um espólio de belezas e raridades arquitectónicas, talvez até mais merecedoras de igual tratamento? A resposta é simples, pois não existe informação exacta sobre as dimensões e volumetrias dos restantes edifícios. Assim, e de acordo com o Director do Museu Monográfico de Conimbriga, foi decidido reconstruir “apenas” o Fórum Flaviano, por não ser intenção do próprio e arqueólogos efectuar recriações baseadas em

suposições. Com efeito, os trabalhos efectuados no terreno durante as escavações de meados da década de 60, permitiram descobrir que “a densidade dos pontos altimétricos registados contribuiu eficazmente para a elaboração da planta esquemática, do mesmo modo que para o desenho dos cortes longitudinais (norte-sul) ou transversais (este-oeste). Assim, o levantamento do conjunto flaviano, executado no terreno pelo topógrafo e pelo desenhador, é uma tradução exacta da realidade” (Alarcão et al., 1994). Deste modo, foram as plantas elaboradas durante esse período e gentilmente cedidas pelos responsáveis do Museu, a nossa base de trabalho para a “edificação” do Fórum Flaviano de Conimbriga. A implementação deste trabalho foi, assim, feita apenas com base nessas plantas esquemáticas, na observação das ruínas do Fórum e no acesso a alguns dos fragmentos retirados da sua construção original.

Tendo em conta as características do projecto e os objectivos a atingir, a tecnologia a utilizar deveria permitir a obtenção de bons resultados visuais, mas ao mesmo tempo fornecer ao autor as potencialidades necessárias a um desenvolvimento eficaz. A escolha recaiu na *Virtual Reality Modeling Language* (VRML), uma linguagem que possibilita a criação de ambientes virtuais, e que permite ao utilizador, agora convertido em visitante, explorar na primeira pessoa os espaços reconstruídos à distância de um simples movimento com o rato. Para além das suas características interactivas, o VRML possui potencialidades, ao nível do realismo, factor crítico do sucesso de um trabalho desta natureza, e em termos de desempenho, aspecto que poderia limitar fortemente o acesso a um maior número de potenciais “visitantes”, devido à elevada capacidade do equipamento informático que seria necessário para a sua visualização.

Nesta linha de raciocínio, definimos um segundo grande objectivo para este trabalho, propiciar esta forma de realidade virtual com alguma qualidade, realisticamente falando, a um público mais heterogéneo e numeroso, contribuindo assim para uma efectiva democratização do saber, não restringindo o acesso a estes paradigmas de visualização a um grupo restrito de pessoas, como muitas vezes acontece. Como o VRML é standard para a publicação de conteúdos tridimensionais na *World Wide Web*, o leque de potenciais utilizadores crescerá certamente ao disponibilizar a reconstrução desta relíquia histórica num servidor Web, a partir do qual alguém no outro extremo do planeta poderá efectuar, sem

grandes dificuldades³, uma viagem no tempo e visitar um dos ex-libris do antigo império romano.

Por outro lado, são conhecidos diversos exemplos de reconstruções em VRML que, pela dimensão dos ficheiros correspondentes, tornam difícil ou impossível a sua descarga e visualização. Para evitar este problema foram utilizadas diversas técnicas de optimização, de modo a possibilitar a visualização e navegação, sem degradação do desempenho, num qualquer equipamento informático doméstico adquirido nos nossos dias. Este é um dos aspectos centrais deste trabalho, tendo sido feito um esforço significativo no sentido de obter um ambiente realista e fácil de navegar sem que isso implicasse a produção de ficheiros de grande dimensão, o que dificultaria significativamente a sua disponibilização na Web.

No últimos anos tem-se verificado uma crescente preocupação, nomeadamente por parte de grandes instituições como a UNESCO ou a União Europeia, com a preservação, interpretação e divulgação da riqueza histórica e cultural legada pelos nossos antepassados. Tal facto despontou uma nova vaga, onde a utilização de novas tecnologias, designadamente de realidade virtual, adquirem um papel cada vez de maior relevo na consecução de tais propósitos. Exemplo disso mesmo são dois grandes projectos europeus de reconstituição histórica que estão a decorrer sob a égide da União Europeia (Archeoguide e 3D Murale, que estão descritos brevemente no segundo capítulo), ou então a utilização na China de tecnologia laser para apurar da fisionomia de estátuas de guerreiros sagrados soterradas à séculos de modo a poder elaborar a sua reconstrução virtual sem ter de as desenterrar, dado que com os meios actuais seria muito difícil retirá-las do seu "jazigo" sem as danificar irremediavelmente, deixando essa tarefa para gerações vindouras que disponham de tecnologia mais avançada.

Este trabalho pretende também contribuir, mesmo que modestamente, para a preservação e melhor conhecimento de um passado que nos é familiar, seja pela proximidade geográfica, seja pelos valores que representa, o Fórum Flaviano de Conimbriga.

As especificidades próprias do monumento a ser recriado, seja pelas suas dimensões físicas, seja pelas suas características geométricas, associadas às

³ Para tal apenas será necessário instalar um pequeno plug-in VRML, disponível gratuitamente na Internet

condicionantes normalmente impostas a trabalhos realizados na área da realidade virtual, mormente ao nível da capacidade do equipamento informático necessário para a sua visualização, acarretaram um enorme desafio na elaboração deste trabalho, principalmente quando, e como foi referido anteriormente, se pretendia abranger um público alvo o mais amplo possível. Essa foi a razão pela qual emergiu um outro tipo de objectivos, estes de carácter mais técnico e relacionados com a tecnologia que iria permitir “dar corpo” ao desenvolvimento do trabalho, o VRML. No caso, foi dado especial enfoque às questões de optimização, para uma melhor e mais fluída navegabilidade e, naturalmente, à problemática da dimensão final do(s) ficheiro(s), de modo a minimizar o tempo de *download*.

Por forma a estruturar de um modo adequado e coerente todos os conteúdos abordados nesta dissertação, a mesma está organizada da seguinte forma:

- ✓ Capítulo 1, “*Introdução*”: Esta introdução. Inicialmente é feito um enquadramento geográfico e temporal da antiga cidade romana de Conimbriga. Posteriormente é efectuada uma descrição do Fórum Flaviano de Conimbriga, onde se alude à sua importância na cidade e às suas características arquitectónicas. Finalmente, são expostos os objectivos principais deste trabalho;
- ✓ Capítulo 2, “*VRML na Reconstrução Histórica*”: Este é um capítulo que apresenta um estado da arte da utilização do VRML em trabalhos ou projectos relacionados com reconstruções históricas de espaços com interesse cultural e patrimonial. Inicialmente neste capítulo, é apresentada uma pequena descrição da linguagem VRML, incluindo as suas origens e evolução, bem como as suas características principais. Nessa secção, é também apresentado um modelo que descreve as principais etapas de desenvolvimento de trabalhos/projectos em VRML;
- ✓ Capítulo 3, “*Reconstrução do Fórum Flaviano de Conimbriga*”: Neste capítulo descrevem-se as diversas etapas de desenvolvimento deste trabalho, incluindo os aspectos de investigação, as principais opções e as dificuldades mais importantes sentidas durante o mesmo;
- ✓ Capítulo 4, “*Aspectos de Optimização*”: Neste capítulo são abordadas questões relacionadas com optimização em VRML. Numa primeira secção são descritos os principais problemas que afectam o desempenho de

ambientes virtuais tridimensionais e as opções de otimização que é possível utilizar para minimizar esses problemas. De seguida são descritas as otimizações possíveis de concretizar em VRML. No final do capítulo, apresentam-se as opções de otimização adoptadas no nosso trabalho e que, na nossa opinião, levaram a um desempenho bastante satisfatório do ambiente virtual construído;

- ✓ Capítulo 5, “*Conclusão*”: Este capítulo pretende sintetizar o trabalho realizado e apresentar algumas reflexões resultantes da experiência adquirida durante o seu desenvolvimento.

2. VRML na Reconstrução Histórica

2.1 VRML

2.1.1 Motivação

No início do século XXI a comunicação assume um papel preponderante, não só como meio de conversação mas também como meio de transporte e acesso a informação, hoje em dia um dos bens mais preciosos nesta aldeia, dita global. As recentes evoluções tecnológicas permitem-nos actualmente dispor de um leque de serviços que nos põe ao corrente de tudo o que se passa em qualquer ponto do mundo.

Um desses meios de comunicação, e provavelmente um dos que mais rapidamente revolucionou os hábitos e costumes da sociedade actual, é a Internet, que permite aos seus utilizadores a possibilidade de trocarem mensagens, efectuar “visitas” a instituições bancárias sem sair de casa, acesso a um infindável conjunto de informação, desde cotações bolsistas, previsões meteorológicas ou notícias on-line.

Paralelamente a alguns desses desenvolvimentos iniciais, surgiu também o conceito de ambientes virtuais. A ideia é permitir às pessoas uma integração com determinado ambiente, seja um museu ou um monumento já inexistente, sem terem de sair de suas casas. Este conceito aliado à expansão da *World Wide Web*, e em consonância com as crescentes solicitações de projectar a Web para outro patamar (tridimensional), levou à criação do VRML.

O VRML – *Virtual Reality Modeling Language* – é uma linguagem que permite a criação de mundos virtuais com os quais é possível interagir fácil e intuitivamente. Estes mundos virtuais poderão conter objectos de vários tipos, como sejam, geometria tridimensional, imagens, sons, ligações para outros mundos, animações, entre outros. O VRML emergiu como a linguagem standard para a visualização de conteúdos tridimensionais na *World Wide Web*. Hoje em dia é uma técnica pouco dispendiosa capaz de levar a realidade virtual, desde sempre associada a grandes *mainframes*, a um qualquer computador doméstico, em

grande parte devido à evolução da capacidade de processamento do hardware verificada nos últimos anos.

2.1.2 Génese

Há alguns anos, as pesquisas na área de interfaces “sensíveis” começaram a atrair a atenção da imprensa e da indústria. Uma grande variedade de tecnologias, que vieram a ser conhecidas como “Realidade Virtual”, começaram uma mudança fundamental na natureza dos interfaces, movendo-os para um *design* centrado no utilizador, num esforço para tornar os computadores mais adequados às pessoas que os utilizavam.

Em finais de 1993, Mark Pesce e Toni Parisi desenvolveram um interface tridimensional para a Web, que fazia uso dos conhecimentos por eles adquiridos em vários anos de investigação nas áreas de realidade virtual e comunicações. Depois de comunicar estas inovações a Berners-Lee (o “pai” da Web), Pesce foi convidado a apresentar um documento na Primeira Conferência Internacional sobre a *World Wide Web*, em Genebra. Durante uma sessão de discussão sobre os interfaces de realidade virtual para a Web, os presentes concordaram na necessidade de existir uma linguagem para especificar cenas tridimensionais em conjunto com *links* Web - o análogo de HTML para realidade virtual. Surgiu então o termo *Virtual Reality Markup*⁴ *Language* (VRML) e o grupo liderado por Pesce e Brian Behlendorf (da revista *Wired*) começou a criar uma especificação para VRML logo a seguir à conferência.

Com o apadrinhamento da *Wired*, Behlendorf criou uma *mailing-list* para facilitar a discussão de uma especificação para o VRML. A resposta superou todas as expectativas. Numa semana subscreveram-se mais de mil utilizadores, que rapidamente chegaram a acordo sobre um conjunto de requisitos para VRML e começaram uma pesquisa de tecnologias que poderiam ser adaptadas para responder a essas necessidades.

⁴ Mais tarde a palavra *Markup* foi alterada para *Modeling* por melhor reflectir a natureza tridimensional do VRML

Após uma grande discussão, os utilizadores da *mailing-list* chegaram a um consenso: o formato de ficheiros ASCII do Open Inventor da Silicon Graphics, Inc. Este formato suporta descrições completas de cenas 3D, tendo todas as características necessárias à produção de trabalhos bastante realistas e de grande qualidade, bem como uma base de ferramentas auxiliares bastante popular.

Gavin Bell, da Silicon Graphics, adaptou o formato de ficheiros do Open Inventor para VRML, seguindo as orientações que vinham da *mailing-list*. A Silicon Graphics disponibilizou então o formato criado ao público em geral, contribuindo também com um interpretador de ficheiros VRML, para facilitar o desenvolvimento de visualizadores de VRML.

Surgiu assim em Novembro de 1994 o primeiro *draft* da especificação do VRML.

2.1.3 Evolução

Em Maio de 1995 ficou completa a especificação da primeira versão do VRML, com a sua concepção a pretender atingir três critérios fundamentais para uma linguagem standard de comunicações em rede: independência da plataforma, extensibilidade e capacidade de responder com eficácia a ligações de baixa velocidade.

Dado o entusiasmo e vontade inicial de um rápido surgimento desta primeira versão do VRML - em apenas 6 meses foi concluída a especificação, foi decidido que esta não iria suportar comportamentos interactivos, e desse modo seria ainda bastante limitativa. A concepção de uma linguagem para descrever comportamentos interactivos é um processo moroso especialmente quando a linguagem necessita de descrever comportamentos de objectos que comunicam através de uma rede. Embora tais linguagens já existissem, aquando da criação do VRML, a escolha de uma delas poderia ter levado a discordâncias sobre a linguagem a utilizar, o que causaria atrasos consideráveis no processo de concepção do VRML.

Após alguns retoques intermédios e muita discussão na comunidade científica, em Agosto de 1996 surge o tão esperado VRML 2.0 (mais tarde denominado

VRML97 após apreciado e aprovado pela ISO - *International Standards Organization* - em Setembro de 1997).

Se o VRML 1.0 fornecia meios de criação e visualização de mundos 3D estáticos, o VRML97 fornece muito mais. O seu principal objectivo é o de propiciar ao utilizador uma experiência mais rica, excitante e interactiva do que a que era possível dentro das fronteiras estáticas do VRML 1.0. O objectivo secundário era o estabelecer alicerces sólidos para futuras expansões do VRML, mantendo as coisas simples, rápidas e eficientes.

Em relação ao VRML 1.0 e além do aperfeiçoamento visual dos mundos estáticos, o VRML97 introduz quatro novos conceitos globais: interactividade, animação, *scripting* e prototipagem.

A nível do aperfeiçoamento visual dos mundos estáticos, há a salientar vários aspectos: a possibilidade de criar planos para o céu e para o solo, adicionar panorâmicas, criar terrenos irregulares e até adicionar nevoeiro às cenas. Com o VRML97 também é possível introduzir nas cenas som especializado 3D (digital ou sintetizado), bem como vídeo. Por outro lado, a reestruturação do formato de uma cena, veio tornar mais simples a sua construção e interpretação pelo *browser*.

Com o VRML97 um utilizador pode agora interagir com objectos. Existem sensores que detectam o movimento e acções do utilizador, gerando eventos que podem ser transmitidos aos objectos para mudar o seu estado. Existem também sensores de tempo que permitem controlar tudo o que se relaciona com o tempo, desde relógios de alarme a animações. Há ainda a salientar a detecção de colisões que assegura que objectos sólidos (como paredes ou muros) reajam como tal, e ainda existência da noção de gravidade que permite acompanhar irregularidades do terreno ou subir e descer escadas.

O VRML97 introduz um conjunto de funções - Interpoladores - que permitem controlar a animação de objectos. Estes interpoladores possibilitam a criação de animações predefinidas de vários aspectos de um mundo, que podem ser activadas em qualquer altura. Com os interpoladores é possível definir objectos com movimento, portas que se abrem, carros em movimento, objectos que mudam a sua cor à medida que o tempo passa e até objectos que se transformem noutros objectos (*morphing*). Podem até ser criadas visitas guiadas ao mundo

virtual usando caminhos predefinidos que permitam apresentar esse novo ambiente ao utilizador.

Se os interpoladores vieram dar movimento aos mundos VRML, os *scripts* vieram dar um semblante de inteligência aos objectos desses mundos. A lógica junta-se então à animação tornando um mundo VRML num ambiente verdadeiramente interactivo. Os *scripts* recebem dados (eventos) enviados pelos sensores e geram eventos, baseados nesses dados, que podem mudar o comportamento dos objectos.

Finalmente, o conceito de prototipagem vem dar corpo às ideias de expansão do VRML. Através da prototipagem é possível encapsular um grupo de objectos definindo um novo tipo de objecto, que pode vir a ser usado por qualquer pessoa. Podem-se então criar instâncias desse novo tipo, cada uma com valores diferentes para os seus diferentes parâmetros.

2.1.4 Características

Algumas das principais características do VRML já foram descritas no final da secção anterior, quando se apresentaram os conceitos introduzidos com a última versão do VRML (melhoria dos mundos estáticos, interactividade, animação, *scripting* e prototipagem). De modo que, nesta secção apenas irão ser focados alguns aspectos genéricos do seu funcionamento, sem entrar em grandes detalhes relacionados com a estrutura interna da linguagem, pois com a utilização de uma ferramenta de desenvolvimento VRML, como o *Cosmo Worlds*, o criador pode focalizar toda a sua atenção e esforço na concepção dos mundos, sem ter de digitar código directamente no ficheiro VRML.

Os ambientes virtuais gerados com VRML, na realidade não são mais que simples ficheiros de texto, normalmente com extensão *.wrl* (de *world*), cujo conteúdo não pressupõe uma prévia compilação, como é comum acontecer com linguagens de programação. Um programa denominado por *browser* é responsável pela interpretação, execução e apresentação do ficheiro, funcionando também como interface entre o utilizador e o ambiente por intermédio de dispositivos de interacção, como são o teclado e o rato por exemplo.

Nota: Apesar de existirem *browsers* exclusivamente para VRML, é mais comum (e estão mais actualizados) a instalação de plug-ins que estendem as capacidades dos *browsers* que utilizamos diariamente para aceder à Internet (Internet Explorer e Netscape Navigator são os mais comuns), para leitura e interpretação de ficheiros VRML

2.1.4.1 Nós e Campos

A unidade básica num ficheiro VRML é o nó. Os nós podem representar desde objectos, materiais, sensores, ou outras instâncias VRML. Cada nó contém campos que armazenam os parâmetros que definem as suas características, estando definidos valores por defeito para esses campos, caso eles não sejam especificados.

Os nós podem ser classificados em diversas categorias:

Categoria	Objectivo	Exemplos⁵
Geometria	Representar a geometria na cena VRML	<i>Box, Cone, IndexedFaceSet, Cylinder, Sphere</i>
Aparência	Usados para definir o aspecto visual da geometria	<i>Material, Color, Appearance, MovieTexture</i>
Grupo	Juntar nós num único grupo	<i>Transform, Collision, Inline, Billboard, LOD</i>
Sensores	Detecção de movimento e acção pelo utilizador	<i>VisibilitySensor, ProximitySensor, TouchSensor</i>
Animação e comportamentos	Animações, actividades temporais, bem como comportamentos programáveis	<i>TimeSensor, PositionInterpolator, ColorInterpolator, Script</i>
Ambiente	Efeitos visuais da cena	<i>DirectionalLight, PointLight, Fog, Sound, Background</i>
Movimentação pela cena	Estabelecer regras e meios de navegação do utilizador pelo mundo	<i>Viewpoint, Anchor, NavigationInfo</i>

⁵ Apenas são indicados alguns nós de cada categoria, não a totalidade

Os nós estão relacionados entre si de forma hierárquica, podendo-se caracterizar por duas classes: nós agrupantes ou nós folha. Os nós agrupantes juntam outros nós entre si, permitindo criar colecções de nós, designados por filhos, que serão consideradas como um único objecto. Alguns nós agrupantes também controlam o modo como os seus filhos são visualizados.

Os nós folha não podem ter filhos. Dentro dos nós considerados folha estão formas, luzes, *viewpoints*, sons, *scripts*, sensores, interpoladores, e nós que fornecem informação ao *browser*.

Esta estrutura hierárquica que resulta da forma como os nós estão relacionados entre si é, como iremos ver mais adiante, um dos conceitos mais importantes associados ao VRML, sendo fundamental para um bom desempenho de qualquer ambiente virtual concebido em VRML.

2.1.4.2 Sistema de coordenadas

O VRML usa um sistema de coordenadas 3D, cartesiano, usando a regra da mão direita. Por defeito, os objectos são visualizados em 2D, projectando-os na direcção positiva do eixo dos Z, com a direcção positiva do eixo dos X para a direita e a do eixo dos Y para cima. Esta projecção pode ser alterada usando uma transformação.

A unidade standard para distâncias é o metro e para ângulos é o radiano.

Conceptualmente, o VRML tem um sistema de coordenadas absoluto (*world coordinate system*). As várias transformações locais (translações, rotações e mudanças de escala) mapeiam os objectos no sistema de coordenadas absoluto, que é onde a cena é criada, passando a definir um sistema de coordenadas locais (ou relativas ao objecto). As transformações são acumuladas à medida que se desce na hierarquia da cena, herdando todas as propriedades dos seus ascendentes. Contudo, é de notar que estas séries de transformações têm efeito partindo dos nós folha e subindo na hierarquia. Isto é, as transformações locais mais próximas do objecto são efectuadas primeiro, seguidas das transformações encontradas à medida que se sobe na hierarquia.

2.1.4.3 Modelo de visualização

Assumindo que existe um utilizador a visualizar e a interagir com o mundo VRML, existe a possibilidade de se criarem vários locais interessantes de onde o utilizador pode ter uma vista sobre o mundo. A estes locais dá-se o nome de *viewpoints*. Cada *viewpoint* é descrito pelo nó *Viewpoint*. Os *viewpoints* existem num sistema de coordenadas particular e tanto o *viewpoint* como o sistema de coordenadas podem ser animados.

Os *browsers* VRML dispõem de mecanismos no interface para permitir aos utilizadores "transportarem-se" de um *viewpoint* para outro, além de poderem disponibilizar animação entre esses *viewpoints*. Do mesmo modo, o *browser* fornece um interface ao utilizador que permite alterar a sua posição de visualização, bem como a sua orientação. No fundo, o *browser* deve permitir ao utilizador que se movimente pelo mundo VRML.

2.1.4.4 Representação do utilizador

A visualização de um qualquer mundo VRML, é feita por intermédio de uma câmara virtual que tenta simular o campo visual de um ser humano normal e regista movimentos concordantes com os efectuados com o dispositivo de interacção utilizado, teclado, rato ou ambos (excepto aquando da animação de um *viewpoint*). O avatar, assim se denomina o suposto ser humano nesse mundo VRML, é o portador dessa câmara virtual que, para uma simulação realista, é situada na zona dos olhos. Em resumo, o avatar tem como função comportar-se como o "representante" do utilizador no mundo virtual, quer na visualização, quer na movimentação.

2.1.5 O processo de desenvolvimento

O desenvolvimento de projectos na área da realidade virtual e multimédia, decorre de um trabalho de perícia e minúcia apurado, sendo desse modo indissociável de uma definição clara e objectiva de etapas perfeitamente delineadas no tempo, cuja finalidade seja o cumprimento de todos os objectivos propostos. É perfeitamente legítimo que no processo de desenvolvimento de projectos em

VRML, cada criador adopte uma metodologia própria, mas que não deve diferir muito da proposta de modelo que a seguir apresentamos. Esta inclui uma série de etapas que abrangem todo o percurso de criação, desde a fase da planificação até à conclusão.

Teoricamente, a elaboração de um projecto de realidade virtual em VRML, não é mais que um processo cíclico de cumprimento de diversas etapas. E dizemos cíclico porque é muito provável que após a conclusão dessas etapas, o resultado final, visualmente falando, ainda não seja totalmente do agrado dos intervenientes directos ou porque ainda há incumprimento, total ou parcial, de alguns objectivos inicialmente propostos, designadamente o desempenho que é um dos mais importantes. A revisão de algumas dessas etapas, eventualmente mais que uma vez, levará certamente à obtenção de melhores resultados.

O modelo sugerido consiste numa série de etapas que devem ser executadas seguindo uma sequência cronológica (Cosmo Worlds 2.0 User's Guide, 1998):

- ✓ **Planificação do projecto.** Fase embrionária onde se tomam decisões que condicionam algumas das fases posteriores do projecto. O que se vai criar, qual o objectivo, qual a interactividade pretendida, a quem se dirige (público alvo), tem animações, áudio ou vídeo, qual a plataforma onde vai ser visualizado e quais os seus recursos. Estas são algumas das questões que devem ser colocadas antes do avanço real para o trabalho;
- ✓ **Modelação.** Após a definição clara do que se pretende, iniciar a modelação dos objectos que irão constituir o mundo. Essa modelação, por questões de optimização e eficácia, deve ser feita com ferramentas de desenvolvimento VRML, mas também poderá ser executada recorrendo a outras ferramentas de modelação que tenham disponível um motor de exportação para VRML (o 3DStudio, por exemplo);
- ✓ **Disposição dos objectos.** Finda a modelação de todos os objectos, há que efectuar a sua distribuição pelo espaço tridimensional, de acordo com os objectivos visuais traçados. Essa disposição pelo espaço é efectuada com recurso a transformações geométricas: translações, rotações e eventualmente mudanças de escala;
- ✓ **Definir a aparência.** Dar a todas as superfícies existentes na cena o aspecto visual pretendido, com recurso à utilização de cores e/ou texturas;

- ✓ **Animar objectos.** Tal como foi estabelecido na planificação, criar as animações definidas para assim dar mais vida ao mundo. Como já foi referido, animações podem ser criadas com interpoladores ou *scripts*;
- ✓ **Adicionar efeitos ambientais.** Focos de luz (quanto mais não seja para simular o sol), sons, *backgrounds* e nevoeiro, são alguns dos efeitos sem os quais o realismo numa cena estaria seriamente comprometido;
- ✓ **Criar auxiliares de navegação.** A inclusão de *viewpoints* (alguns podem ser animados), ligações para outras zonas da cena ou para outros mundos, são medidas que facilitam todo o processo de navegação pelo mundo, especialmente a utilizadores menos experientes;
- ✓ **Revisão.** Trabalho terminado. Será que tudo está como desejado? O desempenho? O realismo? Todos os links funcionam e o seu destino é o pretendido? Em caso afirmativo, bom trabalho, podemos passar à etapa seguinte;
- ✓ **Conclusão.** Tudo pronto. Reunir todos os ficheiros necessários e disponibilizá-los no meio definido na planificação: Internet, CD-Rom, numa máquina para funcionar a nível local ou entregar ao cliente que requisitou o serviço;
- ✓ **Testar e otimizar.** Esta última etapa é a única que não surge na sua ordem cronológica relativamente às restantes, precisamente porque não tem uma ordem estabelecida durante o curso de etapas definidas para o processo de desenvolvimento. Qualquer que seja a etapa actual do trabalho, testes contínuos são fundamentais para um acompanhamento efectivo do trabalho. Mas a importância destes testes vai muito para além da mera verificação visual, é essencial efectuar sucessivamente testes de desempenho para aquilatar no imediato quais as optimizações necessárias. Se este aspecto for descurado durante o processo de criação, é bastante provável que o resultado final tenha níveis de desempenho insuficientes. Daqui resultará a necessidade de refazer algum do trabalho que se julgava concluído, com o conseqüente desfasamento do tempo previsto para a conclusão do projecto, também devido à complexidade acrescida que advém de se ter de operar sobre cenas já montadas. É fulcral que logo desde o início do processo de concepção do projecto, o criador esteja sensibilizado para a problemática da optimização em VRML, pois só assim se atingem os objectivos propostos, mas mais que isso, se

podem obter resultados, leia-se cenas realistas, verdadeiramente surpreendentes e com um desempenho que não se julgaria possível. Uma parte considerável desta dissertação foi precisamente dedicada a esse tópico fundamental para a construção de mundos VRML, tal como se descreve no capítulo 4.

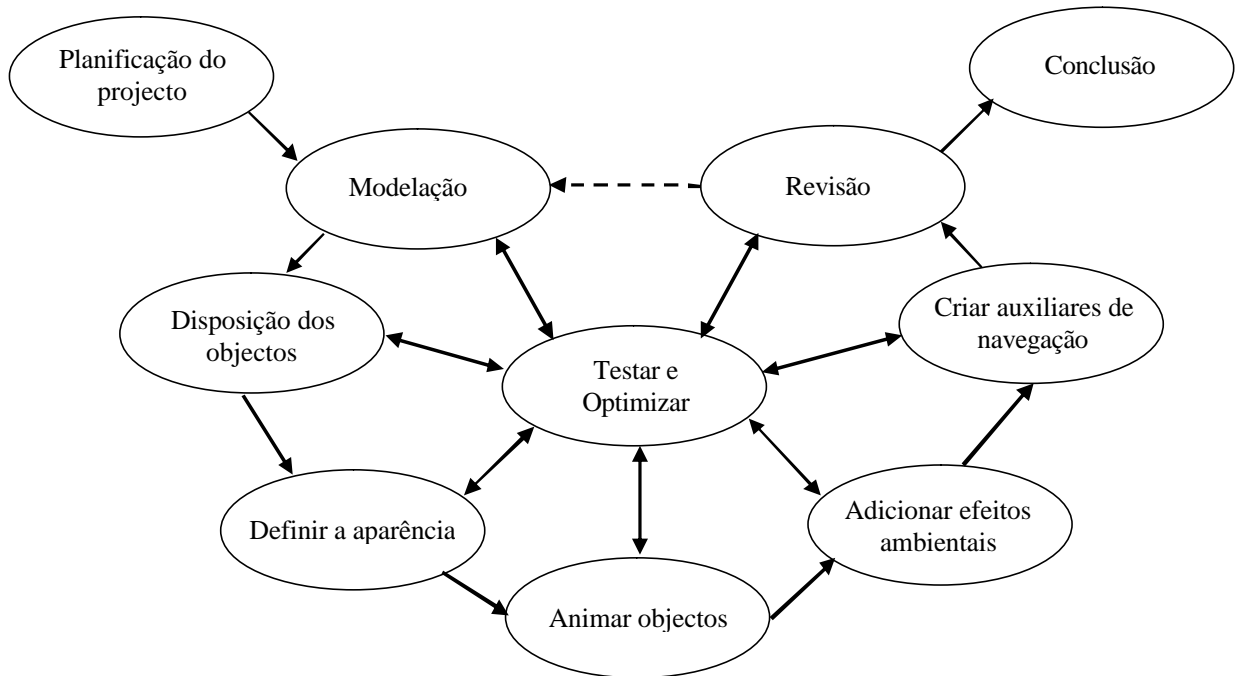


Figura 2.1 Diagrama ilustrativo das etapas no processo de desenvolvimento

2.2 Exemplos

2.2.1 Introdução

Desde a idade média, quiçá até antes, que as questões relacionadas com o passado histórico e cultural, principalmente ao nível de achados arqueológicos, exercem sobre o homem um enorme e misterioso fascínio. Desde então se tentam criar métodos e formas de personificar tais locais, de modo a satisfazer essa curiosidade natural e ancestral reconhecida ao ser humano.

Principalmente nestes últimos anos, o crescimento do número de projectos envolvendo esta temática (reconstrução histórica), tem sido real e efectivo, tal deve-se a dois factores: o acentuado desenvolvimento tecnológico que permite concretizar tais desígnios mais facilmente, com maior impacto e afectando um

maior número de pessoas; e o contínuo e crescente interesse da humanidade nestas questões relacionadas com o seu património cultural e arquitectónico.

É esse o motivo que legitima este trabalho e nos leva a incluir os exemplos a seguir descritos. Todos eles abordam a temática da reconstrução histórica, a uma escala mundial, de um património com grande relevo cultural, usando as denominadas “novas” tecnologias, como por exemplo: GPS, *Head-Mounted Display*, Bases de Dados, GIS, Java, entre outras. No entanto, todos estes exemplos têm algo em comum com o nosso trabalho, que sirva como de elo de ligação. Todos eles têm uma componente, ainda que pequena em alguns casos, da tecnologia que serviu de suporte a este trabalho, o VRML. Também numa tentativa de tentar demonstrar um pouco, as capacidades e potencialidades do VRML, em nossa opinião tão pouco exploradas.

2.2.2 Archeoguide

Archeoguide, abreviatura de *Augmented Reality-Based Cultural Heritage On-Site Guide*, é um dos maiores projectos Europeus de realidade virtual aumentada com o apoio da União Europeia, para o qual foi criado um consórcio constituído por instituições europeias de reconhecida reputação na área (www.archeoguide.com).

Na lista de pretensões deste ousado projecto, está uma tentativa de concepção e implementação de um sistema inovador que estabeleça um ponto de viragem no modo como os visitantes tomam conhecimento do património cultural legado pelos nossos antepassados, por intermédio de formas alternativas de apresentação dessa informação de um modo intuitivo e convincente.

O primeiro protótipo do projecto Archeoguide foi instalado e testado no espaço arqueológico de Olímpia na Grécia, por ser um dos mais importantes espaços arqueológicos a nível mundial, por ser o local de “nascimento” dos Jogos Olímpicos e essencialmente por se encontrar em ruínas.

A complexidade e questões técnicas associadas a este projecto, são de tal ordem, que seria impossível descrever por poucas palavras todo o trabalho e tecnologias envolvidas no desenvolvimento do mesmo. Para um conhecimento

mais profundo recomendamos a leitura de Vlahakis et al. (2002). Desse modo, iremos descrever apenas alguns dos processos e funcionalidades mais importantes associadas ao projecto.

O sistema Archeoguide explora os mais recentes avanços tecnológicos em computação móvel, realidade aumentada, visualização tridimensional, redes de dados, entre outras e recorre a uma arquitectura cliente/servidor, composta por três importantes subsistemas: um servidor central, as infra-estruturas de rede e as unidades móveis.

Uma rede local sem fios é utilizada para a comunicação entre o cliente, as unidades móveis transportadas pelos visitantes, e o servidor central. Com base na posição e orientação do visitante⁶, o sistema representa a informação audiovisual contextualizada com o cenário em causa, ou seja, todo o processamento é realizado em tempo real, mediante o comportamento, leia-se movimentação, do visitante no espaço físico. Técnicas de realidade aumentada permitem a inclusão visual, por sobreposição dos modelos tridimensionais construídos⁷ com a paisagem existente em seu redor, apresentando um cenário visual harmonioso com o meio envolvente e muito próximo da realidade da época.

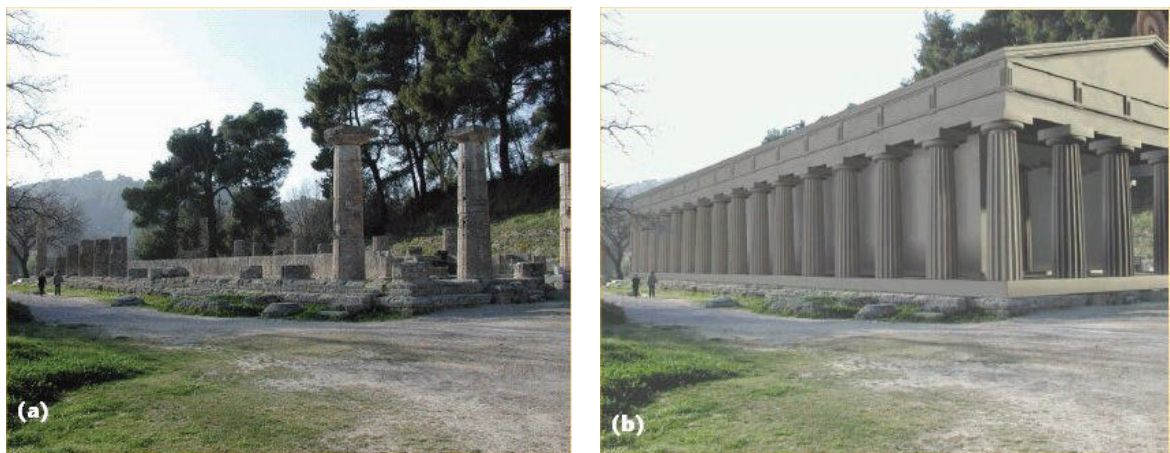


Figura 2.2 Imagem real a). Imagem gerada por realidade aumentada b)

O servidor central é o “coração” de todo o sistema e reside num equipamento de alta performance, com capacidade suficiente para armazenar uma base de dados multimédia. A sua função, além de repositório da base de dados, é a de “gestor”

⁶ Essa posição e orientação é determinada por GPS (*Global Positioning System*)

⁷ VRML foi o formato usado para a modelação tridimensional dos monumentos já desaparecidos ou desprovidos da sua aparência original

de toda a informação multimédia necessária (imagens 2D, modelos 3D, áudio, vídeo e texto), para a construção e representação da visita guiada, que será posteriormente enviada para as unidades móveis.

Em virtude das generosas dimensões de todo o espaço (300 x 500 metros) e dada a necessidade de uma comunicação em rede, a opção recaiu por uma solução prática e de certa forma óbvia, no caso, a utilização de um protocolo de rede de dados sem fios. A escolha recaiu no protocolo IEEE 802.11b, pois além de solucionar o problema, é de fácil extensibilidade e actualização, caso tal seja necessário.

Por fim, as unidades móveis são dispositivos cuja principal característica é a sua portabilidade. São estas unidades que formulam os pedidos ao servidor central e é por seu intermédio que o sistema de GPS calcula a posição e orientação do visitante que o transporta. Para o primeiro protótipo do Archeoguide foram utilizadas três diferentes versões destas unidades móveis, que diferem na portabilidade e funcionalidades apresentadas:

- ✓ A utilização de um portátil, representa a solução mais rica de conteúdos e ao mesmo tempo, também a mais complexa. Associado ao portátil está um capacete, constituído por um *Head-Mounted Display* e uma câmara Web USB. De um modo um pouco simplista, o seu funcionamento é o seguinte: Quando o visitante se posicionar num dos pontos estrategicamente predefinidos no recinto, e direccionar o seu olhar para o monumento supostamente em ruínas, que deverá estar próximo desse local, é desencadeado pelo sistema um pedido ao servidor central para agregar toda informação necessária para a representação desse monumento, e enviá-la para a unidade móvel, neste caso o portátil. O próprio sistema faz a integração do modelo virtual com a imagem recebida da câmara Web, e apresenta o resultado no *Head-Mounted Display*. Ao mesmo tempo, é feita uma sincronização da apresentação visual com uma narração áudio adequada ao monumento em causa. Outra das funcionalidades do sistema, dá a possibilidade de personalizar a visita de acordo com o perfil do visitante, alternando o contexto audiovisual da visita guiada. Esses parâmetros podem diferir com a idade, interesses, educação ou conhecimento arqueológico.

- ✓ Uma outra solução faz uso de um *Pen-Tablet*, que pode ser transportado manualmente como um livro. O utilizador pode interagir com o dispositivo, seleccionando com uma caneta especial as diversas funcionalidades disponíveis, pois o mesmo funciona como um sistema operativo. Os conteúdos no *Pen-Tablet* estão dispostos por secções, mais especificamente, é apresentado um sistema de orientação geográfica, para o visitante ter sempre uma correcta percepção da sua posição dentro do recinto, e a respectiva reconstrução tridimensional, integrada com a paisagem envolvente, do monumento em apreciação pelo visitante, sempre acompanhada por uma narração áudio na língua materna do mesmo. Outra das funcionalidades deste dispositivo, permite apresentar uma vista panorâmica do local onde está posicionado o visitante. A dita vista é automaticamente actualizada (mais uma vez, essa orientação é determinada por GPS), à medida que o utilizador roda sobre si próprio.
- ✓ O *Palmtop* é o mais limitado de todos os dispositivos, e como tal, a informação disponibilizada é mais escassa. Esta unidade não faz uso da orientação do seu portador, mas sim e apenas da sua posição. À medida que o visitante se vai aproximando de um local previamente definido, normalmente perto de um monumento ou uma área importante da antiga cidade, vai-lhe sendo apresentada informação relevante relacionada com esse mesmo local, informação essa que pode ser disponibilizada das mais diversas formas: imagens do modelo virtual reconstruído, vídeos, vistas panorâmicas, narração áudio ou informação textual.

2.2.3 Viagem Virtual a Bracara Augusta

Bracara Augusta foi uma das três grandes “metrópoles” urbanas construídas pelo Imperador Augusto a Noroeste da Ibéria após o final das guerras Cantábricas (19 A.C.). Durante o século I, a cidade passou a ser um importante centro administrativo e económico, altura em que terá atingido a sua máxima extensão, no que diz respeito ao seu urbanismo, arquitectura e vida económica e social. Nos últimos quase que 30 anos, escavações arqueológicas realizadas em Braga pela Unidade Arqueológica da Universidade do Minho, foram o mote para uma

intervenção no local a longo prazo, com vista ao “salvamento” de Bracara Augusta, fazendo-a “renascer” das cinzas.

O projecto *Viagem Virtual a Bracara Augusta* desenvolvido em conjunto pela Universidade do Minho e o Centro de Computação Gráfica, para a Fundação Cultural Bracara Augusta, tinha como objectivos a divulgação junto dos utilizadores leigos parte do registo arqueológico de Bracara Augusta. Tal seria consumado por aplicação de tecnologia multimédia e paradigmas de visualização adequados que contribuíssem para um maior nível de expressividade de informação e que facilitassem o acesso à análise científica de dados e a interpretações arqueológicas (Marcos et al., 2000).

O projecto por si, era bastante abrangente, tendo sido estruturado por quatro áreas temáticas principais: apresentação de um historial das escavações mantidas ao longo destas últimas décadas, as investigações levadas a cabo para a compreensão de como era planificada uma cidade romana, o espólio da cidade, finalmente, e o mais importante de todos, a Reconstrução Virtual de Bracara Augusta, integrando muitos dos edifícios e complexos existentes na cidade acompanhados por elementos multimédia.

O processo de reconstrução de Bracara Augusta foi composto por etapas de modelação cronologicamente encadeadas, nomeadamente, o terreno, os espaços “verticais” (paredes, pilares e colunas), e por fim os telhados. Os modelos elaborados, alguns dos quais em formato VRML, foram usados na concepção final do projecto, onde por exemplo, se poderá visitar uma casa habitacional ou as únicas termas romanas de cariz público conhecidas actualmente.

Outra das mais valias deste projecto, foi a atenção dada para tentar simular a percepção de uma visita virtual, de acordo com as cores e formas do Império Romano. Por outro lado, grande parte dos textos são apresentados com voz, misturada com outros sons e música, aumentando significativamente a espectacularidade do ambiente multimédia para os utilizadores finais.



Figura 2.3 Imagem representativa das termas romanas de Bracara Augusta

As plataformas de suporte para o sistema de informação pretendido, incidiram em três vertentes, dependendo do público alvo a atingir:

- ✓ Por intermédio de quiosques multimédia, disponibilizados ao público em locais estratégicos da cidade de Braga;
- ✓ Elaboração de um CD Multimédia, para uma divulgação mais ampla e cómoda dos conteúdos elaborados no projecto;
- ✓ Desenvolvimento de um site Web de suporte e divulgação ao projecto *Viagem Virtual a Bracara Augusta* (www.cm-braga.pt/bracara_augusta/).

2.2.4 Projecto 3D Murale

A universidade britânica *Brunel University*, lidera um consórcio de outras universidades e empresas que colaboram num projecto apoiado pela União Europeia denominado *3D Murale* (www.brunel.ac.uk/project/murale/). O projecto consiste no uso e desenvolvimento de ferramentas multimédia com apetência tridimensional para medição, reconstrução e visualização de espaços arqueológicos da Europa, em realidade virtual (Cosmas, 2001).

Como primeiro teste estão a ser usados os achados arqueológicos de Sagalassos, uma secular cidade outrora existente na actual Turquia. Estes, correspondem a um dos maiores projectos arqueológicos existentes no mediterrâneo referentes a uma era Greco-Romana, e que abarca um período superior a um milhar de anos, desde o século IV A.C. ao século VII D.C..

O projecto assenta em quatro vertentes principais:

- ✓ Aquisição e armazenamento, com meios adequados e portáteis, das medidas e dimensões quer de pequenos artefactos para catalogação e visualização, quer de cerâmicas, esculturas e edifícios para “restauração” e visualização, seja virtual ou não. Medições topográficas também deverão ser efectuadas para a criação de um modelo tridimensional do terreno com os mesmos propósitos.
- ✓ A criação de uma base de dados multimédia é também essencial para um projecto com estas características e objectivos. Esta base de dados irá armazenar além de informação textual, imagens 2D e informação para a criação de modelos 3D VRML. Entre outras funcionalidades como pesquisas, a base de dados permite definir a posição de cada uma das “peças” no modelo global e a época que retracts. Esta estrutura de armazenamento torna possível, com o auxílio de outra aplicação, gerar automaticamente um mundo virtual que retracte um determinado período, apresentando edifícios, vegetação e objectos desse período.
- ✓ Com base nos achados arqueológicos e nas medições efectuadas serão realizadas reconstruções virtuais e possivelmente mesmo físicas de objectos, edifícios e ornamentos existentes nas diferentes eras da sua existência. Tal reconstrução tem como móbil a sua posterior visualização, e também poderá servir como medida de conservação dos objectos originais.
- ✓ Todo este trabalho e esforço despendido são alicerces basilares para o culminar dos objectivos de um projecto desta natureza, ou seja, a possibilidade de visualizar, desde os seus tempos de metrópole aos dias de hoje, esta secular herança cultural que é proporcionada pelo espaço arqueológico de Sagalassos. Essa visualização deverá ser consubstanciada de dois modos distintos, dependendo do número de pessoas que lhe poderão aceder: Um modelo de alta qualidade para navegação individual em local próprio; e um outro que permita alargar o âmbito do público alvo, ainda que sacrificando alguma qualidade, ao disponibilizar essa informação/visualização através da Internet.

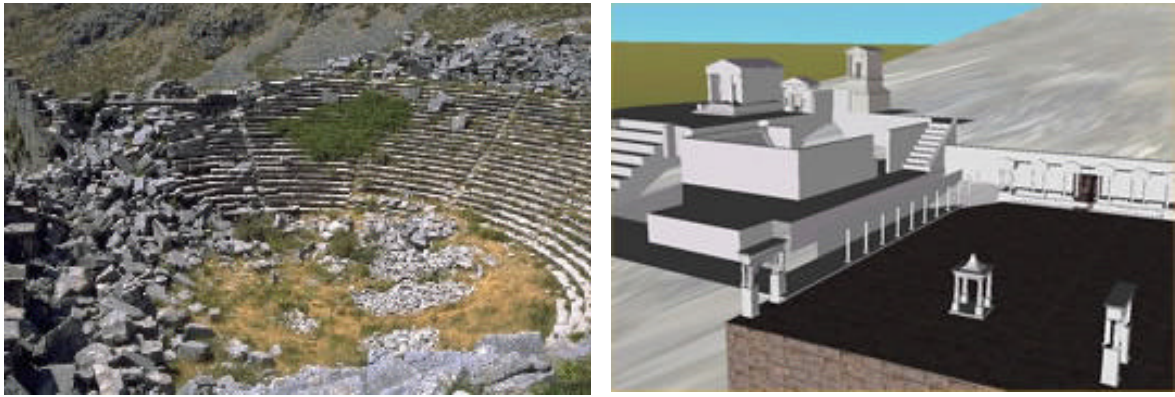


Figura 2.4 O teatro romano de Sagalassos no seu estado actual (esq.) e um modelo VRML de uma parte de Sagalassos (dir.).

Estas imagens foram obtidas num site Web dedicado às ruínas de Sagalassos e sua representação virtual (*Virtual Sagalassos on the WWW* em www.esat.kuleuven.ac.be/sagalassos/).

2.2.5 Wat Ratchaburana

Ayutthaya é uma cidade Tailandesa que dista cerca de 80 Km da capital Bangucoque, sendo ainda hoje considerada por muitos como uma das mais fantásticas cidades existentes no planeta. A cidade fundada em meados do século XIV D.C., foi a capital do reino de Sião durante cerca de 400 anos, até ser atacada e destruída pelos Burmeses. No seu período áureo (século XVI), a cidade foi uma das maiores e mais ricas cidades asiáticas, sendo na altura ponto de confluência, para comércio, visita e missões religiosas, de povos das mais distintas regiões do planeta. Inclusive, oriundos da Europa, os missionários portugueses foram dos primeiros a chegar à região no longínquo ano de 1511. Em 1991 parte significativa da cidade de Ayutthaya, foi nomeada património mundial da humanidade pela UNESCO, em grande parte devido aos os seus magníficos templos budistas, também eles maioritariamente destruídos pelos Burmeses.

No final da década de 90, um projecto conjunto da Universidade de Melbourne na Austrália e da Universidade de Chulalongkorn na Tailândia, foi conduzido com vista a uma recriação fotorealista e em formato tridimensional de um modelo da área de Ayutthaya considerada património mundial (Ogleby, 2001). Paralelamente a esse projecto, um outro foi desenvolvido pela Universidade de Silpakorn,

Tailândia, cujo objectivo era também uma reconstrução tridimensional de um Wat⁸ budista, existente no período áureo de Ayutthaya, chamado Wat Ratchaburana (Tantatsanawong et al., 2001), e que agora se encontra em ruínas.

A implementação desse projecto teve como linhas orientadoras a execução de três módulos distintos:

- ✓ O primeiro módulo pretendia explorar o conceito de realidade virtual, para a apresentação e visualização de espaços históricos. Esse conceito deveria ser potencializado através Internet, permitindo o deslocamento pelo espaço virtual por intermédio de um simples *browser*. Para tal recorreu-se à criação de modelos VRML bem como de imagens panorâmicas;
- ✓ O segundo módulo utilizou um Sistema de Informação Geográfica (GIS – *Geographical Information System*), que foi aplicado para o tratamento de informação histórica, como a idade e o autor de edifícios em análise;
- ✓ O terceiro e último módulo, utilizou um conceito de realidade virtual orientada a objectos. Esta metodologia pretendeu congregar e usufruir das vantagens que lhe estão associadas, como independência, reutilização, herança, manutenção, entre outras. As características dos edifícios foram estabelecidas para uma definição adequada das respectivas classes, sendo as suas propriedades e a hereditariedade entre objectos utilizada para, através da criação de subclasses, cada uma representar um determinado período cronológico referente às diversas fases de construção de um determinado edifício.



Figura 2.5 Imagem inicial (esq.). Imagem após a reconstrução (dir.)

⁸ Normalmente refere-se a um conjunto de edifícios religiosos delimitados por muros, cuja entrada é efectuada através de grandes portões

2.2.6 O Labirinto de Hawara

Por volta de 1800 A.C. o rei Amenemhat III, o último grande rei da 12ª Dinastia Egípcia, mandou edificar na região de Fayum (cerca de 80 Km a Sudeste da actual capital egípcia, Cairo) um enorme complexo de culto enquadrado a sul da pirâmide de Amenemhat III. Hawara, assim se designa o local, tornou-se num local de culto e veneração ao rei ali sepultado, que mais tarde passou a ser considerado um deus. 1500 anos depois, o rei Amenemhat III ainda era venerado nessa região, e foi durante esse período que o complexo de culto ao rei passou a ser conhecido como o “Labirinto” (Shiode e Grajetzki, 2000). Mais tarde, aquando do domínio romano o local foi usado como pedreira e maioritariamente destruído.

Englobado num desígnio mais vasto de ensino à distância, surgiu um projecto piloto para explorar as possibilidades de “reproduzir” locais históricos, total ou parcialmente destruídos, usando tecnologias de realidade virtual. O *Centre for Advanced Spatial Analysis* e o *The Petrie Museum for Egyptian Archaeology* ambos afectos à *University College London*, colaboraram na concepção desse projecto piloto na qual visaram a reconstrução do Labirinto de Hawara, descrito no parágrafo anterior.

Ao contrário de outras reconstruções de espaços arqueológicos, onde o plano original é conhecido, neste caso esse plano não existia e o modelo foi idealizado a partir de literatura e objectos descobertos por diversas escavações feitas ao longo dos anos. No entanto, as divergências entre historiadores, referentes à forma e até à escala do labirinto de Hawara, eram de tal ordem, que se optou por uma solução de compromisso, ou seja, foram modeladas duas versões do referido labirinto. Ambas estão disponíveis em formato VRML e podem ser consultadas no endereço: www.casa.ucl.ac.uk/digital_egypt/hawara/.

Um primeiro modelo foi criado a partir das expedições e publicações de Flinders Petrie dos fins do século XIX e princípio do século XX. A partir dos seus desenhos e de assunções perfeitamente legítimas, foi criado um modelo que se apresenta na Figura 2.6a.

O segundo modelo teve a sua génese em observações e resultados mais recentes (1979), por parte do historiador alemão Arnold. O resultado desse modelo pode ser constatado na Figura 2.6b.

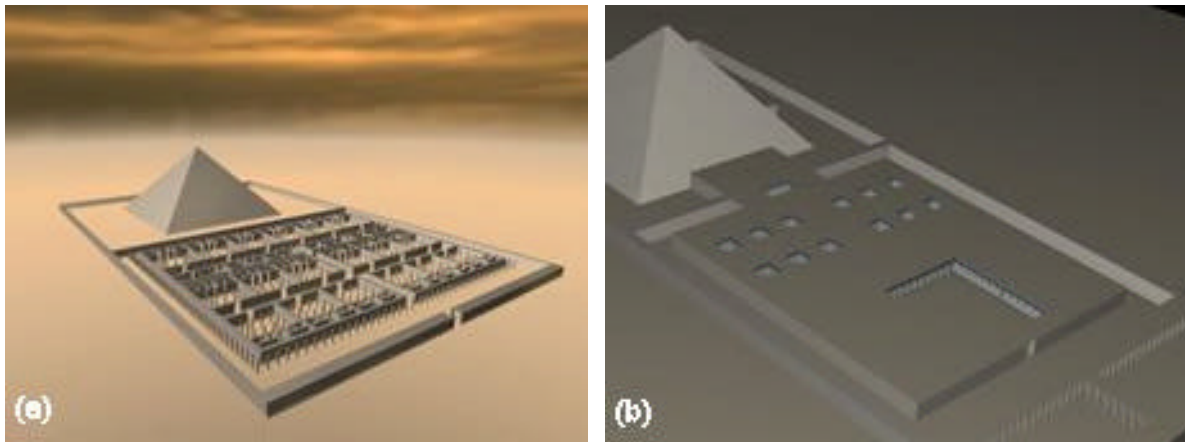


Figura 2.6 Modelo de Petrie a) e modelo de Arnold b)

2.2.7 A Basílica de Fano

Vitruvio, considerado o grande arquitecto romano da época do Imperador Augusto, foi o autor da obra literária *De Architectura*. Este livro de teor arquitectónico é um dos raros exemplares da época que retractam a arquitectura clássica, e foi por seu intermédio que alguma da arquitectura ocidental se influenciou e caracterizou, mesmo até aos dias de hoje. No entanto, a história diz-nos que, apenas um edifício foi, na realidade, planeado e edificado na sua totalidade pelo próprio Vitruvio. A hoje conhecida como Basílica de Fano, construída por volta de 19 A.C., era considerada uma obra prima de extrema importância histórica e arquitectónica na época, na medida em que, e de acordo com a descrição do próprio Vitruvio, as suas características seriam substancialmente diferentes das basílicas romanas de então. Infelizmente, a Basílica foi totalmente destruída quando os Bárbaros Góticos invadiram Fano em 540. Como a “nova” cidade de Fano foi construída sobre as ruínas da antiga, qualquer indício da presença de tal Basílica foi apagado, restando apenas a descrição existente no já mencionado livro de Vitruvio, assim como relatos existentes em alguma da literatura de meados deste milénio, de onde era comum apresentarem uma proposta para um possível modelo do formato da Basílica

(Figura 2.7). Até que em 1840, escavações arqueológicas efectuadas no local, “colocaram a nu” as ruínas da referida Basílica.

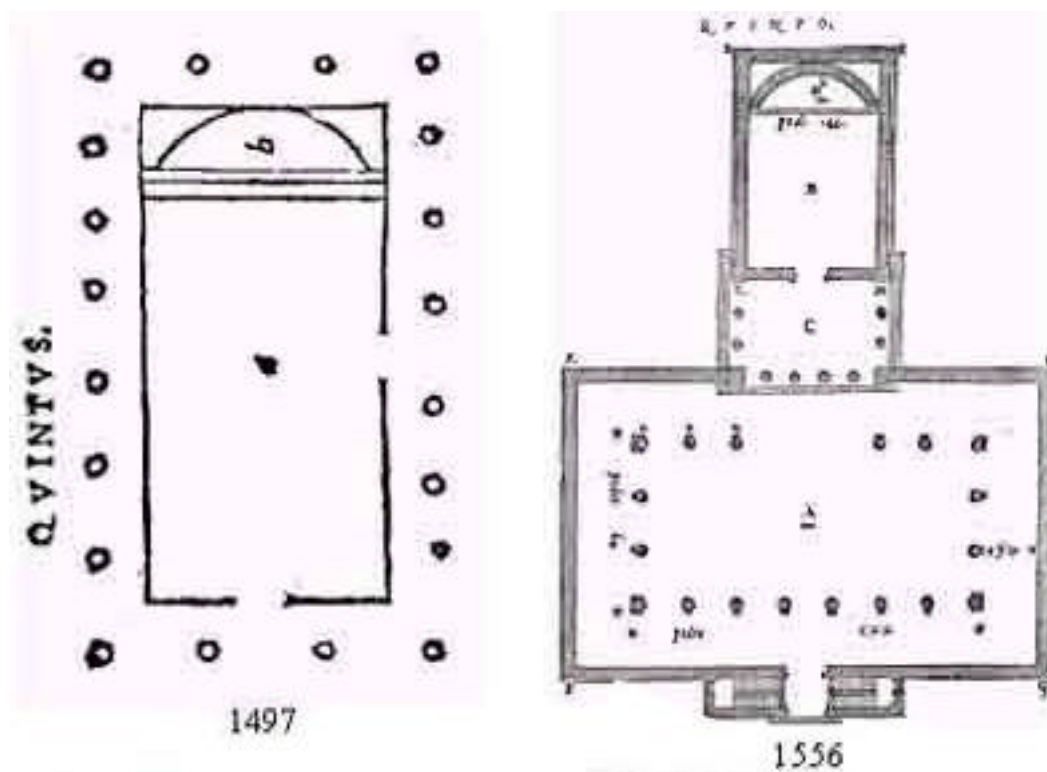


Figura 2.7 Possíveis plantas de historiadores antigos: Frá Giocondo (esq.) e Andrea Palladio (dir.)

Foi com base em toda a informação recolhida, desde os achados arqueológicos, à descrição de Vitruvius, até à restante literatura existente sobre a Basílica, que dois investigadores da Universidade de Ancona, Itália, resolveram levar a cabo a tarefa de reconstruir este património perdido e do qual tão pouco se sabia (Pugnali e Cini, 2001).

O projecto consistiu na elaboração de um CD multimédia, que para além de ambicionar efectuar uma reconstrução total da Basílica, pretendia explorar as capacidades do meio físico de armazenamento para ilustrar todo o trabalho de investigação efectuado até chegar ao modelo final, e servir em simultâneo como repositório de uma base de dados documental permanente e de fácil acesso de toda a informação, fosse ela textual ou não, existente sobre a Basílica, e que antes estaria dispersa por inúmeros livros e bibliotecas.

A componente da representação e visualização da Basílica foi levada a cabo por intermédio de vídeos e animações interactivas, preparadas para o efeito em VRML e *Quick Time*. O utilizador tem a possibilidade de entrar na Basílica,

apreciar o seu aspecto e interactivamente poder aceder a informação textual referente à ala ou compartimento que está a visualizar no momento.

2.2.8 Acampamento Romano

Em matéria de reconstrução de espaços referentes à imperial época romana, a estação britânica BBC disponibiliza na sua página Web uma secção histórica, onde são descritas e retractadas estruturas existentes no passado que revelem interesse histórico e patrimonial (www.bbc.co.uk/history/multimedia_zone/3ds/). Cada uma dessas estruturas, tem disponível uma representação virtual em formato tridimensional, elaborada em VRML, que permite aos seus visitantes adquirir um conhecimento mais sólido, realista e visual, de cada um desses espaços.

Um dos exemplos existentes nessa secção, reporta-se a um acampamento construído pelo império romano ao longo da famosa fronteira de *Hadrian's Wall*, a norte da Inglaterra em meados de 122 D.C.. A sua acção de posto fronteiriço, teria também uma função de persuasão para com as tribos do norte da ilha, ao marcar uma posição bélica no território (Birley, 1973). A sua operacionalidade consistia num controlo permanente do movimento de pessoas dessa região, impedindo-as de penetrar em território romano e que acabava por funcionar também como uma clara demonstração de força do poder de Roma.

A acompanhar a descrição textual, está também disponível um modelo tridimensional para visualização e exploração de todo o complexo urbanístico do acampamento. O modelo elaborado em VRML retracta de forma fidedigna todos os espaços físicos existentes no local, desde o quartel general, o hospital, as casernas dos soldados, entre outros; mas peca por inibir uma das principais potencialidades do VRML, nomeadamente a possibilidade de usufruir de uma interactividade livre e total. Ao utilizador apenas é permitida uma interacção relativa, ao ter a opção de seleccionar algumas posições predefinidas no modelo virtual, que contemplam as principais áreas do acampamento romano. Há, no entanto, uma excepção. Uma dessas opções é consideravelmente menos estática e mais rica de conteúdo, ao realizar uma visita guiada seguindo um trajecto

predefinido, na qual é dada uma panorâmica global de todo o complexo urbanístico do acampamento.



Figura 2.8 Imagem ilustrativa do acampamento romano

2.2.9 Huys Hengelo

Crê-se que nos primórdios do século XIII, um pequeno edifício, provavelmente um alpendre usado para arrumações, tenha sido o “berço” para o que mais tarde viria a ser uma grande mansão, conhecida como o Castelo de Huys Hengelo. É um dado adquirido que os seus proprietários ao longo dos tempos foram sempre pessoas com uma posição social de alguma influência, nomeadamente política, no espaço geográfico circundante. Inclusivamente esse edifício terá sido um forte impulsionador do desenvolvimento acentuado da localidade mais próxima, Hengelo, tornando-se até numa das maiores cidades a leste da Holanda. Mas, grande parte do interesse histórico e cultural deste edifício, advém do facto de ao longo do tempo se ter tornado local privilegiado de decisões e encontros políticos de diversa espécie, do qual sobressai o relato de uma história ocorrida em 1646. Nessa data, os enviados das *United Provinces of the Netherlands*, iam a caminho de Muenster para a assinatura do tratado de paz Hispano/Holandês, quando resolveram pernoitar durante algumas noites no castelo a alinhar os últimos

detalhes do documento. Em 1821, após ter caído em desuso o edifício foi demolido e no seu local foram construídos complexos industriais, que por sua vez foram abatidos entre 1993 e 1995, de onde se escavaram as fundações do antigo Castelo de Huys Hengelo.

Em 1999 um conjunto de alunos e professores da Universidade de Twente, Holanda, em colaboração com peritos em arquitectura e arqueologia iniciaram um projecto de reconstrução, virtual, do Castelo de Huys Hengelo (interior e exterior), bem como toda a sua área circundante (Reidsma et al., 2001).

Para a conclusão do projecto e de acordo com os objectivos previamente delineados, foram implementadas três componentes distintas, embora complementares:

- ✓ O primeiro e mais natural é a construção/modelação do modelo virtual, e que representa a face visível do projecto. O trabalho foi desenvolvido quase na totalidade em VRML, dada a sua natureza interactiva e por ser de fácil integração na Web;
- ✓ Para a modelação e manipulação de algumas peças em VRML a utilizar no modelo global, foram desenvolvidas algumas ferramentas para o auxilio desta tarefa, como por exemplo ferramentas baseadas no conceito de extrusão;
- ✓ Como é comum neste tipo de situações, muito pouco se sabe do formato real do castelo, ao ponto de as escassas fontes históricas existentes chegarem a contradizerem-se. Para lidar com esta pouca fiabilidade da informação existente, foi idealizado um modo de permitir utilizador, interactivamente gerar formas alternativas para a aparência e até para a geografia do castelo de Huys e a sua área envolvente, como seja o deslocar de um portão para possíveis locais de posicionamento ou o visualizar das janelas do castelo, ora num formato circular, ora num formato rectangular. Para tal desiderato, foi elaborada uma consola na linguagem Java, para poder parametrizar esses dados configuráveis interactivamente.

O resultado do projecto, bem como uma sequência cronológica e animada da forma de edificação na época, dos edifícios em madeira existentes nas herdades, pode ser visualizado em: parlevink.cs.utwente.nl/Projects/huyshengelo/.



Figura 2.9 Imagem do modelo virtual do Castelo de Huys Hengelo

2.2.10 Santa Clara-a-Velha Virtual

Com origem no nosso país, este é um dos bons exemplos de um ambiente virtual, que recria tridimensionalmente um espaço com algum simbolismo outrora existente e do qual actualmente apenas existem resquícios. Este trabalho foi desenvolvido pelo Centro de Computação Gráfica, na altura, sediado em Coimbra.

O mosteiro de Santa Clara-a-Velha em Coimbra, foi fundado em 1286 por Dona Maior Dias nas margens do rio Mondego. Pouco tempo após a sua construção foi possível constatar que o local escolhido não terá sido o mais adequado, uma vez que desde cedo sofreu as primeiras cheias do rio, que irremediavelmente traçaram a história deste mosteiro. Apesar dos esforços ao longo dos anos, através de modificações estruturais no edifício, o mosteiro foi abandonado em 1677, tendo o claustro sido “engolido” pouco a pouco, pelas águas e areias do Mondego.

Após três séculos de completo abandono, foi encetada em 1990 uma iniciativa de recuperação e valorização do monumento, até então esquecido. Do manancial de informação resultante dos trabalhos arqueológicos efectuados, emergiram dois projectos no âmbito das “novas” tecnologias:

- ✓ O desenvolvimento de um modelo virtual de todo o mosteiro;
- ✓ A criação de um site Web, onde estivessem descritos os trabalhos efectuados neste espaço secular (sta-clara-a-velha.ccg.pt).

O desenvolvimento do modelo virtual, teve como principal objectivo dar a possibilidade a grupos de visitantes de explorar o monumento gótico nas diversas fases da sua existência, com especial relevância nos espaços já inexistentes, como o claustro, o altar da igreja e o pavimento superior. Na apresentação final do projecto e utilizando os recursos dos ambientes virtuais, grupos de visitantes, conduzidos por guias virtuais, podem visitar e explorar capelas, claustros, pátios, jardins e túmulos, apreciando os pormenores artísticos da arquitectura, escultura e artes decorativas que os artistas medievais nos legaram.

Outro dos aspectos importantes deste ambiente virtual, é o facto de permitir vários níveis e tipos de exploração em função do público alvo. São disponibilizados guias apropriados, que adequam as suas explicações às características dos visitantes e suportam a exploração de um modo passivo, com percursos já predefinidos, ou com a possibilidade de interacção do público com o ambiente.

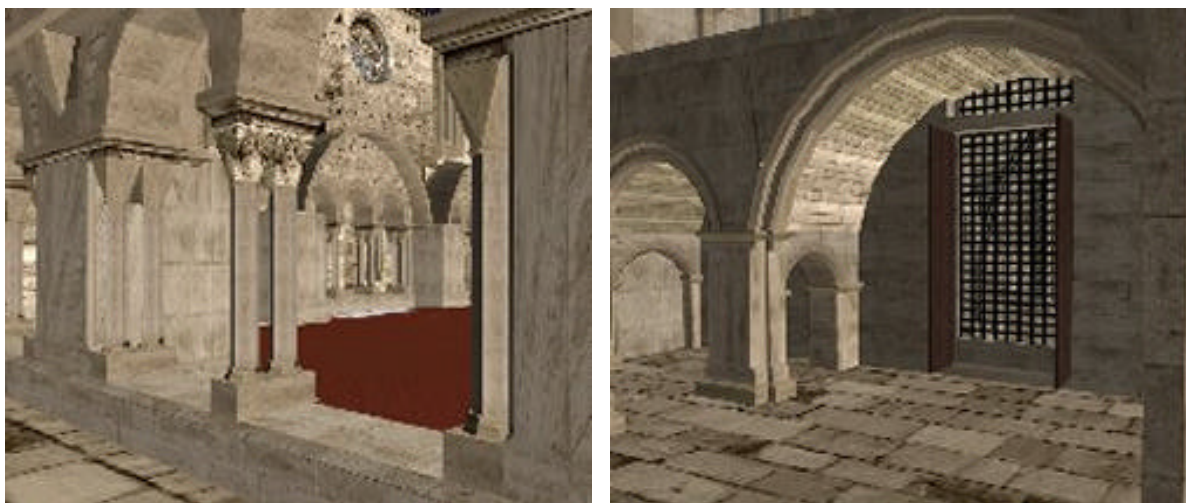


Figura 2.10 Imagens do Mosteiro de Santa Clara-a-Velha Virtual

As especificidades muito particulares, ao nível do realismo, perfeccionismo e objectivos delineados na concepção deste ambiente virtual, implicam que um projecto com estas características não possa estar acessível pela Internet, devido a condicionamentos de largura de banda. Por esse motivo, estão disponíveis para consulta e visualização em VRML no site Web criado para o efeito (sta-clara-a-velha.ccg.pt), vários modelos representativos, retirados do modelo virtual, de alguns elementos do mosteiro de Santa Clara-a-Velha, como a igreja e o claustro. Esses modelos podem servir quer para apreciação geral de um qualquer utilizador interessado e curioso, ou podem ser utilizados como ferramenta de investigação por parte de arqueólogos e arquitectos que não tenham possibilidade de visualizar, por qualquer motivo, o modelo na íntegra.

3. Reconstrução do Fórum Flaviano de Conimbriga

3.1 Descrição

Séculos após a sua destruição, e depois de um completo esquecimento e total abandono, o Fórum Flaviano de Conimbriga “renasce das cinzas” quando em 1964 as escavações levadas a cabo pela Missão Arqueológica Francesa em Portugal e o Museu Monográfico de Conimbriga descobrem vestígios das ruínas do Fórum existente na antiga cidade lusitana. Podemos considerar este nosso projecto como uma consequência de todo o trabalho desenvolvido durante anos por investigadores, arqueólogos e topógrafos, que voltaram a “dar vida” a este espaço emblemático de Conimbriga, e como tal o seu esforço e dedicação também merece ser invocado.

Neste tipo de reconstruções históricas, a questão da veracidade científica do modelo arquitectónico a utilizar é sempre bastante discutida e motivo de profunda análise. Situações há, em que a falta de consenso entre investigadores ocasiona a apresentação de mais que uma proposta para o modelo do espaço ou área a recriar, tal como sucedeu por exemplo, no Labirinto de Hawara descrito na secção 2.2.6. Neste trabalho esta situação não se verifica, pois essas escavações concluídas em 1971, culminaram na obtenção de plantas e alçados fidedignos da totalidade da estrutura arquitectónica do Fórum.

Esses dados desempenharam um papel fundamental e foram o elo necessário para alcançar um acentuado grau de rigor histórico, arquitectonicamente falando, e que deve sempre estar associado a este tipo de recriações ancestrais com algum interesse para o passado cultural da humanidade. Assim, foi desde logo estabelecido que a reconstrução iria ser fiel ao original e seria materializada à escala real, o que permitia:

- i) Ser possível efectuar a visita ao Fórum Flaviano de Conimbriga na perspectiva de um qualquer habitante daquela época, e assim ter uma correcta percepção das cores e formas da arquitectura do antigo Império Romano;

- ii) E em simultâneo, conseguir a partir dessa mesma perspectiva (cerca de 1,70m de altura), transpor para o visitante a real grandiosidade e imponência de todo o espaço envolvente do Fórum. Desde o alto e majestoso templo com as suas colunas colossais; à beleza e simetria dos pórticos do templo, dos quais as suas alas podemos visitar, algo não muito fácil de concretizar na altura, mesmo para os habitantes locais (Alarcão et al., 1994); e por fim a enorme praça de culto do saber, onde as conversas decorriam ao ritmo de uma sociedade austera, mas por vezes pretensiosa da grandiosidade do seu império.

Outra das características fundamentais deste trabalho prende-se com a interactividade. Pretendeu-se criar um espaço virtual onde o “visitante” (utilizador) tivesse uma completa autonomia de movimentos dentro do recinto do Fórum, ou seja, ele desloca-se para onde quiser, seguindo o caminho que bem entender, usando para tal um dispositivo de interacção (teclado ou rato) (Gonçalves e Mendes, 2003). O utilizador não está restringido a um simples visionamento de um vídeo, ou a uma interacção limitada, do tipo: quer seguir em frente ou deseja virar à esquerda, como se encontra frequentemente em trabalhos deste tipo.

Nota: Para precaver eventuais baixos níveis de agilidade e destreza no manuseamento dos dispositivos de interacção que permitem simular os movimentos pelo espaço virtual ou até algum receio “de mexer” no equipamento, por parte de alguns dos potenciais utilizadores; será também disponibilizada uma visita guiada a todos os monumentos que constituem o Fórum Flaviano de Conimbriga. Como se um guião de um filme se tratasse.

Quem optar por efectuar a sua própria visita ao espaço do Fórum será contemplado com outra das mais valias associadas aos mundos virtuais elaborados em VRML, mais concretamente a facilidade de utilização. Na realidade basta um simples pressionar das teclas direccionais do teclado ou um clique no rato acompanhado por arrastamento, para um qualquer leigo na matéria conseguir movimentar-se pelo espaço virtual sem dificuldades de maior, com excepção naturalmente das condicionantes realisticamente impostas pela

linguagem, como seja por exemplo a não transposição de obstáculos físicos (apesar dessa característica ser configurável). Toda a acção é efectuada de modo a propiciar ao visitante uma realista percepção do que na realidade lhe estaria a acontecer fisicamente, como seja a noção de subir escadas ou “cair” numa zona com um relativo desnível altimétrico.

Ainda relativamente à movimentação do visitante virtual pela cena, importa salientar duas questões: o acesso aos pórticos do templo é efectuado por intermédio das duas portas localizadas em cada uma das extremidades ao fundo do pátio. Para tal, o utilizador apenas terá de efectuar um clique com o rato na porta, para ser “transportado” para uma das alas do pórtico, dependendo da porta seleccionada. A saída do pórtico é concretizada do mesmo modo por intermédio de outras portas disponibilizadas para o efeito. A outra questão é relativa ao caso do utilizador “cair” na área da esplanada⁹. Como não foram encontradas indícios da existência de alguma escadaria de acesso a essa mesma esplanada, foi colocada numa das extremidades uma pequena rampa, quase imperceptível, mas suficiente para o utilizador poder abandonar essa área, pois caso contrário estaria impossibilitado de o poder fazer, devido à altura do patamar intermédio de acesso ao templo.

Nota: Apesar das semelhanças com as restantes portas (em virtude de utilizarem a mesma textura), a porta do templo não é “clicável”, fundamentalmente pelos motivos já expostos na introdução desta dissertação: não há dados concretos sobre o aspecto do seu interior, dado que todo o seu conteúdo foi vandalizado e saqueado.

Este trabalho poderá ser apreciado em: <http://lsm.dei.uc.pt/forum>.

Uma nota final para referir que, este projecto não teria sido possível concretizar nos moldes idealizados, sem o apoio e colaboração do Museu Monográfico de Conimbriga, nas pessoas do seu Director o Dr. Virgílio Correia e Técnico Superior o Dr. José Ruivo.

⁹ Espaço físico do solo sagrado, vazio, existente entre o Templo e os seu Pórticos

3.2 O Projecto

3.2.1 Investigação

No início do projecto procurámos consultar exemplos de projectos semelhantes, alguns de índole profissional. Constatou-se que a qualidade e realismo apresentadas deixavam um pouco a desejar, algo que na altura nos deixou de certa forma apreensivos. Tal facto indiciava uma de duas situações: ou a linguagem não tinha as potencialidades que lhe eram atribuídas, ou essas potencialidades não estariam a ser bem exploradas. A culpa não pode ser sempre atribuída à escassa capacidade de processamento gráfico dos equipamentos informáticos de um utilizador comum. Após consulta da muita bibliografia¹⁰ existente relativa ao tema, foi nossa profunda convicção que as potencialidades existiam, estavam era a ser mal aproveitadas.

Depois de uma abordagem mais profunda, facilmente constatámos que em VRML tudo gravita em torno de um binómio basilar: Realismo e Desempenho. Para aperfeiçoar um deles, degradamos o outro e vice-versa, como é facilmente perceptível ao ler outros capítulos desta dissertação. Como para a obtenção dos objectivos que nos propusemos atingir, ambas as componentes desempenham um papel fundamental, procurámos concentrar grande parte da nossa investigação nas questões relacionadas com a optimização, daí a forte componente da mesma existente nesta dissertação.

Este é um aspecto que esteve presente, durante as diversas etapas de desenvolvimento, pois só dessa forma os conhecimentos acumulados ao longo do tempo, puderam ser utilizados num desenvolvimento firme, bem fundamentado e realmente progressivo do projecto.

¹⁰ Essa bibliografia consultada foi sempre posterior ao ano 1996, dado que a de anos anteriores reportava versões do VRML que além de pobres, eram inclusivamente incompatíveis com a actual

3.2.2 Ferramenta de desenvolvimento

Escolhida aquela que iria ser linguagem de suporte do nosso trabalho, e antes de o começar, teria de existir um estudo/pesquisa de como esse mesmo trabalho iria ser implementado. Um ficheiro VRML é um “simples” ficheiro de texto, sendo a linguagem VRML interpretada directamente pelo *browser*. Assim, a edição do código VRML pode ser feita directamente num qualquer editor de texto, como o *notepad*. Logicamente, e pela complexidade do espaço que iríamos criar, essa opção nem sequer foi considerada, também por se saber que existiam alternativas, desde ferramentas especificamente vocacionadas para o VRML até outras que tendo o seu formato nativo, dispunham de opções de exportação para outros formatos, entre os quais o VRML. O 3DStudio era uma das mais conhecidas.

Por esta altura, e por não haver muita variedade, era comum dividir essas ferramentas em diversas categorias, de acordo com a sua funcionalidade específica. Desde os *Scene editors* aos *Script editors*, passando por aplicações de suporte que auxiliavam, inclusive ao nível da optimização, o desenvolvimento do código VRML, havia de tudo um pouco (Scheele, 1998). A escolha, apesar de não muito abundante, existia em três formatos de distribuição: aplicações comerciais, como o *Sony's Community Place Conductor*, aplicações disponibilizadas em *shareware*, como o *Spazz3D*, e mesmo em *freeware*.

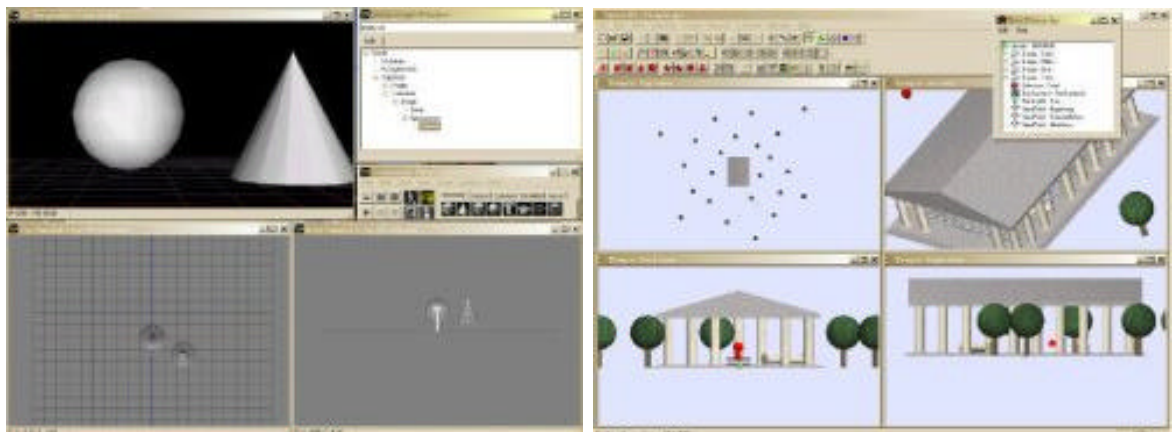


Figura 3.1 Dois exemplos inseridos na categoria dos *Scene editors*. A aplicação da Sony (esq.) e o Spazz3D (dir.)

No entanto, estas aplicações, principalmente as não comerciais, estavam, ainda assim, um pouco desprovidas de funcionalidades que tirassem o devido partido das potencialidades do VRML no seu todo.

Após o lançamento do VRML97, a Silicon Graphics, uma das empresas promotoras do VRML, produziu a sua própria aplicação de desenvolvimento VRML, o Cosmo Worlds. A segunda versão desta aplicação, editada em 1998, permitia uma completa autonomia de outras aplicações e possibilitava o desenvolvimento um projecto VRML na sua globalidade, usufruindo de todas as características da linguagem. Inclusivamente foi galardoada com diversos prémios, entre os quais a melhor ferramenta de VRML no ano de 1998. Apesar de não ser uma aplicação vocacionada para pessoas inexperientes¹¹, como era o nosso caso na altura, esta foi com naturalidade a nossa escolha. Talvez, e em grande parte, devido aos objectivos muito específicos e concretos que nos propunha-mos alcançar com este trabalho.

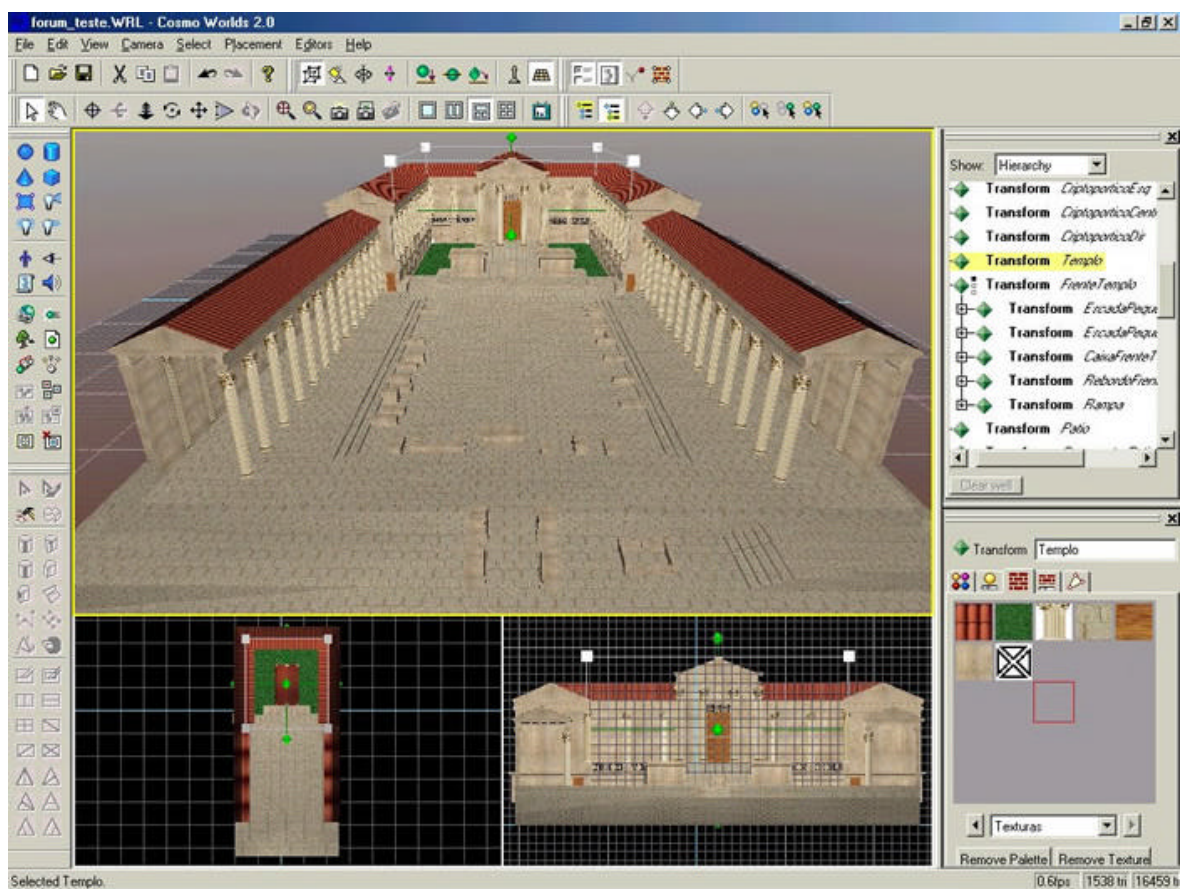


Figura 3.2 Interface do Cosmo Worlds 2.0

¹¹ Para esses seria, na altura, recomendada a aplicação *HomeSpace Builder*

Na realidade esta aplicação revelou ser de excelente qualidade, ao incidir sobre todas as funcionalidades do VRML (modelação, posicionamento, texturas, animações, sensores, áudio, iluminação, propriedades geométricas e visuais, *scripts*, instanciação, importação, transformações geométricas, optimização, entre muitas outras) de um modo prático, simples e eficaz, e sempre com recurso a um interface amigável e após algum treino, de fácil manuseamento. No entanto, não é muito fácil modelar (construir) objectos que em toda a extensão do seu volume exijam uma precisão apurada em determinadas áreas das suas faces, como sucedeu neste projecto, onde para obter uma modelação à escala e realista, as dimensões foram trabalhadas ao centímetro.

3.2.3 Modelação

Como já foi mencionado, todo o conjunto de monumentos pertencentes ao Fórum Flaviano de Conimbriga foi modelado no espaço tridimensional à escala das suas reais dimensões. Tal só foi possível, por recurso às plantas disponibilizadas para o efeito pelo Museu Monográfico de Conimbriga, as quais se revelaram bastante completas e abrangiam a totalidade, e de diversos alçados, dos monumentos pertencentes ao Fórum Flaviano de Conimbriga.

Para o desenvolvimento dos monumentos, foi adoptada uma estratégia de “construção” modelar e gradual, onde cada peça ou objecto era modelado individualmente e em separado do ficheiro principal, e só posteriormente colocado na sua posição na cena, tal como peças de um *puzzle*. Foi dada particular atenção à existência de padrões geométricos repetitivos, pois tal facto poderia simplificar o desenvolvimento do trabalho, ao permitir modelar apenas um desses padrões geométricos, e usar as propriedades do VRML para promover a repetição do padrão, com ganhos evidentes no desempenho (ver secções 4.2.4 e 4.3.4).

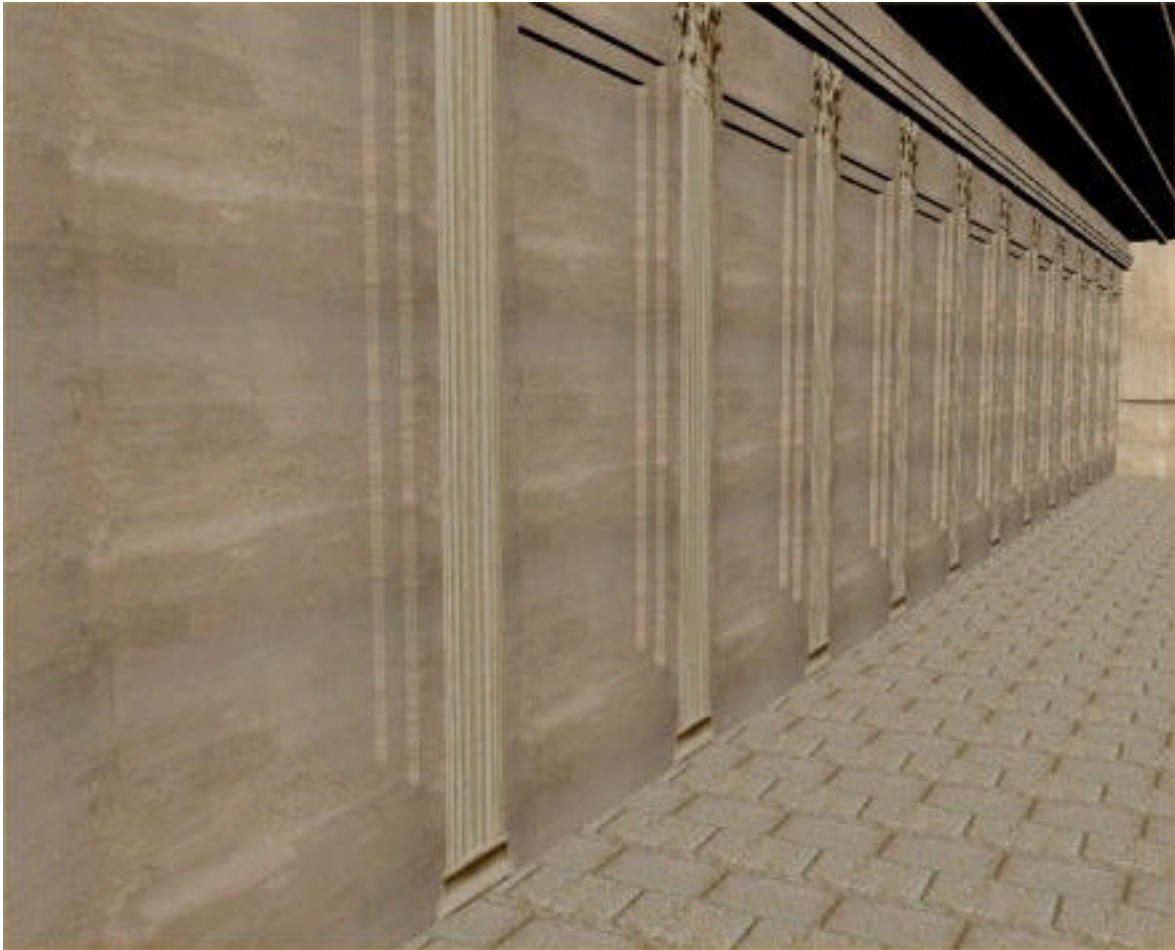


Figura 3.3 A parede deste corredor segue um padrão geométrico repetitivo

O facto das peças serem independentes entre si e o resultado final ser obtido por “assentamento” das peças umas nas outras, reforça o conceito de modularidade que deve existir num projecto desta natureza. Em determinadas situações, esta metodologia poderá ser geradora de benefícios no processo de desenvolvimento, como aconteceu no nosso caso: no momento da colocação das colunas pelo Fórum, e isto já numa fase final do projecto, constatámos que a altura do templo apresentava um erro considerável. Tirando partido dessa estratégia de modelação, essa falha foi corrigida em poucos minutos, em grande parte devido à facilidade de manuseamento das peças de uma forma autónoma.

Em todas as peças e objectos modelados, foi adoptado um cuidado especial, com particular incidência em dois factores: o seu aspecto visual ser o mais possível aproximado com a realidade e o número de polígonos necessários para a modelar¹² com esse grau de perfeição. Situações houve, em que após alguns

¹² Todas as formas geométricas visualizadas num qualquer mundo VRML, são construídas por intermédio de polígonos, normalmente triângulos, posicionados de forma adjacente entre si

testes, foi possível decrementar o número de polígonos, sem degradar as características visuais da peça.

Como foi referido anteriormente, o Cosmo Worlds foi usado na modelação de formas geométricas. No entanto, devido a algumas imprecisões já referidas, houve excepções na modelação de algumas peças que requeriam uma precisão geométrica mais apurada na sua concepção. Por esse motivo, e apenas nesta fase, foi utilizada uma ferramenta de modelação tridimensional mais poderosa que permite criar as peças com as características desejadas, o 3DStudio MAX na sua versão 2.5. As peças foram posteriormente exportadas para VRML, usando o motor de exportação existente no 3DStudio MAX.



Figura 3.4 Exemplo de peça modelada no 3Dstudio. O seu interior é visível, ou seja a peça é oca

3.2.4 Assemblagem

Em simultâneo com o processo de modelação individual de cada um dos componentes do Fórum, desde a grande praça ao mais pequeno degrau, foi-se procedendo ao seu exacto posicionamento na cena, em conjunto com as restantes formas geométricas já alojadas. Esta metodologia permitia apreciar a evolução na “construção” do Fórum. De tal forma, que foi dada uma especial atenção em definir uma sequência temporal na modelação e colocação individual

de cada uma dessas peças, escolhendo nessa ordem, primeiramente as que seriam adjacentes entre si.

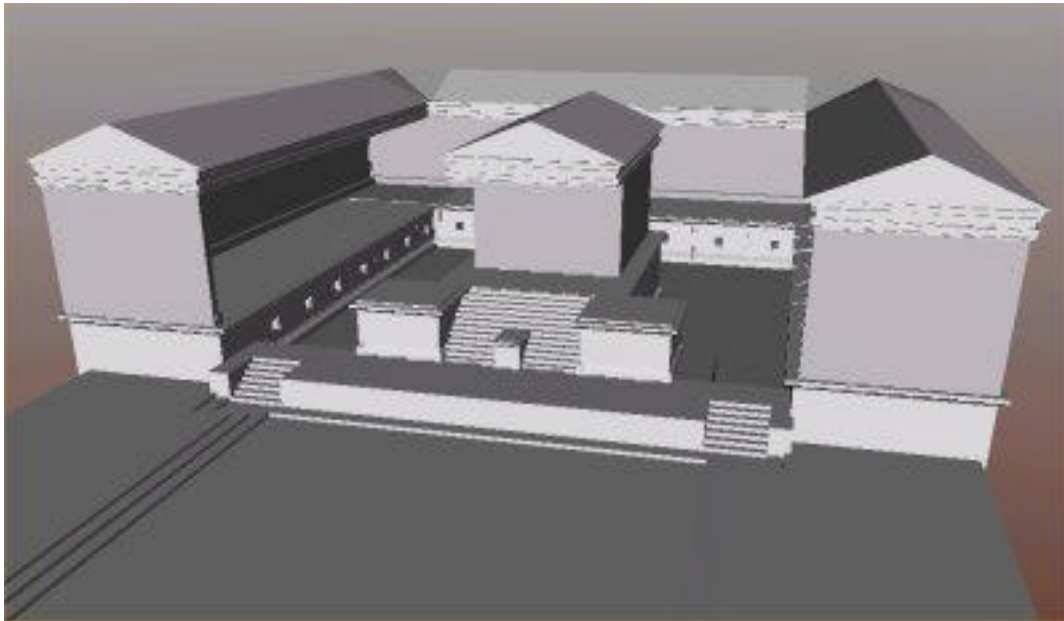


Figura 3.5 Processo de montagem gradual, inicialmente com módulos adjacentes

Não sendo estritamente necessário, e até em determinadas situações não recomendado (ver secção 4.2.5), o produto final do trabalho efectuado, no que ao código VRML diz respeito, ficou confinado a um só ficheiro, sem qualquer comprometimento do realismo e do desempenho. A portabilidade resultante e o reduzido tamanho do ficheiro final, 22Kb, foram as razões que mais contribuíram para a opção por esta solução. A decisão foi tomada sem qualquer receio, mesmo prevendo a possibilidade de, numa fase posterior do projecto, pretendermos migrar para uma outra metodologia, pois tal seria possível sem dificuldades ou problemas acrescidos, em grande parte devido à bem hierarquizada estrutura interna do ficheiro, aliada às potencialidades da ferramenta de desenvolvimento VRML em utilização.

3.2.5 Texturas

O realismo da representação só pode ser conseguido com a utilização adequada de texturas. Apenas com elas nos aproximamos do conhecimento das cores, rugosidades e padrões geométricos usados na época.

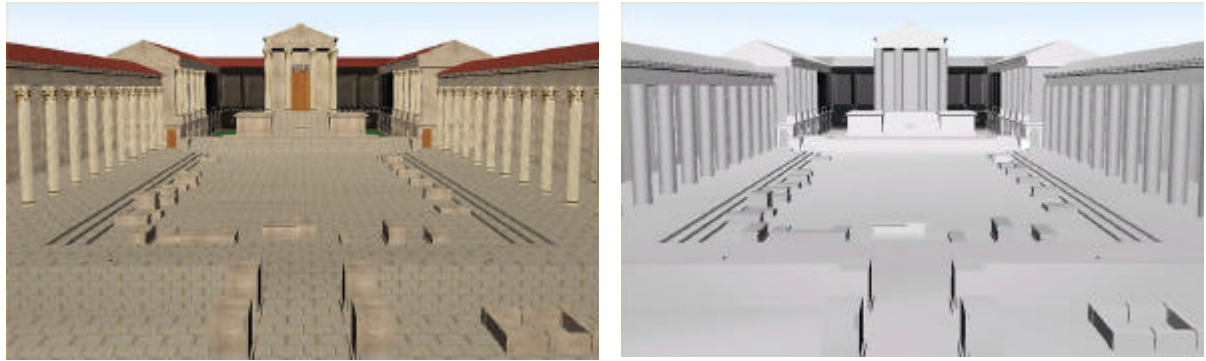


Figura 3.6 Imagem do fórum com e sem texturas

Pela sua importância para o resultado final, esta fase foi objecto de particular cuidado, nomeadamente incluindo: a recolha frequente de fotografias, com uma câmara digital, de fragmentos e pedras da construção original do Fórum; efectuando pesquisas em numerosas bibliotecas e repositórios gráficos, existentes na Web e não só; e por fim uma precisa e minuciosa manipulação gráfica, com *software* adequado para o efeito.

De seguida será feita uma abordagem mais detalhada, a questões relevantes relacionadas com algumas das texturas mais importantes usadas neste projecto.

3.2.5.1 Coluna

Esta foi, sem margem para dúvidas, a textura que mais tempo e atenção mereceu na sua concepção. Um simples olhar pelos desenhos e plantas do Fórum, bastava para mostrar a sua importância, nomeadamente a simetria e cadência do hábil posicionamento das 116 magistras colunas, com diferentes alturas, dispersas pelo recinto (este número traria obviamente consequências no desempenho), e as características muito particulares do capitel¹³ dessas colunas.

O capitel era de fulcral importância, pois ele representava uma característica muito importante na construção desta obra romana, a ordem coríntia (Alarcão et al., 1994).

¹³ Parte superior da coluna



Figura 3.7 Exemplo de um capitel coríntio

Para conseguir representar o capitel realisticamente, uma concepção manual aparentemente estaria descartada à partida, pois além da dificuldade e do tempo necessário para o consumir, seria sempre um desenho manual, e como tal retiraria algum do realismo que pretendíamos atingir neste trabalho. No entanto, todos os capiteis existentes no Museu Monográfico de Conimbriga estavam de tal forma danificados, que uma eventual reconstrução por manipulação gráfica a partir de uma fotografia, também não seria muito viável. Aparentemente este beco sem saída seria fatal para o realismo desejado, até que um dia, ao ver um serviço noticioso numa estação televisiva portuguesa, o autor deste projecto apercebeu-se da semelhança existente entre o capitel que procurava e o capitel das colunas do Supremo Tribunal de Justiça dos Estados Unidos da América. Após alguma investigação confirmou-se que, de facto, essas colunas obedeciam à mesma ordem coríntia que o Fórum Flaviano, situação devidamente atestada pelo Director do Museu. O problema estaria resolvido, não fosse o facto de todas as pesquisas realizadas na Web de imagens das colunas do Supremo Tribunal de Justiça dos Estados Unidos da América, onde fosse bem visível o seu capitel, se terem revelado insuficientes, pois as imagens obtidas eram de fraca qualidade ou bastante pequenas para poderem ser tratadas com eficiência. Para contornar esta situação, foram pesquisados os conteúdos de repositórios fotográficos para venda de algumas empresas existentes na Web. Essa pesquisa foi coroada de sucesso com a aquisição da fotografia catalogada com o número e008357, à empresa *EyeWire, Inc.*. Pelas suas dimensões e qualidade gráfica, esta fotografia já poderia ser manipulada com uma qualquer aplicação de processamento gráfico.

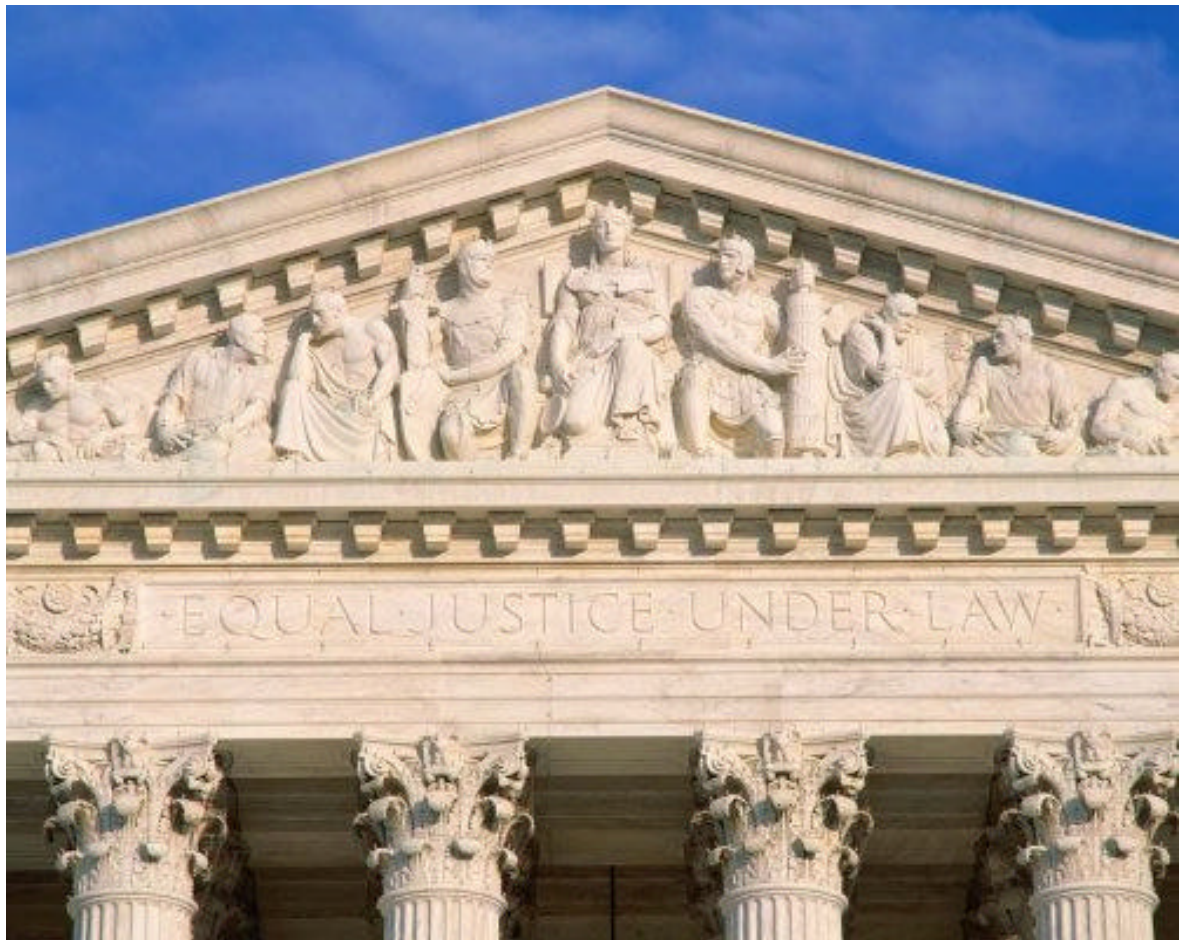


Figura 3.8 Imagem da foto adquirida. Não na sua dimensão original

Findo este processo, o mais moroso, toda a textura foi trabalhada com auxílio de uma aplicação gráfica, o *PhotoShop*, para a junção das suas diversas componentes (base, corpo e capitel), bem como a definição das transparências necessárias, o que levou a que a imagem fosse armazenada no formato PNG¹⁴. O resultado final é a textura apresentada de seguida.

¹⁴ O PNG (*Portable Network Graphics*) é um dos formatos que suporta transparências, com uma boa relação qualidade/espço ocupado

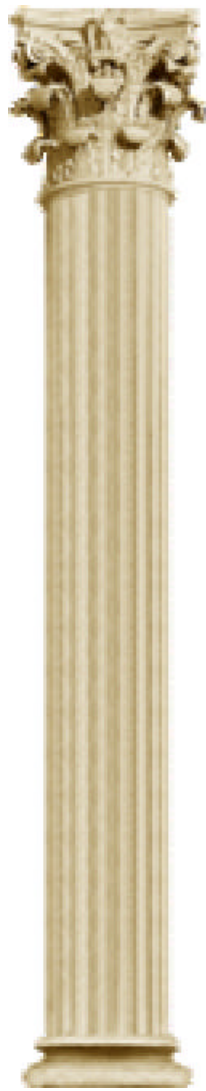


Figura 3.9 Aspecto final da coluna

3.2.5.2 Lajeado

Já fizemos alusão na introdução desta dissertação, que do fórum original da época Augustana apenas se manteve a praça. Essa preservação apenas incidiu nas suas formas e dimensões, dado que o pavimento da praça passou a ser revestido por umas lajes de dimensões consideráveis (80 cm x 60 cm), dispostas em fiadas. Pelas dimensões e características próprias, estas lajes deixam bem vincada a sua contribuição no impacto global proporcionado pelo Fórum, a quem entra no recinto através da entrada principal. Perante tal, o cuidado e atenção tidos para com esta textura foram reforçados. Após muitos testes, fotografias e manipulação gráfica, foi criada uma representação de uma pedra cujas características visuais assentam num bom compromisso, na opinião dos especialistas, com as existentes na época, potencializando assim o realismo da nossa recriação do Fórum.

Após análise do material disponível, constatou-se que esta textura poderia ser utilizada em outras áreas do recinto, nomeadamente, nos muros da entrada e na zona frontal do patamar intermédio de acesso ao templo, apenas com uma modificação das suas dimensões e sem qualquer desrespeito pela construção dos monumentos originais. Com isto, há ganhos no capítulo da optimização ao utilizar a mesma imagem, sendo possível obter as diferentes dimensões através do próprio VRML.

Na criação desta textura, tentou-se gerar uma imagem que permitisse uma repetição em mosaico, sem que a transição entre imagens adjacentes fosse perceptível, proporcionando assim um efeito de alinhamento meticuloso no lajeado. Desse modo consegue-se um bom efeito realístico com uma imagem muito pequena (2Kb), ver Figura 3.10.



Figura 3.10 Textura usada no lajeado da praça

3.2.5.3 Parede

A par com a coluna e a laje, esta seria das texturas mais importantes deste trabalho. Todas as paredes e “peças” individuais, como pódios ou plintos, deveriam ser revestidos com esta pedra de aspecto rebocado, que pelas suas propriedades e grande área de exposição, caracteriza de forma indelével a aparência geral do Fórum, pois onde quer que nos “encontremos” dentro do recinto, é a tonalidade cromática dessa pedra que predomina.

Mais uma vez foram necessários muitos testes e análises, para em conjunto com os especialistas, conceber uma textura representativa da pedra rebocada da época. Também aqui se tentou obter uma textura passível de compor repetitivamente de forma adjacente, sem comprometer o aspecto visual pretendido.



Figura 3.11 Textura usada como padrão das paredes

3.2.5.4 Gradeamento

Devido ao desnível existente entre o pórtico do templo e a esplanada¹⁵, havia um gradeamento entre as bases das colunas do referido pórtico. Esse gradeamento era constituído por uma trama de elementos cruzados, e tinha uma altura de 90 cm (Alarcão et al., 1994).

Dado não existirem exemplares, mesmo danificados, de uma parcela desse gradeamento, esta foi a única textura concebida de raiz para este trabalho. Para tal, partiu-se da digitalização de uma das plantas onde esse gradeamento surgia. Posteriormente foram eliminados ruídos e aperfeiçoados os traços, até obter um efeito realista da trama de elementos cruzados. Por fim foram aplicadas transparências para esse efeito realista ser completo, ao permitir a visualização por entre os elementos cruzados do gradeamento.

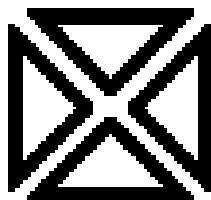


Figura 3.12 Textura do gradeamento

¹⁵ Espaço físico do solo sagrado, vazio, existente entre o Templo e os seu Pórticos

3.2.6 Coloração

Apesar das muitas fotografias das pedras originais obtidas com uma câmara digital, tal não significou que a busca das cores e padrões usadas na época, ficasse concluída facilmente. Fomos alertados pelos especialistas, que devido à erosão natural ocorrida ao longo dos séculos, as pedras já não apresentavam as tonalidades e cores originais (apesar de manterem a textura original). Assim, foi necessária a colaboração dos especialistas na busca das cores originais. Estas foram posteriormente aplicadas às diversas texturas utilizadas. Como texturas aplicadas em zonas diferentes da cena, teriam tonalidades diferentes, ainda que com uma coloração semelhante, essa pesquisa teve de ser repetida para cada uma dessas texturas, nomeadamente, as paredes, o chão e as colunas.



Figura 3.13 O antes e depois da coloração

Nesse processo exaustivo e meticuloso foi utilizada uma aplicação gráfica, o Paint Shop Pro 7.04¹⁶. As características visuais das texturas foram facilmente manipuladas, através da alteração de diversos parâmetros de várias propriedades associadas à cor, existentes na aplicação.

3.2.7 Iluminação

O uso de iluminação num qualquer ambiente virtual tem como finalidade a simulação dos efeitos visuais gerados pelas fontes emissoras de luz, sejam elas naturais ou não, existentes no espaço que se está a recriar. Essa simulação visa

¹⁶ Inicialmente considerámos a possibilidade de utilizar o Photoshop, unanimemente considerada a melhor aplicação gráfica do mercado, mas as potencialidades das últimas versões do Paint Shop Pro, aliadas a uma maior destreza do autor no manuseamento desta aplicação, levaram à sua escolha

essencialmente a obtenção de um acréscimo no realismo global da cena, tentando proporcionar ao seu “visitante” uma melhor percepção de toda a ambiência que o rodeia. Se possível tal deve ser conseguido minimizando o mais possível os seus efeitos no desempenho (ver secção 4.1.6). Por esse motivo, as nossas escolhas consideraram os seguintes aspectos:

- i) Das diferentes formas de emissão de luz existentes no VRML, fazer uma escolha cuidadosa de qual, ou quais, as que deveriam ser utilizadas (ver secção 4.2.6). A escolha recaiu na utilização de um *DirectionalLight* e *PointLight*;
- ii) O número fontes emissoras de luz existente na cena (dois);
- iii) O seu posicionamento pela cena, de modo a proporcionar o efeito pretendido.

Como os cálculos de iluminação são efectuados a toda a estrutura geométrica dos objectos directamente afectados por esse(s) foco(s), um posicionamento adequado desses focos pela cena, pode possibilitar a visualização de alguns efeitos de sombreamento na superfície dos objectos. No entanto, para acautelar a existência permanente de, pelo menos, uma fonte de luz na cena (de outro modo nada se visualizaria), o avatar “traz consigo” um foco que ilumina na direcção da sua área de visualização, ao jeito de um capacete de mineiro. Como por defeito esse foco está sempre activo (ligado), o que aconteceria é que os efeitos de sombreamento, referidos anteriormente, seriam eliminados, e consequentemente perder-se-ia realismo, sem usufruto de todas as potencialidades da iluminação colocada na cena. Deste modo, considerámos ser boa opção manter esse foco “desligado” por defeito, podendo a qualquer altura um utilizador “ligá-lo” por interacção com o *browser* VRML.

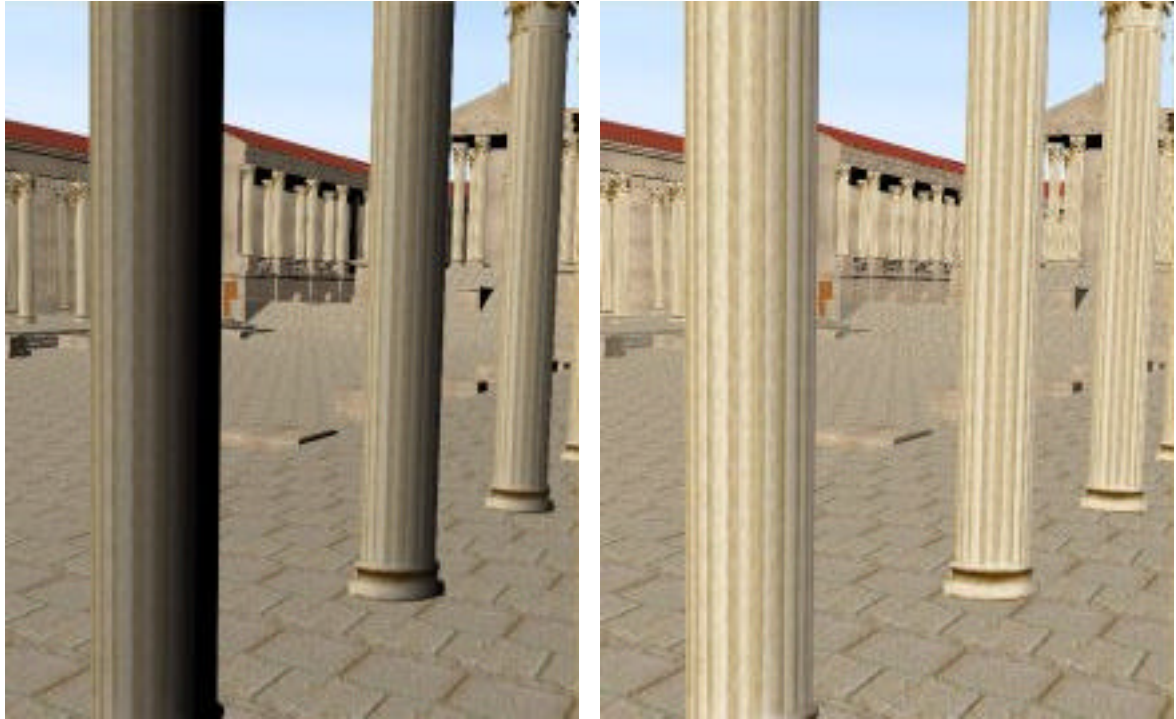


Figura 3.14 Com o foco desligado (esq.). Com ele ligado (dir.)

3.2.8 Optimização

A optimização dos mundos criados em VRML é um aspecto fundamental para conseguir apresentar um desempenho satisfatório. Esta preocupação esteve presente em todo o nosso trabalho. Pela sua importância e pelos resultados obtidos, os aspectos de optimização introduzidos neste trabalho são descritos no próximo capítulo.

4. Aspectos de Optimização

4.1 Visualização 3D (em VRML)

4.1.1 Introdução

Foi a utilização de uma terceira dimensão que projectou para um outro nível de realismo algumas áreas do meio informático, como os videojogos, Internet ou a realidade virtual. Tal facto, propicia ao utilizador o acesso a informação potencialmente mais rica, quer em conteúdo quer em interactividade.

À excepção destes últimos anos, a evolução da visualização tridimensional ao longo do tempo, não seguiu o mesmo ritmo de outras áreas do mundo informático. Vários factores podem ser apontados, mas todos eles se resumem a esta conclusão óbvia: a visualização de cenas tridimensionais difere substancialmente da visualização de imagens num contexto bidimensional. É precisamente essa discrepância e as suas consequências que pretendemos abordar neste sub-capítulo, para melhor inteirar o leitor das especificidades deste tipo de visualização.

A primeira questão a considerar prende-se com a visualização. Apesar dos objectos estarem posicionados no espaço tridimensional, eles são visualizados, neste contexto, bidimensionalmente no periférico de visualização (ecrã). Como os objectos coabitam num espaço tridimensional, tem de haver forma de os visualizar em qualquer ponto do espaço. Numa analogia com uma câmara fotográfica (não digital), quando se pretende tirar uma fotografia, primeiro posiciona-se a câmara na posição desejada, e só depois é tirada a fotografia, dando origem ao armazenamento numa película plana, logo bidimensional, do conteúdo da imagem captada, essa sim tridimensional. Analogamente, para visualizar uma cena tridimensional num equipamento informático, é necessária a existência de uma câmara virtual na cena, de modo a visualizar determinada parte da mesma. De seguida essa imagem tridimensional captada pela câmara virtual é representada na área destinada para o efeito, o ecrã. Este processo é conhecido como *rendering*.

Para ter uma perspectiva da cena a partir de outro ângulo simplesmente move-se a câmara virtual para nova posição, faz-se o *rendering* da cena e visualiza-se no ecrã. Se este processo for contínuo por cada movimento da câmara, por mais pequeno que seja, é conseguida uma simulação do efeito de deslocação pelo espaço tridimensional, tal qual acontece na realidade. Isto é algo que não pode ser conseguido num universo bidimensional, e destaca-se como a grande potencialidade do 3D. Mas este é um processo de difícil execução e bastante pesado computacionalmente. A realização de um *rendering* suficientemente rápido, de modo a obter um movimento fluido e contínuo pela cena tridimensional, requer uma capacidade de processamento substancial quando comparada com outras aplicações informáticas. Até algum tempo atrás, alguma da visualização de ambientes tridimensionais só era possível em poderosos e dispendiosos equipamentos informáticos. Com o advento dos novos *chips* gráficos, o atraso no processo evolutivo da visualização tridimensional, mencionado há pouco, está a ser de certa forma recuperado e até compensado com os excelentes avanços tecnológicos conseguidos por alguns fabricantes, com particular destaque para a *nVidia*.

As próximas secções desta dissertação, irão ser dedicadas à apresentação de alguns dos mais importantes e dispendiosos cálculos, maioritariamente matemáticos, efectuados durante o processo de visualização tridimensional. Com particular ênfase, como não podia deixar de ser, às questões relacionadas com VRML. Este é o motivo, pelo qual, por exemplo, não se abordará a questão do *rendering* de polígonos¹⁷ que, apesar de também influir na globalidade do processamento interno necessário à visualização de um qualquer espaço tridimensional, é comum a praticamente todos os tipos de visualização tridimensional, qualquer que seja a tecnologia utilizada, não sendo uma problemática exclusiva do VRML.

4.1.2 Detecção de colisões

A necessidade de proporcionar um maior realismo ao comportamento interactivo dos mundos elaborados em VRML, deu origem a uma completa reformulação do

¹⁷ Responsável pela existência, logo visualização, de superfícies tridimensionais numa cena, dado que todas elas são formadas pela disposição adjacente de polígonos, normalmente triângulos

VRML da 1ª para a 2ª versão, de tal modo que ficheiros de diferentes versões tornaram-se incompatíveis. De entre muitas, uma das mais importantes inovações introduzida foi a detecção de colisões, que colocou um ponto final na movimentação irrealista, tipo fantasma, que era permitida aos utilizadores da 1ª versão, como a possibilidade de atravessar paredes ou os muros servirem apenas como adereço cénico. Realisticamente, o ser humano não consegue transpor sólidos, sendo essa a função da detecção de colisões: impedir a progressão do movimento do utilizador virtual sempre que uma superfície sólida se lhe deparar pela frente.

Mas naturalmente e como seria de esperar, o acréscimo de realismo é normalmente um factor condicionante do desempenho, essencialmente devido ao aumento de processamento necessário para acompanhar esse acréscimo de realismo. A detecção de colisões é de todas as funcionalidades introduzidas, uma das que tem maior exigência de capacidade de processamento.

Implementar a detecção de colisões é uma tarefa que não tem nada de simplista e trivial como eventualmente poderá parecer à primeira vista. Os testes de colisões não são mais que fórmulas matemáticas, que calculam a distância entre o visitante virtual e as superfícies sólidas existentes na cena, cálculos esses que se tornam bastante mais complexos por estarem a ser efectuados no espaço tridimensional. Quando essa distância atingir um valor mínimo - um raio de colisão definido pelo autor -, o visitante é impedido de prosseguir o movimento, pelo menos no sentido em que o estava a fazer (Carey et al., 1997).

Foi já referido que a sensação de movimento pela cena é transmitida pelo constante *rendering* que é efectuado por entre os pequenos deslocamentos da câmara virtual controlada pelo utilizador. Mas, antes da execução de cada um desses deslocamentos, os testes de colisão têm de ser realizados, de modo a aferir da validade ou não do movimento que se está a tentar efectuar. Numa tentativa de contabilização desses testes, os movimentos da câmara são realizados várias vezes por segundo e os testes têm de ser efectuados a todos os objectos passíveis de colisão¹⁸ existentes na cena. Por sua vez cada objecto é

¹⁸ De modo a reduzir o número de testes, é possível desactivar os testes de colisão a algumas superfícies

constituído por dezenas ou centenas de polígonos que definem a sua superfície, sendo a esses polígonos que são efectuados os testes¹⁹.

Como o VRML tenta fazer uma aproximação bastante precisa da realidade, o visitante virtual que se desloca pela cena, não é um ponto no espaço, mas sim um volume tridimensional que simula um visitante real e, portanto, tem altura e largura²⁰. É possível que o percurso entre duas superfícies esteja livre de obstáculos, e, no entanto, haja a possibilidade do movimento entre elas ser travado algures num ponto intermédio. O volume ocupado pelo visitante virtual também é considerado nos testes de colisão, e apesar do caminho em linha recta estar aparentemente livre, o espaço disponível num dado ponto pode não ser suficiente para permitir a passagem do volume representativo do avatar (pode ser imaginado como um cilindro).

Além destes, ainda outros cálculos menos significativos também têm que ser efectuados pelo *browser*, como por exemplo a definição do comportamento do avatar quando ocorrer uma colisão. Mas a questão mais importante é o significativo processamento necessário para implementar a detecção de colisões. Este aspecto deve merecer particular atenção por parte dos autores e todas as medidas que possibilitem uma redução do número de testes a efectuar devem ser analisadas e implementadas sempre que possível (algumas dessas medidas são indicadas na secção 4.2.3).

4.1.3 Gravidade

A par da detecção de colisões, uma outra funcionalidade de inquestionável importância para o aperfeiçoamento do realismo foi introduzida com a 2ª versão do VRML: a noção de gravidade, que tem como finalidade manter o utilizador virtual sempre “colado” ao chão, com repercussões na sua movimentação pela cena, designadamente ao acompanhar as irregularidades do terreno. Estas duas funcionalidades geralmente surgem associadas uma à outra, pois nem fazia sentido a existência de uma sem a outra, dada a similaridade nos cálculos a efectuar e a finalidade de ambas.

¹⁹ Também aqui há forma de simplificar os testes usando o *ColisionProxy*. Ver secção 4.2.3

²⁰ Estes parâmetros podem ser definidos no campo *avatarSize* do nó *NavigationInfo*

Tal como acontece na detecção de colisões, a gravidade não é algo que surja naturalmente, pois têm de ser realizados cálculos para o ajuste, pelo *browser*, da posição do avatar de acordo com as condições da superfície sobre a qual se está a deslocar. A simulação da gravidade é feita ao “forçar” o acompanhamento dos movimentos do avatar ao valor mínimo da coordenada em y da superfície que ele está a “pisar” em cada momento (Carey et al., 1997). Se, por exemplo, o avatar cair do peitoril de uma janela, ele é impelido para “baixo” pela gravidade - o *browser* mantém sempre a noção do vector gravitacional, normalmente no sentido negativo do eixo dos y's, uma vez que a gravidade é aplicada no sentido desse mesmo vector – até encontrar uma superfície virtualmente sólida, altura em que esse movimento descendente é interrompido, tal como aconteceria na realidade. O teste que detecta esta situação é muito similar com o efectuado na detecção de colisões, quando em movimento pela cena, com excepção do volume de colisão ser aplicado na vertical em vez de o ser na horizontal.

Associado a estes cálculos gravitacionais, um outro teste também ele condicionador da capacidade de movimentação pela cena tem de ser efectuado. Quando alguém se dirige a um muro com dimensões (altura) consideráveis, a sua progressão é travada após a aproximação ao mesmo, em virtude da incapacidade para transpor obstáculos com tal altura, sem meios auxiliares. Se, por outro lado, o avatar for em direcção a umas escadas, o movimento já não é travado, como acontecia com o muro. O que diferencia as duas situações é a altura do obstáculo. No caso do muro, este impede a transposição sem meios auxiliares. No caso das escadas, cada degrau que as constitui, tem uma altura suficientemente baixa para poder ser transposto sem grande dificuldade por uma pessoa normal, não constituindo por isso impedimento à progressão do deslocamento. O VRML possibilita a definição de uma altura máxima dos objectos transponíveis pelo avatar. Qualquer objecto com uma altura superior à definida irá gerar uma colisão entre o avatar e a superfície, impedindo o seu deslocamento. É este o teste referido no início deste parágrafo, no qual se define se um objecto é transponível ou gerador de colisão, mediante a sua altura relativamente à superfície sobre a qual o avatar se está a deslocar.

Para finalizar este tópico da gravidade, a referência a mais dois procedimentos efectuados, embora estes não consumam tantos recursos como os já descritos. Um dos parâmetros configuráveis no VRML permite definir uma altura para o

avatar, logo é necessário efectuar cálculos de distância ao solo da câmara virtual que simula os olhos do avatar. Deste modo é possível simular as características reais de um visitante no mundo virtual. E, por último, o facto da gravidade só ser aplicada quando existe uma superfície virtual sob o avatar, pois caso isso não suceda, ele fica suspenso no “ar”. Cabe ao *browser* a constante verificação da existência, ou não, de uma superfície virtualmente sólida sob o avatar, para a aplicação, ou não, dos cálculos de gravidade.

4.1.4 *Culling*

Na “arte” da representação tridimensional, qualquer que seja a técnica usada, a quantidade de informação (geométrica) a visualizar é, por norma, inversamente proporcional ao desempenho obtido durante essa mesma visualização. O VRML não é diferente nesse aspecto, e, por conseguinte, os *browsers* utilizam estratégias baseadas nas características do VRML, que limitem a quantidade de informação geométrica a ser representada num dado momento, sem prejuízo da qualidade da imagem apresentada ao utilizador. Essa estratégia que possibilita a não representação de geometria não essencial à visualização é denominada por *culling*, e os cálculos efectuados para a concretizar incidem em duas vertentes:

- i) O primeiro teste consiste em determinar quais os objectos (geométricos) que estão totalmente fora do volume de visão do avatar, pelo que não se torna necessária a sua representação. O teste consiste em comparar a *bounding box*²¹ de cada um dos objectos existentes na cena, com o volume de visão actual; se os dois volumes não se intersectarem significa que o objecto em causa não está visível e pode ser ignorado na representação da cena (Cosmo Worlds 2.0 User’s Guide, 1998). Para uma melhor optimização de todo este processo é essencial uma boa estrutura hierárquica de todos os objectos da cena (ver secção 4.2.2). O cálculo da intersecção de volumes num espaço tridimensional não é uma operação simples em virtude dos cálculos matemáticos envolvidos e de estes serem efectuados exaustivamente, dado o constante alterar do volume de visão resultante da movimentação do utilizador pela cena.

²¹ Paralelepípedo que delimita todo o volume de um qualquer objecto tridimensional

O volume de visão que é responsável pela quantidade de informação visível pelo avatar, é um parâmetro configurável no VRML, por intermédio do campo *fieldOfView*²². Este valor tenta aproximar os limites da capacidade visual humana, e quando alterado o efeito visual obtido pode-se assemelhar à utilização de lentes panorâmicas num equipamento fotográfico. Há, portanto, cálculos (matemáticos) que é necessário efectuar para não desproporcionar os objectos relativamente às suas dimensões.

- ii) *Backface culling* é a terminologia usada para designar o outro teste, e surge como mais uma medida que aumenta a optimização de ficheiros VRML. Dada a natureza poligonal de todas as superfícies geométricas existentes num mundo virtual, os polígonos representativos de um sólido nunca serão visíveis a partir do seu interior. Desse modo é evidente que não é necessário representar esse lado “invisível” dos polígonos. Num qualquer sólido existente numa cena VRML, este procedimento representa uma economia de 50% do número de polígonos a representar e a considerar nos cálculos de iluminação (responsáveis pela visualização), com os consequentes ganhos no desempenho geral do mundo virtual. Este é também um raciocínio válido para objectos que não sendo sólidos fechados, tenham faces que nunca sejam visíveis pelo visitante virtual.

Em termos de cálculos efectuados, *backface culling* não é mais que um processo que evita a representação pelo *browser* das faces poligonais cujas normais “apontem” no sentido oposto ao da posição do observador virtual, com os cálculos a serem efectuados da seguinte forma (Hartman e Wernecke,1996): seja V a posição do observador no espaço coordenado; seja N o vector que representa a normal do polígono, e P um qualquer ponto pertencente ao plano definido pelos vértices do polígono (excepto a origem). Se a subtracção do produto interno de V e N , com o produto interno de N e P for maior que zero ($(V.N)-(N.P)>0$), a face do polígono será visível, caso contrário essa mesma face será invisível, e desse modo ignorada nos cálculos de iluminação.

²² Campo pertencente ao nó *Viewpoint*

4.1.5 Remoção de superfícies ocultas

Ainda respeitante ao *rendering* da cena num qualquer momento, devem ser considerados os cálculos envolvidos na determinação das superfícies, ou parte delas, que mesmo estando dentro do volume de visão não devem ser representadas pelo *browser*, devido a estarem total, ou parcialmente, encobertas por outra(s) superfície(s). Os cálculos que implementam este procedimento são efectuados por algoritmos de remoção de superfícies ocultas, cuja função se resume a determinar qual a superfície dentro do espaço tridimensional, que estando em linha de vista com o visitante virtual, se encontra mais próxima do mesmo, para desse modo ser ela a representada pelo *browser*. Estes cálculos são efectuados continuamente, especialmente quando em movimento pela cena, devido à constante alteração do campo visual do avatar.

Nota: Por norma, o algoritmo usado para implementar esta operação, é conhecido como Algoritmo *Z-buffer* (Talvez o algoritmo de remoção de superfícies ocultas mais conhecido).

4.1.6 Iluminação

Qualquer ambiente virtual criado tridimensionalmente deve simular tanto quanto possível, a realidade do mundo que nos envolve. Os objectos são modelados de modo a reflectir o real, as posições que ocupam entre si expressam a realidade, e são-lhe aplicadas cores e texturas que melhoram o seu realismo. Mas, nada disto seria suficiente, se o processo que simula a visão humana fosse inadequado, ineficaz e pouco realista.

O sistema visual humano recebe a luz reflectida pelos objectos do mundo, luz essa que estabelece a forma, a localização e a cor dos objectos. Ora, num mundo virtual, o cálculo das fontes emissoras de luz incidente nas superfícies existentes nesse mundo, não é praticável num *rendering* em tempo real (como acontece no VRML). Assim são necessárias optimizações no processo, para um visionamento mais eficaz e realista em tempo real. Deste modo, os cálculos de iluminação, designadamente a cor, são efectuados nos vértices dos polígonos que definem a

superfície. A coloração final de toda a superfície, é feita por interpolação das cores definidas para os vértices (Marrin e Campbell, 1997). Esta aproximação é geradora de resultados bastante satisfatórios, no que ao realismo diz respeito.

Mas antes do cálculo da cor de um qualquer vértice, é necessário identificar todas as fontes emissoras de luz existentes na cena. Alguma dessa luz é espontaneamente gerada pelo próprio objecto (*emissiveColor* e *ambientIntensity*²³), mas a cor final do objecto provém essencialmente das fontes de luz externas, como por exemplo, a luz solar ou um candeeiro numa secretária.

A pertinência da abordagem deste tópico neste capítulo, prende-se com o número dessas fontes externas de luz, existentes numa cena. Como se poderá constatar na fórmula a seguir apresentada, há um significativo aumento dos cálculos a realizar, por cada fonte de luz adicionada. Este processo normalmente implica, em quase todos os casos, um decréscimo significativo do desempenho, em virtude do consumo de recursos originado por este processamento.

Assim, para dar uma ideia da complexidade dos cálculos efectuados com a iluminação, apresenta-se de seguida a fórmula de cálculo da cor num dado ponto da superfície (Carey et al., 1997). Este modelo de iluminação em VRML, supostamente ideal, é composto por equações detalhadas de cálculo de cada uma das componentes da cor (R(ed), G(reen), B(lue)) a aplicar nos objectos geométricos existentes na cena a ser representada:

$$I_{rgb} = I_{Frgb} \times (1 - f_0) + f_0 \times (O_{Ergb} + \sum (on_i \times atenuante_i \times spot_i \times I_{Lrgb} \times (ambiente_i + difusa_i + especular_i)))$$

- I_{rgb} - Intensidade final de cada um dos componentes (RGB);
- I_{Frgb} - Cor do nevoeiro (caso exista);
- f_0 - Interpolador do nevoeiro (pode ter de efectuar mais cálculos);
- O_{Ergb} - Cor emissiva das propriedades da superfície;
- \sum - Somatório para todas fontes emissoras de luz externas;
- on_i - 0 ou 1 (se o fonte de luz afecta ou não a geometria);
- $atenuante_i$ - Factor de atenuação (mais cálculos são efectuados);

²³ Parâmetros do nó *Material*

- $spot_i$ - Afecção do *Spotlight* (tipo de iluminação usado no VRML). São efectuados cálculos de acordo com os parâmetros específicos deste nó;
- I_{Lrgb} - Cor da fonte emissora de luz a ser processada actualmente;
- $ambiente_i$ - Componente de luz ambiente ($=I_{ia} \times O_{Drgb} \times O_a$);
- $difusa_i$ - Componente difusa ($=I_i \times O_{Drgb} \times (N \cdot L)$);
- $especular_i$ - Comp. especular ($=I_i \times O_{Srgb} \times (N \cdot ((L + V / |L + V|))^{Brilho \times 128}$);
 - I_{ia} - Intensidade de luz ambiente da fonte emissora de luz;
 - O_{Drgb} - Cor difusa das propriedades da superfície (textura ou cor);
 - O_a - Luz ambiente emanada pela superfície;
 - I_i - Intensidade da fonte emissora de luz;
 - Brilho - Brilho associado à superfície;
 - N - Normal (normalizada) ao ponto da superfície;
 - L - Vector normalizado direccionado à fonte emissora de luz;
 - V - Vector normalizado direccionado ao observador;

O uso de iluminação num mundo virtual tridimensional é vital para a obtenção dos resultados pretendidos, mas em virtude dos cálculos que acarreta, o seu uso deve ser feito com moderação, procurando um realismo adequado, sem prejuízo do desempenho. Mais detalhes sobre o correcto uso da iluminação num mundo VRML são apresentados na secção 4.2.6.

4.1.7 Normais

Basta uma análise, ainda que superficial, à secção anterior para aferir da importância do papel que as normais desempenham no cálculo da iluminação. Como tal, o modo como essas normais são geradas reveste-se de particular importância para o grau de realismo de um mundo VRML.

A provisão da equação da iluminação, com as normais necessárias aos cálculos envolvidos, pode ser efectuada de duas formas distintas: ser o próprio autor a

definir as normais necessárias (este procedimento é efectuado por intermédio do nó *Normal*), ou então, e como acontece na maior parte dos casos, serem essas normais geradas automaticamente (Carey et al., 1997). Isto é concretizado por intermédio de processamento interno, que é efectivado de acordo com vários pressupostos, alguns dos quais configuráveis.

Alguns nós de modelação geométrica usados no VRML²⁴, dispõem de um campo designado por *creaseAngle*, onde é indicado um ângulo (em radianos) que afecta o modo automático de geração das normais. Se numa superfície com esse parâmetro definido, o ângulo de incidência entre dois polígonos adjacentes for menor que o ângulo definido nesse campo, apenas uma normal será gerada nos vértices em causa, originando uma suave transição do cálculo da iluminação na fronteira definida (aresta) entre esses polígonos adjacentes; se, pelo contrário, o ângulo de incidência for superior ao valor do *creaseAngle*, irão ser geradas nos vértices afectos aos polígonos, tantas normais quantas os polígonos, resultando neste caso numa descontinuidade entre esses polígonos aquando do *rendering* da superfície que os contém (Marrin e Campbell, 1997). O efeito típico desta situação, é a visualização de um vinco entre os polígonos. A Figura 4.2 da página 91 ilustra bem o efeito do *creaseAngle*.

De referir ainda que as normais, como vectores que são, possuem um comprimento. Em muitas das situações, mas mais especificamente nos cálculos de iluminação, esse comprimento deve ser igual à unidade. A isso chama-se vector normalizado. Essa normalização, também tem de ser efectuada durante o processo de geração das normais.

4.1.8 Sensores

No que toca à obtenção de realismo no VRML, os sensores podem desempenhar um papel de certa relevância, na medida que é por intermédio deles que alguns dos comportamentos mais comuns do dia a dia são implementados. Por exemplo, a porta que necessita de ser aberta antes de se entrar em casa.

²⁴ Os nós em causa são: *IndexedFaceSet*, *ElevationGrid* e *Extrusion*

Os sensores são mecanismos de captação de eventos que permitem definir comportamentos dentro da cena. Estes podem ser manipulados/controlados de três modos distintos: pela movimentação do visitante virtual pela cena, temporalmente ou por intervenção directa do utilizador através do dispositivo de interacção, normalmente o rato.

Existem duas classes de sensores no VRML (Carey et al., 1997):

- i) Sensores Ambientais²⁵ – A quase totalidade dos sensores pertencentes a esta classe monitorizam a movimentação do visitante virtual pela cena e são activados sem qualquer outro tipo de intervenção do utilizador. Há um outro sensor que se incorpora nesta classe, mas com características diferentes dos anteriores. Este é gerido por tempo (*TimeSensor*), e o seu comportamento resume-se a um gerar de eventos à medida que o tempo passa.
- ii) Sensores de Interacção Directa²⁶ – Estes apenas são activados por intervenção directa do utilizador por intermédio do dispositivo de interacção.

É em virtude deste contínuo estado de alerta e monitorização dos passos do visitante pelo mundo virtual que incluímos este tópico neste capítulo, pois a sua inclusão é geradora do processamento necessário para implementar o controlo dos eventos.

Será que o avatar já está dentro do volume definido para o início desta animação? Será que o dispositivo de interacção está sobre uma superfície controlada por um sensor e já foi accionado o mecanismo de activação desse mesmo dispositivo? Será que o avatar já embateu com determinada superfície para assim poder emitir o som associado a esse embate? Será que já passou tempo suficiente para iniciar esta animação? Estas são algumas das questões que emergem com a utilização dos sensores. O facto destas questões e consequentemente os cálculos necessários à sua efectivação, se colocarem num universo tridimensional, fazem com que este processo seja algo complexo e, por conseguinte, mais consumidor de recursos.

²⁵ Os nós em causa são: *Collision*, *ProximitySensor*, *VisibilitySensor* e *TimeSensor*

²⁶ Os nós em causa são: *Anchor*, *TouchSensor*, *CylinderSensor*, *PlaneSensor* e *SphereSensor*

Apesar dos cálculos necessários e consequente consumo de recursos, a utilização de sensores é em várias situações aconselhada como meio de optimização em VRML (secção 4.2.12).

4.1.9 Áudio

A capacidade de incorporar áudio é outra das características do VRML que facilita a obtenção de realismo. Apesar de não estar directamente relacionada com a visualização, a opção de incluir este tópico neste capítulo, deve-se não apenas à complementaridade resultante da incorporação do áudio em ambientes virtuais, mas também ao facto da sua utilização implicar uma disponibilidade de recursos por parte do equipamento responsável pela visualização.

A inclusão de som no VRML não é concretizada apenas e em exclusivo com um nó, mas há no entanto um (*Sound*), que é o responsável pelas características da propagação do conteúdo sonoro no espaço tridimensional, como seja o alcance de audibilidade ou a detecção da proveniência do som no espaço tridimensional, recorrendo às capacidades estereofónicas. O primeiro ponto é resolvido por intermédio da indicação de duas elipsóides, sendo uma delas interior à outra. No volume definido pela elipsóide interna, região onde deve estar localizada a fonte emissora, o som é audível na totalidade da sua intensidade²⁷. É na zona compreendida entre as duas elipsóides que se efectua a atenuação da intensidade do registo sonoro. Essa atenuação sonora é concretizada por intermédio de uma fórmula matemática de interpolação linear, que vai diminuindo o registo até ele deixar de ser audível, o que acontece quando se atinge a fronteira definida pela elipsóide exterior (Carey et al., 1997).

O campo *spatialize* do nó *Sound* permite configurar por intermédio de uma referência booleana, a capacidade do visitante poder ter uma percepção da origem da emissão do som relativamente à sua própria posição no mundo virtual, daí a referência de uma audição em estéreo, pois para simular a posição da fonte emissora, o som deve surgir com mais intensidade num dos canais de saída. Este processo é implementado com base no ângulo formado entre a posição do avatar e da fonte emissora. Inicialmente é efectuada uma projecção do vector direccional

²⁷ Este parâmetro também é configurável através do campo *intensity*

da fonte emissora (definido no campo *direction*) no plano XZ, e o ângulo pretendido é o ângulo formado entre a componente Z dessa projecção com a posição actual do avatar. Deste ângulo é calculado um valor de referência compreendido no intervalo [0.0, 1.0], que irá permitir estabelecer os respectivos níveis do canal esquerdo e do canal direito, por intermédio das seguintes equações (Carey et al., 1997):

$$\text{nívelEsquerdo} = 1 - \text{valorRef}^2$$

$$\text{nívelDireito} = 1 - (1 - \text{valorRef})^2$$

Os valores resultantes destas equações serão posteriormente usados para o cálculo final da intensidade de saída dos dois canais, em conjugação com a distância à fonte emissora.

4.2 Optimização em VRML

4.2.1 Introdução

Cedo neste projecto se percebeu que para o cumprimento dos objectivos traçados: realismo, desempenho e dimensão dos ficheiros, principalmente para estes dois últimos, seria fundamental utilizar esquemas de optimização. Desde logo concentrou-se a leitura da bibliografia existente nesta questão de primordial importância na elaboração de ambientes virtuais em VRML. No entanto, essa tarefa não foi simples, linear e de todo célere, dado que a informação ia surgindo gradualmente, dispersa ao longo dos textos, e raramente abordava uma quantidade significativa de tópicos relacionados com a temática, visto que, aquando da leitura de uma outra referência bibliográfica, novas questões eram afloradas. Foi essa a razão que motivou o surgimento deste sub-capítulo que tenta “condensar”, num único local, todo esse conhecimento adquirido durante um período bastante longo, facilitando a sua transmissão a potenciais criadores de “mundos” VRML, libertando-os do ónus de efectuar tal trabalho de investigação, bastante moroso para quem não domina a linguagem ou então é “novo” nesta área.

4.2.2 Hierarquização

Já foi mencionado que um ficheiro VRML é constituído por nós, alguns dos quais podem conter outros nós, que por sua vez podem ainda conter outros nós. Esta estrutura interna do ficheiro, sugere que os ficheiros VRML têm, ou podem ter, uma estrutura hierárquica. E este é, precisamente, um dos conceitos mais importantes do VRML.

Mas afinal, o que se quer dizer quando se fala em hierarquia num ficheiro VRML? Sendo um dos principais propósitos de um mundo virtual em VRML a representação de objectos num espaço tridimensional, é a forma como esses objectos estão relacionados, internamente, entre si que definem a hierarquia.

O corpo humano é um excelente exemplo de uma estrutura hierárquica. Quando movemos o braço pelo cotovelo, toda a zona desde o cotovelo até à última falange dos dedos acompanha esse movimento. Quando movemos o pulso, apenas a mão e os dedos seguem esse movimento, o mesmo raciocínio se pode ir aplicando até à última falange de um dedo. Assim, é lógico pensar nos dedos como descendentes hierárquicos da mão, a mão por sua vez é descendente do braço, onde este descende do tronco. Desse modo, pode-se considerar todo o corpo humano como uma enorme estrutura hierárquica. Na figura 4.1 apresenta-se, através de um diagrama em árvore, uma possível hierarquização do braço humano:

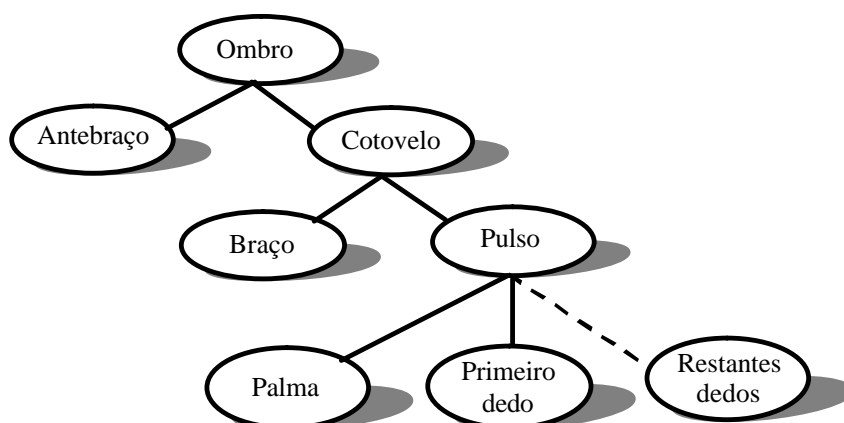


Figura 4.1 Esquema em árvore da estrutura hierárquica do braço humano

As folhas desta árvore indicam cada um dos objectos necessários à representação do corpo humano. Este diagrama mostra como esses objectos

deveriam estar definidos/relacionados na estrutura interna do ficheiro VRML, no caso de querermos representar o corpo humano num espaço virtual.

Mas, porquê tudo isto? Não se conseguiria representar o corpo humano exactamente com os mesmos objectos, mas estando todos ao mesmo nível na hierarquia? Claro que sim. De facto, e como se irá mostrar de seguida, esse não é, de todo, o procedimento mais adequado. Na secção 4.1.4, escreveu-se que um dos testes realizados pelo *browser* durante a representação de um mundo virtual, era o teste de visualização (*culling*), ou seja, verificar quais os objectos que estão completamente fora do volume de visão²⁸, para assim poder determinar quais deles estão visíveis no momento. O teste consiste em intersectar o volume de visão com a *bounding box* representativa do objecto e se esses dois volumes não se intersectarem, significa que o objecto não está visível, logo pode ser ignorado durante o *rendering* da cena (Cosmo Worlds 2.0 User's Guide, 1998). Como se pode imaginar, este processo de intersecção de volumes é um cálculo pesado em termos de processamento interno, pois envolve cálculos matemáticos bastante apurados. Se o movimento efectuado for contínuo, por exemplo andar numa rua de uma cidade, esses testes estão a ser feitos continuamente. Facilmente se verifica que, quantos mais objectos existirem mais testes têm de ser feitos. No exemplo do corpo humano, que em termos de dimensão espacial até nem ocupa uma grande área, é possível ter umas largas dezenas de objectos. Numa cena mais complexa e que espacialmente ocupe uma área consideravelmente superior, pode-se atingir sem dificuldade alguns milhares de objectos. Assim, cada movimento levaria à realização dos testes de visualização a todos eles, ou seja, milhares de testes de intersecção de volumes teriam de ser realizados. Como num movimento contínuo, o volume de visão está constantemente a variar, todos esses cálculos teriam que ser repetidos na sua totalidade várias vezes por segundo. É fácil verificar que este processo acarreta repercussões óbvias ao desempenho de um mundo virtual.

É através de uma correcta estrutura hierárquica que esses testes poderão ser reduzidos significativamente. Como exemplo, imagine-se um carro numa rua. Para representar esse veículo seriam necessários umas largas dezenas de objectos que constituiriam os diversos componentes do mesmo. Isto implicaria que só para o carro teriam de estar constantemente a ser realizados dezenas de

²⁸ Volume de visualização do avatar

testes de visualização. Mas se todos esses componentes estivessem definidos/agrupados numa estrutura hierárquica, onde no topo teríamos o grupo “carro”, seria a *bounding box* do grupo “carro”, que engloba todas os nós dos descendentes hierárquicos, a ser usada nos testes de visualização. Deste modo verifica-se que, num modelo bem definido hierarquicamente apenas um teste é feito, e que num modelo não definido hierarquicamente, dezenas de testes que teriam de ser efectuados exactamente para o mesmo objecto e para o mesmo objectivo. Se esta situação se verifica num exemplo muito simples de um veículo, imagine-se na cidade inteira.

É esse o motivo pelo qual foi referido no início desta secção que a hierarquização é um dos conceitos mais importantes no VRML, pois como se demonstrou, ao reduzir significativamente os testes de *culling* a realizar os ganhos em termos de desempenho são substanciais.

Para além do desempenho outros benefícios podem advir de um modelo bem definido hierarquicamente, nomeadamente no caso de se pretender disponibilizar animações no mundo virtual. Voltando ao exemplo do corpo humano, sem uma estrutura hierárquica qualquer movimento, por exemplo da mão, faria com que estivesse ao cargo do criador do ficheiro VRML o cuidado de definir as transformações necessárias, por forma a que os dedos acompanhem o movimento que a mão está a realizar. Pelo contrário, num modelo hierárquico, onde os dedos sejam descendentes da mão, qualquer movimento aplicado à mão, fará com que todos os seus descendentes, neste caso os dedos, acompanhem o movimento que está a ser efectuado, sem qualquer intervenção adicional.

É uma boa política idealizar logo à partida uma estrutura adequada e definir quais as animações que se pretendem introduzir, para mais tarde não ter de efectuar alterações no ficheiro.

Para finalizar, uma correcta estrutura hierárquica ainda facilita a divisão do ficheiro principal em ficheiros mais pequenos, os quais poderão ser usados no ficheiro principal através da técnica de *inlining* (ver secção 4.2.5). Basta fazer essa divisão pelos níveis hierárquicos, principalmente os superiores, da estrutura.

Não há fórmulas mágicas para estabelecer as hierarquias, este é um processo que está a cargo do(s) criador(es), e que pode ir sofrendo alterações ao longo do

processo, de acordo com as necessidades do momento. No entanto, algumas sugestões poderão ser referidas:

- ✓ Devem ser construídos objectos agrupando outros objectos mais pequenos, onde estes por sua vez agrupam objectos ainda mais pequenos;
- ✓ Devem-se agrupar os objectos que estejam próximos uns dos outros no mundo virtual (Cosmo Worlds 2.0 User's Guide, 1998). Se agruparmos objectos dispostos em zonas opostas na cena, o *browser* tem de efectuar os testes de visibilidade e representá-los sempre a ambos, mesmo quando apenas um está visível no momento. O *browser* apenas pode "ignorar/esconder" os objectos quando a *bounding box* do grupo está completamente fora do volume de visão, o que pode nunca acontecer com um agrupamento incorrecto;
- ✓ Definir inicialmente as animações a introduzir no mundo virtual e para cada uma delas verificar se há objectos que possam ter movimentos dependentes de outros. Por exemplo, o movimento de um braço tem de originar o acompanhamento desse movimento por parte de todos os restantes componentes que compõem o braço: mãos, dedos, antebraço, etc. É essa análise das dependências que reflecte precisamente a estrutura hierárquica a estabelecer;
- ✓ Dividir objectos muito grandes em objectos com dimensões mais reduzidas. Um objecto de representação complexa e de dimensões consideráveis estará, muito provavelmente, visível durante uma grande parte do tempo da movimentação pelo mundo virtual. Isso implica a sua total representação, mesmo quando a sua área visível possa ser muito pequena. Se o objecto for bem dividido, todos os cálculos necessários ao *rendering* são apenas efectuados às zonas visíveis, ignorando as restantes.

4.2.3 Colisões

O motivo pelo qual existem portas nas habitações é a impossibilidade do ser humano atravessar sólidos, neste caso paredes. Se o tentar é impedido de o fazer, devido à colisão que é gerada entre a pessoa e o sólido, seja um banco de jardim, um poste, etc. Uma das principais características do VRML (a partir da sua

versão 2.0), numa tentativa de aproximação à realidade, consiste na realização de testes de colisão, para aquilatar da viabilidade do deslocamento que se pretende efectuar.

Por defeito todos os objectos criados num mundo VRML são sólidos, e por esse motivo impenetráveis. O *browser* efectua testes de colisão a todos os objectos e impede o utilizador de os “atravessar”. No entanto, esses testes requerem uma grande capacidade de processamento, como referido na secção 4.1.2. Como um ficheiro VRML pode ter umas dezenas ou centenas de objectos, totalizando alguns milhares de polígonos, facilmente se constata o dispêndio de recursos apenas para efectuar estes testes de colisão.

Existe no VRML um nó (*Collision*) que permite minorar o dispêndio de recursos ao conseguir não só reduzir os testes de colisão a efectuar, como facilitar a essa detecção.

Normalmente na construção de mundos virtuais é tentada uma simulação da realidade, e é com base nessa realidade que se pode extrair algo que permita melhorar o desempenho de um mundo VRML. O ser humano por si só não consegue voar, e por esse motivo é impossível que consiga colidir com algo cuja distância ao solo seja superior à sua altura. Foi já referido que o *browser* efectua testes de colisão a todos os objectos da cena. No entanto, como acontece na vida real, é comum que existam objectos com os quais o utilizador nunca chega a colidir, pelo que é um desperdício de recursos efectuar testes de colisão a esses objectos. Por exemplo, num movimento normal o utilizador nunca colide com o telhado de uma casa, pelo que seria útil poder desactivar o teste de colisão com essa superfície. Como é normal que existam várias superfícies nesta situação a desactivação dos testes de colisão a todas elas, resultará numa redução considerável do número de testes de colisão a efectuar pelo *browser*, aumentando assim os recursos disponíveis e consequentemente o desempenho geral do mundo. Na realidade, a desactivação dos testes de colisão é possível através da colocação do nó *Collision* no(s) objecto(s) em causa, e alterar o campo *collide* para FALSE.

O nó *Collision* também permite facilitar a detecção de colisões por parte do *browser*, na medida em que pode acelerar e simplificar essa mesma detecção. O

nó dispõe de um outro campo, *proxy*, onde se pode definir uma outra geometria, que irá ser usada no teste de colisão em substituição da geometria original do objecto em si. Imagine-se a modelação de um carro a ser colocado na avenida de uma cidade. Devido ao elevado grau de perfeição desejado na visualização desse carro, a construção do mesmo inclui uma largas centenas de polígonos. No entanto, efectuar os testes de colisão a todos esses polígonos torna-se ineficiente e na realidade desnecessário. O grau de perfeição é desejável para a visualização, mas não o é com certeza para as colisões. Se, por exemplo, se definisse uma caixa (paralelepípedo), que se adequasse à forma da parte inferior do carro (motor, bagageira, rodas, luzes, etc.) e outra para a parte superior (tejadilho, vidros, etc.), seria suficiente para resolver esta questão. Desta forma, o *browser* apenas teria de efectuar os testes de colisão a uma simples caixa com 12 polígonos²⁹ em vez das iniciais centenas de polígonos, sem prejuízo algum de realismo. Deste modo, é simplificado o cálculo dos testes e acelerada a sua execução. Este processo deve ser usado em formas mais complexas e compactas pois permite reduzir consideravelmente o processamento necessário à detecção de colisões.

Nota: Há *browsers* que tornam este processo ainda mais rápido, quando são usadas como *proxy*, primitivas simples: caixa, cilindro, esfera, cone (Hartman e Wernecke, 1996).

Uma referência final para outra das funcionalidades do nó *Collision*, o gerar eventos aquando da ocorrência de uma colisão com a superfície que lhe está associada (Marrin e Campbell, 1997). Por exemplo, a emissão de um registo sonoro após a colisão do utilizador com uma porta, simulando o embate.

4.2.4 Instanciação (Clonagem)

Imagine-se um pedido de um cliente para modelar o interior de uma casa. Na altura de modelar a sala de jantar, é construída uma cadeira e uma mesa de formato rectangular para 4 pessoas, o que implica a necessidade de existirem mais 3 cadeiras iguais. Num raciocínio análogo à quase totalidade das aplicações

²⁹ Uma caixa: 6 faces + 2 polígonos (triângulos) por face = 12 polígonos

informáticas que regularmente se utilizam, sempre que pretende copiar algo, a técnica do copiar/colar é por norma a mais usada, pois é fácil de usar, prática e poupa trabalho ao autor. Em VRML tal também é possível. No entanto, na maioria das situações, este procedimento compromete o desempenho. Ao efectuar uma cópia da cadeira, todos os nós usados na sua representação, quer geométrica, quer visualmente (cor ou textura), são copiados, obrigando a uma reconstrução total das cadeiras por parte do *browser*. Este procedimento de reconstrução dos objectos que se repetem na cena, devido à repetição de código, é bastante ineficiente, em virtude de se estarem a criar novos objectos em vez de reutilizar os já existentes. A solução deste problema passa então por uma reutilização dos objectos já existentes, usando uma técnica denominada por instanciação (também conhecida por clonagem). Esta técnica difere da cópia, na medida em que há uma partilha de todas as características (geometria e aparência) do original, e dessa forma a informação não precisa de estar duplicada no ficheiro. O seu funcionamento é análogo a um ponteiro numa linguagem de programação (Marrin e Campbell, 1997): aquando da leitura da informação do ficheiro, todos os nós representativos dos diversos objectos são colocados em memória, e posteriormente representados pelo *browser*. Sempre que se dá uma reutilização de um objecto por instanciação, o que acontece não é a ocupação de outra zona de memória com esse objecto, mas sim uma referência à zona de memória onde está localizado o original. Esta situação implica que qualquer alteração das características, geométricas ou visuais, a qualquer um dos objectos, se reflecta em todos os restantes.

Há no entanto uma aparente alteração geométrica que pode ser feita a clones sem afectar os restantes. As cadeiras, clonadas, em falta da sala de jantar são colocadas em redor da mesa por intermédio de translações e rotações, usando para esse efeito o nó *Transform*. Esse nó dispõe de uma outra transformação geométrica que pode ser aplicada aos seus descendentes, designadamente a mudança de escala. Assim, e sem alterar qualquer das características do objecto, é possível modificar as suas dimensões, sem qualquer interferência nos outros clones.

O rol de vantagens associadas à utilização desta técnica é vasto, repercutindo-se no desempenho global do mundo:

- ✓ Redução do tamanho final do ficheiro, pois não há duplicação de informação no mesmo;
- ✓ Redução do tempo de *download*, quer inicialmente para a memória, quer depois por parte do *browser*;
- ✓ Libertação de recursos a nível de memória, devido à referência da mesma posição de memória em vez da ocupação de uma nova posição;
- ✓ Redução no tempo de *rendering*, na medida em que há um aumento da eficiência por parte do *browser*, devido a repetir as mesmas operações (Marrin e Campbell, 1997);
- ✓ A haver necessidade de efectuar alterações nos objectos (geometria ou aparência), elas podem ser efectuadas apenas a um deles, pois os restantes partilham as mesmas características;
- ✓ Facilita o trabalho ao autor, pois o processo é de implementação muito simples;

Para além da clonagem de objectos, é possível ir ainda um pouco mais além. De facto, os objectos não são mais do que nós em VRML, pelo que mantendo a mesma linha de raciocínio concluí-se que qualquer nó pode ser clonado. Esta possibilidade é útil numa cena de objectos distintos, mas que partilhem as mesmas características visuais (cor e/ou textura). Em vez de ter que definir as mesmas propriedades visuais em cada objecto, poderá fazer-se em apenas um deles e clonar, com os restantes objectos, apenas o nó que define essas propriedades (*Appearance*).

4.2.5 Inlining

É comum que durante o desenvolvimento de um projecto seja necessário usar objectos já modelados em projectos anteriores, seja uma casa, uma cadeira ou um carro, por exemplo. Seria um desperdício de tempo e energia modelar novamente esses objectos, pelo que a sua reutilização se afigura aconselhável. Procurar os projectos antigos e copiar código de um ficheiro para outro é possível, mas não é a solução mais eficiente. Uma solução mais lógica será definir o objecto num ficheiro separado e utilizá-lo sempre que necessário. Esta solução permite que cada vez que seja necessário usar o objecto, baste indicar no ficheiro

onde se está a trabalhar, uma referência (analogamente a um ponteiro numa linguagem de programação) para o ficheiro onde esse objecto está definido. Desse modo pode ser criada uma espécie de base de dados de objectos de uso frequente, onde se pode também incluir modelos desenvolvidos por outros criadores.

Nota: Ao utilizar modelos desenvolvidos por outro criador, pode acontecer que a escala usada nesse modelo seja diferente da que se está a utilizar. Uma solução prática e eficaz de resolução desse problema é a colocação do nó que permite realizar transformações geométricas (*Transform*) como ascendente do modelo a utilizar e aplicar uma mudança de escala, para adequar as dimensões do modelo importado ao mundo em construção.

Esta técnica, que permite separar do ficheiro principal o conteúdo de algumas partes da cena denomina-se por *inlining*, e o nó que a implementa chama-se *inline*. O seu funcionamento é muito simples: localiza o ficheiro referenciado no campo *url* do nó e coloca o seu conteúdo numa dada posição, indicada pelo autor, da cena que está a ser representada.

Nota: O uso desta técnica apenas faz uso da referência ao ficheiro, não importa o seu conteúdo para o ficheiro principal.

Este é um conceito muito simples, mas o seu uso tem diversas vantagens:

- ✓ Modularidade. Durante o desenvolvimento de mundos grandes e complexos, é por vezes difícil operar num só ficheiro, seja pela capacidade do equipamento, seja pela dificuldade em manusear uma cena já demasiadamente complexa. A divisão do ficheiro principal em pequenos módulos surge como uma alternativa bastante viável. A facilidade em operar nesses pequenos módulos (ficheiros) torna-se evidente. Inclusive, no caso de serem necessárias alterações, essas alterações apenas serão efectuadas no módulo correspondente, sem o perigo de afectar inadvertidamente outros componentes do mundo virtual. A sugestão para uma adequada divisão do ficheiro principal, centra-se no uso da estrutura

hierárquica (secção 4.2.2) definida para este mundo, pois ela já reflecte uma correcta divisão da cena em módulos devidamente agrupados.

- ✓ Rapidez de download do ficheiro principal. O ficheiro principal, que é o responsável pela interligação de todos os restantes, tem o seu tamanho reduzido consideravelmente, proporcionando assim um *download* mais rápido e eficiente, não só dele próprio, como dos restantes ficheiros que compõem a totalidade da cena. Além disso, a redução de tamanho do ficheiro principal e consequente diminuição do tempo de *download*, origina uma diminuição do tempo de espera do utilizador até ter algum *feedback* visual, do mundo virtual.
- ✓ Rapidez de utilização. Assim que o ficheiro principal esteja totalmente em memória, e em alguns *browsers*, o utilizador pode começar a explorar a cena, enquanto os ficheiros definidos como *inline*, continuam a ser carregados (Cosmo Worlds 2.0 Tutorial, 1998). O que não sucedia sem esta divisão, pois só quando a totalidade da cena, com a complexidade e dimensões associadas, estivesse em memória, é que se poderia começar a explorar. Por este motivo e para uma mais rápida interacção do utilizador com o mundo virtual, sugere-se a inclusão no ficheiro principal dos objectos que constituem a vista inicial do utilizador aquando da entrada no mundo.
- ✓ Repetição da reutilização. Nada impede que a técnica de *inlining* seja utilizada mais do que uma vez num mesmo ficheiro. Por exemplo, considere-se que se estava a projectar a modelação de um bairro de uma qualquer cidade e que foi criado um poste de iluminação num ficheiro em separado. Como as ruas do bairro contêm diversos postes de iluminação, faz sentido utilizar esta técnica repetidamente, colocando a referência para o ficheiro que contém o poste o número de vezes necessário e nas diversas posições. É com este procedimento que se consegue colocar todos os postes na cena usando apenas o conteúdo de um pequeno ficheiro. Deste processo duas vantagens se evidenciam claramente:
 - i. Não é necessário voltar a modelar objectos que se repetem na cena, bastando referenciar o objecto já existente, indicando apenas a sua nova posição;

- ii. O *download* do ficheiro só é feito na altura de representar o objecto (poste no exemplo) pela primeira vez, já que posteriormente o ficheiro já se encontra em memória e o acesso a ele é consideravelmente mais rápido. Torna-se evidente a economia de recursos e conseqüente aumento do desempenho.
- ✓ Além do campo *url*, o nó *inline* possui mais dois campos, *bboxSize* e *bboxCenter*, que quando usados convenientemente potenciam ganhos na optimização da cena. Basicamente estes dois campos permitem indicar ao *browser* as dimensões e posição da *bounding box* do conteúdo do ficheiro *inline*. É precisamente essa *bounding box* que é usada nos cálculos de visualização aquando da entrada inicial na cena. Caso a *bounding box* esteja fora do volume de visão, não é necessário fazer o *download*, no imediato, do ficheiro referenciado, libertando algum processamento para outras tarefas (Hartman e Wernecke, 1996).
- ✓ Combinar o uso de *inlines* com LOD's (secção 4.2.11) ameniza o impacto inicial do mundo quando está a ser carregado (Vacca, 1988). Especialmente quando usado em objectos mais distantes da perspectiva inicial do utilizador, este processo é vantajoso no sentido da garantia de um *download* da cena mais rápido e uma navegação mais pronta e eficaz, uma vez que há uma economia inicial de polígonos devido ao uso dos LOD's, implicando um menor consumo de memória por parte do *browser*.

Uma referência final, para indicar que toda a informação não geométrica da cena VRML também deve ser incluída no ficheiro principal, nomeadamente dados do ficheiro e autor (*WorldInfo*), propriedades do avatar (*NavigationInfo*), *backgrounds* e *viewpoints*. Alguma desta informação é necessária para a movimentação e visualização, especialmente no caso do utilizador desejar iniciar a exploração do mundo virtual antes de terminar o *download* de todos os ficheiros definidos como *inline*.

4.2.6 Iluminação

Na idealização e construção de um ambiente virtual, alguns dos efeitos mais realistas conseguidos no aspecto final da cena, são obtidos através de um eficaz

uso da iluminação, simulando algumas das fontes emissoras de luz existentes na vida real, como sejam o sol ou um poste de iluminação de uma avenida. Para esse efeito o VRML disponibiliza três formas diferentes de emissão de luz na cena, cada uma delas com as suas características próprias, nomeadamente, *DirectionalLight*, *PointLight* e *SpotLight*.

A utilização de um número elevado de focos de luz numa cena não é aconselhável, uma vez que implica uma redução significativa de desempenho do mundo. Isto acontece devido ao facto dos cálculos de iluminação, que são bastante pesados (secção 4.1.6), serem efectuados em todos os vértices que eventualmente possam ser afectados pelo(s) foco(s) de iluminação. Por cada novo foco inserido são necessários novos cálculos, de tal modo que pode afectar significativamente o desempenho. Portanto, é importante racionalizar o número de focos de luz espalhados pela cena, bem como o tipo de emissão usado. Mais uma vez, e também neste ponto, deve ser estabelecido um compromisso entre realismo e desempenho.

DirectionalLight é um nó que permite simular a emissão de luz a partir do infinito numa determinada direcção, assemelhando-se ao sol. Em virtude do paralelismo dos “raios luminosos”, pelo foco estar supostamente no infinito, os cálculos envolvidos na iluminação são simplificados, em virtude do vector direccionado ao foco de luz, usado nesses cálculos, ser sempre o mesmo em todos os vértices afectados pela iluminação. Este tipo de iluminação apenas afecta os objectos que lhe estejam associados no mesmo grupo³⁰ (Marrin e Campbell, 1997). Esta é uma medida que visa permitir ao criador um controlo mais efectivo da iluminação sobre os objectos, dado que o VRML não efectua cálculos de sombras, e poderiam estar a ser iluminados objectos sem que tal fosse desejado. Por outro lado, o próprio *rendering* é mais eficiente, pois se diversos focos apenas afectarem diferentes partes da cena, menos entram em simultâneo nos cálculos de iluminação, durante o deslocamento pela cena. Como há, com certeza, objectos da cena que não estão visíveis no momento, os cálculos de iluminação do foco que afecta esses objectos não são realizados. Para concluir, um foco de iluminação do tipo *DirectionalLight*, não requer tanta capacidade de processamento como os restantes e consegue um bom efeito geral de iluminação com um menor impacto no desempenho.

³⁰ Pode haver browsers que não cumpram esta norma da especificação

Um *PointLight* é um nó que permite colocar, numa dada posição da cena, um foco que emite luz em todas as direcções a partir da sua posição, simulando uma lâmpada. Neste caso, para cada vértice onde se efectuem cálculos de iluminação, tem de ser recalculado o vector direccionado ao foco de luz, para assim conseguir proporcionar o efeito luminoso desejado. Este facto resulta num processamento mais pesado que o *DirectionalLight*, mas ainda assim proporciona um *rendering* razoavelmente rápido.

Finalmente, o *SpotLight*, que com uma localização específica na cena, emite luz numa dada direcção e limitada por um cone finito. A intensidade da iluminação diminui exponencialmente até às extremidades do cone, cujo ângulo de abertura também é controlado. Este tipo de iluminação permite simular, por exemplo, uma lanterna ou um candeeiro. No entanto, devido aos cálculos que é necessário realizar, implica um processamento mais exaustivo e consumidor de recursos, com incidência directa no desempenho. Assim, o seu uso deve ser motivo alguma ponderação e cuidado.

Durante esta breve descrição dos 3 tipos de focos de luz existentes no VRML, deu-se particular ênfase à especificidade de alguns dos cálculos envolvidos em cada uma deles. Este facto foi propositado e serve para realçar a importância que um cauteloso, mas eficaz, uso da iluminação tem no desempenho e no realismo do mundo. É um facto que os efeitos mais realistas são conseguidos por intermédio de *PointLights* e *SpotLights*, mas também é verdade que são estes dois os que mais contribuem para uma degradação mais acelerada do desempenho, especialmente o *SpotLight*.

Existe uma forma de se conseguir simular efeitos de iluminação e até sombras, sem o peso computacional associado a esses cálculos de iluminação. Em superfícies com texturas pode ser aplicada directamente na textura, com uma qualquer aplicação gráfica com potencialidades para tal, a simulação do efeito que a incidência da luz teria na superfície. Ao visualizar a textura na superfície seria transmitida a sensação que ela está sob influência de um qualquer foco de luz, quando na realidade não está. Este método pode ser ainda mais proveitoso, especialmente no caso de ser usado para simular um *SpotLight*.

Uma referência ainda para o facto de, especialmente quando há iluminação presente, o desempenho do mundo poder aumentar significativamente com a redução das dimensões da janela de visualização (Marrin e Campbell, 1997). Como o número de pixels a representar baixa consideravelmente, também se reduzem os cálculos a efectuar e desta forma há uma maior disponibilidade de todos os agentes responsáveis pela representação da cena.

Em jeito de recomendação final relativamente ao uso de iluminação: usar *DirectionalLight* sempre que possível, *PointLight* caso contrário e *SpotLight* apenas quando estritamente necessário.

4.2.7 Número de Polígonos

Um pouco antes do início deste milénio, em pleno fervilhar do constante desenvolvimento tecnológico de equipamentos informáticos, uma nova vaga colocou em autêntico frenesim alguns dos fabricantes desses equipamentos. Motivados essencialmente pelo emergente e cada vez mais importante mercado dos jogos de vídeo, onde o desempenho gráfico, especialmente a nível tridimensional, era factor preponderante, assistiu-se, e continua-se a assistir, num curtíssimo espaço de tempo a um notável desenvolvimento das capacidades dos *chips* gráficos. Nas especificações desses *chips* discute-se: memória interna (por vezes até superior à do próprio computador), AGP 8x, *vertex e pixel shader*, *Transform & Lighting*, etc. Há, no entanto, um item que também é mencionado nessas especificações, o número de triângulos “renderizáveis” por segundo. Por exemplo, nas placas GeForce de 1ª geração, esse valor rondava os 15 milhões de triângulos, nas gerações actuais já ultrapassa os 300 milhões de triângulos.

Serve esta introdução para enfatizar que o número de polígonos numa cena VRML é um dos factores críticos de sucesso, e neste caso sucesso é sinónimo de desempenho (Cosmo Worlds 2.0 User's Guide, 1998). Assim sendo, deve ser preocupação do autor, logo desde o início, tentar manter o número de polígonos o mais baixo possível, sem comprometer o realismo pretendido. Esse equilíbrio pode ser muitas vezes difícil, mas é sempre de grande importância. Os objectos que irão compor o mundo devem ser realistas, mas utilizar na sua construção o menor número de polígonos possível. Apesar de nas próximas secções irem ser

discutidas algumas técnicas que indirectamente permitem reduzir o número de polígonos, conseguindo manter um realismo aproximado, nomeadamente o uso de texturas, LOD's, *crease angle* e *billboards*, mais algumas sugestões poderão ser feitas:

- ✓ Durante o processo de criação do mundo, é normal recorrer a outras aplicações que possam auxiliar na modelação de alguns objectos. Antes de exportar o objecto, do formato nativo da aplicação em causa, para VRML, deve-se aplicar algum comando disponível nessa aplicação que permita efectuar optimizações nesse ponto. No exemplo concreto do 3DStudio, uma aplicação bem conhecida na área da Computação Gráfica, e que foi usada na elaboração deste trabalho, existe um comando, *Optimize* (Santos e Barata, 1998) que permite reduzir o número de polígonos do objecto mantendo uma qualidade igual ou aproximada na visualização;
- ✓ Eliminar os polígonos que não sejam visíveis na cena, eventualmente por estarem encobertos por outros objectos. Deve-se, no entanto, evitar o uso deste processo quando estiverem em causa primitivas geométricas simples (esfera, caixa, cilindro e cone), pois os *browsers* já dispõem de algoritmos optimizados para a representação destas primitivas (secção 4.2.14).
- ✓ Por vezes durante o processo de modelação de um objecto, existe a necessidade apenas de uma face, para colocar uma textura por exemplo. Por uma questão de comodismo, é por vezes é comum usar uma caixa com uma profundidade quase nula. Este procedimento é bastante ineficaz não só porque existe no VRML um nó que permite criar faces isoladas (*IndexedFaceSet*), mas também porque ao usarmos uma caixa para simular uma face estamos a desperdiçar polígonos, que não estão a ter qualquer tipo de utilidade;

É comum encontrar em bibliografia relativa a este tópico, sugestões para um número máximo de polígonos visíveis em simultâneo numa cena. Com o crescente avanço tecnológico, especialmente dos *chips* gráficos, esses valores estão completamente desajustados. Estabelecer um limite para esse número máximo é uma tarefa muito complicada, pois está dependente do equipamento em utilização, mas também e especialmente do tipo de optimizações que foram efectuadas no ficheiro VRML. Por exemplo, durante a nossa investigação foram testados num mesmo equipamento, diversos ficheiros/trabalhos em VRML

existentes na Web. Comparativamente com o nosso trabalho, alguns desses ficheiros totalizavam um número de polígonos inferior e mesmo assim o tamanho do ficheiro era cerca de 20 vezes superior e o desempenho consideravelmente inferior, com uma fluidez de movimentos muito insuficiente para um efeito realista.

4.2.8 Crease Angle

A redução de polígonos num modelo virtual VRML, não tem necessariamente que resultar numa redução da qualidade. Com uma adequada iluminação e sombreamento, efeitos similares podem ser criados sem o incremento da complexidade no modelo. Já se referiu que todas as entidades geométricas são construídas por intermédio de polígonos, mesmo aquelas que incluem superfícies curvas. Nestes casos, para aumentar a perfeição dessas superfícies, os polígonos devem ser mais pequenos e, por consequência, em maior número. Isto não é muito aconselhável, pois o desempenho tem como um dos factores inibitórios, precisamente o número de polígonos. Este é um dos maiores dilemas na modelação de superfícies curvas, o atingir um ponto de equilíbrio entre realismo e desempenho. Mas como foi dito no início desta secção, a redução de polígonos no modelo não tem necessariamente que resultar numa redução da qualidade. *Crease angle* é o ângulo de incidência entre dois polígonos adjacentes numa superfície, que for se usado de forma adequada pode melhorar o aspecto visual de uma superfície curva sem o incremento de novos polígonos. Dito de outra forma, cabe ao criador definir o ângulo a partir do qual se torna perceptível o “vinco” entre duas faces adjacentes. Sem entrar em grandes pormenores, pode-se resumir este processo ao modo como as normais usadas nos cálculos de iluminação são geradas (Marrin e Campbell, 1997). Por defeito, cada vértice tem associadas as normais de cada uma das faces que lhe são comuns, o que origina, quando dos cálculos de iluminação a cada uma dessas faces, o efeito de “vinco” entre as mesmas. Se o ângulo entre duas faces for inferior ao definido no campo *creaseAngle*³¹, é gerada apenas uma normal no vértice, proporcionando assim nos cálculos de iluminação uma suave transição da cor entre as faces, causando um efeito de alisamento na superfície (Ames et al., 1997). Nas imagens

³¹ Os nós que dispõem deste campo são: *IndexedFaceSet*, *ElevationGrid* e *Extrusion*

seguintes pode-se verificar este efeito, o objecto é o mesmo, só variou o *creaseAngle* adoptado entre ambas.

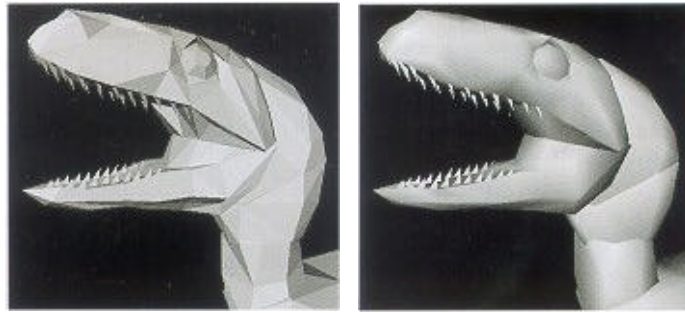


Figura 4.2 Modelo facetado (esq.). Mesmo modelo, apenas com alteração do *creaseAngle* (dir.)

4.2.9 Texturas

A aplicação de texturas em superfícies é uma das medidas mais eficazes na representação realista, sendo usada desde os primórdios da computação gráfica. Naturalmente que, tendo o VRML sido criado com o objectivo de tentar representar tridimensionalmente ambientes virtuais que consigam proporcionar ao utilizador sensações tão realistas quanto possível, a utilização de texturas no VRML é vivamente recomendada. Ao contrário do que acontece, por exemplo, com a composição de imagens fotorealistas, onde somente se aplica a textura e visualiza o resultado final, no VRML, pelas suas características tridimensionais e interactivas, onde o utilizador se move livremente pelo mundo, questões de desempenho têm de ser fortemente consideradas e analisadas. As imagens usadas como texturas, na realidade, são grandes consumidoras de recursos, seja pelo tempo de *download*, a memória consumida ou ainda a velocidade de *rendering* por parte do *browser*. No entanto, o seu uso no VRML pode servir como factor de optimização, na medida em que possibilita a economia de centenas ou milhares de polígonos.

Imagine-se uma superfície complexa e bastante detalhada, cuja modelação só seria possível com recurso a um número considerável de polígonos. Se fosse possível representar essa mesma superfície com tanto ou mais realismo, com uma redução do número de polígonos, seria óptimo. É precisamente neste ponto que a utilização de texturas é mais eficaz no VRML, pois com um certo cuidado no tratamento das imagens, consegue-se mais realismo, melhor desempenho e uma redução significativa de polígonos. Com as potencialidades actuais das

aplicações gráficas (PhotoShop, 3DStudio, PaintShopPro, por exemplo), é possível criar texturas com sofisticados efeitos de sombreamento e iluminação que simulam com um elevado grau de perfeição, efeitos tridimensionais. Se essas texturas forem aplicadas numa simples face rectangular, é proporcionado ao utilizador uma visão mais realista do objecto e com um custo de dois polígonos da face onde foi aplicada textura. Além do mais, com a mesma textura e uma simples alteração de algumas das propriedades de alguns nós do VRML³², conseguem-se obter efeitos ligeiramente diferentes na visualização dessa mesma textura na cena.

Esta descrição optimista da utilização de texturas é correcta, mas convém lembrar que as texturas também podem ser grandes consumidoras de recursos se não usadas convenientemente. Eis algumas sugestões na utilização de texturas:

- ✓ Texturas muito grandes e de alta resolução/qualidade têm dimensões físicas (espaço que ocupam no suporte de armazenamento) muito elevadas, repercutindo-se num *download* lento, bem como na sua representação por parte do *browser*. Por esse motivo, essas texturas devem ser tratadas com uma aplicação gráfica adequada, que permita, se não reduzir o tamanho, pelo menos a qualidade. É frequente obter economias de 30% a 40% nas dimensões de uma imagem ao reduzir a qualidade da mesma, sem que essa redução seja muito notória na qualidade visual;
- ✓ Sempre que possível, reutilizar uma textura. Quando é feito o seu *download* para representação, ela é colocada em memória e uma eventual reutilização dessa mesma textura é muito mais célere ao aceder à mesma zona de memória;
- ✓ Usar texturas em escala de cinzento. Ocupam menos espaço e têm um *rendering* mais rápido. Se usadas em parceria com a cor da superfície (*diffuseColor* do nó *Material*), conseguem-se criar efeitos interessantes, pois numa textura em escala de cinzento apenas o preto não deixa transparecer a cor da superfície (Cosmo Worlds 2.0 Tutorial, 1998);
- ✓ O uso de texturas é mais efectivo, quando usado para simular superfícies que seriam grandes consumidoras de recursos, devido ao

³² Com os nós *TextureTransform* e *ImageTexture*

elevado número de polígonos necessários para representar com realismo, a sua complexidade;

- ✓ O autor não está limitado ao mapeamento realizado, por defeito, das texturas pelas superfícies. Existem dois nós (*TextureTransform* e *ImageTexture*) que permitem dispor a textura pela superfície, com grande flexibilidade para o criador;
- ✓ Usar, caso se justifique, transparências nas texturas. Dada a imutabilidade do formato geométrico das texturas (rectangular), quando se pretende simular algo que não ocupe toda a área da textura, pode e deve-se colocar uma transparência na área não relevante. Por exemplo os cantos superiores de uma imagem que simule uma árvore;
- ✓ Por questões de desempenho no processamento interno do equipamento, as dimensões de uma textura devem ser de preferência potências de 2 (Cosmo Worlds 2.0 User's Guide, 1998);
- ✓ Dada a relevância que as dimensões físicas de uma imagem têm no desempenho, importa alertar para o facto do formato das imagens ser, por vezes, o responsável pelas dimensões exageradas de uma imagem. Por exemplo, em imagens fotorealistas onde normalmente há grandes variações de cor, deve ser usado o formato JPG. Quando a paleta de cores usada numa imagem for reduzida, deve ser usado o formato GIF. Apesar do GIF também suportar transparências, a sua limitação de cores não é ideal para imagens com alguma qualidade e que tenham transparências. Nesse caso recomenda-se o formato PNG;
- ✓ Quando uma textura representar uma grande área e seguir um padrão repetitivo, deve-se criar uma textura pequena, que por repetição simule na perfeição esse padrão. Desse modo, consegue-se gerar essa grande área texturada com uma imagem muito pequena.

Uma outra forma de enriquecer o mundo sem grandes custos, é através da utilização de texturas animadas. Têm dimensões reduzidas e consomem poucos recursos. Mas esse tema será abordado, com mais ênfase, na secção das animações (4.2.12).

4.2.10 *Bilboards*

Desde a sua génese que o VRML é vocacionado para a representação virtual de entidades tridimensionais. No entanto, o VRML também possibilita, e até encoraja, o uso na cena de entidades bidimensionais, que de alguma forma possam aumentar o realismo e melhorar o desempenho. Para isso, utilizam-se *billboards*, um nó do VRML que tem como função colocar os objectos que lhe estão associados, por intermédio de rotações definidas por um eixo, numa posição sempre frontal perante o utilizador, independentemente da sua localização.

A utilização mais comum de um *billboard* e com resultados mais satisfatórios, verifica-se na aplicação de uma textura numa face simples. Deste modo, é conseguida uma simulação realista de um efeito tridimensional, e com uma considerável economia de polígonos (ver secção anterior, Texturas). Por exemplo, caso fosse necessário incluir na cena uma(s) árvore(s), seriam necessários milhares de polígonos, isto porque não é nada fácil conseguir modelar uma árvore com um aspecto minimamente realista apenas e só com polígonos. Se, por outro lado, fosse utilizada uma textura de uma árvore real, aplicada numa face e colocada na cena virada para o utilizador, teria um efeito bastante realista de uma árvore num espaço tridimensional. Mas o problema dos objectos bidimensionais é que não têm espessura e ao mover-se o utilizador aperceber-se-ia que afinal aquela árvore não passa de uma textura numa face. Agora imagine-se que, cada movimento do utilizador é acompanhado, sem que ele se aperceba, por uma rotação da face que contém essa árvore de modo a que esta esteja sempre de frente para o utilizador. Ao efectuar movimentos circulares em torno da árvore constata-se que ela está sempre visível, proporcionando assim uma sensação de tridimensionalidade. Explicado de uma forma um pouco simplista, é este o comportamento de um *billboard*.

Como facilmente se pode deduzir, a utilização de *billboards* tem uma série de vantagens:

- ✓ Forma económica de acrescentar realismo à cena;
- ✓ Redução significativa de polígonos;

- ✓ Melhora o desempenho, devido a um *rendering* mais rápido e eficaz;
- ✓ Reduz o tamanho do ficheiro.

Da análise da funcionalidade de um *billboard*, concluí-se que a sua utilização é ideal em objectos cuja a geometria tenha uma certa simetria, onde a visualização por diversos ângulos seja bastante idêntica.

Uma referência final para o facto de, devido à utilização de *billboards*, uma das faces do polígono onde está aplicada a textura estar sempre do lado oposto ao utilizador. Assim sendo, é desnecessária a sua representação por parte do *browser*, podendo o autor “desligar” a sua representação, para os testes de *culling* (secção 4.2.18).

4.2.11 LOD

Desde o início desta dissertação que se tem dado uma especial atenção a uma das questões fulcrais relativas ao VRML, a necessidade de encontrar um ponto de equilíbrio no binómio Realismo vs Desempenho, sendo comum que um acréscimo de realismo na cena normalmente implica um degradar do desempenho. Mas nem sempre esta situação se verifica. A nossa acuidade visual só nos permite distinguir certos detalhes e pormenores em formas da vida real, sejam grandes (um edifício) ou pequenas (uma flor), à medida que nos vamos aproximando delas. Este é o princípio básico subjacente à utilização de LOD's (*Level Of Detail*). Basicamente o que se faz é ter diversas representações, com diferentes níveis de detalhe do mesmo objecto, que vão sendo apresentadas de acordo com certos critérios, nomeadamente de distância. Assim, apenas quando da aproximação aos objectos é que vão sendo visualizadas as suas representações mais detalhadas.

O uso desta técnica permite melhorar a eficiência, na medida em que podem ser omitidos certos detalhes visuais não perceptíveis ao utilizador quando situado a determinadas distâncias. Essa omissão de detalhes é conseguida através da redução do número de polígonos necessários para representar o objecto, o que possibilita uma maior fluidez da navegação na cena por parte do *browser*.

Em VRML existem dois tipos de LOD's, o de distância e o de desempenho. No de distância cabe ao autor definir, além das diversas representações do objecto em causa, as distâncias entre as quais é feita a transição entre uma representação e a outra. No de desempenho, o autor apenas cria os diversos níveis de detalhe, não especificando distâncias. A tarefa de definir qual a representação do objecto a apresentar cabe ao *browser* que dinamicamente escolhe a representação mais adequada para não comprometer o desempenho do mundo (Cosmo Worlds 2.0 User's Guide, 1998). Este procedimento comporta algo de vantajoso: é apresentado ao utilizador o mundo mais adequado às características do equipamento que está a utilizar. Mas o uso desta técnica também acarreta uma desvantagem: o consumo de mais recursos, pois o *browser* tem de carregar todas as representações possíveis do objecto, para assim poder escolher qual deve apresentar.

4.2.12 Animações

A inclusão de animações num ambiente tridimensional pode revelar-se como uma das medidas eficaz no combate a algum desinteresse que pode surgir na visualização desses ambientes, proporcionando ao utilizador uma integração com o ambiente virtual muito mais natural. Imagine-se uma cena onde seja visível uma nora num curso de água. Seria com certeza muito mais interessante e realista visualizar a nora em rotação impelida pela água. Este tipo de animações, e outras que serão abordadas mais adiante, quando usadas com destreza, podem elevar o nosso mundo para o patamar qualitativo que por vezes separa um mundo interessante, onde o utilizador vê, gosta e tem vontade de explorar mais vezes, de um outro monótono, que não desperta grande interesse. Claro que é necessário também levar em conta aspectos de desempenho. As animações requerem uma significativa capacidade de processamento do equipamento, capacidade essa que em primeira instância tem que ser dedicada ao essencial, ou seja, à visualização e navegabilidade e só depois a outros aspectos. Este alerta não têm o intuito de desencorajar o uso de animações. Pelo contrário, é nossa convicção que um mundo bem optimizado, talvez com algumas das sugestões deste capítulo, tem capacidade de incorporar animações, sem degradar muito o desempenho. Mas em todo o caso, convém levar em conta alguns aspectos:

- ✓ Usar sensores para iniciar as animações. Voltando ao exemplo da nora em movimento, só há necessidade de a ter em movimento quando estiver visível. Este é um procedimento ideal, quando se têm animações, e deve ser utilizado no caso de se estar a visualizar zonas da cena sem animações e, também, quando da entrada no mundo, altura em que todos os recursos disponíveis são necessários. Existe no VRML um sensor adequado a esta situação (*VisibilitySensor*). Este sensor gera eventos quando uma determinada área, definida pelo autor, entra no campo visual. Este sensor é perfeito para as animações, pois permite “ligar” e “desligar” animações, caso estejam visíveis, ou não;
- ✓ Evitar, na medida do possível, o uso de animações em objectos bastante complexos, pois a capacidade de processamento requerida é directamente proporcional à complexidade do objecto que se está a animar (Marrin e Campbell, 1997);
- ✓ Ter cuidado quando se está a animar apenas parte de um objecto. Imagine-se que se pretende animar uma porta, que é parte integrante da forma geométrica de uma habitação. Apesar da animação apenas ser visível na porta, todos os pontos (vértices) pertencentes ao objecto fazem parte da animação (Cosmo Worlds 2.0 User's Guide, 1998). Os cálculos de interpolação são efectuados em todos os vértices do objecto, quer se movam ou não. Como facilmente se constata, esta situação é bastante ineficaz, pois requer um processamento muito mais intensivo ao ter de interpolar desnecessariamente, todos os vértices do objecto, quando na realidade apenas os vértices pertencentes à porta, necessitavam de ser animados. Para solucionar esta situação, basta separar a porta do objecto *habitação*, para outro independente, e animar apenas esse novo objecto agora criado;
- ✓ As animações são geradas por intermédio de fórmulas matemáticas, que determinam as posições intermédias dos objectos que estão a ser animados. A escolha dessas fórmulas, denominadas por fórmulas de interpolação, deve ser a mais adequada para o comportamento cinético desejado para o objecto (Hartman e Wernecke, 1996). No entanto há fórmulas mais pesadas computacionalmente que outras, apesar de gerarem comportamentos similares. Por esse motivo devem ser estas últimas as escolhidas. No *Cosmo Worlds*, por exemplo, estão

disponíveis 5 tipos de interpolação: *Constant*, *Linear*, *Hermite*, *Split* e *Mixed*.

Pode-se considerar como animação, qualquer tipo de movimento existente na cena, sem intervenção do utilizador. Desse modo não estamos apenas restringidos à animação de formas geométricas. Existem mais dois tipos de animação, simples de usar e de uma eficácia comprovada, que até nem provocam grande degradação de desempenho:

- i. É frequente encontrar em mundos virtuais, talvez precavendo alguma inexperiência do utilizador na navegação ou porque se deseja “forçar” o utilizador a visitar alguns locais, disponibilizar visitas guiadas, em que o utilizador é “transportado” automaticamente numa visita pela cena, sem que tenha poder de intervenção. Esta é uma forma muito eficaz de dar a conhecer o mundo em toda a sua plenitude ao visitante virtual e não envolve custos muito elevados, pois é conseguida por intermédio de uma simples animação de um *Viewpoint* (ver secção 4.2.19);
- ii. Animar objectos estáticos, é algo que também é possível. Pode-se simular em VRML uma ida ao cinema ou então uma estada no conforto do lar a ver televisão. Este processo é conseguido tal como na aplicação de uma textura a uma face, mas neste caso associa-se à face um filme (em formato MPEG), através do nó *MovieTexture*. Ainda associada a uma face, podemos criar outra forma de animação (Vacca, 1998). Em vez de um filme, uma textura animada. Como se mostrou nas texturas (secção 4.2.9), o autor tem uma grande flexibilidade no posicionamento da textura na face. Em vez de ter esse processo estático, é possível torná-lo dinâmico ao efectuar constantes alterações no mapeamento da textura pela face, simulando o movimento, água a deslizar numa cascata, por exemplo.

4.2.13 Áudio

De encontro aos desejos comuns de dotar os nossos mundos com características cada vez mais semelhantes à realidade, a incorporação num ambiente virtual de efeitos sonoros e áudio, perfila-se como uma das medidas com resultados mais

satisfatórios e consensuais, ao conseguir proporcionar ao utilizador uma envolvência mais realista com o ambiente virtual. Por exemplo, quando nas animações (secção anterior) se mencionou o efeito mais realista que era gerado pela visualização de uma nora em rotação num curso de água, imagine-se que essa rotação era acompanhada pela emissão do som gerado pelo curso de água, talvez acompanhado pelo chilrear de uns pássaros. É indiscutível que o cenário propiciado ao visitante seria com toda a certeza, muito mais agradável e realista.

Há contudo um senão no uso de áudio, o seu elevado custo em termos de recursos consumidos e tempo de *download*, que não é desprezível dada a tendência para grandes dimensões dos ficheiros de áudio. Assim sendo, e tal como foi recomendado na iluminação (secção 4.2.6), a utilização de áudio deve ser doseada, quer em número, quer na definição das suas características. Eis algumas recomendações:

- ✓ Tal como foi sugerido nas animações, conseguem-se optimizações quando o despoletar dos sons é executado por intermédio de eventos, um sensor de proximidade (*ProximitySensor*), por exemplo. Deste modo não só se garante que o ficheiro áudio é carregado e executado apenas quando for necessário, como se consegue delimitar a audição do som a uma região. Esta característica tem a sua utilidade, ao conseguir simular um efeito de ouvir ou não o som, por exemplo quando se entra ou sai de um edifício com musica ambiente no seu interior. Pois caso contrário o registo sonoro seria sempre audível dentro da elipsóide definida para o alcance do som, o que poderia incluir as paredes do edifício;
- ✓ A qualidade sonora é directamente proporcional ao tamanho dos ficheiros e essa qualidade é definida pela frequência da amostra sonora. Por exemplo a frequência de um CD áudio é de 44,1 Khz. Consegue-se reduzir o tamanho dos ficheiros, usando uma frequência mais baixa, mas que ainda assim mantenha uma qualidade sonora que se defina como aceitável;
- ✓ Todos os ficheiros de áudio usados no mundo devem usar a mesma frequência de amostragem. Assim evita-se a necessidade de uma reassemblagem da mesma por parte do *browser*, para os restantes ficheiros (Cosmo Worlds 2.0 User's Guide, 1998);

- ✓ O número de sons audíveis em simultâneo é limitado, não só por alguns *browsers*, pela latência permitida entre o *rendering*, ou ainda pelos recursos disponíveis do equipamento (Hartman e Wernecke, 1996). Pelo que, para assegurar que determinado registo sonoro seja audível, pode-se definir sons prioritários. O nó *Sound* tem um campo para esse efeito, *priority*;
- ✓ Para obter um realismo mais aproximado do atenuar da propagação do som com a distância, deve-se definir a elipsóide exterior 10 vezes maior que a interior, ou seja, $maxFront^* = 10 \times minFront^*$ e $maxBack^* = 10 \times minBack^*$ (Cosmo Worlds 2.0 User's Guide, 1998);
- ✓ Há uma optimização na utilização do áudio, quando não se define uma direcção particular para a propagação do som, isto é, o som parece surgir de todo o lado. Este efeito de som omnidireccional é conseguido através da conversão da elipsóide numa esfera ($maxFront = maxBack$ e $minFront = minBack$) (Carey et al., 1997). Este processo não proporciona um efeito muito realista, mas como reduz os cálculos internos realizados, optimiza.

4.2.14 Primitivas simples

Das formas geométricas existentes na vida real, exactamente aquelas que se tenta simular em ambientes virtuais, há algumas que surgem com frequência, eventualmente com diferentes dimensões, conferindo aos mundos uma certa semelhança. O VRML definiu algumas dessas formas como primitivas (nós), que irão servir de base ou suporte na modelação dos mundos, designadamente, a esfera, o cilindro, a caixa (paralelepípedo) e o cone. No entanto, para uma modelação realista de ambientes, o uso destas primitivas é manifestamente insuficiente na modelação de objectos mais complexos, existindo outros nós criados para esse efeito.

Apesar de, aparentemente, o uso dessas primitivas simples na modelação tridimensional ser bastante limitativo, tal não tem necessariamente de ser verdade. Com um pouco de imaginação e astúcia, conseguem-se resultados surpreendentes na modelação de algumas superfícies complexas, usando apenas

* Parâmetros do nó *Sound*

primitivas simples, intersectadas ou adjacentes por exemplo. Pode parecer pouco eficiente, usar várias primitivas (nós) para representar uma superfície, quando bastaria indicar todos os polígonos necessários à sua representação num único nó (*IndexedFaceSet*). Esta situação pode não ser verdadeira, e na generalidade dos casos não o é, devido aos algoritmos otimizados de representação das primitivas simples existentes nos *browsers* (Lemay et al., 1996). Estas primitivas são muito eficientes de usar e de representar, precisamente porque não necessitam de ter definidos os polígonos necessários à sua representação, pois os *browsers* dispõem de algoritmos internos para as representar eficazmente. Como esses algoritmos estão muito otimizados, o uso dessas primitivas é muito aconselhável do ponto de vista do desempenho. Por exemplo, pode-se conseguir modelar uma esfera, manualmente, com menos polígonos que a representada pela primitiva *Sphere*, mas esta última, seria com certeza interpretada e representada mais rapidamente, que a outra definida manualmente com menos polígonos.

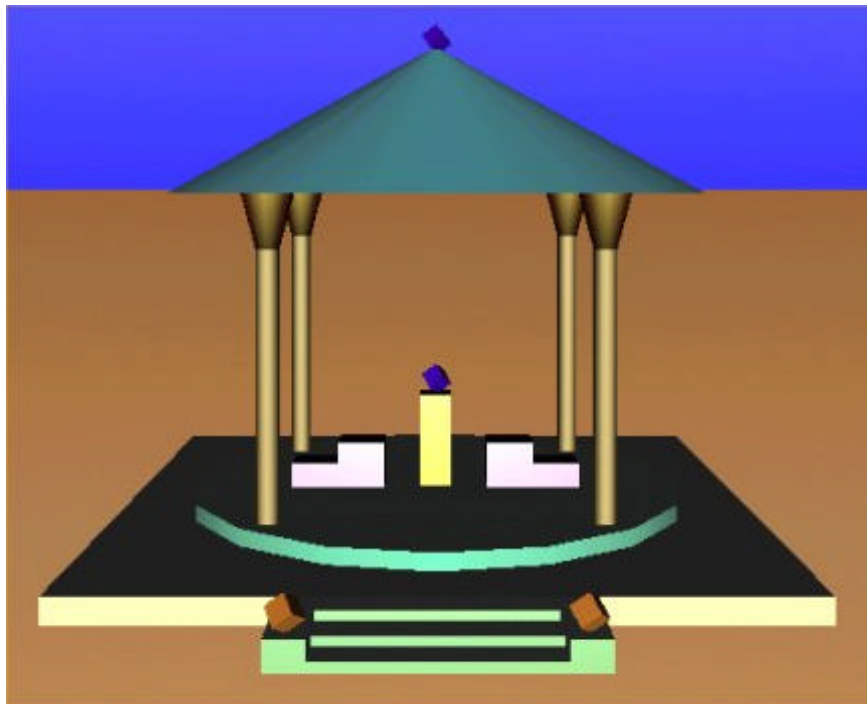


Figura 4.3 Exemplo construído apenas com recurso a primitivas simples

Em suma, as primitivas simples devem ser usadas sempre que possível, em virtude de os *browsers* estarem otimizados para as representar.

4.2.15 Propriedades do Avatar

Avatar é o termo usado para designar a personagem virtual que representa o utilizador no ambiente que está a explorar. A interacção do avatar com o meio envolvente é assente em algumas características físicas (altura e largura por exemplo) que possam simular esse visitante virtual. Dada a necessidade do *browser* definir antes do início da visita, o comportamento do avatar em algumas situações, nomeadamente a distância ao solo (altura) e detecção de colisões (largura), essas características têm valores predefinidos, por norma valores aproximados aos de um ser humano. Mas, estas e outras características podem e devem ser definidas pelo autor do espaço virtual, pois quando usadas convenientemente não só conseguem transmitir ao utilizador uma sensação mais realista, como podem melhorar o desempenho. Para esse efeito, existe o nó *NavigationInfo* que permite definir estas características. Em seguida apresentam-se algumas das mais importantes:

Dimensões do avatar (<i>avatarSize</i>)	Raio de colisão, altura e altura máxima de um obstáculo transponível pelo avatar;
Velocidade (<i>speed</i>)	Velocidade de deslocação pelo mundo;
Limite visual (<i>visibilityLimit</i>)	Distância máxima de visualização do avatar;
Tipo de movimento (<i>type</i>)	Movimentos permitidos ao avatar. Os dois mais comuns são: FLY, em que é permitido ao utilizador “voar” na cena e WALK, onde o único movimento possível é um deslocamento gravitacional, sempre “colado” ao solo.

Nota: A maior parte dos *browsers* têm no seu interface opções que permitem redefinir algumas destas propriedades, durante a navegação.

Um dos parâmetros a definir é o limite visual, que por defeito não estabelece qualquer limite à capacidade visual do avatar (para obter este efeito o valor do campo é 0), ou seja, todos os objectos dentro do campo visual são representados pelo *browser*, mesmo que se encontrem a uma distância considerável. No

entanto, objectos mais distantes podem ser grandes consumidores de recursos, devido à sua complexidade e a sua contribuição para o efeito visual da cena pode ser insignificante ou nula, em virtude de se encontrarem demasiado afastados. Neste caso, pode ser dispensável a representação desses objectos mais distantes do utilizador. O limite visual tem precisamente essa finalidade ao permitir definir qual a distância máxima do alcance visual³³ do explorador virtual, fazendo com que tudo o que esteja para além dessa distância, não seja renderizado, aumentando assim o desempenho do mundo virtual.

É importante que o valor definido para este limite visual seja razoável, pois caso contrário pode-se acabar por ter um efeito pouco realista. Por exemplo, se se estabelecer um limite visual relativamente pequeno, pode-se obter o efeito visual pretendido quando se está imóvel na cena. No entanto, quando iniciado qualquer movimento os objectos e superfícies vão surgindo gradualmente do vazio, como se algo estivesse penetrar no nosso mundo, o que não é muito realista. De facto, ganhou-se no desempenho, mas perdeu-se em realismo. Para evitar este tipo de situação sugere-se também o uso de um efeito atmosférico, como o nevoeiro, que tem um procedimento relativamente semelhante a este, mas proporciona um efeito muito mais realista (ver secção seguinte).

Normalmente o desempenho de um mundo avalia-se pela fluidez do movimento conseguido na cena por parte do utilizador. Se este movimento for lento e aos “soluços”, pode significar uma de três hipóteses (ou uma combinação delas):

- i. O mundo não está optimizado;
- ii. O equipamento não será o mais adequado para este tipo de visualização;
- iii. A velocidade definida para o avatar é muito baixa.

Praticamente a totalidade dos *browsers* dispõe de uma opção no seu interface que permite alterar a velocidade de deslocamento do avatar na cena. Como já foi referido, existe um parâmetro no nó *NavigationInfo* que permite definir uma velocidade para o deslocamento no mundo. A existência deste parâmetro no VRML tem em vista permitir um aumento de realismo, ao simular uma correcta movimentação no mundo. Mas, com uma certa imaginação, este parâmetro pode ser usado para disfarçar eventuais lacunas existentes na optimização do mundo,

³³ Distância em metros

ou a sua utilização num equipamento obsoleto. Ao aumentar-se a velocidade para um valor razoável, obtido através de testes sucessivos, consegue-se um aumento fictício de desempenho, apesar de apenas se ter aumentado a velocidade de interacção do utilizador com o mundo. Note-se, no entanto, que esta opção pode ajudar, mas não resolve a questão, pois os problemas de desempenho podem ser inultrapassáveis.

4.2.16 Nevoeiro

O grau de realismo conseguido na elaboração de um ambiente virtual é um dos factores primordiais para a satisfação do utilizador. Esse realismo é obtido através da inclusão de efeitos visuais que permitem simular a realidade. Mas, normalmente, o incremento de efeitos visuais é sinónimo de um decréscimo de desempenho. No entanto, isso nem sempre acontece. Existe no VRML um nó (*Fog*) que, ao reduzir gradualmente a visibilidade, permite simular efeitos de nevoeiro, fumo ou qualquer outro efeito atmosférico. Este efeito provoca que à medida que os objectos vão estando mais distantes do utilizador, ficam mais envoltos pelo nevoeiro e, por esse motivo, menos visíveis.

Ao contrário do que possa parecer, este é um efeito visual que melhora o desempenho e simultaneamente dá mais realismo à atmosfera visual do mundo. A redução de visibilidade implica a não visualização de objectos, ou parte deles, libertando o peso computacional associado à sua representação por parte do *browser* (Cosmo Worlds 2.0 User's Guide, 1998).

A distância a partir da qual os objectos passam a estar totalmente envoltos pelo efeito atmosférico e, deste modo, completamente invisíveis é definida no campo *visibilityRange*. Este pode ser usado em benefício do desempenho, pois quanto menor for esta distância, mais depressa os objectos são cobertos pelo nevoeiro e, portanto, deixam de ser representados.

Um outro parâmetro deve ser considerado, quer por questões de desempenho quer de realismo. Este nó também permite indicar um tipo de nevoeiro (*fogType*), que define o comportamento da densidade do nevoeiro com a distância, existindo duas opções:

- a) LINEAR, o avolumar da densidade de nevoeiro é directamente proporcional à distância. Os cálculos necessários à implementação deste tipo de nevoeiro são de fácil aplicação por parte do *browser*, mas não proporciona um efeito atmosférico muito realista
- b) EXPONENTIAL, a diminuição da visibilidade é mais acentuada com o aumento da distância. Este tipo provoca um efeito mais realista, mas tem um processamento mais pesado

Um nota final para sugerir a utilização do nó *Fog* em conjunção com o campo *visibilityLimit* do nó *NavigationInfo* (secção 4.2.15). Nessa secção indicou-se que quando os objectos ultrapassam a distância definida no campo *visibilityRange* deixam de ser perceptíveis na cena. Se se estabelecer o mesmo valor para o *visibilityLimit*, todos os objectos para lá dessa distância não são representados pelo *browser*, com proveitos significativos no desempenho. Mas é importante notar que só é conseguido um efeito realista se a cor do *background* for coincidente com a do nevoeiro

4.2.17 Âncoras

O tempo de *download* e a fluidez conseguida na navegação de um mundo são, como já foi referido por diversas vezes, dois dos factores primordiais ao sucesso do VRML. Estes factores são consequência directa, quer das dimensões dos ficheiros, no caso do *download*, quer do número de polígonos existentes em simultâneo na cena, no caso do desempenho.

A utilização de âncoras na cena é outra das medidas que possibilita conseguir, em simultâneo, reduzir o tamanho dos ficheiros e o número de polígonos na cena. Uma âncora tem um funcionamento em tudo idêntico ao de uma ligação numa página HTML, com a diferença de, neste caso, o que está do “outro lado” da ligação, poder ser um outro ficheiro VRML, que é carregado e visualizado quando o utilizador seleccionar com o rato uma determinada área.

Imagine-se uma visita virtual ao interior de uma casa no qual deseja conhecer algumas das divisões da mesma. A entrada para cada uma dessas divisões é feita por uma porta opaca, o que implica a não visualização do interior das

mesmas. Assim sendo, não há motivo para os interiores das referidas divisões serem representados, em virtude da sua invisibilidade. Neste caso é perfeitamente razoável adoptar o princípio da modularidade, discutido na secção 4.2.5, que conjugado com o uso de âncoras, pode resultar numa boa combinação para uma melhoria geral da eficiência. A metodologia a aplicar passa pela colocação da geometria de cada uma das divisões em ficheiros individuais, que apenas serão carregados para a memória e, conseqüente, visualizados, caso o utilizador accione a ligação da porta correspondente à divisão pretendida (Cosmo Worlds 2.0 Tutorial, 1998). Facilmente se constata a economia de recursos resultantes deste procedimento. A redução de informação no ficheiro principal dá origem a uma redução do número de polígonos existentes em simultâneo na cena e à conseqüente redução das dimensões do ficheiro em memória, seja o ele o principal ou os “secundários”. Desta forma saem beneficiados quer o desempenho, quer o tempo de *download*.

Há todavia um problema resultante da utilização de âncoras. Quando o utilizador activa uma âncora, o ficheiro correspondente tem de ser carregado para a memória e representado, em substituição do anterior. Este é um processo que leva o seu tempo, pelo que é inevitável a existência de uma descontinuidade muito pouco realista perante o utilizador, pois diante dos seus olhos vê desaparecer uma cena e surgir-lhe outra completamente diferente, não tendo, por exemplo, a sensação contínua de accionar a porta da divisão e entrar.

4.2.18 Culling

No início deste sub-capítulo referiu-se que uma boa estrutura hierárquica permite otimizar os testes de *culling*. Estes testes também incidem numa outra vertente: durante a modelação e após a disposição das superfícies pela cena, é natural que existam polígonos ou faces de polígonos (não esquecer que um polígono é constituído por duas faces opostas), que nunca serão visíveis pelo utilizador, ou por nunca estarem numa posição frontal perante ele (uma das faces de um *billboard*, por exemplo), ou por estarem encobertos com outros objectos. Como a quantidade de polígonos a representar é um dos factores inibitórios do desempenho, torna-se evidente que estes polígonos não devem ser representados. Desta forma consegue-se uma melhoria de desempenho ao

diminuir o número de polígonos a representar. Quando falamos em não representar polígonos ou faces de polígonos, duas perspectivas se enquadram: o polígono pura e simplesmente deixa de existir na geometria da superfície, ou então o polígono está definido e existe, apenas é indicado ao *browser* que não é necessária a sua representação. Por exemplo, o nó *IndexedFaceSet* tem um campo, *solid*, que quando tem o valor TRUE implica a não representação de um dos lados, normalmente o interior, do objecto, diminuindo para metade o número de polígonos a representar e afectar pela iluminação.

Nota: Quando os objectos em causa envolvam primitivas simples (caixa, esfera, cone e cilindro), este procedimento não deve ser adoptado, para não descaracterizar a representação dessas primitivas. Como apresentado na secção 4.2.14, os *browsers* dispõem de algoritmos optimizados para as representar.

4.2.19 Viewpoints

Na construção de ambientes é natural a existência de alguns locais de particular importância na cena. É normal que se pretenda estimular o visitante a observar o mundo a partir desses pontos. Essa pretensão é, de certa forma, viabilizada pela utilização de *viewpoints*.

Um *viewpoint* permite predefinir uma posição e orientação específica do avatar na cena, garantindo assim a visibilidade de determinado local. Uma cena pode ter definidos tantos *viewpoints* quantos forem necessários. Durante uma visita basta seleccionar o *viewpoint* desejado para que o utilizador seja “transportado” automaticamente para a posição definida por ele. Normalmente é disponibilizado pelo *browser* um menu com todos os *viewpoints* disponíveis na cena, de modo a facilitar a sua selecção.

Nota: É o primeiro *viewpoint* existente no ficheiro, que vai definir a posição inicial do avatar, quando da entrada no mundo (Carey et al., 1997).

Os *viewpoints* também podem servir como meio orientador do utilizador durante a visita, ao sugerir pontos de passagem pelo mundo, como se de um guia se tratasse. Mas, talvez a maior vantagem no uso de *viewpoints*, é o modo como facilita a navegação pelo mundo. Esta pode ser dificultada pela falta de destreza do utilizador a manusear o dispositivo de interacção (normalmente o rato), ou pelo facto de a visualização estar a ser efectuada num sistema com recursos limitados, onde o “salto” entre *viewpoints*, além de ser extremamente simples e eficaz, permite otimizar o *rendering* ao “transportar” o utilizador directamente para uma nova posição, sem ter de representar o espaço intermédio³⁴.

Em vez de ser o utilizador a ter que escolher os pontos de passagem para a visita, seleccionando os *viewpoints* sucessivos, é possível facultar-lhe uma visita guiada pelo mundo, já com passagem por todos esses pontos e com uma sequência definida. Pois bem, essa visita guiada é possível e bastante simples de concretizar, bastando para tal definir um *viewpoint* e aplicar-lhe uma animação cujo percurso abranja todos os locais desejados para a visita guiada.

4.2.20 Organização espacial

A tarefa de distribuir e posicionar os objectos modelados pela cena exige um grau de precisão assinalável. Uma boa organização dos objectos quer na cena, quer no próprio ficheiro, pode contribuir para um aumento da eficácia geral do mundo.

Uma boa organização espacial no ficheiro VRML implica que a proximidade dos objectos na cena, se reflecta na proximidade com que estão definidos no próprio ficheiro, principalmente em ficheiros muito extensos. Desta forma, optimizam-se os movimentos do *browser* pelo ficheiro durante a representação, limitando esses movimentos ao mínimo, proporcionando assim um *rendering* mais eficiente (Cosmo Worlds 2.0 Tutorial, 1998).

Uma boa organização espacial dos objectos pela cena também pode contribuir para uma melhoria do desempenho. Se possível, a distribuição de partes da cena mais complexas pelo espaço tridimensional deve ser feita tentando manter uma

³⁴ Os browsers normalmente, dispõem de uma opção que permite alterar este comportamento

certa distância entre elas. Dessa forma consegue-se limitar a simultaneidade da visualização dessas áreas, melhorando assim o desempenho.

4.2.21 Mudança de escala

Após a modelação dos objectos, durante a sua disposição pelo espaço tridimensional, é frequente realizar pequenos ajustes às dimensões dos mesmos, de maneira a que o “encaixe” entre objectos adjacentes seja perfeito. Este é um processo bastante simplificado pelo nó *Transform* que permite esse ajuste sem redefinir as características geométricas do objecto. Este nó permite modificar dimensões em qualquer dos eixos coordenados, designadamente por intermédio de uma mudança de escala.

No entanto, este procedimento deve ser usado com moderação, pois por cada mudança de escala aplicada a um objecto, as normais que lhe estão associadas - como dependem do número de polígonos, podem ser em número considerável -, têm de ser recalculadas e normalizadas, sendo este último um processo que envolve cálculos mais intensivos, com o peso computacional a reflectir-se na eficácia (Foley et al., 1990). Esta não é uma situação totalmente desprezível, pois as normais são fundamentais na visualização da cena, ao intervirem, por exemplo, nos cálculos de iluminação.

A atitude mais correcta a tomar para evitar esta situação, consiste em, sempre que possível, modelar os objectos já com as suas dimensões finais, para não dar origem a posteriores reajustamentos.

4.2.22 Compressão

Um dos factores condicionantes da visualização em VRML centra-se, como já foi abordado anteriormente, no tempo que o utilizador tem de esperar até ter algum *feedback* visual da cena e dar início à visita. Claro que é o tamanho dos ficheiros associados à cena, sejam ficheiros VRML, texturas ou áudio, o principal responsável por este condicionalismo.

Desde os primórdios da Web que se fazem estudos sobre a implicação que o tempo de espera de um utilizador até ter algum retorno, tem no grau de sucesso de uma página. Os resultados indicavam o óbvio, quanto mais lento o *download*, menos interesse despertava a página e, por conseguinte, era pouco provável que um utilizador a revisitasse ou aconselhasse. O VRML não é muito diferente neste aspecto. Quanto mais tempo demorar o utilizador a poder desfrutar das potencialidades do ambiente virtual, menos interesse este desperta perante um utilizador ou um potencial cliente. Esse é motivo mais que suficiente para também focar a atenção dos autores nas dimensões dos ficheiros, que devem ser o mais diminutas possível. Nas secções anteriores foram abordadas essencialmente questões de desempenho durante a visualização, embora algumas das técnicas usadas, acabassem também por contribuir para uma redução do tamanho dos ficheiros. Mas, ainda assim, outras questões merecem ser discutidas sobre este último aspecto.

Um ficheiro VRML, na realidade não passa de um ficheiro de texto e muitas vezes estes ficheiros têm dimensões superiores ao que seria de esperar para a quantidade de texto existente no ficheiro. Esta situação verifica-se porque, para os ficheiros de texto serem legíveis, normalmente são formatados, ora com espaçamentos, ora com indentações. Esta legibilidade do ficheiro pode ser-nos útil e necessária enquanto se trabalha no mesmo, mas não o será de todo quando o trabalho estiver concluído e o ficheiro apenas for necessário para visualização no *browser*. Perante a necessidade de reduzir o mais possível as dimensões dos ficheiros, é comum compactar os ficheiros VRML, através de um utilitário de compressão, o *gzip* (Lemay et al., 1996). Este utilitário é bastante eficaz em ficheiros de texto, pois consegue taxas de compressão na ordem das 10 vezes. No nosso caso, o ficheiro VRML passou de cerca de 200Kb, para 22Kb. Actualmente, grande parte dos *browsers* disponibilizam a descompactação do formato *gzip*, em tempo real, permitindo assim a leitura e visualização imediata do ficheiro compactado sem qualquer dificuldade.

O *Cosmo Worlds*, a ferramenta de desenvolvimento VRML usada na elaboração deste trabalho, tem um comando, *Publish*, que para além de outros procedimentos, efectua este processo de forma automática. A utilização deste comando é vivamente recomendada, pois, para além do já referido, também

efectua outro tipo de optimizações e organização (*Cosmo Worlds 2.0 User's Guide, 1998*).

Actualmente, a maioria das aplicações gráficas com capacidades de modelação tridimensional tem uma opção de exportação para VRML. No entanto, além dessa exportação ser muitas vezes ineficaz em questões de desempenho (secção 4.2.24), também o é em termos das dimensões finais dos ficheiro VRML. Durante o nosso trabalho, tivemos oportunidade de testar um ficheiro VRML oriundo de uma outra aplicação, ocupando 8Mb. Após a compressão ficou com cerca de 1Mb. Apesar deste valor final ser ainda muito elevado, a economia de espaço e tempo é notável.

4.2.23 Scripting

Um *script* em VRML é um programa escrito numa linguagem suportada pelo *browser*, normalmente Java³⁵ ou JavaScript, que recebe eventos gerados pelo mundo ao qual está associado, e executa uma acção em conformidade com a informação recebida. É frequente utilizar um *script* para definir um comportamento na cena, normalmente associado a uma animação. Tal acontece sempre que se pretende um comportamento mais sofisticado do que o proporcionado pelos interpoladores. Por exemplo, para que a porta de um cofre só abra após a introdução de um código secreto; é necessário um factor externo ao VRML, que possa verificar se o código introduzido está correcto ou não. Daí a utilidade dos *scripts*.

Nota: Não usar *scripts* para definir comportamentos que também possam ser gerados com interpoladores, nomeadamente animações. Animações criadas com interpoladores, são executadas mais rapidamente e consomem menos recursos, do que se forem criadas com *scripts* (*Cosmo Worlds 2.0 User's Guide, 1998*).

³⁵ Apesar do Java não ser uma linguagem de scripting, ainda assim os programas denominam-se por scripts

O uso de *scripts* pode, de facto, elevar as capacidades do VRML para um outro nível, mas o seu uso deve ser moderado. É que os *scripts* tornam o desempenho do *browser* mais lento, na medida em que para a execução de um *script* implica o tempo necessário para carregar o interpretador da linguagem usada e o tempo de execução do próprio *script* (Marrin e Campbell, 1997). Será ainda de considerar o tempo de *download*, necessariamente mais elevado, em virtude do aumento das dimensões do(s) ficheiro(s) devido ao código necessário para implementar o *script*. Desta forma, o uso de *scripts* é aconselhável apenas quando o seu uso for uma mais valia para o mundo, pois o processamento extra necessário para carregar, correr e executar o *script* pode limitar a capacidade de enriquecer o mundo com outras características, também elas relevantes.

Uma referência final para as duas linguagens mais usadas na criação de *scripts*. O JavaScript é uma linguagem interpretada, em que o código pode estar embutido no próprio ficheiro VRML, sendo interpretado e executado em *runtime*. O Java já necessita de uma prévia compilação do código, dando origem a um ficheiro binário. Neste caso o *download* do(s) ficheiro(s) pode ser mais lento, em virtude das dimensões superiores dos ficheiros binários.

4.2.24 Exportar para VRML

Após o reconhecimento das vantagens e potencialidades do VRML, grande parte do *software* de modelação tridimensional existente no mercado passou a disponibilizar aos seus utilizadores uma opção de exportação do formato nativo da aplicação para o VRML. Esta foi uma característica que se revelou importante, principalmente numa fase inicial, em virtude da inexistência, na altura, de uma ferramenta eficaz de desenvolvimento VRML. Actualmente já existem ferramentas de desenvolvimento VRML, mas continuam a ser bastante utilizados os motores de exportação para VRML oriundos de outras aplicações, provavelmente devido à simplicidade e rapidez desse processo e da familiaridade que os autores já têm com essas aplicações.

O problema principal desses motores de exportação é produzirem ficheiros VRML ineficientes quer em desempenho, quer nas dimensões finais dos ficheiros gerados. Tivemos oportunidade de testar um ficheiro proveniente da exportação

de uma aplicação de modelação 3D (*SolidWorks*), em que as suas dimensões finais ascendiam a mais de 8Mb. Esta situação resulta da fraca capacidade dos algoritmos dessas aplicações para introduzir optimizações, limitando-se a fazer a conversão do modelo tridimensional directamente para VRML, não havendo preocupação, por exemplo, com o número de polígonos, sendo normal serem gerados mais polígonos que os necessários para representar uma superfície.

Na nossa actividade académica deparamos com situações elucidativas relativas a esta situação: alunos a quem foi pedido a elaboração de um projecto em VRML, concretizaram a sua implementação exclusivamente com uma aplicação de modelação bastante conhecida, no caso o 3DStudio, posteriormente recorreram ao comando *Export* existente na referida aplicação para obter o trabalho final em VRML. Como seria de prever, visualmente o resultado final era bastante aceitável, mas o desempenho, medido pela navegabilidade na cena, era de tal forma limitado, que apenas num equipamento de elevado desempenho gráfico era possível alguma fluidez na navegação. Mais tarde, um outro grupo de alunos teve a incumbência de prosseguir o mesmo trabalho, fazendo incidir os seus esforços nas questões de optimização. Como resultado foi mantido o modelo geométrico, mas obteve-se uma economia de polígonos na ordem dos 80% a 90% (Narciso, 1999) com o consequente acentuado decréscimo nas dimensões dos ficheiros.

Nota: Existem no mercado algumas aplicações e plug-in's que permitem minorar esta situação. O *Internet Model Optimizer* da *Parallel Graphics* é uma aplicação vocacionada para a optimização em VRML de modelos tridimensionais elaborados em ferramentas CAD/CAM/CAE. Outro exemplo é o *Mark's 3d studio max to VRML97/X3D Export Plug-In*, que é um pequeno plug-in que substitui o existente no 3DStudio, e que além de acrescentar novas potencialidades à exportação, corrige alguns dos erros resultantes de uma exportação efectuada com o plug-in original.

Um outro problema que pode acontecer com os algoritmos de exportação, é uma incorrecta geração das normais associadas aos polígonos de uma superfície, com possíveis consequências futuras, em virtude das normais serem usadas, entre

outros, nos cálculos de iluminação, e desse modo poderem surgir erros na representação e conseqüentemente na visualização (Lemay et al., 1996).

Logicamente que a estratégia ideal na elaboração de mundos em VRML, consiste na sua total implementação directamente em VRML, usando uma ferramenta adequada para o efeito, o Cosmo Worlds por exemplo. Mas essa solução é por vezes de difícil concretização, nomeadamente no Cosmo Worlds não é muito fácil modelar pequenas peças individuais que possuam um grau de detalhe e precisão geométrica bastante apurado. Nestes casos, e este procedimento foi adoptado numa fase inicial do nosso trabalho, a modelação dessas pequenas peças pode ser feita com uma ferramenta auxiliar, que permita ultrapassar algumas dessas limitações. Posteriormente toda a disposição e assemblagem dos objectos modelados pela cena é feita já na ferramenta de desenvolvimento VRML, efectuando as optimizações desejadas. Desta forma, consegue-se obter um compromisso entre eficácia e o processo natural de desenvolvimento.

4.2.25 Memória

A memória é um dos recursos mais preciosos na visualização de ambientes tridimensionais e todas as medidas que levem a um menor consumo da mesma³⁶ serão importantes, uma vez que têm repercussões directas no desempenho.

Além de alguns procedimentos já discutidos ao longo deste capítulo, todos eles directamente relacionadas com VRML, outros há, não directamente relacionados com VRML, que contribuem também para uma poupança de memória.

Durante a visualização, uma parte da memória é reservada para armazenar a informação dos chamados *front* buffer e *back* buffer, responsáveis pelo estado actual da cena (Cosmo Player 2.1.1 Documentation, 1999). A quantidade de informação armazenada nestes buffers e, conseqüentemente, o consumo de memória, é variável, dependendo de alguns factores:

- ✓ A área de visualização do mundo, é um dos principais factores. Quanto maior for, mais pixels têm que ser representados, mais informação tem de ser armazenada nos *buffers* e mais memória é ocupada. Uma

³⁶ Com as placas gráficas de última geração, a memória consumida é a da própria placa gráfica

diminuição do espaço ocupado pode passar pela redução da área de visualização, através da diminuição da janela de visualização (*browser*), ou reduzindo a resolução de visualização do periférico de saída, o monitor;

- ✓ A profundidade da paleta de cores é também um dos grandes responsáveis pelo consumo de memória do *front* e *back* buffers. Por exemplo, uma profundidade de 24-bits consome mais 50% de memória que a de 16-bits (Cosmo Player 2.1.1 Documentation, 1999). Esse consumo de memória não tem correspondência proporcional nos ganhos de qualidade na visualização, pelo que muitas vezes é preferível utilizar uma profundidade mais baixa.

4.2.26 *Browser*

Todas estas questões relacionadas com melhorias de desempenho e memória ocupada, acabam por enviar para segundo plano um outro aspecto importante, o interface de visualização, ou seja, o *browser* ou plug-in VRML. É este quem vai ler, interpretar e representar a informação contida no ficheiro, de acordo com os seus algoritmos internos.

Dado o número considerável de *browsers* VRML existentes, é natural que a sua eficácia seja distinta, implicando diferentes desempenhos para o mesmo ficheiro VRML. A utilização de OpenGL ou DirectX, e a forma como cada um tira partido da aceleração por hardware e das potencialidades das últimas gerações de *chips* gráficos³⁷, são alguns dos aspectos que influenciam o desempenho do *browser*.

É importante que em cada projecto concreto se teste o desempenho de diversos *browsers* ou plug-ins na situação concreta, procurando sempre utilizar a última versão de cada um, pois esta normalmente está melhorada e corrige erros das versões anteriores. Desse modo, é possível verificar melhor qual o que apresenta resultados mais satisfatórios para o caso concreto. No caso do nosso trabalho, foi testado o *Cosmo Player*, um dos clientes VRML mais populares, verificando-se que não representa na perfeição alguns pormenores do Fórum, como por

³⁷ Muitas destas características são configuráveis no interface de cada browser

exemplo as portas de acesso aos pórticos do templo³⁸. Ao contrário deste, que já não é actualizado há alguns anos, a *Parallel Graphics* tem investido no desenvolvimento e actualização do seu próprio cliente VRML, o *Cortona*, que além de maximizar o desempenho por meio dos seus algoritmos internos, tira um melhor partido das características do VRML não apresentando a limitação verificada com o *Cosmo Player*.

Em suma, uma má escolha do *browser* pode deitar por terra algum do cuidado colocado na construção do mundo.

4.3 Optimizações efectuadas neste trabalho

4.3.1 Introdução

No sub-capítulo anterior foi apresentado um conjunto de estratégias que visam melhorar o desempenho e até o realismo de ambientes virtuais criados em VRML, tanto no desenvolvimento, como na visualização. De seguida irão ser apresentadas as optimizações efectuadas neste trabalho. Foram estas que ditaram os resultados finais obtidos, nomeadamente ao nível do realismo, desempenho e dimensões finais dos ficheiros (Gonçalves e Mendes, 2003).

As características muito específicas do Fórum Flaviano de Conimbriga e alguns dos objectivos propostos com este trabalho implicaram que dos itens abordados no sub-capítulo anterior, nem todos fossem implementados, porque:

- ✓ diminuía o realismo. Apesar da optimização ser um dos objectivos primordiais, havia um nível mínimo de realismo a conseguir, o que inviabilizou, por exemplo, a utilização de LOD's;
- ✓ a sua utilização não seria viável. Como praticamente a totalidade do Fórum está visível a partir da sua entrada, algumas das recomendações são inviáveis;
- ✓ não fazia sentido neste trabalho, como por exemplo o *scripting*;
- ✓ ou simplesmente por opção do autor, como os *inlines*. Apesar de todas as vantagens reconhecidas à utilização de *inlines*, a sua não utilização foi

³⁸ Tal situação não aconteceu em *chips* gráficos mais poderosos

consciente e propositada. A portabilidade inerente a um único ficheiro VRML e a pequena dimensão final desse mesmo ficheiro, foram os fundamentos de tal decisão. No caso de mais tarde se pretender modularizar componentes do Fórum em ficheiros independentes, a boa estrutura hierárquica interna do ficheiro e as características da ferramenta de desenvolvimento (Cosmo Worlds 2.0), permitiam concretizar esse procedimento muito facilmente e em poucos minutos.

4.3.2 Hierarquização

Como foi referido, este é um dos conceitos mais importantes no desenvolvimento de ambientes virtuais em VRML. Por esse motivo, e dado os objectivos muito específicos a alcançar com este trabalho, este tópico teve grande relevância no desenvolvimento do mesmo.

Em consonância com as directrizes sugeridas na secção 4.2.2, todo o ficheiro VRML é uma enorme estrutura hierárquica, onde cada objecto engloba outros objectos. Sendo também muito provável que este último, também ele seja parte integrante da hierarquia de um outro ainda maior.

As recomendações apresentadas nessa secção foram seguidas, de tal modo que:

- ✓ Grandes objectos foram divididos noutros mais pequenos. Por sua vez, estes foram divididos em outros ainda mais pequenos, e assim sucessivamente. Talvez o exemplo mais claro desta situação neste trabalho, sejam os Pórticos do Templo. Como é uma estrutura que envolve toda a largura do Fórum, se fosse modelado e agrupado em conjunto, tornaria ineficiente a visualização, a navegação e a própria modelação. Assim, numa primeira abordagem foi “dividido” em ala esquerda, ala direita e ala central, prosseguindo em cada uma delas esta subdivisão por áreas modulares, como se pode constatar na figura seguinte referente à ala esquerda;

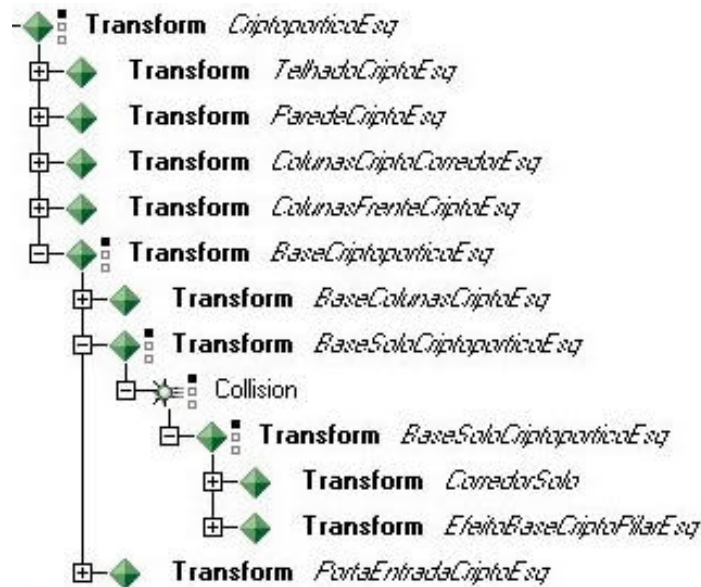


Figura 4.4 Estrutura hierárquica expressa no Cosmo Worlds

- ✓ Em toda esta estrutura hierárquica existente no ficheiro VRML, qualquer nó de um determinado nível, tem todos os seus filhos próximos entre si, para facilitar e otimizar os testes de visualização e detecção de colisões.

Uma boa estrutura hierárquica, também poderá trazer benefícios durante a fase de desenvolvimento de um projecto em VRML. A manipulação dos objectos existentes na cena é mais prática, cómoda, célere e menos sujeita a erros, em virtude de se poder manipular como um todo, um conjunto de objectos, eventualmente numeroso ou complexo.

Em suma, julgamos ter definido uma boa estrutura hierárquica, que permitiu maximizar a optimização, particularmente ao nível dos cálculos matemáticos que são efectuados com recurso às *bounding boxes* dos objectos existentes no ambiente VRML.

4.3.3 Colisões

Dado que os testes de colisão são geralmente pesados, as questões de optimização ligadas com este aspecto foram objecto de atenção particular, nomeadamente:

- ✓ Desactivação dos testes de colisão com as superfícies com as quais não é possível colidir. Como apenas é permitido ao utilizador um “andar”³⁹ gravitacional pelo recinto do Fórum, ele nunca irá colidir com muitas das superfícies existentes no mesmo.
- ✓ Para facilitar e simplificar os testes de colisão, foram definidas, através das propriedades do VRML, formas geométricas menos complexas, como primitivas simples por exemplo, para serem usadas nos cálculos da detecção de colisões em substituição das formas originais, estas mais complexas e logo geradoras de cálculos mais intensivos. Esse método optimiza, por intermédio de uma redução significativa dos testes a efectuar, todo o processo da verificação da existência ou não de uma colisão entre o avatar e uma qualquer superfície com essas características. Esse foi um procedimento concretizado sempre que possível em todo o espaço do Fórum. Na figura seguinte pode-se apreciar um exemplo. Toda a zona da base do pórtico, assinalada a vermelho na figura, é constituída por inúmeras “peças” individuais. No entanto, os testes de colisão são efectuados com um paralelepípedo que abrange toda a sua extensão.

³⁹ O VRML possibilita outras formas de movimentação pelo espaço virtual, mas foram inactivas por opção do autor

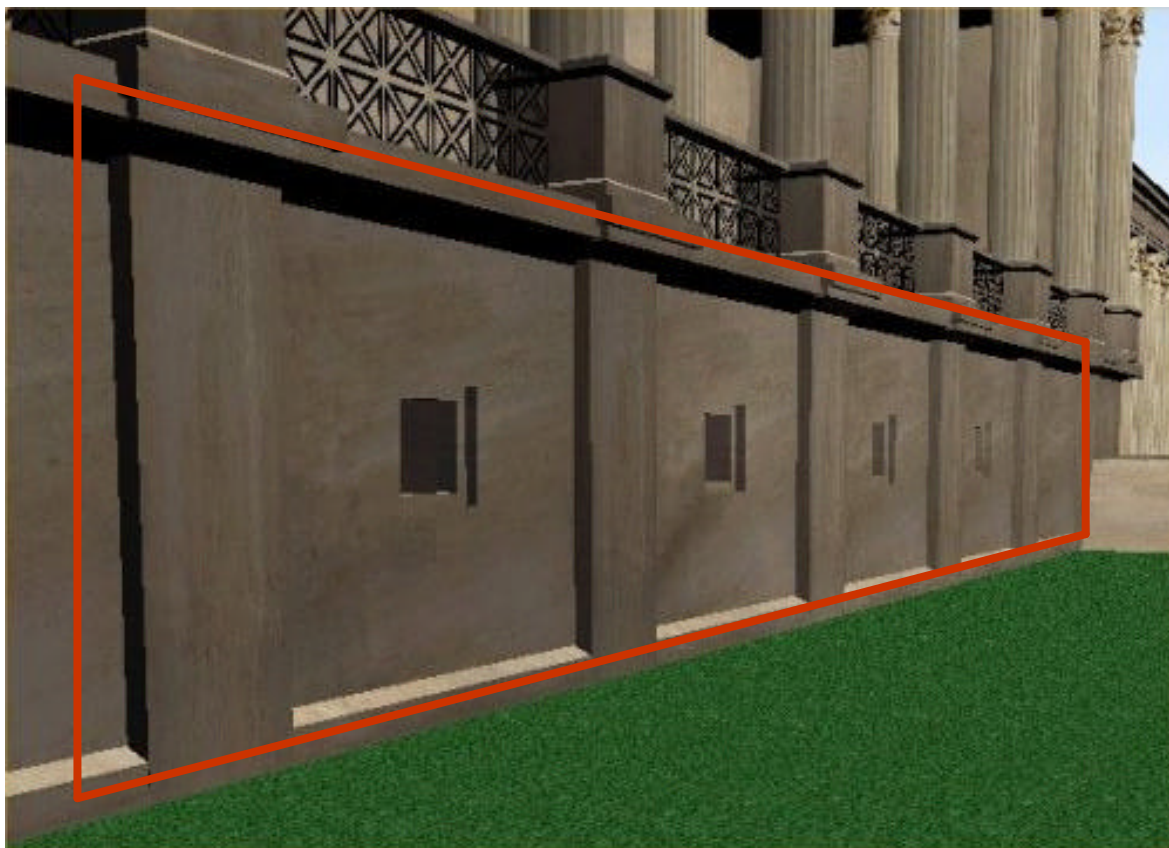


Figura 4.5 Zona, indicada a vermelho, sujeita à optimização das colisões

Relembra-se que, uma boa estrutura hierárquica também contribui para a optimização dos testes de colisão.

4.3.4 Instanciação (Clonagem)

Tendo em conta as características do Fórum Flaviano, esta foi a técnica mais utilizada no desenvolvimento deste trabalho, tendo sido uma das grandes responsáveis pelos resultados finais obtidos. As vantagens da sua utilização foram amplamente discutidas na secção 4.2.4. O Fórum Flaviano de Conimbriga acabou por se revelar perfeito para a utilização desta técnica, dado que o mesmo era uma grande sequência repetitiva de padrões geométricos que, inclusivamente, se podem definir hierarquicamente.

Por exemplo, toda a parede do corredor esquerdo existente no pátio segue um determinado padrão geométrico. Assim sendo, apenas um desses “blocos” foi modelado (ver Figura 4.6), sendo a parede gerada por clonagem desse bloco e por disposição de forma adjacente dos blocos assim criados. Esta metodologia é

muito mais eficaz e fácil de implementar, comparativamente com a modelação total e de uma vez só da parede como um bloco único.

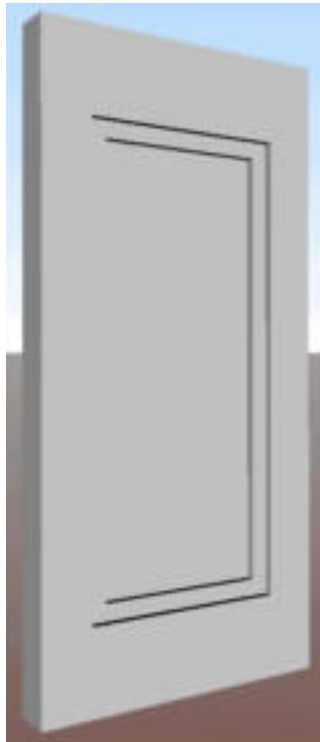


Figura 4.6 Bloco, único, modelado para usar nas paredes do pátio

Após a modelação de todo o corredor esquerdo, constatou-se que o corredor direito é idêntico, diferindo apenas numa rotação de 180° acompanhada de uma translação. Assim sendo, todo o corredor esquerdo, que já é constituído por diversos clones, foi clonado para dar origem ao corredor direito. Com esta metodologia, conseguiu-se conceber duas estruturas relativamente complexas apenas com a modelação de três objectos individuais: o “bloco” individual da parede, o telhado e uma coluna (também ela depois clonada ao longo do corredor).

O recinto do Fórum inclui 116 colunas de três tipos distintos. Sem a utilização da instanciação seria impossível atingir os resultados que foram obtidos, principalmente ao nível do desempenho e da dimensão final do ficheiro VRML, pois, por exemplo, o elevado número de colunas e as suas características geométricas, comprometeriam certamente o propósito deste trabalho.

De referir ainda que a instanciação também pode ser usada em objectos que, sendo iguais geometricamente, não possuam as mesmas dimensões (como aconteceu com as colunas). Tal facto, dá a possibilidade de otimizar mesmo onde aparentemente não seria possível fazê-lo. Exemplo disso mesmo é

apresentado na figura seguinte. O pequeno objecto situado no início e ao centro da grande escadaria do templo é um clone dos pódios laterais.



Figura 4.7 A peça indicada foi obtida por clonagem

4.3.5 Iluminação

Dadas as consequências que poderiam advir de uma má utilização da iluminação (ver secção 4.2.6), o nosso cuidado incidiu em duas vertentes fundamentais: quais os tipos de iluminação a utilizar e o número de pontos de iluminação a utilizar. Relativamente ao primeiro aspecto, a escolha recaiu nos dois menos dispendiosos em termos de processamento. Foi utilizado um *DirectionalLight* e um *PointLight*. O número de focos utilizados, foi o mínimo necessário para obter um bom efeito visual. Através de um correcto posicionamento dos focos na cena, foi possível conseguir um bom compromisso realismo/desempenho com a inclusão de apenas dois focos na cena.

4.3.6 Número de Polígonos

Durante a modelação de todas as peças usadas neste trabalho, a preocupação com o número de polígonos foi sempre uma constante. No entanto, a redução do número de polígonos não poderia lesar significativamente o realismo pretendido.

Assim, mais uma vez, teve de prevalecer algum equilíbrio entre o realismo de cada peça e o número de polígonos necessários para a representar. Por vezes conseguiu-se uma redução significativa no número de polígonos de uma peça, sem que ela tenha implicado um decréscimo, na mesma proporção, do realismo da mesma. Este é, sem dúvida, um dos caminhos a seguir na modelação de formas e objectos, em projectos desta natureza.

A representação do Fórum Flaviano de Conimbriga inclui perto de 17000 polígonos. Aparentemente, e dado que quase todo o Fórum está visível logo no início da visualização, este número seria incomportável para um bom desempenho da representação, particularmente quando fosse usado um equipamento informático sem grandes potencialidades gráficas. Na realidade, tal não acontece devido às optimizações efectuadas e ao facto já referido de uma boa parte dos objectos que constituem o Fórum serem clones entre si.

4.3.7 Texturas

Fazendo uma análise a este trabalho do ponto de vista do produto final e na perspectiva do utilizador, este item das texturas é, sem dúvida, o mais importante de todos. Sem texturas este trabalho não teria um grau de realismo aceitável, pois são elas que dão “vida” ao Fórum e nos fazem recuar no tempo para as cores e formas do antigo Império Romano.

A importância das texturas no aspecto final do trabalho levou a ter sido dada uma atenção muito especial a este item. Mas, se o realismo é fundamental, as texturas podem ter um impacto negativo no desempenho, pelo que também aqui urge optimizar. Assim e seguindo as recomendações já referidas anteriormente:

- ✓ Quase todas as texturas têm dimensões (altura e largura) potências de 2. Apenas a textura da coluna, pela importância e especificidades muito próprias, não obedece a esta regra;
- ✓ Usar texturas para simular superfícies complexas, que de outro modo necessitariam de centenas ou milhares de polígonos para as representar. Apesar dos vários casos existentes neste trabalho, o exemplo mais flagrante e que maior dificuldade gerou foi o capitel das colunas;

- ✓ Ainda associado a este último ponto, para conseguir simular superfícies complexas, por vezes é necessária a utilização de transparências. O que aconteceu no nosso caso com as texturas da coluna, da grelha do pórtico do templo e do frontão do telhado;
- ✓ Todas as texturas foram tratadas de modo a terem uma boa relação tamanho/qualidade. Muitas delas não ultrapassam os 2Kb, sem que esse facto seja perceptível na qualidade geral do Fórum;
- ✓ Com vista a conseguir esta relação tamanho/qualidade e devido à utilização de transparências, os formatos utilizados nessas texturas foram o PNG para as texturas com transparências e o JPG para as restantes;
- ✓ Para uma maior economia no tamanho das texturas, algumas delas foram tratadas de modo a poderem ser dispostas entre si em mosaico. Um dos melhores exemplos é o lajeado da praça, que foi obtido com uma textura simples com 1.68Kb (ver Figura 4.8);



Figura 4.8 Textura usada no lajeado da praça

- ✓ Usando a mesma textura, logo sem necessidade de carregar outra(s) para a memória, conseguem-se diferentes efeitos visuais, apenas por alteração de alguns parâmetros do próprio VRML. Por exemplo, as dimensões da pedra do lajeado da praça são diferentes das do patamar intermédio de acesso ao templo. No entanto, a textura é a mesma.

4.3.8 *Bilboards*

A utilização de *billboards* neste trabalho resumiu-se a uma única entidade geométrica existente no Fórum. No entanto, a sua utilização foi um dos grandes responsáveis pelos resultados obtidos ao nível do realismo, desempenho e tamanho final do ficheiro VRML.

Essa única entidade geométrica sobre o qual incidiu a utilização do *billboard*, foi a coluna coríntia do Fórum Flaviano de Conimbriga. O seu apurado detalhe

geométrico e o elevado número de colunas existente por todo o recinto do Fórum, faziam com que fosse importante otimizar a sua representação. De acordo com o referido na secção 4.2.10, esta coluna revelava-se perfeita para a utilização de um *billboard*, nomeadamente:

- ✓ Forma geométrica bastante complexa;
- ✓ Utilização de uma textura para a representar;
- ✓ Existência de uma simetria nas suas formas.

Como os testes experimentais com esta metodologia revelaram-se satisfatórios, essencialmente ao nível do realismo proporcionado, esta foi a técnica usada numa das colunas do Fórum, pois todas as restantes foram obtidas por clonagem (secção 4.2.4), mesmo apesar de algumas terem dimensões diferentes entre si.

Com vista a um acréscimo no realismo, uma das recomendações da secção 4.2.10, não foi seguida: aplicar o *billboard* com uma textura a uma face simples de dois polígonos. Partindo de um cilindro sem base nem topo e “cortando-o ao meio”, utilizou-se uma das metades para, não só colocar a textura, como aplicar o próprio *billboard*. Desse modo, houve melhorias na visualização a dois níveis:

- ✓ como os cálculos de iluminação são aplicados a uma superfície curva, o efeito proporcionado pela mesma é superior;
- ✓ a visualização da coluna, especialmente na zona do capitel, proporciona um efeito mais realista se aplicado numa superfície curva, em vez de plana.

Nesta situação, foi preferível comprometer um pouco o desempenho em favor do realismo. Logicamente, a decisão foi tomada após a realização de alguns testes com ambas as possibilidades.



Figura 4.9 Como se pode ver todas as colunas estão de frente para o utilizador/visitante

4.3.9 Animações

Tendo em conta as especificidades do monumento reconstruído, apenas foi incluída uma animação em todo o espaço virtual. A que efectua uma visita guiada pelos pontos principais do Fórum Flaviano de Conimbriga.

4.3.10 Primitivas simples

Pelo exposto na secção 4.2.14, sempre que possível foram utilizadas primitivas simples⁴⁰ na modelação dos diferentes objectos do Fórum, mesmo quando tal não se afigurava necessário. Por exemplo, as paredes dos pórticos do templo, bem como o próprio templo (dado que não se representou o seu interior), não são mais que caixas (paralelepípedos). No entanto, houve outros objectos mais complexos, que foram “construídos” com recurso a primitivas simples, ora por intersecção das mesmas, ora pela disposição adjacente entre si. Por exemplo, todos os efeitos existentes nas portas, foram conseguidos através da disposição adjacente de paralelepípedos com dimensões (altura, largura e profundidade) distintas.

⁴⁰ Cilindros, cones, esferas e paralelepípedos



Figura 4.10 Porta usada no Fórum

No entanto, a técnica de intersecção/sobreposição de paralelepípedos também foi amplamente utilizada. Por exemplo, com excepção de uma pequena peça, os pódios foram obtidos dessa forma, bem como muitos dos plintos existentes na praça. Alguns exemplos são apresentados nas figuras seguintes.

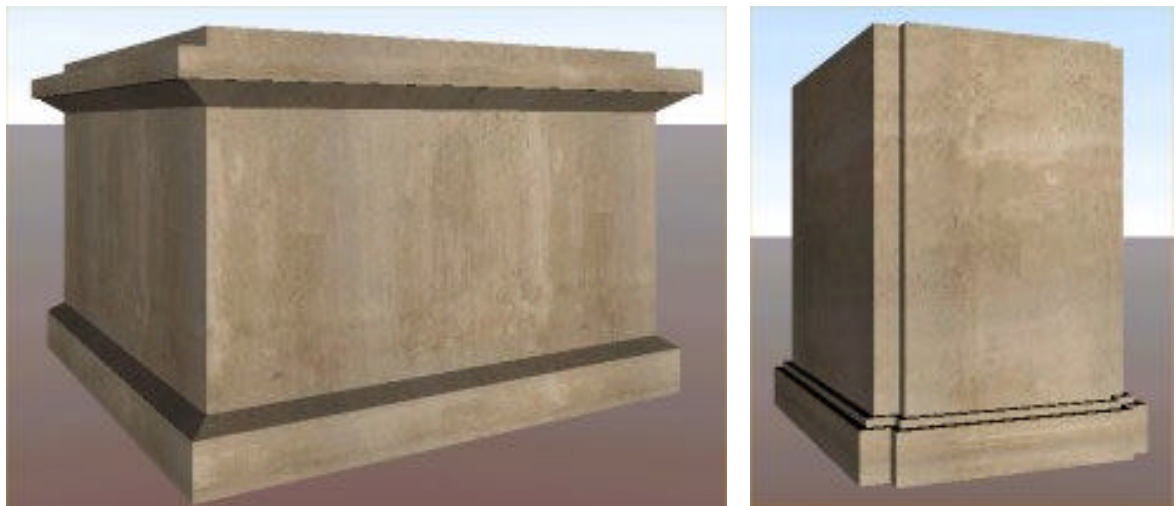


Figura 4.11 Exemplos de peças construídas por intersecção de primitivas simples. Um pódio (esq.) e um ornamento da entrada (dir.)

4.3.11 Propriedades do Avatar

Dos quatro parâmetros principais do Avatar (discutidos na secção 4.2.15), apenas foram modificados os valores por defeito de dois deles. Os restantes não foram alterados, dado que isso retiraria realismo ao espaço virtual.

O tipo de movimento (*type*), foi alterado apenas para *Walk*, para impedir o visitante de “voar” pela cena, pois esse cenário de visualização não era desejado.

A velocidade do movimento (*speed*) é uma questão mais sensível, sendo complexo estabelecer um valor fixo para ela. Foi já referido que este parâmetro pode ser usado para disfarçar lacunas na capacidade de processamento dos equipamentos, mas a questão é estabelecer um valor satisfatório para a maioria dos equipamentos. No nosso caso, foi definido um valor para a velocidade numa determinada máquina, mas após a visualização num outro equipamento com características diferentes, o valor escolhido já não se revelava satisfatório, pecando, ora por excesso, no caso de equipamentos mais recentes, ora por defeito, noutros mais antiquados. Assim, e após inúmeros testes com diversos equipamentos de diferentes capacidades de processamento, gráfico ou não, determinou-se um valor intermédio que pudesse ser utilizado em todos eles.

4.3.12 Âncoras

Por motivos que se prendem com as características do Fórum, nomeadamente a visualização de todo o recinto em simultâneo, a utilização de âncoras neste trabalho não foi concretizada nos moldes descritos na secção 4.2.17. Apenas foram utilizadas para permitir o acesso às alas dos pórticos do templo a partir da praça, ou para regressar à praça a partir dessas mesmas alas. Esse acesso é efectuado por intermédio de um clique com o rato na porta correspondente.

4.3.13 Viewpoints

Dado que é disponibilizada uma visita guiada através de uma animação que parte de um *viewpoint*, não sentimos a necessidade de criar mais pontos específicos, além dos necessários ao correcto funcionamento das âncoras.

4.3.14 Organização espacial

Este foi um princípio adoptado durante a evolução do ficheiro VRML usado na “reconstrução” do Fórum. A figura seguinte representa a estrutura interna dos nós no Cosmo Worlds. Como se pode verificar, a proximidade entre objectos na cena, manteve-se, sempre que possível, na estruturação interna dos nós usados para a definição desses objectos no ficheiro VRML.

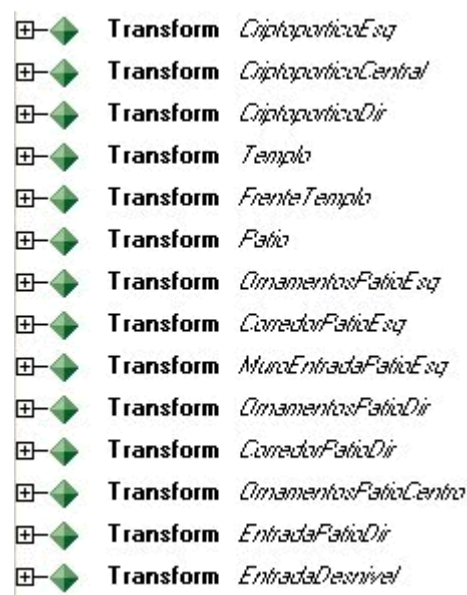


Figura 4.12 Posicionamento interno dos objectos no Cosmo Worlds

4.3.15 Compressão

O conjunto de pessoas que tiveram contacto com este trabalho já numa fase terminal, algumas delas profissionais nesta área, é unânime em considerar que um dos resultados mais surpreendentes foi a dimensão final do ficheiro VRML, cerca de 22Kb. No entanto, este valor só foi obtido após a compressão final

efectuada, pois antes disso o ficheiro de trabalho ocupava aproximadamente 200Kb.

No nosso caso o processo de compressão foi muito simples, pois a ferramenta de trabalho (Cosmo Worlds) possui um comando que o faz automaticamente e sem dificuldades. No final todos os *browsers* VRML estão aptos a interpretar, sem dificuldades, o ficheiro comprimido gerado.

4.3.16 Exportar para VRML

Como já foi mencionado na secção 3.2.3, existiram algumas peças que foram modeladas numa outra ferramenta de modelação tridimensional, o 3DStudio. Como eram objectos individuais para posteriormente incorporar no Cosmo Worlds, a preocupação com optimização neste caso centrou-se apenas no número de polígonos resultante da peça modelada. Para tal, foi aplicado à peça, após a sua modelação, o comando *Optimize*⁴¹ para tentar reduzir o número de polígonos sem deformação do seu aspecto inicial. Em algumas peças foram conseguidas melhorias nesse ponto.

Após a conclusão da peça, foi necessário exportá-la para VRML usando o motor de exportação existente no 3DStudio. Ao efectuar essa exportação e após alguns testes, constatou-se que existiam diferenças no número de polígonos gerados, como resultado da escolha de valores de alguns dos parâmetros existentes na janela de exportação. Como todas elas originavam resultados visuais idênticos, foram escolhidas as opções que produziam num número de polígonos mais reduzido.

4.3.17 Browser

No caso concreto do nosso trabalho, as considerações tidas na secção 4.2.26, revelaram-se ajustadas. Durante grande parte do desenvolvimento deste trabalho, foi utilizado como *browser* VRML o Cosmo Player na sua versão 2.1.1 (a última existente). Já com o Fórum quase no seu estado final, observava-se uma falha de

⁴¹ Comando existente na versão 2.5 do 3DStudio Max. Versão utilizada neste trabalho

renderização nas portas de acesso aos pórticos do templo, quando visualizadas a grande distância. Como essa falha não acontecia num outro equipamento mais poderoso, nomeadamente ao nível do *chip* gráfico utilizado (uma placa GeForce de 3ª geração), pensou-se inicialmente ser uma falha desse componente. Mais tarde, quando foram efectuados testes de visualização e desempenho com outros *browsers* VRML, nomeadamente o Blaxxun Contact e o Cortona⁴², verificámos que tal problema já não acontecia quando era utilizado o Cortona, mesmo em equipamentos bastante antiquados. Devido aos esforços da Parallel Graphics (empresa responsável pelo produto), o Cortona tem sido regularmente actualizado, com melhorias ao nível de *rendering* e desempenho, tentando potencializar ao máximo as capacidades gráficas dos componentes informáticos (processadores, *chips* gráficos, memória, etc.) de última geração. Como tal, é o Cortona da Parallel Graphics o *browser* VRML recomendado para a visualização do Fórum Flaviano de Conimbriga virtual.

⁴² Considerados pela comunidade científica na área, os melhores *browsers* a par do Cosmo Player

5. Conclusão

Um património passado, em alguns casos degradado, noutros mesmo desaparecido, pode ser hoje “recuperado” pelas novas tecnologias, lançando pistas para o futuro na exploração desses marcos da nossa cultura. O Fórum Flaviano de Conimbriga serviu de base a este nosso trabalho que visou a construção de um ambiente virtual que recriasse com exactidão, esse imponente monumento já desaparecido. Este é um trabalho, que só é possível devido ao labor dos arqueólogos e historiadores, e pretende ser fiel à realidade do monumento e ao desenho do seu criador. Além disso, constitui um meio de permitir ao grande público tomar um contacto com este conjunto monumental. A representação tridimensional interactiva permite um contacto mais profundo, melhor do que é possível com um desenho ou através de qualquer descrição escrita. O trabalho tem assim uma dupla função, pedagógica e social; pois torna possível uma “leitura” do passado a vários níveis e assegura a transmissão da cultura.

Assim e para a concretização deste desígnio, toda a área pertencente ao Fórum Flaviano foi modelada tridimensionalmente (com excepção do interior do templo, pelos motivos já descritos) à escala real e sempre com um especial cuidado em preservar as suas características geométricas, para que o rigor histórico, sempre essencial nestas situações, seja inquestionável. Mas foi após a aplicação das texturas, também elas rigorosamente tratadas, de modo a propiciar um maior realismo, que se pôde realmente contemplar a beleza e o requinte arquitectónico daquele que foi um dos espaços mais emblemáticos e representativos da antiga cidade romana de Conimbriga.

São inúmeros os casos por todo o planeta de trabalhos arqueológicos, historiadores em investigação e projectos nas mais diversas áreas, que visam a divulgação e preservação de muitas relíquias históricas. E por preservação também nos referimos à recriação por via de maquetas, desenhos ou representações virtuais de espaços irremediavelmente desaparecidos. Todas essas actividades perspectivam o futuro, não esquecem o passado, pois é sobre ele que assenta o presente. Este nosso trabalho pretende dar uma pequena

contribuição a esta crescente consciencialização das questões relacionadas com a herança cultural da humanidade, ao disponibilizar ao público em geral, via Web e não só, o acesso a este espaço virtual.

É justamente nos aspectos de acessibilidade, que este trabalho difere da quase que totalidade dos trabalhos conhecidos do mesmo tipo, pois todo o cuidado colocado na procura de um bom equilíbrio entre realismo, desempenho e dimensão dos ficheiros, resultou num ficheiro final de reduzidas dimensões, fácil de descarregar e com um bom desempenho em qualquer computador minimamente recente.

Das opiniões recolhidas de quem teria legitimidade e competência para o fazer e pelas inúmeras análises e visualizações de trabalhos nesta área, podemos afirmar que os objectivos propostos foram cumpridos:

- ✓ Realismo: comprovado por especialistas, particularmente pelo Director do Museu Monográfico de Conimbriga;
- ✓ Desempenho: funciona em qualquer equipamento informático doméstico adquirido nos nossos dias;
- ✓ Dimensão dos ficheiros: o ficheiro principal (VRML), ocupa apenas 22Kb e o trabalho na sua totalidade, incluindo as texturas, não ultrapassa os 130Kb;
- ✓ Acessibilidade: por via da linguagem utilizada, o VRML, e deste último factor (dimensão), este trabalho pode ser acedido facilmente por qualquer pessoa com acesso à Internet.

Debruçando-nos agora um pouco sobre a linguagem utilizada na elaboração deste trabalho, a *Virtual Reality Modeling Language* (VRML), existe, em nossa opinião, um aparente paradoxo: se com esta linguagem se conseguem realizar trabalhos com uma relação qualidade/desempenho bastante aceitável, com um baixo custo, e sem existirem grandes alternativas⁴³, porque não existirão mais projectos em VRML, principalmente na Web? Essa é uma questão que não tem uma resposta objectiva e clara, podendo imiscuir-se em áreas e motivações que não nos compete abordar no momento. No entanto, gostaríamos de efectuar um breve comentário/análise relacionado com esta questão. Vivendo nós numa

⁴³ Estamo-nos a referir à disponibilização via Web

sociedade de consumo desenfreado, onde o lucro é quem mais ordena e como tal, tempo é dinheiro, raros são os casos, em quase todas as actividades, em que o método do *best effort* não é o que está presente em maior número de situações. É mais rápido, prático e por vezes também consegue produzir bons resultados. Isto para dizer, inserido no nosso contexto, que é nossa convicção que são poucos os trabalhos de VRML, elaborados de facto em VRML. Será, porventura, pela falta de ferramentas suficientemente poderosas e intuitivas e pela maior familiaridade dos criadores com ferramentas mais “universais”. As consequências dessa situação são principalmente os ficheiros de grande dimensão e de navegabilidade reduzida, se não mesmo impossível, em equipamentos informáticos domésticos.

Uma outra questão fundamental é a optimização do “código” VRML. Esta requer uma consciencialização profunda e o dispêndio de algum tempo dos criadores no processo de desenvolvimento de um qualquer projecto elaborado nesta linguagem.

Também aqui neste ponto, esta dissertação tenta contribuir para diminuir uma das maiores dificuldades tidas na fase inicial deste trabalho: o que optimizar, quando optimizar e como optimizar. Foi assim, e após algumas dificuldades na selecção e agregação da informação/conhecimento necessária para suprir tal dificuldade, que decidimos “condensar” num só capítulo desta dissertação as questões relacionadas com optimização em VRML. Estas estavam dispersas por uma vasta bibliografia, o que dificultava o acesso a essa informação de grande relevância na elaboração dos “mundos” VRML. É nossa firme convicção que com uma referência bibliográfica deste tipo disponível no início do nosso trabalho, reduziríamos o tempo de execução do mesmo para menos de metade. Este, julgamos nós, também é uma importante contribuição que deixamos, principalmente aos interessados em iniciar-se nesta “arte” do Virtual (*Reality Modeling Language*).

A incorporação das questões da optimização desde uma fase embrionária do trabalho e o facto do mesmo ter sido elaborado usando nativamente o VRML em praticamente a sua totalidade, são numa perspectiva mais técnica os pontos mais fortes deste trabalho.

Em matéria de trabalho futuro, todas as considerações que possamos efectuar, não podem estar indissociáveis dos avanços tecnológicos dos componentes necessários à visualização e “distribuição” destes conteúdos vulgarmente denominados por Realidade Virtual. Com efeito, a rivalidade existente entre algumas empresas do sector, tem proporcionado ao consumidor cada vez melhores meios, por um preço cada vez mais acessível. Ainda há bem pouco tempo a velocidade de relógio dos processadores situava-se , na altura, em uns excelentes 1Ghz, sendo que actualmente já ultrapassa os 3Ghz. A “luta” pelo domínio na área dos *chips* gráficos está ao rubro, com uma cadência incrível de lançamento no mercado de *chips* cada vez mais poderosos e rápidos, originando uma redução nos preços das gerações imediatamente anteriores, já de si com um avanço tecnológico considerável. Finalmente a gradual generalização da banda larga, seja por ADSL, ou por Cabo, que permite “levar” a casa do utilizador comum, aqueles conteúdos multimédia que até agora estavam um pouco arredados da Internet. Tudo isto tem como efeito imediato, a possibilidade de visualização de mais e melhores conteúdos, por via do hardware e uma maior capacidade de “transporte” de mais informação por unidade de tempo, por via da banda larga. Pelo exposto, somos tentados a idealizar uma cidade de Conimbriga totalmente recriada, acompanhada por uma sequência de áudio que retracte o quotidiano da época, e eventualmente até com personagens trajadas a rigor com a indumentária do período romano a movimentarem-se pelas ruas cheias de vida... Teremos exagerado? Daqui a alguns anos, talvez menos do que possamos imaginar, responderemos a essa questão.

Em conclusão, não existem fórmulas mágicas nem varinhas de condão, mas com um pouco de perseverança e uma consciencialização de alguns cuidados a ter na concepção, principalmente ao nível da optimização, conseguem-se conceber bons trabalhos em VRML. Além do mais é uma “tecnologia” barata, fácil de utilizar, interactiva, standard para a Web, acessível a qualquer um, funciona num qualquer equipamento doméstico, é independente da plataforma, etc., pelo que deixamos este pensamento: VRML... porque não?

6. Bibliografia

- Alarcão, J., Etienne, R. e Golvin, J.-C. (1994), *A Maqueta do Centro Monumental Flaviano*, pp. 11-43, Instituto Português de Museus/Museu Monográfico de Conimbriga
- Ames, A., Nadeau, D. e Moreland, J. (1997), *VRML 2.0 Sourcebook*, pp. 220-222, 2nd Edition, John Wiley & Sons
- Birley, R. (1973), *Housesteads - Roman Fort*, Frank Graham
- Carey, R., Bell, G. e Marrin, C. (1997), *ISO/IEC 14772-1:1997 Virtual Reality Modeling Language (VRML97)*
- Cosmas, J. (Outubro 2001), *3D Measurement & Virtual Reconstruction of Ancient Lost Worlds of Europe*, Cultivate Interactive
- Cosmo Player 2.1.1 Documentation* (1999), Silicon Graphics
- Cosmo Worlds 2.0 Tutorial* (1998), pp. 98-101, 170-178, 334-349, Silicon Graphics
- Cosmo Worlds 2.0 User's Guide* (1998), pp. 6-12, 132, 176, 190-201, 216-224, Silicon Graphics
- Direcção Geral dos Edifícios e Monumentos Nacionais (1948), *Oppidum Romano de Conimbriga*, pp. 5-22, Ministério das Obras Públicas
- Foley, J., Dam A., Feiner, S. e Hughes J. (1990), *Computer Graphics Principles and Practice*, Addison-Wesley Publishing Company
- Gonçalves, A., Mendes, A. (2003), The rebirth of a Roman Forum: The case study of the Flavian Forum of Conimbriga, aceite para publicação in *Proceedings of Enter the Past - The E-way into the four Dimensions of Cultural Heritage, 8-12 Abril 2003, Viena, Áustria*
- Hartman, J. e Wernecke, J. (1996), *The VRML 2.0 handbook: Building moving worlds on the web*, Addison-Wesley Publishing Company

- Lemay, L., Murdock, K. e Couch, J. (1996), *3D Graphics & VRML 2.0*, pp. 208-210, 219-223, 294-295, Sams.net Publishing
- Marcos, A., Bernardes, P., Ferreira, R., Lains, H. e Martins, M. (2000), Virtual Tour to Bracara Augusta, *Topics*, 5, pp. 9-10, 2000
- Marrin, C. e Campbell, B. (1997), *Teach yourself VRML 2 in 21 days*, pp. 43,46, 89-102, 275-289, 311-314, Sams.net Publishing
- Narciso, S. (1999), *Edifício Principal da ESTG em Realidade Virtual II*, pp. 6-12, Escola Superior de Tecnologia e Gestão/Instituto Politécnico de Leiria
- Ogleby, C. (2001), The ancient city of Ayutthaya – explorations in virtual reality and multimedia, in *Proceedings of International Workshop on Recreating the Past - Visualization and Animation of Cultural Heritage*, 26 Fevereiro – 1 Março 2001, Ayuttaya, Tailândia
- Pugnaloni, F. e Clini, P. (2001), Vitruvius Basilica in Fano, Italy, journey through the virtual space of the reconstructed memory, in *Proceedings of International Workshop on Recreating the Past - Visualization and Animation of Cultural Heritage*, 26 Fevereiro – 1 Março 2001, Ayuttaya, Tailândia
- Reidsma, D., Kragtwijk, M. e Nijholt, A. (2001), Building “Huys Hengelo” in VRML, in *Proceedings of the International Cultural Heritage Informatics Meeting 2001*, Setembro 2001, Milão, Itália
- Santos, J. e Barata, J. (1998), *3D Studio Max 2 Curso Completo*, pp. 263-264, FCA – Editora de Informática
- Scheele, V. (1998), *VRML - Authoring Tools*, Universität Mannheim
- Shiode, N. e Grajetzki, W. (2000), *Developing a Reconstructive Model of Hawara Labyrinth Pyramid Complex*
- Tantatsanawong, P., Pipitpotjanakarn, R., Lertlum, S. e Sriarun, K. (2001), Recreating the past using object oriented virtual reality for Ayutthaya world heritage historical park: A case study of Wat Ratchaburana, in *Proceedings of*

International Workshop on Recreating the Past - Visualization and Animation of Cultural Heritage, 26 Fevereiro – 1 Março 2001, Ayuttaya, Tailândia

Vacca, J. (1998), *VRML Clearly explained 2nd Edition*, pp. 768-773, AP Professional

Vlahakis, V., Almeida, L., Ioannidis, N., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., Gleue, T. e Daehne, P. (2002), Archeoguide: An Augmented Reality Guide for Archaeological Sites, *IEEE Computer Graphics and Applications*, 22, 5, pp. 52-60, Setembro/Outubro